User's Manual

32

SH7786 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC Engine Family / SH7786 Series

R8A77860

Renesas Electronics
www.renesas.com

Rev.1.00 Nov 2010

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 in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions may occur due to the false recognition of the pin state as an input signal. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

   The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Preface

This LSI is a RISC (Reduced Instruction Set Computer) microcomputer which includes a Renesas Electronics-original RISC CPU (SH-4A) and various 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.

Notes on reading this manual:

- In order to understand the overall functions of the chip

  Read the manual according to the contents. This manual consists of parts on the CPU, system control functions, and peripheral functions.
- In order to understand individual instructions in detail

  Read the separate manuals SH-4A Extended Functions Software Manual and SH-4A Software Manual.

Rules:		
	Bit order:	The MSB is on the left and the LSB is on the right.
	Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx.
	Signal notation:	An overbar is added to active-low signals: xxxx

Abbreviations

ALU	Arithmetic Logic Unit
ASID	Address Space Identifier
BGA	Ball Grid Array
CPG	Clock Pulse Generator
CPU	Central Processing Unit
DBSC3	DDR3-SDRAM Interface
DDR	Double Data Rate
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
FIFO	First-In First-Out
FPU	Floating-point Unit
HAC	Audio Codec
H-UDI	User Debugging Interface
INTC	Interrupt Controller
JTAG	Joint Test Action Group
LBSC	Local Bus State Controller
LRAM	L Memory
LRU	Least Recently Used
LSB	Least Significant Bit
MMU	Memory Management Unit
MSB	Most Significant Bit
PC	Program Counter

PCI	Peripheral Component Interconnect
PCIEC	PCI Express Controller
PFC	Pin Function Controller
RISC	Reduced Instruction Set Computer
SCIF	Serial Communication Interface with FIFO
SSI	Serial Sound Interface
TAP	Test Access Port
TLB	Translation Lookaside Buffer
TMU	Timer Unit
UART	Universal Asynchronous Receiver/Transmitter
UBC	User Break Controller
WDT	Watchdog Timer

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

Contents

Section 1 Overview

The SH7786 is an integrated system-on-a-chip microprocessor of SuperH™ RISC engine family SH7780 series that is designed as a high performance and embedded processor aimed at the multimedia, information, networking, infotainment and CIS (car information system) market.

The SH7786 features a PCI Express interface, USB 2.0 interface, and Ethernet MAC, DDR3 interface that can be coupled to the DDR3-SDRAM. And also features a DMA controller, timers, serial communications functions with an audio interface, and a display unit (DU) that supports digital RGB display. These independent external bus interfaces enable large-size data transfer and streaming data processing.

The SH7786 has a dual SH-4A processor core which is a 32-bit RISC (reduced instruction set computer) multiprocessor including an FPU as well as a CPU, providing upward compatibility (instruction set level) with the SH-1, SH-2, SH-3, and SH-4 microcomputers. The CPU and FPU run at 533 MHz (1.9 GIPS, 7.4 GFLOPS). The processor also includes an instruction cache, an operand cache for which copy-back or write-through mode is selectable, a four-entry fully associative instruction TLB (translation look-aside buffer), and an MMU (memory management unit) with a 64-entry fully associative unified TLB

Note: The CPU-related functions of the SH7786 are significantly extended from those of the SH7785. Those extensions are summarized in "Appendix A".

1.1 SH7786 Features

Table 1.1 SH7786 Features

Item	Features
LSI	• Operating frequency: 533MHz (1.92GIPS,7.46GFLOPS) • Voltage: 1.25 V (internal), 1.5 V (DDR3-SDRAM/PCI Express), 3.3V (I/O /PCI Express) • Pakages: 593-pin BGA (size: 25 × 25mm, pitch: 0.8mm) • Local bus interface (external bus) Separate 26-bit address and 32-bit data buses External bus frequency: 88.9 MHz • DDR3-SDRAM bus interface (external bus) Separate 16-bit address and 32-bit data buses External bus frequency: 533 MHz (maximum data transmission rate 1066 Mbps) • PCI Express bus interface (external bus) 4/2/1 lanes + 1 lane Root point/end point selectable
CPU (each CPU)	• Renesas Electronics original architecture • 32-bit internal data bus • General-register file: Sixteen 32-bit general registers (eight 32-bit shadow registers) Seven 32-bit control registers Four 32-bit system registers • RISC-type instruction set (providing upward compatibility with code for the SH-1, SH-2, SH-3 and SH-4 processors) Instruction length: 16-bit fix length for improved code efficiency Load/store architecture Delayed branch instructions Conditional instruction execution Instruction-set design on the C language • Super-scalar architecture covering both the FPU and CPU provides for the simultaneous execution on any two instructions • Instruction-execution time: Two instructions per cycle (max) • Virtual address space: 4 Gbytes • ASIDs (address-space identifiers): 8 bits, for 256 virtual address spaces • Internal multiplier

Item	Features
FPU (each CPU)	• On-chop floating-point coprocessor • Supports single (32-bit) and double (64-bit) precisions • Supports IEEE754 compliant data types and exceptions • Two rounding modes. Round to nearest and Round to Zero. • Handling of denormalized numbers: Truncation to zero or interrupt generation for IEEE754 compliance • Floating-point registers: 32-bit × 16 registers × 2 banks (single-precision × 16 registers or double-precision × 8 registers) × 2 • 32-bit CPU-FPU floating-point communications register (FPUL) • FMAC (multiply and accumulate) instruction • FDIV (divide) and FSQRT (square root) instructions • FLDI0/FLDI1 (load constants 0 and 1) instructions • Instruction execution times: Latency (FADD/FSUB): 3 cycles (single-precision), 5 cycles (double-precision) Latency (FMAC/FMUL): 5 cycles (single-precision), 7 cycles (double-precision) Pitch (FADD/FSUB): 1 cycle (single-precision/double-precision) Pitch (FMAC/FMUL): 1 cycle (single-precision), 3 cycles (double-precision) Note: FMAC only supports single-precision operands • 3-D graphics instructions (single-precision only) 4-dimensional vector conversion and matrix operations (FTRV): 4 cycles (pitch), 8 cycles (latency) 4-dimensional vector (FIPR) inner product: 1 cycle (pitch), 5 cycles (latency)
Memory management unit (MMU) (each CPU)	• 4-Gbyte address space, 256 address space identifiers (8-bit ASID) • Single virtual memory mode and multiple virtual memory mode • Multiple page sizes: 1, 4, 8, 64, or 256 Kbytes, or 1, 4, or 64 Mbytes • 4-entry fully associative TLB for instructions • 64-entry fully associative TLB for instructions ond operands • Selection of software-driven or random-counter replacement algorithms • The TLB is address-mapped, making its contents directly accessible • 29-bit and 32-bit physical address modes

Item	Features
Cache memory (each CPU)	• Instruction cache (IC) 32-Kbyte 4-way set associative 256 entries/way, 32-byte block length • Operand cache (OC) 32-Kbyte 4-way set associative 256 entries/way, 32-byte block length • Selectable write method (copy-back or write-through) • Storage queue (32-byte × 2 entries) • 1-stage copy-back buffer and 1-stage write-through buffer • Supports Cache coherency • Write method: Copy-back or write-through
LRAM (each CPU)	• IL memory 8-Kbyte high-speed memory Three independent read/write ports 8/16/32/64-bit access from CPU or FPU 8/16/32/64-bit access and 16/32-byte access in response to external requests Support for protection of memory from CPU or FPU access • OL memory 16-Kbyte high-speed memory Three independent read/write ports 8/16/32/64-bit access from CPU or FPU 8/16/32-bit access and 16/32-byte access in response to external requests Support for protection of memory from CPU or FPU access
L2 cache / Shared memory (each CPU)	• 256-Kbyte L2 cache memory • 4-way set associative • Write method: Write-through • Compatible with IC only mode, OC only mode and IC/OC common mode • Available for use as a shared memory (L2 256K, shared memory 256K and L2 128K/shared memory 128K selectable)

Item	Features
Interrupt controller (INTC)	• Two interrupt distribution methods: Static: Configurable routing to dedicated processor cores Dynamic: Broadcasted to all cores and ACK'ed exclusively • Nine independent external interrupts: NMI and IRQ7 to IRQ0 NMI: Falling/rising edge selectable IRQ: Falling/rising edge or high level/low level selectable • 15-level encoded external interrupts: IRL3 to IRL0, or IRL7 to IRL4 • On-chip module interrupts: A priority level can be set for each module The following modules can issue on-chip module interrupts: TMU, DU, SCIF, WDT, H-UDI, DMAC0, DMAC1, HPB-DMAC, HAC, PCIE, USB, HSPI, SDIF, SSI, FLCTL, I2C, GPIO
Local bus state controller (LBSC)	• A dedicated Local bus interface Controls the external memory space divided into five 64-Mbyte (max) areas The interface type, bus width, and wait-cycle insertion can be set for each area • SRAM interface Wait-cycle insertion can be set by register values Wait-cycle insertion by the RDY pin Connectable as any area (CS0/CS1/CS2/CS3/CS4/CS5/CS6) Selectable bus width: 32/16/8-bit • Burst ROM interface Wait-cycle insertion can be set by register values Number of units in burst transfers can be set by register values Connectable as any area(CS0/CS1/CS2/CS3/CS4/CS5/CS6) Selectable bus width: 32/16/8-bit • MPX interface Address/data multiplexing Connectable as any area(CS0/CS1/CS2/CS3/CS4/CS5/CS6) Selectable bus width: 32-bit • SRAM interface with byte-control Connectable as area CS1, CS4 Selectable bus width 32/16-bit • PCMCIA interface (only for little-endian mode) Wait-cycle insertion can be set by register values Bus-sizing function for adaptation to the I/O bus width Connectable as area CS5, CS6 Selectable bus width: 16/8-bit • Supports transfer to and from E-IDE/ATAPI devices (ATA3) Supports PIO mode 4 and multi-word DMA mode 2 Connectable as area CS5, CS6 • Big or little endian selectable

Item	Features
DDR3-SDRAM bus controller (DBSC)	• A dedicated DDR3-SDRAM bus interface Multi-bank support: Supports multi-bank (four banks) operation Number of banks: Supports four or eight banks (however, no more than four banks can be opened concurrently) Bus width: 32-bit Supports preceding precharging and activation Burst length: Eight Burst type: Sequential (fixed) CAS latency: 5, 6, 7, 8, 9, 10 cycles • Auto-refresh mode An average interval is selectable by a register setting. Preceding refresh operations are performed when there are no pending requests. • Self-refresh mode • Connectable memory capacity: Up to 2 Gbytes With a 32-bit bus width 16M × 16bit (256Mbit) × 2, 32M × 16bit (512Mbit) × 2, 64M × 16bit (1Gbit) × 2, 128M × 16bit (2Gbit) × 2, 32M × 8bit (256Mbit) × 4, 64M × 8bit (512Mbit) × 4, 128M × 8bit (1Gbit) × 4, 256M × 8bit (2Gbit) × 4
PCI Express controller (PCIEC)	• Subsets PCI Express Base Specification, revision 1.1 Lane: 4-lane (×4) + 1-lane (×1), bus frequency: 2.5 GHz • Operate as PCI Express Root point or End point • Supports multiple lanes (4/2/1) • Virtual channel (VC): 1-channel • Supports automatically Retry (re-transmission) function by Ack/Nak • Maximum payload size: 1-Kbytes(PCIEC0), 512-bytes(PCIEC1 and PCIEC2)

Item	Features
USB controller (USB)	• Compatible with the Universal Serial Bus Ver.2.0 Specifications • Ports: 2 ports (one is host/function and the other is host only) • Host: EHCI version 1.0, OHCI version 1.0a • Function: On-chip UDC (USB device controller) compatible with USB 2.0 • Signaling bit rate: High-speed (480 Mbps), full-speed (12 Mbps) • Equipped each one module for host/function, and host 2 port works with a built-in root hub • OTG is not supported
Ethernet MAC controller (Ether)	• IEEE802.3u MAC layer • 10M/100Mbps transfer • Flow control (802.3x/back pressure) • IEEE802.3u MII interface • Magic packet detection • Dedicated DMA controller
Direct memory access controller 0 (DMAC0)	• Number of channels: 6-channel • 6-channel physical address DMA controller • Four channels support external requests (channels 0 to 3) • Address space: 4 Gbytes (physical address) • Transfer data size: 8, 16, or 32 bits, 16 or 32 bytes • Address mode: Dual address mode • Transfer request: External (channels 0 to 3), on-chip peripheral module or auto-requests • DACK/DRAK selectable (four external pins) • Bus mode: Cycle-still mode or burst mode • Channel priority fixed priority mode or round robin mode • Supports scatter-gather mechanism

Item	Features
Direct memory access controller 1 (DMAC1)	• Number of channels: 4-channel • 4-channel psysical address DMA controller • Adress space: 4 Gbytes • Transfer data size: 32 bits, 8, 16 or 32 bytes (channels 0 and 1) 8, 16, 32 bits or 8,16, 32bytes (channels 2 and 3) • Address mode: Dual address mode • Transfer request: Auto-requests • Supports command chain (only on channels 0 and 1) • Supports cache coherency (only on channels 0 and 1) • Supports scatter-gather mechanism (only on channels 0 and 1)
Clock pulse generator (CPG)	• CPU clock selection: 533 MHz (max) • Local bus: 88.9MHz (max) • DDR3-SDRAM interface: 533MHz (max) • Peripheral bus: 66MHz (max) • Power down mode CPU sleep mode (each CPU) Module standby mode DDR power supply backup mode
Watchdog timer (WDT)	• Number of channels: 3-channel (local 1-channel, each CPU 1-channel) • Local watchdog timer (only power-on reset) (operation in watchdog-timer mode or interval-timer mode is selectable) • Each CPU watchdog timer (only manual reset)
Timer Unit (TMU)	• Number of channels: 12-channel • 12-channel auto-reloading 32-bit down-counter • Input-capture function (only on channel 2) • Up to six input signals to clock of counter are each selectable from external or peripheral clock

Item	Features
Display unit (DU)	• Display plane 4 planes (a maximum number at 854 dots × 480 dots) 3 planes (a maximum number at 800 dots × 600 dots) • CRT scanning method: Non-interlaced, interlaced, interlaced sync & video • Synchronization modes: Master mode (internal synchronization mode), TV synchronization mode (external synchronization mode), synchronization-mode switching mode • Incorporates color palettes Displays 256 colors from among 260 thousand possible colors Four palettes (one can be set for each layer) • Blending ratio setting Number of color-palette planes with blending ratios: 4 planes α plane: (used in common with the display plane) • Digital RGB output: 6-bit precision for each of R, G, and B • Dot clock: Can be switched between external input and internal clock (division ratio: 1 to 32)
Serial communications interface with FIFO (SCIF)	• Number of channels: 6-channel (max) • On-chip 64-byte (8 bits × 64) FIFO for each of the 6 channels • 2 full-duplex channels • Selectable mode: Asynchronous mode or synchronous mode • Any bit rate that can be generated by the on-chip baud-rate generator is selectable • On-chip modem-control function (RTS and CTS) for channel 0 • Internal clock signal from the baud-rate generator or external clock signal from the SCK pin is selectable
I^2C Bus interface (I^2C)	• Number of channels: 2-channel • Subset with Philips IIC bus interface • Supports master/slave mode • Supports multi-master capability • Supports fast mode • Clock frequency can be selectable Note: Does not comply with I^2C timing specification because I^2C buffers are 3.3 V LVTTL interface

Item	Features
Serial peripheral interface (HSPI)	• Number of channels: 1-channel • Supports full duplex transceiver • Supports master/slave mode • Any bit rate that can be generated by the on-chip baud-rate generator is selectable
SD host interface (SDIF)	• Number of channels: 2-channel (max) • SD memory I/O card interface (1-/4-bits SD bus) • Error check function: CRC7 (command), CRC16 (data) • Interrupt requests: Card access interrupt, SDIO access interrupt, card detect interrupt • DMAC transfer request: SD_BUF write, SD_BUF read • Card detection and write protection supported • The SD host interface manual is available, subject to a confidentially agreement. (Please contact a Renesas Technology sales representative for details.)
Audio CODEC interface (HAC)	• Number of channels: 2-channel (max) • Digital interface for audio CODECs • Supports transfer via slots 1 to 4 • DMAC transfer rates for transmission/reception are selectable from 16- or 20-bit • Supports various sampling rates by adjusting the allocation of data to slots
Serial sound interface (SSI)	• Number of channels: 4-channel (max) • Supports transfer of compressed and non-compressed data • Selectable frame size
NAND flash memory controller (FLCTL)	• Number of channels: 1-channel (max) • Exclusively for NAND-type flash memory • Operating modes: Command-access mode, sector-access mode • Data transfer FIFOs On-chip 224-byte FIFO for transfer of data to and from flash memory On-chip 32-byte FIFO for transfer of control codes Flag bit to indicate overruns and underruns during access from the CPU or DMA
General purpose I/O (GPIO)	• General purpose I/O port pins: 103 • Some GPIO pins are configurable as interrupts

Item	Features
User break controller (UBC) (each CPU)	• Supports user-break interrupts as a facility for debugging • Two break channels • Addresses, data values, types of access, and widths of data are all specifiable as break conditions • Supports a sequential break function
User debug interface (H-UDI)	• JTAG interface (TCK, TMS, TRST, TDI, TDO) • Supports the E10A emulator • Realtime branch tracing
Package	• 593-pin FC-BGA, 25 mm × 25 mm, ball pitch: 0.8 mm
Power supply voltage	• Core (VDD), PCI Express Core (VDD-PCIE) , PLL2 (VDDAI): 1.25V • DDR3 I/O (VCCQ-DDR15), PCI Express I/O (VCCQ-PCI15): 1.5V • I/O (VCCQ), PCI Express I/O (VCCQ-PCI): 3.3V

1.2 Block Diagram

A block diagram of the SH7786 is given as figure 1.1.

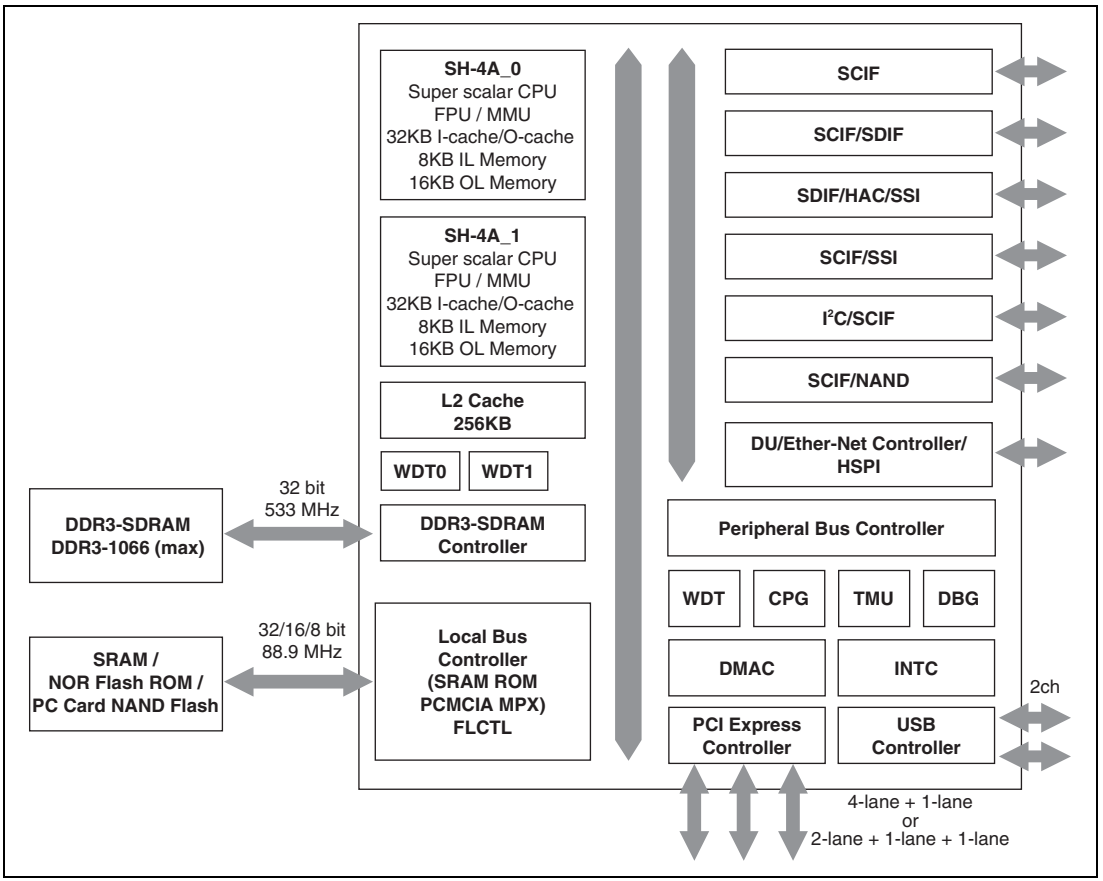


Figure 1.1 SH7786 Block Diagram

1.3 Pin Arrangement Table

No.	Pin name	I/O	Function
1	MDQ0	IO	DDR data 0
2	MDQ1	IO	DDR data 1
3	MDQ2	IO	DDR data 2
4	MDQ3	IO	DDR data 3
5	MDQ4	IO	DDR data 4
6	MDQ5	IO	DDR data 5
7	MDQ6	IO	DDR data 6
8	MDQ7	IO	DDR data 7
9	MDQ8	IO	DDR data 8
10	MDQ9	IO	DDR data 9
11	MDQ10	IO	DDR data 10
12	MDQ11	IO	DDR data 11
13	MDQ12	IO	DDR data 12
14	MDQ13	IO	DDR data 13
15	MDQ14	IO	DDR data 14
16	MDQ15	IO	DDR data 15
17	MDQ16	IO	DDR data 16
18	MDQ17	IO	DDR data 17
19	MDQ18	IO	DDR data 18
20	MDQ19	IO	DDR data 19
21	MDQ20	IO	DDR data 20
22	MDQ21	IO	DDR data 21
23	MDQ22	IO	DDR data 22
24	MDQ23	IO	DDR data 23
25	MDQ24	IO	DDR data 24
26	MDQ25	IO	DDR data 25
27	MDQ26	IO	DDR data 26
28	MDQ27	IO	DDR data 27
29	MDQ28	IO	DDR data 28
30	MDQ29	IO	DDR data 29
31	MDQ30	IO	DDR data 30
32	MDQ31	IO	DDR data 31
33	MDM0	O	DDR data mask 0
34	MDM1	O	DDR data mask 1
35	MDM2	O	DDR data mask 2
36	MDM3	O	DDR data mask 3
37	MDQS0	IO	DDR data strobe 0
38	MDQS1	IO	DDR data strobe 1
39	MDQS2	IO	DDR data strobe 2
40	MDQS3	IO	DDR data strobe 3
41	$\overline{\text{MDQS0}}$	IO	DDR data strobe 0
42	$\overline{\text{MDQS1}}$	IO	DDR data strobe 1
43	$\overline{\text{MDQS2}}$	IO	DDR data strobe 2
44	$\overline{\text{MDQS3}}$	IO	DDR data strobe 3
45	MA0	O	DDR address 0
46	MA1	O	DDR address 1
47	MA2	O	DDR address 2
48	MA3	O	DDR address 3
49	MA4	O	DDR address 4
50	MA5	O	DDR address 5
51	MA6	O	DDR address 6
52	MA7	O	DDR address 7
53	MA8	O	DDR address 8
54	MA9	O	DDR address 9
55	MA10	O	DDR address 10
56	MA11	O	DDR address 11
57	MA12	O	DDR address 12
58	MA13	O	DDR address 13
59	MA14	O	DDR address 14
60	MA15	O	DDR address 15
61	MBA0	O	DDR bank address 0
62	MBA1	O	DDR bank address 1
63	MBA2	O	DDR bank address 2

No.	Pin name	I/O	Function
64	MCK0	O	DDR clock 0
65	$\overline{\text{MCK0}}$	O	DDR clock 0
66	MCK1	O	DDR clock 1
67	$\overline{\text{MCK1}}$	O	DDR clock 1
68	$\overline{\text{MCS0}}$	O	DDR chip select
69	$\overline{\text{MCS1}}$	O	DDR chip select
70	$\overline{\text{MRAS}}$	O	DDR row address select
71	$\overline{\text{MCAS}}$	O	DDR column address select
72	$\overline{\text{MWE}}$	O	DDR write enable
73	MODT0	O	DDR on chip terminator
74	MODT1	O	DDR on chip terminator
75	MCKE0	O	DDR clock enable
76	MCKE1	O	DDR clock enable
77	MVREF0	I	DDR reference voltage DQ
78	MVREF1	I	DDR reference voltage ADR/CMD
79	$\overline{\text{MBKPRST}}$	I	DDR backup reset
80	MZQ	I	DDR MZQ
81	$\overline{\text{MRESET}}$	I	DDR reset
82	SDBUP	I	DDR backup mode input
83	D0/FD0	IO/ IO	Local bus data 0/ NAND flash data 0
84	D1/FD1	IO/ IO	Local bus data 1/ NAND flash data 1
85	D2/FD2	IO/ IO	Local bus data 2/ NAND flash data 2
86	D3/FD3	IO/ IO	Local bus data 3/ NAND flash data 3
87	D4/FD4	IO/ IO	Local bus data 4/ NAND flash data 4
88	D5/FD5	IO/ IO	Local bus data 5/ NAND flash data 5
89	D6/FD6	IO/ IO	Local bus data 6/ NAND flash data 6
90	D7/FD7	IO/ IO	Local bus data 7/ NAND flash data 7
91	D8	IO	Local bus data 8
92	D9	IO	Local bus data 9
93	D10	IO	Local bus data 10
94	D11	IO	Local bus data 11
95	D12	IO	Local bus data 12
96	D13	IO	Local bus data 13
97	D14	IO	Local bus data 14
98	D15	IO	Local bus data 15
99	D16	IO	Local bus data 16
100	D17	IO	Local bus data 17
101	D18	IO	Local bus data 18
102	D19	IO	Local bus data 19
103	D20	IO	Local bus data 20
104	D21	IO	Local bus data 21
105	D22	IO	Local bus data 22
106	D23	IO	Local bus data 23
107	D24	IO	Local bus data 24
108	D25	IO	Local bus data 25
109	D26	IO	Local bus data 26
110	D27	IO	Local bus data 27
111	D28	IO	Local bus data 28
112	D29	IO	Local bus data 29
113	D30	IO	Local bus data 30
114	D31	IO	Local bus data 31
115	A0	O	Local bus address 0
116	A1	O	Local bus address 1
117	A2	O	Local bus address 2
118	A3	O	Local bus address 3
119	A4	O	Local bus address 4
120	A5	O	Local bus address 5
121	A6	O	Local bus address 6

No.	Pin name	I/O	Function
122	A7	O	Local bus address 7
123	A8	O	Local bus address 8
124	A9	O	Local bus address 9
125	A10	O	Local bus address 10
126	A11	O	Local bus address 11
127	A12	O	Local bus address 12
128	A13	O	Local bus address 13
129	A14	O	Local bus address 14
130	A15	O	Local bus address 15
131	A16	O	Local bus address 16
132	A17	O	Local bus address 17
133	A18	O	Local bus address 18
134	A19	O	Local bus address 19
135	A20	O	Local bus address 20
136	A21	O	Local bus address 21
137	A22	O	Local bus address 22
138	A23	O	Local bus address 23
139	A24	O	Local bus address 24
140	A25	O	Local bus address 25
141	$\overline{\text{CS0}}$	O/O	Chip select 0
142	$\overline{\text{CS1}}$/A26	O/O	Chip select 1/Local bus address 26
143	$\overline{\text{CS2}}$	O	Chip select 2
144	$\overline{\text{CS3}}$	O	Chip select 3
145	$\overline{\text{CS4}}$	O	Chip select 4
146	$\overline{\text{CS5}}$	O	Chip select 5
147	$\overline{\text{CS6}}$	O	Chip select 6
148	$\overline{\text{RD}}$/$\overline{\text{FRAME}}$/ $\overline{\text{FRE}}$	O/O	Read strobe/MPX IF FRAME/NAND flash read enable
149	R/$\overline{\text{W}}$	O	Read/Write
150	$\overline{\text{BS}}$	O	Bus start
151	$\overline{\text{WE0}}$/$\overline{\text{REG}}$	O/O	Write enable 0/ PCMCIA IF REG
152	$\overline{\text{WE1}}$/$\overline{\text{FWE}}$	O/O	Write enable 1/NAND flash write enable
153	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	O/O	Write enable 2/ PCMCIA IF IORD
154	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	O/O	Write enable 3/ PCMCIA IF IOWR
155	$\overline{\text{RDY}}$	I	Bus ready
156	CLKOUT	O	Clock out
157	CLKOUTENB	O	Clock out enable
158	GCLK	I	PCI Express differential clock input
159	$\overline{\text{GCLK}}$	I	PCI Express differential clock input
160	PCIE_TX0_0	O	PCI Express transmit differential data 0 ch0
161	$\overline{\text{PCIE\_TX0\_0}}$	O	PCI Express transmit differential data 0 ch0
162	PCIE_RX0_0	I	PCI Express receive differential data 0 ch0
163	$\overline{\text{PCIE\_RX0\_0}}$	I	PCI Express receive differential data 0 ch0
164	PCIE_TX0_1	O	PCI Express transmit differential data 1 ch0
165	$\overline{\text{PCIE\_TX0\_1}}$	O	PCI Express transmit differential data 1 ch0
166	PCIE_RX0_1	I	PCI Express receive differential data 1 ch0
167	$\overline{\text{PCIE\_RX0\_1}}$	I	PCI Express receive differential data 1 ch0
168	PCIE_TX0_2	O	PCI Express transmit differential data 2 ch0
169	$\overline{\text{PCIE\_TX0\_2}}$	O	PCI Express transmit differential data 2 ch0
170	PCIE_RX0_2	I	PCI Express receive differential data 2 ch0
171	$\overline{\text{PCIE\_RX0\_2}}$	I	PCI Express receive differential data 2 ch0
172	PCIE_TX0_3	O	PCI Express transmit differential data 3 ch0
173	$\overline{\text{PCIE\_TX0\_3}}$	O	PCI Express transmit differential data 3 ch0

No.	Pin name	I/O	Function
174	PCIE_RX0_3	I	PCI Express receive differential data 3 ch0
175	$\overline{\text{PCIE\_RX0\_3}}$	I	PCI Express receive differential data 3 ch0
176	PCIE_TX1_0	O	PCI Express transmit differential data 0 ch1
177	$\overline{\text{PCIE\_TX1\_0}}$	O	PCI Express transmit differential data 0 ch1
178	PCIE_RX1_0	I	PCI Express receive differential data 0 ch1
179	$\overline{\text{PCIE\_RX1\_0}}$	I	PCI Express receive differential data 0 ch1
180	USB_DP0	IO	USB ch0 differential data
181	USB_DM0	IO	USB ch0 differential data
182	USB_DP1	IO	USB ch1 differential data
183	USB_DM1	IO	USB ch1 differential data
184	USB_REXT	I	USB external register
185	USB_OVC0	I	USB ch0 over current control
186	USB_VBUS1_ OVC1	I	USB ch1 over current control / Voltage bus
187	USB_EXTAL	I	USB crystal resonator
188	USB_XTAL	O	USB crystal resonator
189	USB_PENC0	O	USB ch0 power enable control
190	USB_PENC1	O	USB ch1 power enable control
191	VTHREF	I	Temperature sensor
192	VTHSENSE	I	Temperature sensor
193	EXTAL	I	External input clock/crystal resonator
194	XTAL	O	Crystal resonator
195	$\overline{\text{PRESET}}$	I	Power on reset
196	NMI	I	Nonmaskable interrupt
197	$\overline{\text{IRL0}}$	I	IRL interrupt request 0
198	$\overline{\text{IRL1}}$	I	IRL interrupt request 1

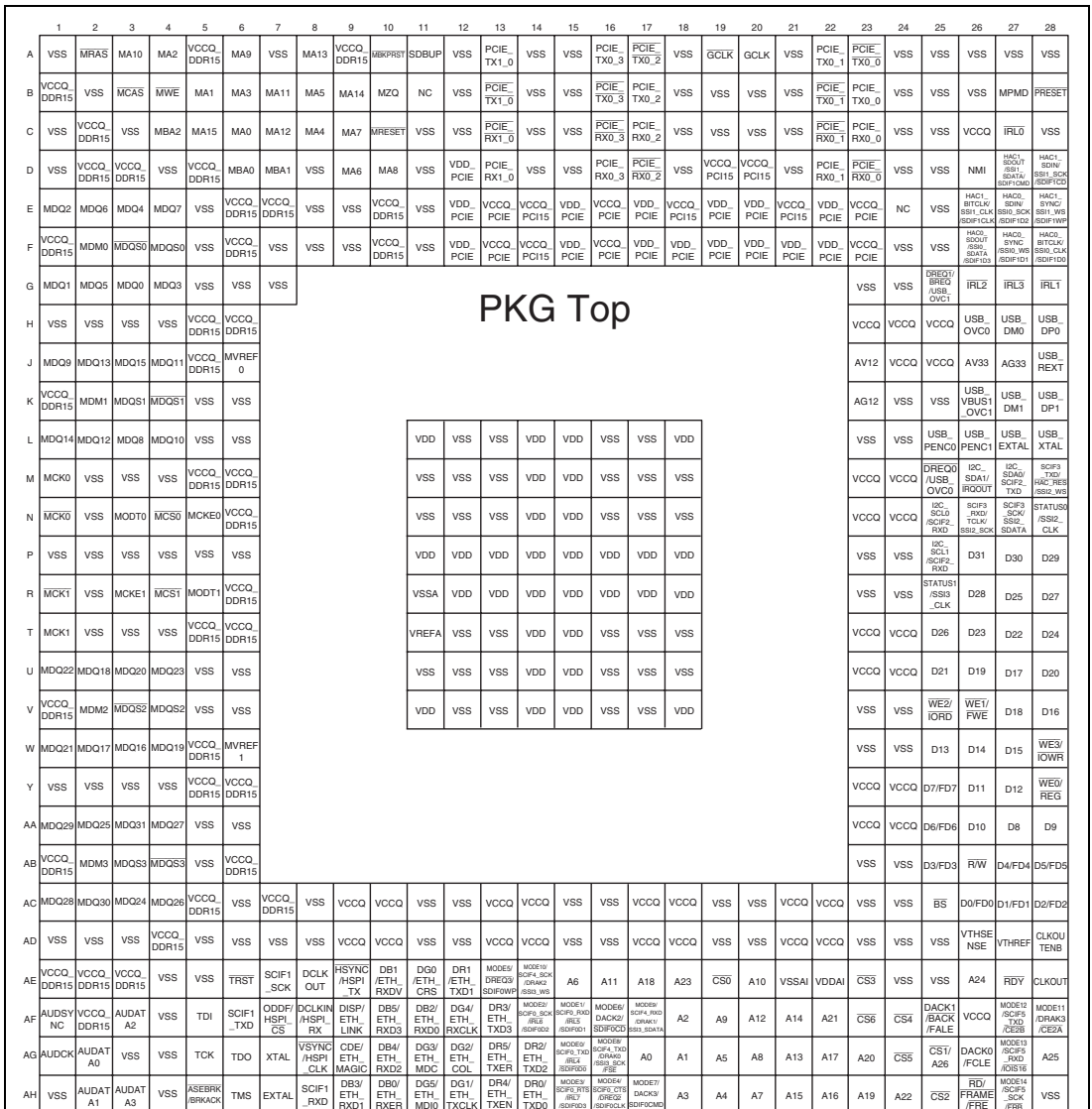
No.	Pin name	I/O	Function
199	$\overline{\text{IRL2}}$	I	IRL interrupt request 2
200	$\overline{\text{IRL3}}$	I	IRL interrupt request 3
201	TCK	I	H-UDI clock
202	TMS	I	H-UDI emulator
203	TDI	I	H-UDI data
204	TDO	O	H-UDI data
205	$\overline{\text{TRST}}$	I	H-UDI emulator
206	$\overline{\text{ASEBRK}}$/ BRKACK	I/O	H-UDI emulator
207	MPMD	I	H-UDI emulator mode
208	AUDCK	IO	H-UDI emulator clock
209	AUDSYNC	IO	H-UDI emulator
210	AUDATA0	IO	H-UDI emulator data 0
211	AUDATA1	IO	H-UDI emulator data 1
212	AUDATA2	IO	H-UDI emulator data 2
213	AUDATA3	IO	H-UDI emulator data 3
214	SCIF1_TXD	O	SCIF1 transmit data
215	SCIF1_RXD	I	SCIF1 receive data
216	SCIF1_SCK	IO	SCIF1 serial clock
217	DR0/ETH_TXD0	O/O	Digital red 0/ ether transmit data
218	DR1/ETH_TXD1	O/O	Digital red 1/ ether transmit data
219	DR2/ ETH_TXD2	O/O	Digital red 2/ ether transmit data
220	DR3/ ETH_TXD3	O/O	Digital red 3/ ether transmit data
221	DR4/ ETH_TX_EN	O/O	Digital red 4/ether transmit data enable
222	DR5/ ETH_TX_ER	O/O	Digital red 5/ ether transmit error
223	DG0/ETH_CRS	O/I	Digital green 0/ ether carrier sense
224	DG1/ ETH_TX_CLK	O/I	Digital green 1/ ether transmit clock
225	DG2/ETH_COL	O/I	Digital green 2/ ether collision detect

No.	Pin name	I/O	Function
226	DG3/ETH_MDC	O/O	Digital green 3/Ether managed data clock
227	DG4/ ETH_RX_CLK	O/I	Digital green 4/ Ether receive clock
228	DG5/ETH_MDIO	O/ IO	Digital green 5/ Ether managed data
229	DB0/ ETH_RX_ER	O/I	Digital blue 0/ Ether receive error
230	DB1/ ETH_RX_DV	O/I	Digital blue 1/ Ether receive data valid
231	DB2/ETH_RXD0	O/I	Digital blue 2/ Ether receive data
232	DB3/ETH_RXD1	O/I	Digital blue 3/ Ether receive data
233	DB4/ETH_RXD2	O/I	Digital blue 4/ Ether receive data
234	DB5/ETH_RXD3	O/I	Digital blue 5/ Ether receive data
235	DISP/ETH_LINK	O/I	DU display/ Ether PHY Link
236	CDE/ ETH_MAGIC	O/O	DU color detect/Ether magic packet detect
237	ODDF/ $\overline{\text{HSPI\_CS}}$	IO/ IO	DU OddEven frame/HSPI chip selection
238	$\overline{\text{VSYNC}}$/ HSPI_CLK	IO/ IO	DU VSYNC/HSPI serial clock
239	$\overline{\text{HSYNC}}$/ HSPI_TX	IO/ O	DU HSYNC/HSPI transmit data
240	DCLKIN/ HSPI_RX	I/I	DU Dot clock in/HSPI receive data
241	DCLKOUT	O	DU dot clock out
242	$\overline{\text{DREQ0}}$/ USB_OVC0	I	DMA channel 0 request/USB ch0 over current control
243	$\overline{\text{DREQ1}}$/$\overline{\text{BREQ}}$/ USB_OVC1	I/I/I	DMA channel 1 request/Bus request/USB ch1 over current control
244	DACK0/FCLE	O/O	DMA acknowledgment 0/NAND flash command latch enable
245	DACK1/$\overline{\text{BACK}}$/ FALE	O/O /O	DMA acknowledgment 1/Bus acknowledgement/ NAND flash address latch enable
246	I2C_SCL0/ SCIF2_RXD	IO/I	I2C0 clock/SCIF2 receive data
247	I2C_SDA0/ SCIF2_TXD	IO/ O	I2C0 data/SCIF2 transmit data
248	I2C_SCL1/ SCIF2_SCK	IO/ IO	I2C1clock/SCIF2 serial clock
249	I2C_SDA1/ $\overline{\text{IRQOUT}}$	IO/ O	I2C1data/Interrupt request output
250	HAC0_BITCLK/ SSI0_CLK/ SDIF1D0	I/I/ IO	HAC0 bit clock/SSI0 serial bit clock/Data bus SD1 data bus signal 0
251	HAC0_SYNC/ SSI0_WS/ SDIF1D1	O/ IO/ IO	HAC0 flame synchronous/SSI0 word select/Data bus SD1 data bus signal 1
252	HAC0_SDIN/ SSI0_SCK/ SDIF1D2	I/IO/ IO	HAC0 serial data incoming to Rx frame/SSI0 serial bit clock/Data bus SD1 data bus signal 2
253	HAC0_SDOUT/ SSI0_SDATA/ SDIF1D3	O/ IO/ IO	HAC0 serial data/SSI0 serial data/Data bus SD1 data bus signal 3
254	HAC1_BITCLK/ SSI1_CLK/ SDIF1CLK	I/I/O	HAC1 bit clock/SSI1 serial bit clock/Output Clock SD1 clock output pin
255	HAC1_SYNC/ SSI1_WS/ SDIF1WP	O/ IO/ IO	HAC1 synchronous/SSI1 serial bit clock/SD1 write-protection signal
256	HAC1_SDIN/ SSI1_SCK/ $\overline{\text{SDIF1CD}}$	I/IO/ I	HAC1 serial data/SSI1 serial data/SD1 card detection signal
257	HAC1_SDOUT/ SSI1_SDATA/ SDIF1CMD	O/ IO/ IO	HAC1 serial data/SSI1 serial data/ SD1command output and response input signal
258	SCIF3_TXD/ $\overline{\text{HAC\_RES}}$/ SSI2_WS	O/O /IO	SCIF3 transmit data/SSI2 word select/HAC Reset

No.	Pin name	I/O	Function
259	SCIF3_RXD/ TCLK/SSI2_SCK	O/ IO/ IO	SCIF3 receive data/SSI2 serial clock/TMU clock
260	SCIF3_SCK/ SSI2_SDATA	IO/ IO	SCIF3 synchrounous/ SSI2 serial data
261	STATUS0/ SSI2_CLK	O/I	Status 0/ SSI2 serial bit clock
262	STATUS1/ SSI3_CLK	O/I	Status 1/ SSI3 serial bit clock
263	MODE0/ SCIF0_TXD/ IRL4/SDIF0D0	I/O/I /IO	Mode 0/IRL interrupt request 4/SCIF0 transmit data/Data bus SD0 data bus signals 0
264	MODE1/ SCIF0_RXD/ IRL5/SDIF0D1	I/I/I/ IO	Mode 1/IRL interrupt request 5/SCIF0 receive data/Data bus SD0 data bus signals 1
265	MODE2/ SCIF0_SCK/ IRL6/SDIF0D2	I/IO/ I/IO	Mode 2/IRL interrupt request 6/SCIF0 serial clock/Data bus SD0 data bus signals 2
266	MODE3/ SCIF0_RTS/ IRL7/ SDIF0D3	I/IO/ I/IO	Mode 3/IRL interrupt request 7/SCIF0 modem control/Data bus SD0 data bus signals 3
267	MODE4/ SCIF0_CTS/ DREQ2/ SDIF0CLK	I/IO/ I/O	Mode 4/DMA channel 2 request/SCIF0 modem control/Output clock SD0 clock output pin
268	MODE5/ DREQ3/ SDIF0WP	I/I/I	Mode 5/DMA channel 3 request/SD0 write-protection signal
269	MODE6/DACK2/ SDIF0CD	I/O/I	Mode 6/DMA acknowledgment 2/SD0 card detection signal
270	MODE7/DACK3/ SDIF0CMD	I/O/ IO	Mode 7/DMA acknowledgment 3/SD0 command output and response input signal
271	MODE8/ SCIF4_TXD/ DRAK0/ SSI3_SCK/FSE	I/O/ O/ IO/ O	Mode 8/SCIF4 transmit data/DMA transfer request acknowledge 0/SSI3 serial bit clock/NAND FSE
272	MODE9/SCIF4_ RXD/DRAK1/SSI 3_SDATA	I/I/O /IO	Mode 9/SCIF4 receive data/DMA transfer request acknowledge
			1/SSI3 serial data
273	MODE10/ SCIF4_SCK/ DRAK2/ SSI3_WS	I/IO/ O/ IO	Mode 10/SCIF4 synchronous/DMA transfer request acknowledge 2/SSI3 word select
274	MODE11/ DRAK3/CE2A	I/O/ O	Mode 11/DMA transfer request acknowledge 3/PCMCIA CE2A
275	MODE12/ SCIF5_TXD/ CE2B	I/O/ O	Mode 12/SCIF5 transmit data/PCMCIA CE2B
276	MODE13/ SCIF5_RXD/ IOIS16	I/I/O	Mode 13/SCIF5 receive data/PCMCIA IOIS16
277	MODE14/ SCIF5_SCK/FRB	I/IO/ I	Mode 14/SCIF5 synchronous/NAND flash ready or busy

1.4 Pin Arrangement

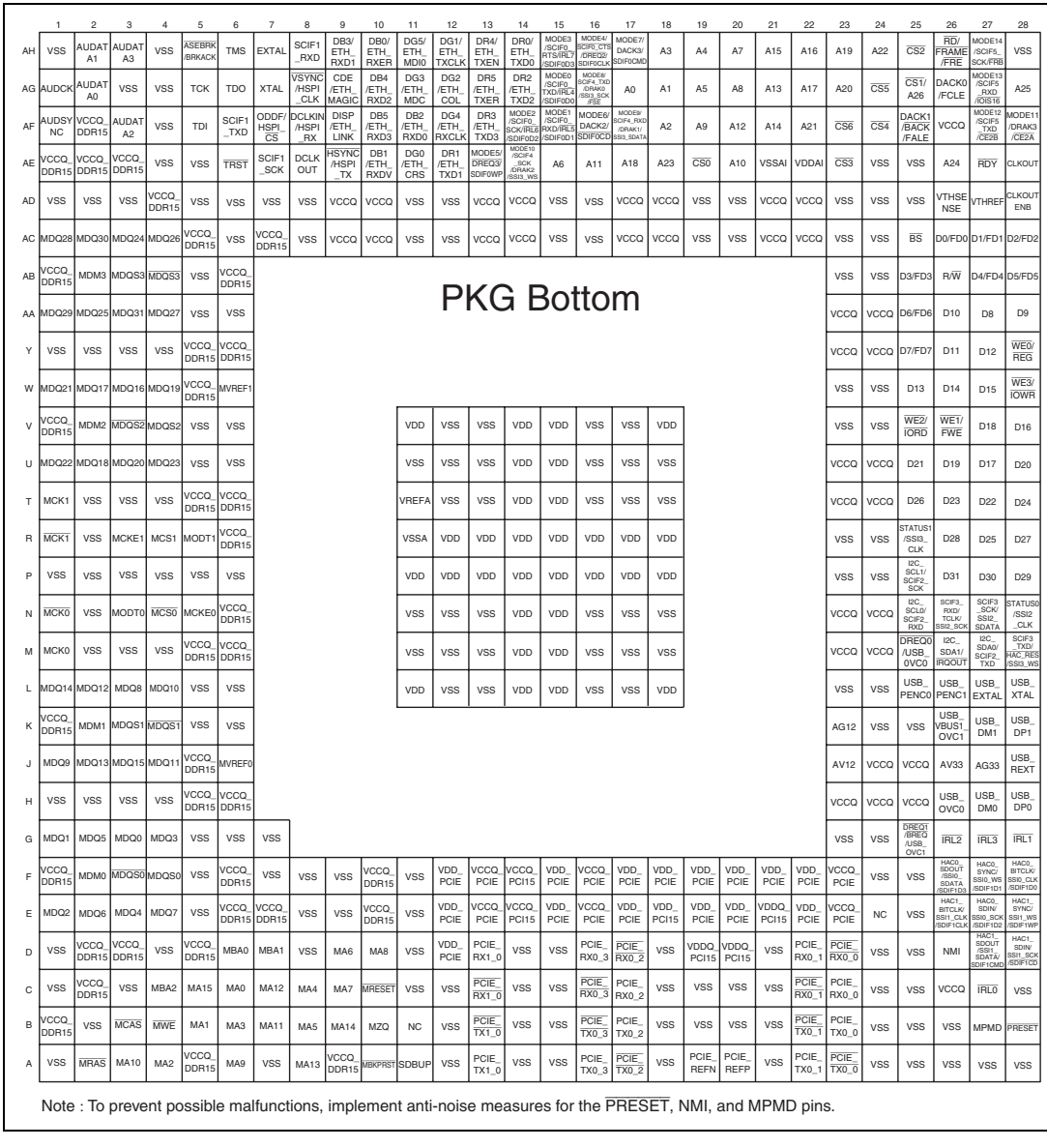
Package: 593-pin FC-BGA, 25 mm × 25 mm, ball pitch: 0.8 mm



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
A	VSS	MRAS	MA10	MA2	VCCQ_DDR15	MA9	VSS	MA13	VCCQ_DDR15	MBKPRST	SDBUP	VSS	PCIE_TX1_0	VSS	VSS	PCIE_TX0_3	PCIE_TX0_2	VSS	GCLK	GCLK	VSS	PCIE_TX0_1	PCIE_TX0_0	VSS	VSS	VSS	VSS	VSS
B	VCCQ_DDR15	VSS	MCAS	MWE	MA1	MA3	MA11	MA5	MA14	MZQ	NC	VSS	PCIE_TX1_0	VSS	VSS	PCIE_TX0_3	PCIE_TX0_2	VSS	VSS	VSS	VSS	PCIE_TX0_1	PCIE_TX0_0	VSS	VSS	VSS	MPMD	PRESET
C	VSS	VCCQ_DDR15	VSS	MBA2	MA15	MA0	MA12	MA4	MA7	MRESET	VSS	VSS	PCIE_RX1_0	VSS	VSS	PCIE_RX0_3	PCIE_RX0_2	VSS	VSS	VSS	VSS	PCIE_RX0_1	PCIE_RX0_0	VSS	VSS	VCCQ	IRL0	VSS
D	VSS	VCCQ_DDR15	VCCQ_DDR15	VSS	VCCQ_DDR15	MBA0	MBA1	VSS	MA6	MA8	VSS	VDD_PCIE	PCIE_RX1_0	VSS	VSS	PCIE_RX0_3	PCIE_RX0_2	VSS	VCCQ_PCI15	VCCQ_PCI15	VSS	PCIE_RX0_1	PCIE_RX0_0	VSS	VSS	NMI	HAC1_SDOUT/SSI1_SDATA/SDIF1CMD	HAC1_SDIN/SSI1_SCK/SDIF1CD
E	MDQ2	MDQ6	MDQ4	MDQ7	VSS	VCCQ_DDR15	VCCQ_DDR15	VSS	VSS	VCCQ_DDR15	VSS	VDD_PCIE	VCCQ_PCIE	VCCQ_PCI15	VDD_PCIE	VCCQ_PCIE	VDD_PCIE	VCCQ_PCI15	VDD_PCIE	VDD_PCIE	VCCQ_PCI15	VDD_PCIE	VCCQ_PCIE	NC	VSS	HAC1_BITCLK/SSI1_CLK/SDIF1CLK	HAC2_SDIN/SSI0_SCK/SDIF1D2	HAC1_SYNC/SSI1_WS/SDIF1WP
F	VCCQ_DDR15	MDM0	MDQS0	MDQS0	VSS	VCCQ_DDR15	VSS	VSS	VSS	VCCQ_DDR15	VSS	VDD_PCIE	VCCQ_PCIE	VCCQ_PCI15	VDD_PCIE	VCCQ_PCIE	VDD_PCIE	VDD_PCIE	VDD_PCIE	VDD_PCIE	VDD_PCIE	VDD_PCIE	VCCQ_PCIE	VSS	VSS	HAC0_SDOUT/SSI0_SDATA/SDIF1D3	HAC0_SYNC/SSI0_WS/SDIF1D1	HAC0_BITCLK/SSI0_CLK/SDIF1D0
G	MDQ1	MDQ5	MDQ0	MDQ3	VSS	VSS	VSS																VSS	VSS	DREQ7/BREQ/USB_OVC1	IRL2	IRL3	IRL1
H	VSS	VSS	VSS	VSS	VCCQ_DDR15	VCCQ_DDR15																	VCCQ	VCCQ	VCCQ	USB_OVC0	USB_DM0	USB_DP0
J	MDQ9	MDQ13	MDQ15	MDQ11	VCCQ_DDR15	MVREF0																	AV12	VCCQ	VCCQ	AV33	AG33	USB_REXT
K	VCCQ_DDR15	MDM1	MDQS1	MDQS1	VSS	VSS																	AG12	VSS	VSS	USB_VBUS1_OVC1	USB_DM1	USB_DP1
L	MDQ14	MDQ12	MDQ8	MDQ10	VSS	VSS																	VSS	VSS	USB_PENC0	USB_PENC1	USB_EXTAL	USB_XTAL
M	MCK0	VSS	VSS	VSS	VCCQ_DDR15	VCCQ_DDR15																	VCCQ	VCCQ	DREQ0/USB_OVC0	I2C_SDA1/IRQOUT	I2C_SDA0/SCIF2_TXD	SCIF3_TXD/HAC_RES/SSI2_WS
N	MCK0	VSS	MODT0	MCS0	MCKE0	VCCQ_DDR15																	VCCQ	VCCQ	I2C_SCL0/SCIF2_RXD	SCIF3_RXD/TCLK/SSI2_SCK	SCIF3_SCK/SSI2_SDATA	STATUS0/SSI2_CLK
P	VSS	VSS	VSS	VSS	VSS	VSS																	VSS	VSS	I2C_SCL1/SCIF2_RXD	D31	D30	D29
R	MCK1	VSS	MCKE1	MCS1	MODT1	VCCQ_DDR15																	VSS	VSS	STATUS1/SSI3_CLK	D28	D25	D27
T	MCK1	VSS	VSS	VSS	VCCQ_DDR15	VCCQ_DDR15																	VCCQ	VCCQ	D26	D23	D22	D24
U	MDQ22	MDQ18	MDQ20	MDQ23	VSS	VSS																	VCCQ	VCCQ	D21	D19	D17	D20
V	VCCQ_DDR15	MDM2	MDQS2	MDQS2	VSS	VSS																	VSS	VSS	WE2/IORD	WE1/FWE	D18	D16
W	MDQ21	MDQ17	MDQ16	MDQ19	VCCQ_DDR15	MVREF1																	VSS	VSS	D13	D14	D15	WE3/IOWR
Y	VSS	VSS	VSS	VSS	VCCQ_DDR15	VCCQ_DDR15																	VCCQ	VCCQ	D7/FD7	D11	D12	WE0/REG
AA	MDQ29	MDQ25	MDQ31	MDQ27	VSS	VSS																	VCCQ	VCCQ	D6/FD6	D10	D8	D9
AB	VCCQ_DDR15	MDM3	MDQS3	MDQS3	VSS	VCCQ_DDR15																	VSS	VSS	D3/FD3	R/W	D4/FD4	D5/FD5
AC	MDQ28	MDQ30	MDQ24	MDQ26	VCCQ_DDR15	VSS	VCCQ_DDR15	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	BS	D0/FD0	D1/FD1	D2/FD2
AD	VSS	VSS	VSS	VCCQ_DDR15	VSS	VSS	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VSS	VTHSENSE	VTHREF	CLKOUTENB
AE	VCCQ_DDR15	VCCQ_DDR15	VCCQ_DDR15	VSS	VSS	TRST	SCIF1_SCK	DCLKOUT	HSYNC/HSPI_TX	DB1/ETH_RXDV	DG0/ETH_CRS	DR1/ETH_TXD1	MODE5/DREQ3/SDIF0WP	MODE10/SCIF4_SCK/DRAK2/SSI3_WS	A6	A11	A18	A23	CS0	A10	VSSAI	VDDAI	CS3	VSS	VSS	A24	RDY	CLKOUT
AF	AUDSYNC	VCCQ_DDR15	AUDATA2	VSS	TDI	SCIF1_TXD	ODDF/HSPI_CS	DCLKIN/HSPI_RX	DISP/ETH_LINK	DB5/ETH_RXD3	DB2/ETH_RXD0	DG4/ETH_RXCLK	DR3/ETH_TXD3	MODE2/SCIF3_SCK/IRL3/SDIF0D2	MODE1/SCIF3_RXD/IRL2/SDIF0D1	MODE6/DACK2/SDIF0CD	MODE9/SCIF4_RXD/DRAK1/SSI3_SDATA	A2	A9	A12	A14	A21	CS6	CS4	DACK1/BACK/FALE	VCCQ	MODE12/SCIF5_TXD/CE2B	MODE11/DRAK3/CE2A
AG	AUDCK	AUDATA0	VSS	VSS	TCK	TDO	XTAL	VSYNC/HSPI_CLK	CDE/ETH_MAGIC	DB4/ETH_RXD2	DG3/ETH_MDC	DG2/ETH_COL	DR5/ETH_TXER	DR2/ETH_TXD2	MODE8/SCIF3_TXD/IRL4/SDIF0D0	MODE4/SCIF2_RTS/IRL1/SDIF0D3	A0	A1	A5	A8	A13	A17	A20	CS5	CS1/A26	DACK0/FCLE	MODE13/SCIF5_RXD/IOIS16	A25
AH	VSS	AUDATA1	AUDATA3	VSS	ASEBRK/BRKACK	TMS	EXTAL	SCIF1_RXD	DB3/ETH_RXD1	DB0/ETH_RXER	DG5/ETH_MDIO	DG1/ETH_TXCLK	DR4/ETH_TXEN	DR0/ETH_TXD0	MODE3/SCIF3_RTS/IRL1/SDIF0D3	MODE7/SCIF2_CTS/DREQ2/SDIF0CLK	MODE7/DACK3/SDIF0CMD	A3	A4	A7	A15	A16	A19	A22	CS2	RD/FRAME/FRE	MODE14/SCIF5_SCK/FRB	VSS

Note : To prevent possible malfunctions, implement anti-noise measures for the PRESET, NMI, and MPMD pins.

Figure 1.2 SH7786 Pin Arrangement (Top View)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
AH	VSS	AUDAT A1	AUDAT A3	VSS	ASEBRK /BRKACK	TMS	EXTAL	SCIF1 _RXD	DB3/ ETH_ RXD1	DB0/ ETH_ RXER	DG5/ ETH_ MDI0	DG1/ ETH_ TXCLK	DR4/ ETH_ TXEN	DR0/ ETH_ TXD0	MODE3 /SCIF0_ RTS/IRL7 /SDIF0D3	MODE4/ SCIF0_CTS /DREQ2/ SDIF0CLK	MODE7/ DACK3/ SDIF0CMD	A3	A4	A7	A15	A16	A19	A22	CS2	RD/ FRAME /FRE	MODE14 /SCIF5_ SCK/FRB	VSS
AG	AUDCK	AUDAT A0	VSS	VSS	TCK	TDO	XTAL	VSYNC /HSPI _CLK	CDE /ETH_ MAGIC	DB4 /ETH_ RXD2	DG3 /ETH_ MDC	DG2 /ETH_ COL	DR5 /ETH_ TXER	DR2 /ETH_ TXD2	MODE0 /SCIF0_ TXD/IRL4 /SDIF0D0	MODE5 /SCIF4_ TXD /DRAK3 /SSI3_SCK /FSE	A0	A1	A5	A8	A13	A17	A20	CS5	CS1/ A26	DACK0 /FCLE	MODE13 /SCIF5 _RXD /IOIS16	A25
AF	AUDSY NC	VCCQ_ DDR15	AUDAT A2	VSS	TDI	SCIF1 _TXD	ODDF/ HSPI_ CS	DCLKIN /HSPI _RX	DISP /ETH_ LINK	DB5 /ETH_ RXD3	DB2 /ETH_ RXD0	DG4 /ETH_ RXCLK	DR3 /ETH_ TXD3	MODE2 /SCIF0_ SCK/IRL6 /SDIF0D2	MODE1 /SCIF0_ RXD/IRL5 /SDIF0D1	MODE6/ DACK2/ SDIF0CD	MODE8/ SCIF4_RXD /DRAK1/ SSI3_SDATA	A2	A9	A12	A14	A21	CS6	CS4	DACK1 /BACK /FALE	VCCQ	MODE12 /SCIF5 _TXD /CE2B	MODE11 /DRAK3 /CE2A
AE	VCCQ_ DDR15	VCCQ_ DDR15	VCCQ_ DDR15	VSS	VSS	TRST	SCIF1 _SCK	DCLK OUT	HSYNC /HSPI _TX	DB1 /ETH_ RXDV	DG0 /ETH_ CRS	DR1 /ETH_ TXD1	MODE5/ DREQ3/ SDIF0WP	MODE10 /SCIF4 _SCK /DRAK2 /SSI3_WS	A6	A11	A18	A23	CS0	A10	VSSAI	VDDAI	CS3	VSS	VSS	A24	RDY	CLKOUT
AD	VSS	VSS	VSS	VCCQ_ DDR15	VSS	VSS	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VSS	VTHSE NSE	VTHREF	CLKOUT ENB
AC	MDQ28	MDQ30	MDQ24	MDQ26	VCCQ_ DDR15	VSS	VCCQ_ DDR15	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	VCCQ	VCCQ	VSS	VSS	BS	D0/FD0	D1/FD1	D2/FD2
AB	VCCQ_ DDR15	MDM3	MDQS3	MDQS3	VSS	VCCQ_ DDR15																	VSS	VSS	D3/FD3	R/W	D4/FD4	D5/FD5
AA	MDQ29	MDQ25	MDQ31	MDQ27	VSS	VSS																	VCCQ	VCCQ	D6/FD6	D10	D8	D9
Y	VSS	VSS	VSS	VSS	VCCQ_ DDR15	VCCQ_ DDR15																	VCCQ	VCCQ	D7/FD7	D11	D12	WE0/ REG
W	MDQ21	MDQ17	MDQ16	MDQ19	VCCQ_ DDR15	MVREF1																	VSS	VSS	D13	D14	D15	WE3/ IOWR
V	VCCQ_ DDR15	MDM2	MDQS2	MDQS2	VSS	VSS			VDD	VSS	VSS	VDD	VDD	VSS	VSS	VDD							VSS	VSS	WE2/ IORD	WE1/ FWE	D18	D16
U	MDQ22	MDQ18	MDQ20	MDQ23	VSS	VSS			VSS	VSS	VSS	VDD	VDD	VSS	VSS	VSS							VCCQ	VCCQ	D21	D19	D17	D20
T	MCK1	VSS	VSS	VSS	VCCQ_ DDR15	VCCQ_ DDR15			VREFA	VSS	VSS	VDD	VDD	VSS	VSS	VSS							VCCQ	VCCQ	D26	D23	D22	D24
R	MCK1	VSS	MCKE1	MCS1	MODT1	VCCQ_ DDR15			VSSA	VDD	VDD	VDD	VDD	VDD	VDD	VDD							VSS	VSS	STATUS1 /SSI3_ CLK	D28	D25	D27
P	VSS	VSS	VSS	VSS	VSS	VSS			VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD							VSS	VSS	I2C_ SCL1/ SCIF2_ SCK	D31	D30	D29
N	MCK0	VSS	MODT0	MCS0	MCKE0	VCCQ_ DDR15			VSS	VSS	VSS	VDD	VDD	VSS	VSS	VSS							VCCQ	VCCQ	I2C_ SCL0/ SCIF2_ RXD	SCIF3_ RXD/ TCLK/ SSI2_SCK	SCIF3 _SCK/ SSI2_ SDATA	STATUS0 /SSI2 _CLK
M	MCK0	VSS	VSS	VSS	VCCQ_ DDR15	VCCQ_ DDR15			VSS	VSS	VSS	VDD	VDD	VSS	VSS	VSS							VCCQ	VCCQ	DREQ0 /USB_ OVC0	I2C_ SDA1/ IRQOUT	I2C_ SDA0/ SCIF2_ TXD	SCIF3 _TXD/ HAC_RES /SSI3_WS
L	MDQ14	MDQ12	MDQ8	MDQ10	VSS	VSS			VDD	VSS	VSS	VDD	VDD	VSS	VSS	VDD							VSS	VSS	USB_ PENC0	USB_ PENC1	USB_ EXTAL	USB_ XTAL
K	VCCQ_ DDR15	MDM1	MDQS1	MDQS1	VSS	VSS																	AG12	VSS	VSS	USB_ VBUS1_ OVC1	USB_ DM1	USB_ DP1
J	MDQ9	MDQ13	MDQ15	MDQ11	VCCQ_ DDR15	MVREF0																	AV12	VCCQ	VCCQ	AV33	AG33	USB_ REXT
H	VSS	VSS	VSS	VSS	VCCQ_ DDR15	VCCQ_ DDR15																	VCCQ	VCCQ	VCCQ	USB_ OVC0	USB_ DM0	USB_ DP0
G	MDQ1	MDQ5	MDQ0	MDQ3	VSS	VSS	VSS																VSS	VSS	DREQ1 /BREQ /USB_ OVC1	IRL2	IRL3	IRL1
F	VCCQ_ DDR15	MDM0	MDQS0	MDQS0	VSS	VCCQ_ DDR15	VSS	VSS	VSS	VCCQ_ DDR15	VSS	VDD_ PCIE	VCCQ_ PCIE	VCCQ_ PCI15	VDD_ PCIE	VCCQ_ PCIE	VDD_ PCIE	VDD_ PCIE	VDD_ PCIE	VDD_ PCIE	VDD_ PCIE	VDD_ PCIE	VCCQ_ PCIE	VSS	VSS	HAC0_ SDOUT /SSI0_ SDATA /SDIF1D3	HAC0_ SYNC/ SSI0_ WS /SDIF1D1	HAC0_ BITCLK/ SSI0_CLK /SDIF1D0
E	MDQ2	MDQ6	MDQ4	MDQ7	VSS	VCCQ_ DDR15	VCCQ_ DDR15	VSS	VSS	VCCQ_ DDR15	VSS	VDD_ PCIE	VCCQ_ PCIE	VCCQ_ PCI15	VDD_ PCIE	VCCQ_ PCIE	VDD_ PCIE	VDD_ PCI15	VDD_ PCIE	VDD_ PCIE	VDDQ_ PCI15	VDD_ PCIE	VCCQ_ PCIE	NC	VSS	HAC1_ BITCLK/ SSI1_CLK /SDIF1CLK	HAC0_ SDIN/ SSI0_SCK /SDIF1D2	HAC1_ SYNC/ SSI1_WS /SDIF1WP
D	VSS	VCCQ_ DDR15	VCCQ_ DDR15	VSS	VCCQ_ DDR15	MBA0	MBA1	VSS	MA6	MA8	VSS	VDD_ PCIE	PCIE_ RX1_0	VSS	VSS	PCIE_ RX0_3	PCIE_ RX0_2	VSS	VDDQ_ PCI15	VDDQ_ PCI15	VSS	PCIE_ RX0_1	PCIE_ RX0_0	VSS	VSS	NMI	HAC1_ SDOUT /SSI1_ SDATA/ SDIF1CMD	HAC1_ SDIN/ SSI1_SCK /SDIF1CD
C	VSS	VCCQ_ DDR15	VSS	MBA2	MA15	MA0	MA12	MA4	MA7	MRESET	VSS	VSS	PCIE_ RX1_0	VSS	VSS	PCIE_ RX0_3	PCIE_ RX0_2	VSS	VSS	VSS	VSS	PCIE_ RX0_1	PCIE_ RX0_0	VSS	VSS	VCCQ	IRL0	VSS	
B	VCCQ_ DDR15	VSS	MCAS	MWE	MA1	MA3	MA11	MA5	MA14	MZQ	NC	VSS	PCIE_ TX1_0	VSS	VSS	PCIE_ TX0_3	PCIE_ TX0_2	VSS	VSS	VSS	VSS	PCIE_ TX0_1	PCIE_ TX0_0	VSS	VSS	VSS	MPMD	PRESET	
A	VSS	MRAS	MA10	MA2	VCCQ_ DDR15	MA9	VSS	MA13	VCCQ_ DDR15	MBKPRST	SDBUP	VSS	PCIE_ TX1_0	VSS	VSS	PCIE_ TX0_3	PCIE_ TX0_2	VSS	PCIE_ REFN	PCIE_ REFP	VSS	PCIE_ TX0_1	PCIE_ TX0_0	VSS	VSS	VSS	VSS	VSS	

Note : To prevent possible malfunctions, implement anti-noise measures for the PRESET, NMI, and MPMD pins.

Figure 1.3 SH7786 Pin Arrangement (Bottom View)

1.5 Address map

The SH7786 supports 32-bit virtual address space, and supports both 29-bit and 32-bit physical address spaces (29-bit or 32-bit address space is selected by the mode pin settings). For details of the virtual address space and the physical address spaces, see section 7, Memory Management Unit.

Figure 1.4 shows the memory address map of the SH7786. The memory address map of the SH7786 is decided by the MM_SEL bit of MMSELR in the LBSC, the spaces from CS2 to CS5 are changed.

For details of internal memory of the CPU0/1 (IL Memory/OL Memory), see section 9, On-Chip Memory.

		000	001	010	011	100	101	110	111
H'00000000 to H'03FFFFFF	CS0	LBSC							
H'04000000 to H'07FFFFFF	CS1	LBSC							
H'08000000 to H'0BFFFFFF	CS2	DBSC (2)							
H'0C000000 to H'0FFFFFFF	CS3	DBSC (3)							
H'10000000 to H'13FFFFFF	CS4	LBSC	PCI Express 0ch		DBSC (4)	LBSC	PCI Express 0ch		DBSC (4)
H'14000000 to H'147FFFFF	CS5	LBSC	LBSC	CPU0 IL Memory/ OL Memory	DBSC (5)	LBSC	LBSC	CPU0 IL Memory/ OL Memory	DBSC (5)
H'14800000 to H'14FFFFFF				CPU1 IL Memory/ OL Memory				CPU1 IL Memory/ OL Memory	
H'15000000 to H'17FFFFFF				RESERVED				RESERVED	
H'18000000 to H'1BFFFFFF	CS6	LBSC							
H'1C000000 to H'1FFFFFFF		RESERVED							
H'20000000 to H'2FFFFFFF		PCI Express 2ch							
H'30000000 to H'3FFFFFFF		PCI Express 1ch							
H'40000000 to H'43FFFFFF		DBSC (0)							
H'44000000 to H'47FFFFFF		DBSC (1)							
H'48000000 to H'4BFFFFFF		DBSC (2)							
H'4C000000 to H'4FFFFFFF		DBSC (3)							
H'50000000 to H'53FFFFFF		DBSC (4)							
H'54000000 to H'57FFFFFF		DBSC (5)							
H'58000000 to H'5BFFFFFF		DBSC (6)							
H'5C000000 to H'5FFFFFFF		DBSC (7)							
H'60000000 to H'63FFFFFF		DBSC (8)							
H'64000000 to H'67FFFFFF		DBSC (9)							
H'68000000 to H'6BFFFFFF		DBSC (10)							
H'6C000000 to H'6FFFFFFF		DBSC (11)							
H'70000000 to H'73FFFFFF		DBSC (12)							
H'74000000 to H'77FFFFFF		DBSC (13)							
H'78000000 to H'7BFFFFFF		DBSC (14)							
H'7C000000 to H'7FFFFFFF		DBSC (15)							
H'80000000 to H'83FFFFFF		PCI Express 2ch							DBSC (16)
H'84000000 to H'87FFFFFF									DBSC (17)
H'88000000 to H'8BFFFFFF									DBSC (18)
H'8C000000 to H'8FFFFFFF									DBSC (19)
H'90000000 to H'93FFFFFF									DBSC (20)
H'94000000 to H'97FFFFFF									DBSC (21)
H'98000000 to H'9BFFFFFF									DBSC (22)
H'9C000000 to H'9FFFFFFF									DBSC (23)
H'A0000000 to H'A3FFFFFF		PCI Express 1ch							DBSC (24)
H'A4000000 to H'A7FFFFFF									DBSC (25)
H'A8000000 to H'ABFFFFFF									DBSC (26)
H'AC000000 to H'AFFFFFFF									DBSC (27)
H'B0000000 to H'B3FFFFFF									DBSC (28)
H'B4000000 to H'B7FFFFFF									DBSC (29)
H'B8000000 to H'BBFFFFFF									DBSC (30)
H'BC000000 to H'BFFFFFFF									DBSC (31)
H'C0000000 to H'DFFFFFFF		PCI Express 0ch							
H'E0000000 to H'E3FFFFFF		CPU Internal Modules							
H'E4000000 to H'E4FFFFFF		Shared Memory *							
H'E5000000 to H'F7FFFFFF		CPU Internal Modules							
H'E8000000 to H'E87FFFFF		CPU0 IL Memory/OL Memory							
H'E8800000 to H'E8FFFFFF		CPU1 IL Memory/OL Memory							
H'E9000000 to H'FBFFFFFF		CPU Internal Modules							
H'FC000000 to H'FFFFFFFF		Peripheral Modules							

Note: * When the secondary cache is used as shared memory, the area from H'E400 0000 to H'E403 FFFF (256 KB) is in RAM.

Figure 1.4 Memory Address Map of SH7786

Section 2 Programming Model

The programming model of this LSI is explained in this section. This LSI has registers and data formats as shown below.

2.1 Data Formats

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

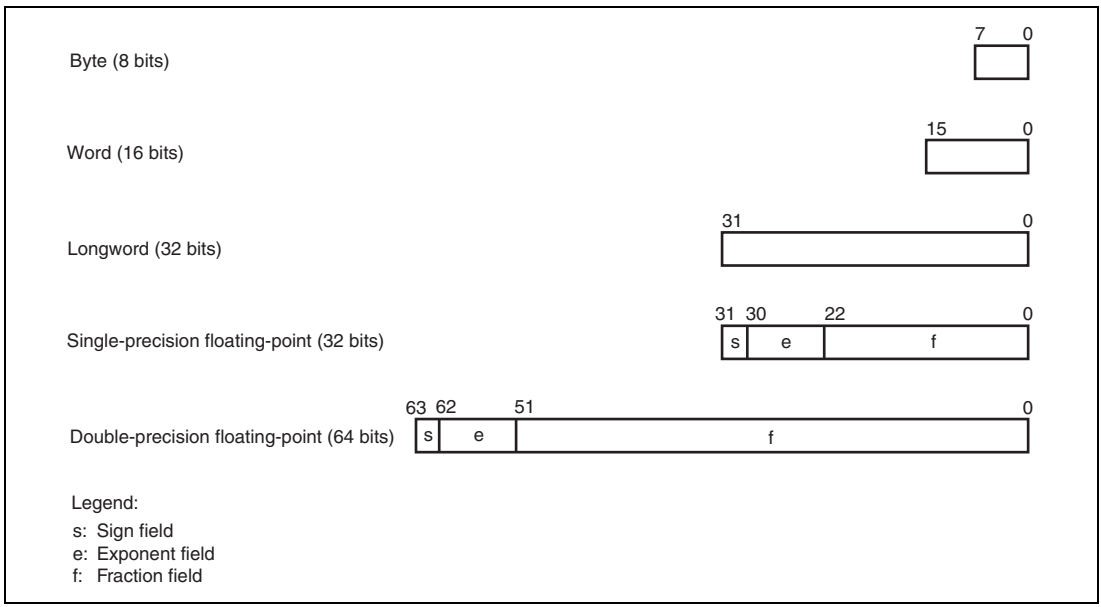


Figure 2.1 Data Formats

2.2 Register Descriptions

2.2.1 Privileged Mode and Banks

(1) Processing Modes

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

(2) General Registers

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

- Privileged mode

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

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

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

(3) Control Registers

Control registers comprise the global base register (GBR) and status register (SR), which can be accessed in both processing modes, and the saved status register (SSR), saved program counter (SPC), vector base register (VBR), saved general register 15 (SGR), and debug base register

(DBR), which can only be accessed in privileged mode. Some bits of the status register (such as the RB bit) can only be accessed in privileged mode.

(4) System Registers

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

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

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

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

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

Register values after a reset are shown in table 2.1.

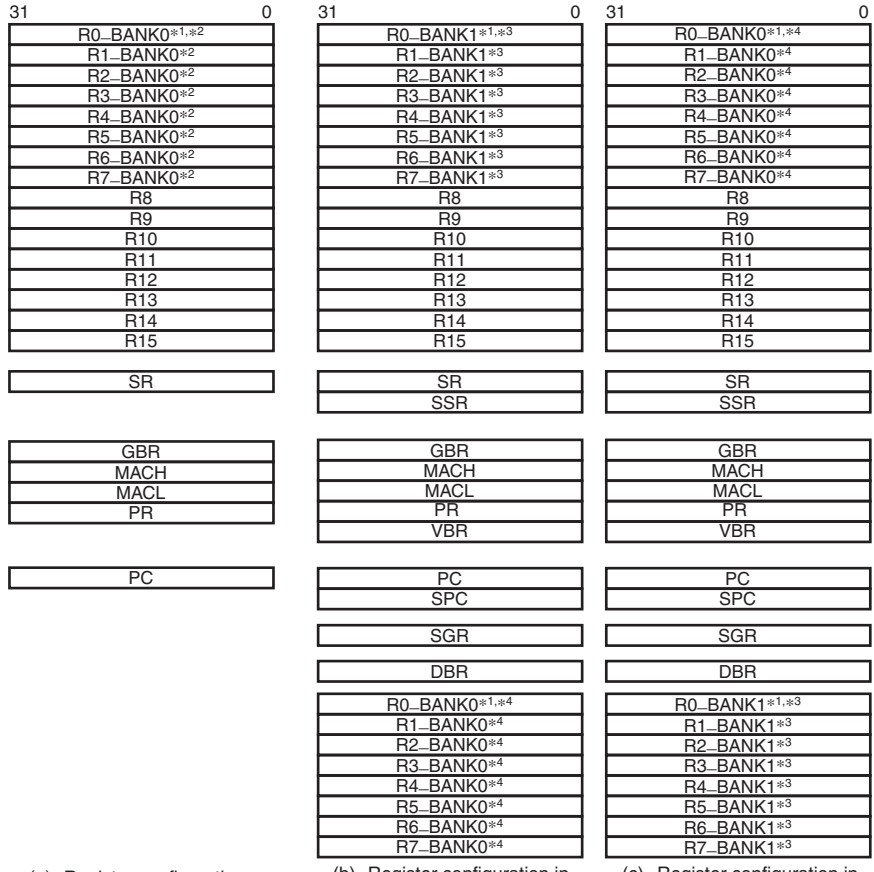
Table 2.1 Initial Register Values

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

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

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

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



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

Figure 2.2 CPU Register Configuration in Each Processing Mode

2.2.2 General Registers

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

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

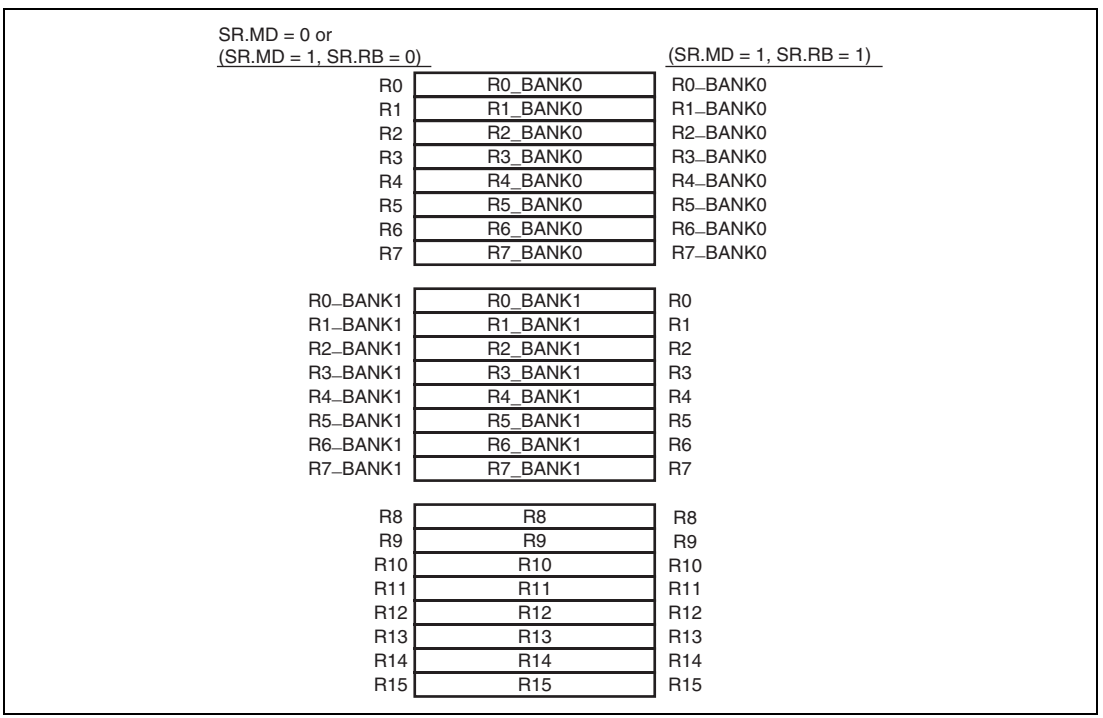


Figure 2.3 General Registers

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

2.2.3 Floating-Point Registers

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

1. Floating-point registers, FPRn_BANKj (32 registers)
   FPR0_BANK0 to FPR15_BANK0
   FPR0_BANK1 to FPR15_BANK1
2. Single-precision floating-point registers, FRi (16 registers)
   When FPSCR.FR = 0, FR0 to FR15 are assigned to FPR0_BANK0 to FPR15_BANK0;
   when FPSCR.FR = 1, FR0 to FR15 are assigned to FPR0_BANK1 to FPR15_BANK1.
3. Double-precision floating-point registers or single-precision floating-point registers, DRi (8 registers): A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14, FR15}
4. Single-precision floating-point vector registers, FVi (4 registers): An FV register comprises four FR registers.
   FV0 = {FR0, FR1, FR2, FR3}, FV4 = {FR4, FR5, FR6, FR7},
   FV8 = {FR8, FR9, FR10, FR11}, FV12 = {FR12, FR13, FR14, FR15}
5. Single-precision floating-point extended registers, XFi (16 registers)
   When FPSCR.FR = 0, XF0 to XF15 are assigned to FPR0_BANK1 to FPR15_BANK1;
   when FPSCR.FR = 1, XF0 to XF15 are assigned to FPR0_BANK0 to FPR15_BANK0.
6. Double-precision floating-point extended registers, XDi (8 registers): An XD register comprises two XF registers.
   XD0 = {XF0, XF1}, XD2 = {XF2, XF3}, XD4 = {XF4, XF5}, XD6 = {XF6, XF7},
   XD8 = {XF8, XF9}, XD10 = {XF10, XF11}, XD12 = {XF12, XF13}, XD14 = {XF14, XF15}

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

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

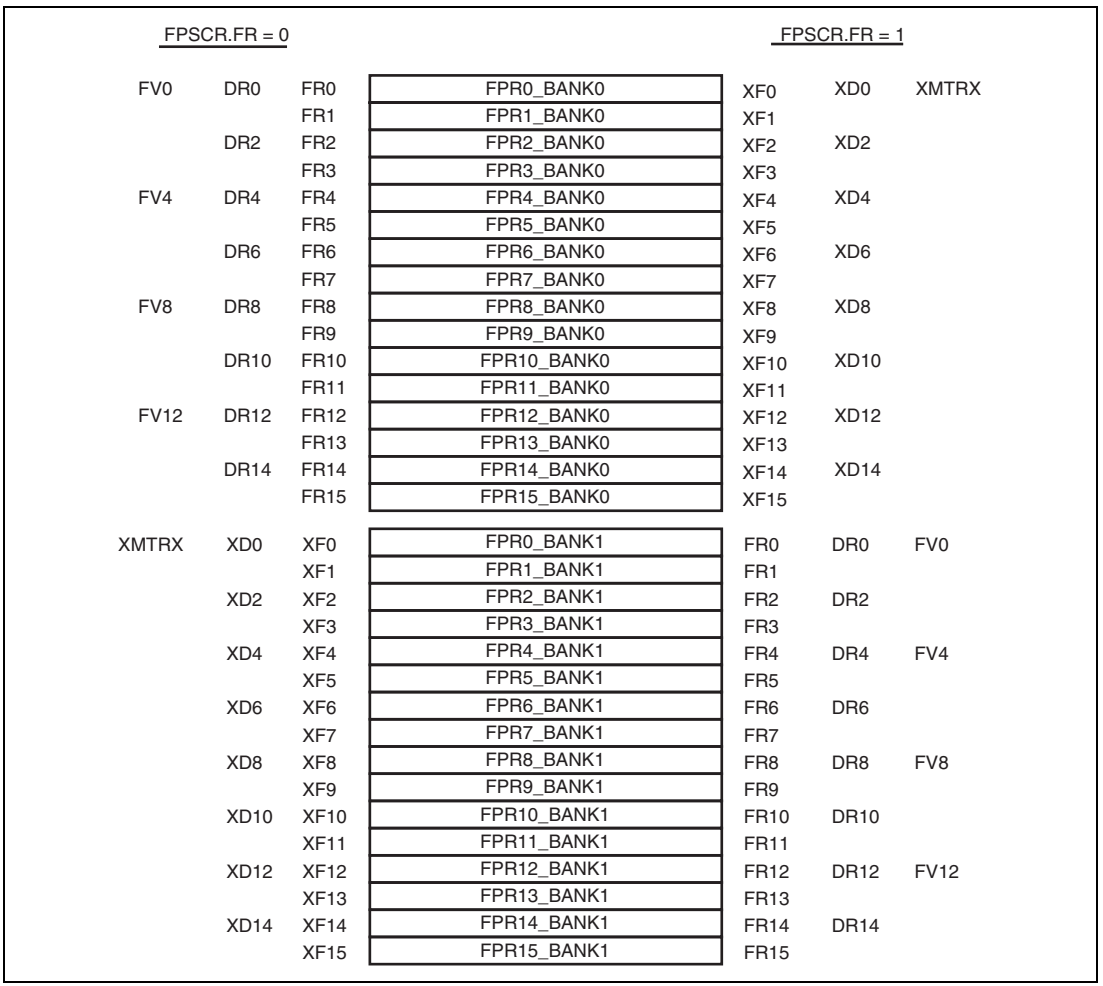


Figure 2.4 Floating-Point Registers

2.2.4 Control Registers

(1) Status Register (SR)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
27 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	FD	0	R/W	FPU Disable Bit When this bit is set to 1 and an FPU instruction is not in a delay slot, a general FPU disable exception occurs. When this bit is set to 1 and an FPU instruction is in a delay slot, a slot FPU disable exception occurs. (FPU instructions: H'F*** instructions and LDS (.L)/STS(.L) instructions using FPUL/FPSCR)
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	M	0	R/W	M Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
8	Q	0	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
7 to 4	IMASK	1111	R/W	Interrupt Mask Level Bits An interrupt whose priority is equal to or less than the value of the IMASK bits is masked. It can be chosen by CPU operation mode register (CPUOPM) whether the level of IMASK is changed to accept an interrupt or not when an interrupt is occurred. For details, refer to section 5.2.4, CPU Operation Mode Register (CPUOPM).
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	S	0	R/W	S Bit Used by the MAC instruction.
0	T	0	R/W	T Bit Indicates true/false condition, carry/borrow, or overflow/underflow. For details, see section 3, Instruction Set.

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

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

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

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

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

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

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

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

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

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

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

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

2.2.5 System Registers

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

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

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

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

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

PC indicates the address of the instruction currently being executed.

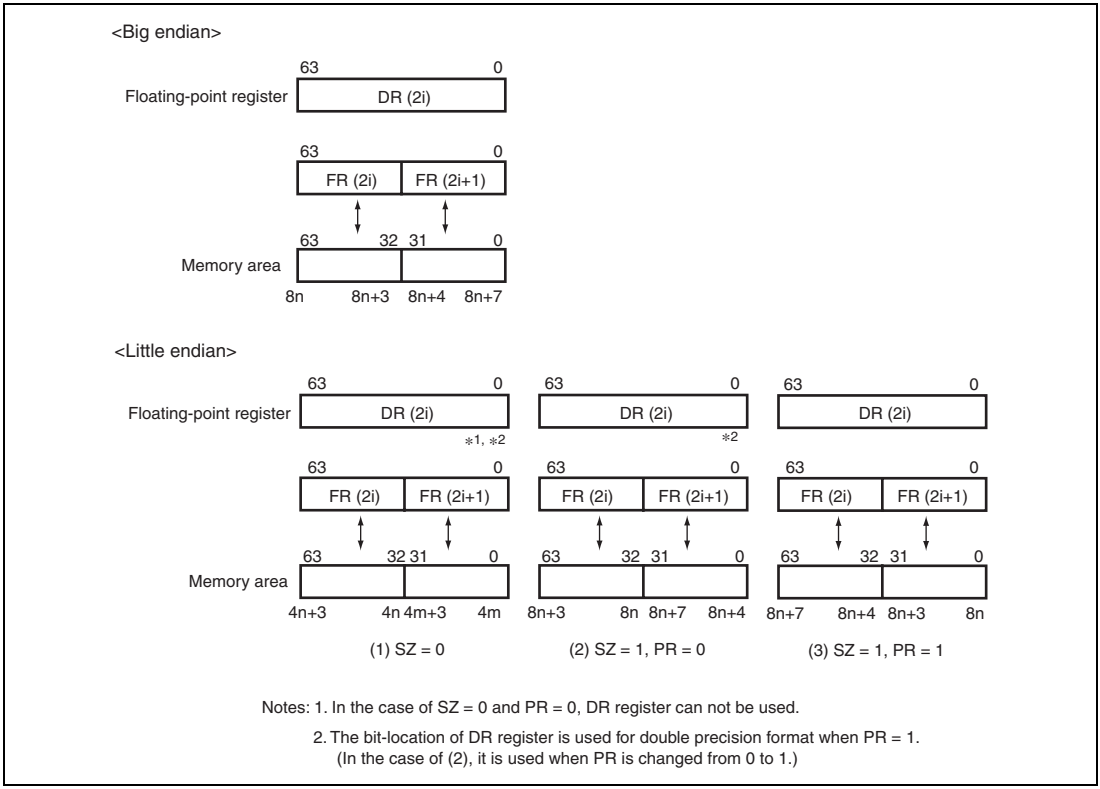
(4) Floating-Point Status/Control Register (FPSCR)

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

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

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

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



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

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

Figure 2.5 Relationship between SZ bit and Endian

Table 2.2 Bit Allocation for FPU Exception Handling

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

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

Information is transferred between the FPU and CPU via FPUL.

2.3 Memory-Mapped Registers

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

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

These two areas are used as follows.

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

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

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

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

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

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

2.4 Data Formats in Registers

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

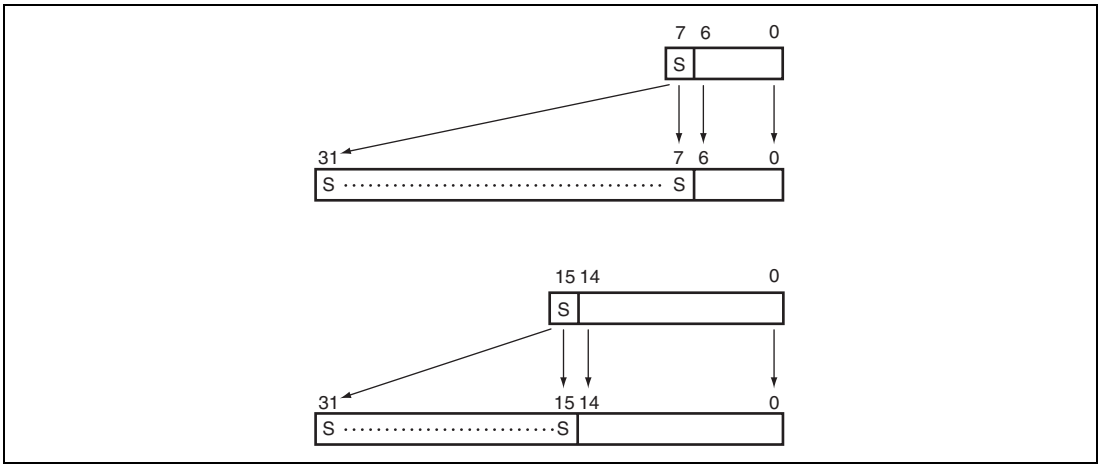


Figure 2.6 Formats of Byte Data and Word Data in Register

2.5 Data Formats in Memory

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

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

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

The data format in memory is shown in figure 2.7.

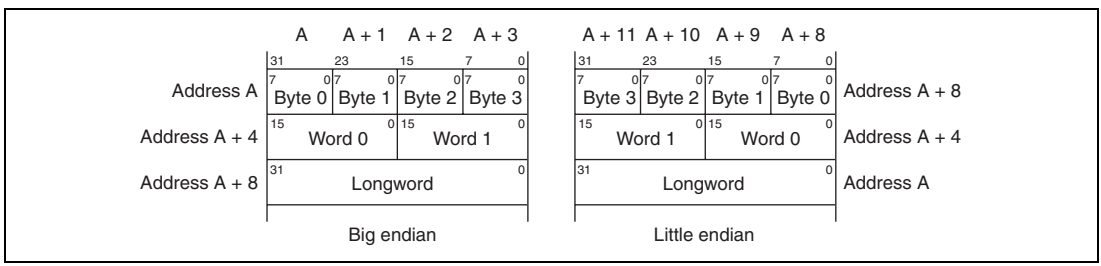


Figure 2.7 Data Formats in Memory

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

2.6 Processing States

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

(1) Reset State

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

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

(2) Instruction Execution State

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

(3) Power-Down State

In a power-down state, CPU halts operation and power consumption is reduced. The power-down state is entered by executing a SLEEP instruction. There are two modes in the CPU power-down state: sleep mode and light sleep mode. For details, see section 20, Power-Down mode.

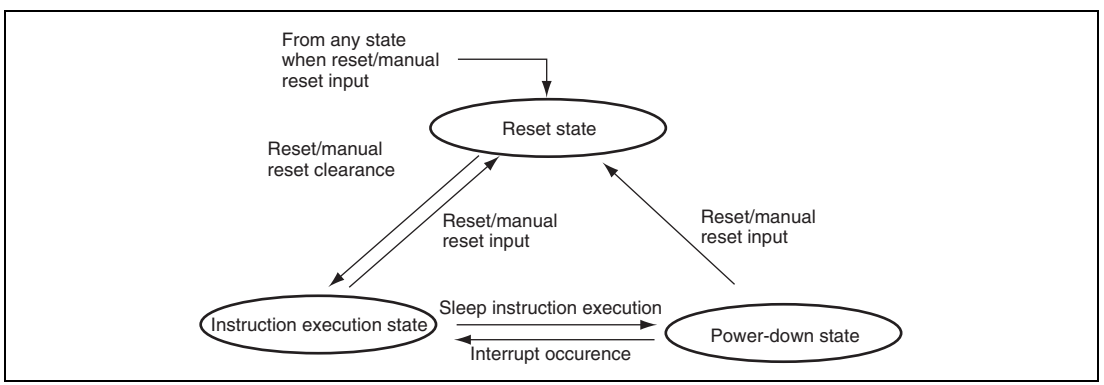


Figure 2.8 Processing State Transitions

2.7 Usage Notes

2.7.1 Notes on Self-Modifying Code

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

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

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

```
SYNCO
ICBI @Rn
```

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

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

```
SYNCO
ICBI @Rn
```

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

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

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

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

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

2.7.2 Instruction Prefetching and Its Side Effects

This LSI is provided with an internal buffer for holding pre-read instructions, and always performs pre-reading. Therefore, program code must not be located in the last 64-byte area of any memory space. If program code is located in these areas, a bus access for instruction prefetch may occur exceeding the memory areas boundary. A case in which this is a problem is shown below.

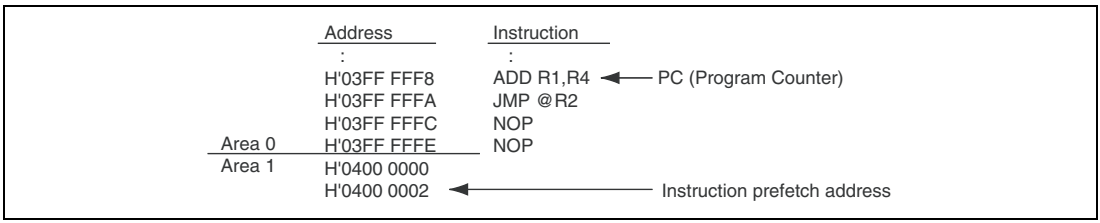


Figure 2.9 Instruction Prefetch

Figure 2.9 presupposes a case in which the instruction (ADD) indicated by the program counter (PC) and the address H'04000002 instruction prefetch are executed simultaneously. It is also assumed that the program branches to an area other than area 1 after executing the following JMP instruction and delay slot instruction.

In this case, a bus access (instruction prefetch) to area 1 may unintentionally occur from the programming flow.

(1) Instruction Prefetch Side Effects

1. It is possible that an external bus access caused by an instruction prefetch may result in misoperation of an external device, such as a FIFO, connected to the area concerned.

2. If there is no device to reply to an external bus request caused by an instruction prefetch, hangup will occur.

(2) Remedies

1. These illegal instruction fetches can be avoided by using the MMU.

2. The problem can be avoided by not locating program code in the last 64 bytes of any area.

Section 3 Instruction Set

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

3.1 Execution Environment

(1) PC

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

(2) Load-Store Architecture

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

(3) Delayed Branches

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

(4) Delay Slot

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

Table 3.1 Execution Order of Delayed Branch Instructions

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

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

(5) T Bit

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

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

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

(6) Constant Values

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

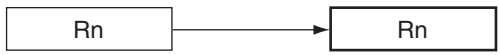
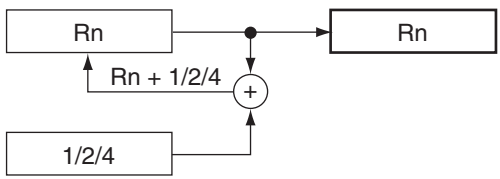
```
MOV.W   @(disp, PC), Rn
MOV.L   @(disp, PC), Rn
```

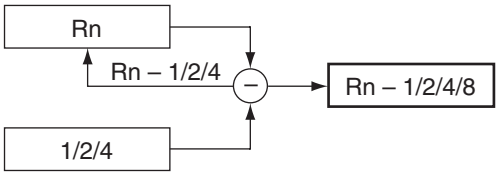
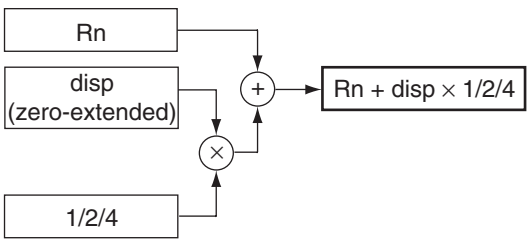
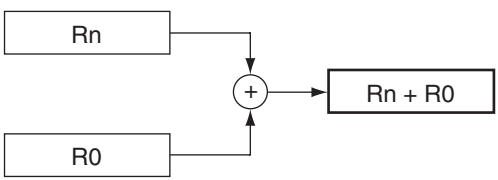
There are no PC-relative load instructions for floating-point operations. However, it is possible to set 0.0 or 1.0 by using the FLDI0 or FLDI1 instruction on a single-precision floating-point register.

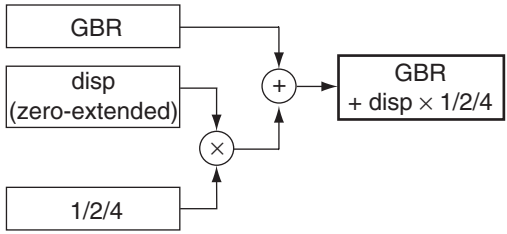
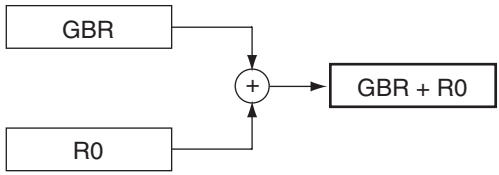
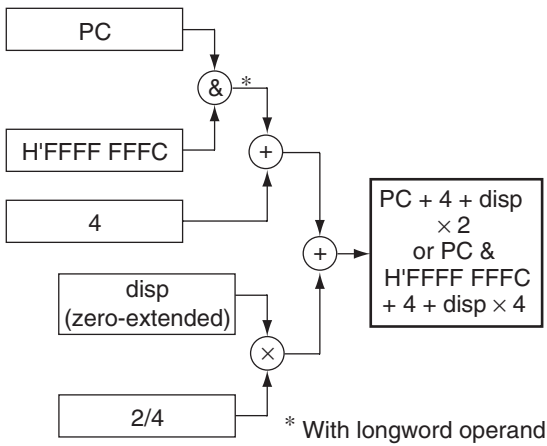
3.2 Addressing Modes

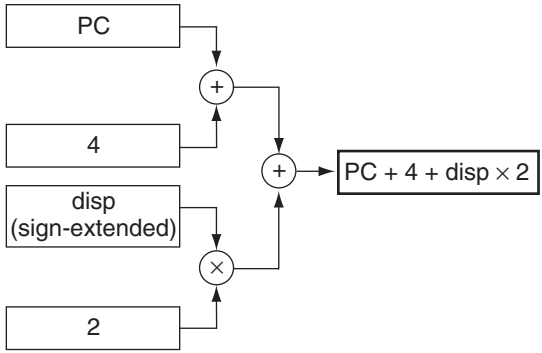
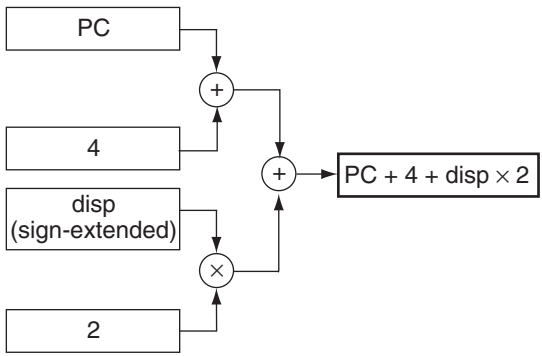
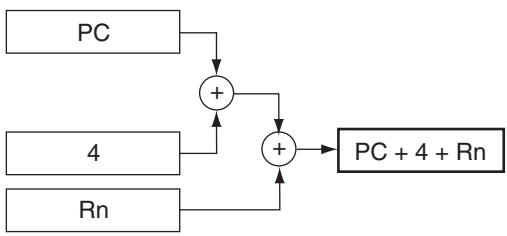
Addressing modes and effective address calculation methods are shown in table 3.2. When a location in virtual memory space is accessed (AT in MMUCR = 1), the effective address is translated into a physical memory address. If multiple virtual memory space systems are selected (SV in MMUCR = 0), the least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

Table 3.2 Addressing Modes and Effective Addresses

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

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

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

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

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

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

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

3.3 Instruction Set

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

Table 3.3 Notation Used in Instruction List

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

Item	Format	Description
Privileged mode		"Privileged" means the instruction can only be executed in privileged mode.
T bit	Value of T bit after instruction execution	—: No change
New	—	"New" means the instruction which has been newly added in the SH-4A with H'20-valued VER bits in the processor version register (PVR).

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

Table 3.4 Fixed-Point Transfer Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
MOV	#imm,Rn	imm → sign extension → Rn	1110nnnniiiiiiii	—	—	—
MOV.W	@(disp*,PC), Rn	(disp × 2 + PC + 4) → sign extension → Rn	1001nnnndddddddd	—	—	—
MOV.L	@(disp*,PC), Rn	(disp × 4 + PC & H'FFFF FFFC + 4) → Rn	1101nnnndddddddd	—	—	—
MOV	Rm,Rn	Rm → Rn	0110nnnnmmmm0011	—	—	—
MOV.B	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0000	—	—	—
MOV.W	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0001	—	—	—
MOV.L	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0010	—	—	—
MOV.B	@Rm,Rn	(Rm) → sign extension → Rn	0110nnnnmmmm0000	—	—	—
MOV.W	@Rm,Rn	(Rm) → sign extension → Rn	0110nnnnmmmm0001	—	—	—
MOV.L	@Rm,Rn	(Rm) → Rn	0110nnnnmmmm0010	—	—	—
MOV.B	Rm,@-Rn	Rn-1 → Rn, Rm → (Rn)	0010nnnnmmmm0100	—	—	—
MOV.W	Rm,@-Rn	Rn-2 → Rn, Rm → (Rn)	0010nnnnmmmm0101	—	—	—
MOV.L	Rm,@-Rn	Rn-4 → Rn, Rm → (Rn)	0010nnnnmmmm0110	—	—	—
MOV.B	@Rm+,Rn	(Rm)→ sign extension → Rn, Rm + 1 → Rm	0110nnnnmmmm0100	—	—	—
MOV.W	@Rm+,Rn	(Rm) → sign extension → Rn, Rm + 2 → Rm	0110nnnnmmmm0101	—	—	—
MOV.L	@Rm+,Rn	(Rm) → Rn, Rm + 4 → Rm	0110nnnnmmmm0110	—	—	—
MOV.B	R0,@(disp*,Rn)	R0 → (disp + Rn)	10000000nnnndddd	—	—	—
MOV.W	R0,@(disp*,Rn)	R0 → (disp × 2 + Rn)	10000001nnnndddd	—	—	—
MOV.L	Rm,@(disp*,Rn)	Rm → (disp × 4 + Rn)	0001nnnnmmmmdddd	—	—	—
MOV.B	@(disp*,Rm),R0	(disp + Rm) → sign extension → R0	10000100mmmmdddd	—	—	—
MOV.W	@(disp*,Rm),R0	(disp × 2 + Rm) → sign extension → R0	10000101mmmmdddd	—	—	—
MOV.L	@(disp*,Rm),Rn	(disp × 4 + Rm) → Rn	0101nnnnmmmmdddd	—	—	—
MOV.B	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0100	—	—	—
MOV.W	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0101	—	—	—
MOV.L	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
MOV.B	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1100	—	—	—
MOV.W	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1101	—	—	—
MOV.L	@(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	—	—
MOV.B	R0,@(disp*,GBR)	R0 → (disp + GBR)	11000000dddddddd	—	—	—
MOV.W	R0,@(disp*,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	—	—
MOV.L	R0,@(disp*,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	—	—
MOV.B	@(disp*,GBR),R0	(disp + GBR) → sign extension → R0	11000100dddddddd	—	—	—
MOV.W	@(disp*,GBR),R0	(disp × 2 + GBR) → sign extension → R0	11000101dddddddd	—	—	—
MOV.L	@(disp*,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	—	—
MOVA	@(disp*,PC),R0	disp × 4 + PC & H'FFFF FFFC + 4 → R0	11000111dddddddd	—	—	—
MOVCO.L	R0,@Rn	LDST → T If (T == 1) R0 → (Rn) 0 → LDST	0000nnnn01110011	—	LDST	New
MOVLI.L	@Rm,R0	1 → LDST (Rm) → R0 When interrupt/exception occurred 0 → LDST	0000mmmm01100011	—	—	New
MOVUA.L	@Rm,R0	(Rm) → R0 Load non-boundary alignment data	0100mmmm10101001	—	—	New
MOVUA.L	@Rm+,R0	(Rm) → R0, Rm + 4 → Rm Load non-boundary alignment data	0100mmmm11101001	—	—	New
MOVT	Rn	T → Rn	0000nnnn00101001	—	—	—
SWAP.B	Rm,Rn	Rm → swap lower 2 bytes → Rn	0110nnnnmmmm1000	—	—	—
SWAP.W	Rm,Rn	Rm → swap upper/lower words → Rn	0110nnnnmmmm1001	—	—	—
XTRCT	Rm,Rn	Rm:Rn middle 32 bits → Rn	0010nnnnmmmm1101	—	—	—

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

Table 3.5 Arithmetic Operation Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
ADD	Rm,Rn	Rn + Rm → Rn	0011nnnnmmmm1100	—	—	—
ADD	#imm,Rn	Rn + imm → Rn	0111nnnniiiiiiii	—	—	—
ADDC	Rm,Rn	Rn + Rm + T → Rn, carry → T	0011nnnnmmmm1110	—	Carry	—
ADDV	Rm,Rn	Rn + Rm → Rn, overflow → T	0011nnnnmmmm1111	—	Overflow	—
CMP/EQ	#imm,R0	When R0 = imm, 1 → T Otherwise, 0 → T	10001000iiiiiiii	—	Comparison result	—
CMP/EQ	Rm,Rn	When Rn = Rm, 1 → T Otherwise, 0 → T	0011nnnnmmmm0000	—	Comparison result	—
CMP/HS	Rm,Rn	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	0011nnnnmmmm0010	—	Comparison result	—
CMP/GE	Rm,Rn	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	0011nnnnmmmm0011	—	Comparison result	—
CMP/HI	Rm,Rn	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	0011nnnnmmmm0110	—	Comparison result	—
CMP/GT	Rm,Rn	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	0011nnnnmmmm0111	—	Comparison result	—
CMP/PZ	Rn	When Rn ≥ 0, 1 → T Otherwise, 0 → T	0100nnnn00010001	—	Comparison result	—
CMP/PL	Rn	When Rn > 0, 1 → T Otherwise, 0 → T	0100nnnn00010101	—	Comparison result	—
CMP/STR	Rm,Rn	When any bytes are equal, 1 → T Otherwise, 0 → T	0010nnnnmmmm1100	—	Comparison result	—
DIV1	Rm,Rn	1-step division (Rn ÷ Rm)	0011nnnnmmmm0100	—	Calculation result	—
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M^Q → T	0010nnnnmmmm0111	—	Calculation result	—
DIV0U		0 → M/Q/T	0000000000011001	—	0	—
DMULS.L	Rm,Rn	Signed, Rn × Rm → MAC, 32 × 32 → 64 bits	0011nnnnmmmm1101	—	—	—

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

Table 3.6 Logic Operation Instructions

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

Table 3.7 Shift Instructions

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

Table 3.8 Branch Instructions

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

Table 3.9 System Control Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
CLRMAC		0 → MACH, MACL	0000000000101000	—	—	—
CLRS		0 → S	0000000001001000	—	—	—
CLRT		0 → T	0000000000001000	—	0	—
ICBI	@Rn	Invalidates instruction cache block	0000nnnn11100011	—	—	New
LDC	Rm,SR	Rm → SR	0100mmmm00001110	Privileged	LSB	—
LDC	Rm,GBR	Rm → GBR	0100mmmm00011110	—	—	—
LDC	Rm,VBR	Rm → VBR	0100mmmm00101110	Privileged	—	—
LDC	Rm,SGR	Rm → SGR	0100mmmm00111010	Privileged	—	New
LDC	Rm,SSR	Rm → SSR	0100mmmm00111110	Privileged	—	—
LDC	Rm,SPC	Rm → SPC	0100mmmm01001110	Privileged	—	—
LDC	Rm,DBR	Rm → DBR	0100mmmm11111010	Privileged	—	—
LDC	Rm,Rn_BANK	Rm → Rn_BANK (n = 0 to 7)	0100mmmm1nnn1110	Privileged	—	—
LDC.L	@Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	Privileged	LSB	—
LDC.L	@Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	—	—	—
LDC.L	@Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	Privileged	—	—
LDC.L	@Rm+,SGR	(Rm) → SGR, Rm + 4 → Rm	0100mmmm00110110	Privileged	—	New
LDC.L	@Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	Privileged	—	—
LDC.L	@Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	Privileged	—	—
LDC.L	@Rm+,DBR	(Rm) → DBR, Rm + 4 → Rm	0100mmmm11110110	Privileged	—	—
LDC.L	@Rm+,Rn_BANK	(Rm) → Rn_BANK, Rm + 4 → Rm	0100mmmm1nnn0111	Privileged	—	—
LDS	Rm,MACH	Rm → MACH	0100mmmm00001010	—	—	—
LDS	Rm,MACL	Rm → MACL	0100mmmm00011010	—	—	—
LDS	Rm,PR	Rm → PR	0100mmmm00101010	—	—	—
LDS.L	@Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	—	—	—
LDS.L	@Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	—	—	—
LDS.L	@Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	—	—	—
LDTLB		PTEH/PTEL (/PTEA) → TLB	0000000000111000	Privileged	—	—
MOVCA.L	R0,@Rn	R0 → (Rn) (without fetching cache block)	0000nnnn11000011	—	—	—
NOP		No operation	0000000000001001	—	—	—
OCBI	@Rn	Invalidates operand cache block	0000nnnn10010011	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
OCBP	@Rn	Writes back and invalidates operand cache block	0000nnnn10100011	—	—	—
OCBWB	@Rn	Writes back operand cache block	0000nnnn10110011	—	—	—
PREF	@Rn	(Rn) → operand cache	0000nnnn10000011	—	—	—
PREFI	@Rn	Reads 32-byte instruction block into instruction cache	0000nnnn11010011	—	—	New
RTE		Delayed branch, SSR/SPC → SR/PC	0000000000101011	Privileged	—	—
SETS		1 → S	0000000001011000	—	—	—
SETT		1 → T	0000000000011000	—	1	—
SLEEP		Sleep	0000000000011011	Privileged	—	—
STC	SR,Rn	SR → Rn	0000nnnn00000010	Privileged	—	—
STC	GBR,Rn	GBR → Rn	0000nnnn00010010	—	—	—
STC	VBR,Rn	VBR → Rn	0000nnnn00100010	Privileged	—	—
STC	SSR,Rn	SSR → Rn	0000nnnn00110010	Privileged	—	—
STC	SPC,Rn	SPC → Rn	0000nnnn01000010	Privileged	—	—
STC	SGR,Rn	SGR → Rn	0000nnnn00111010	Privileged	—	—
STC	DBR,Rn	DBR → Rn	0000nnnn11111010	Privileged	—	—
STC	Rm_BANK,Rn	Rm_BANK → Rn (m = 0 to 7)	0000nnnn1mmm0010	Privileged	—	—
STC.L	SR,@-Rn	Rn – 4 → Rn, SR → (Rn)	0100nnnn00000011	Privileged	—	—
STC.L	GBR,@-Rn	Rn – 4 → Rn, GBR → (Rn)	0100nnnn00010011	—	—	—
STC.L	VBR,@-Rn	Rn – 4 → Rn, VBR → (Rn)	0100nnnn00100011	Privileged	—	—
STC.L	SSR,@-Rn	Rn – 4 → Rn, SSR → (Rn)	0100nnnn00110011	Privileged	—	—
STC.L	SPC,@-Rn	Rn – 4 → Rn, SPC → (Rn)	0100nnnn01000011	Privileged	—	—
STC.L	SGR,@-Rn	Rn – 4 → Rn, SGR → (Rn)	0100nnnn00110010	Privileged	—	—
STC.L	DBR,@-Rn	Rn – 4 → Rn, DBR → (Rn)	0100nnnn11110010	Privileged	—	—
STC.L	Rm_BANK, @-Rn	Rn – 4 → Rn, Rm_BANK → (Rn) (m = 0 to 7)	0100nnnn1mmm0011	Privileged	—	—
STS	MACH,Rn	MACH → Rn	0000nnnn00001010	—	—	—
STS	MACL,Rn	MACL → Rn	0000nnnn00011010	—	—	—
STS	PR,Rn	PR → Rn	0000nnnn00101010	—	—	—
STS.L	MACH,@-Rn	Rn – 4 → Rn, MACH → (Rn)	0100nnnn00000010	—	—	—
STS.L	MACL,@-Rn	Rn – 4 → Rn, MACL → (Rn)	0100nnnn00010010	—	—	—
STS.L	PR,@-Rn	Rn – 4 → Rn, PR → (Rn)	0100nnnn00100010	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
SYNCO		Data accesses invoked by the following instructions are not executed until execution of data accesses which precede this instruction has been completed.	0000000010101011	—	—	New
TRAPA	#imm	PC + 2 → SPC, SR → SSR, R15 → SGR, 1 → SR.MD/BL/RB, #imm << 2 → TRA, H'160 → EXPEVT, VBR + H'0100 → PC	11000011iiiiiiii	—	—	—

Table 3.10 Floating-Point Single-Precision Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
FLDI0	FRn	H'0000 0000 → FRn	1111nnnn10001101	—	—	—
FLDI1	FRn	H'3F80 0000 → FRn	1111nnnn10011101	—	—	—
FMOV	FRm,FRn	FRm → FRn	1111nnnnmmmm1100	—	—	—
FMOV.S	@Rm,FRn	(Rm) → FRn	1111nnnnmmmm1000	—	—	—
FMOV.S	@(R0,Rm),FRn	(R0 + Rm) → FRn	1111nnnnmmmm0110	—	—	—
FMOV.S	@Rm+,FRn	(Rm) → FRn, Rm + 4 → Rm	1111nnnnmmmm1001	—	—	—
FMOV.S	FRm,@Rn	FRm → (Rn)	1111nnnnmmmm1010	—	—	—
FMOV.S	FRm,@-Rn	Rn-4 → Rn, FRm → (Rn)	1111nnnnmmmm1011	—	—	—
FMOV.S	FRm,@(R0,Rn)	FRm → (R0 + Rn)	1111nnnnmmmm0111	—	—	—
FMOV	DRm,DRn	DRm → DRn	1111nnn0mmm01100	—	—	—
FMOV	@Rm,DRn	(Rm) → DRn	1111nnn0mmmm1000	—	—	—
FMOV	@(R0,Rm),DRn	(R0 + Rm) → DRn	1111nnn0mmmm0110	—	—	—
FMOV	@Rm+,DRn	(Rm) → DRn, Rm + 8 → Rm	1111nnn0mmmm1001	—	—	—
FMOV	DRm,@Rn	DRm → (Rn)	1111nnnnmmm01010	—	—	—
FMOV	DRm,@-Rn	Rn-8 → Rn, DRm → (Rn)	1111nnnnmmm01011	—	—	—
FMOV	DRm,@(R0,Rn)	DRm → (R0 + Rn)	1111nnnnmmm00111	—	—	—
FLDS	FRm,FPUL	FRm → FPUL	1111mmmm00011101	—	—	—
FSTS	FPUL,FRn	FPUL → FRn	1111nnnn00001101	—	—	—
FABS	FRn	FRn & H'7FFF FFFF → FRn	1111nnnn01011101	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
FADD	FRm,FRn	FRn + FRm → FRn	1111nnnnmmmm0000	—	—	—
FCMP/EQ	FRm,FRn	When FRn = FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0100	—	Comparison result	—
FCMP/GT	FRm,FRn	When FRn > FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0101	—	Comparison result	—
FDIV	FRm,FRn	FRn/FRm → FRn	1111nnnnmmmm0011	—	—	—
FLOAT	FPUL,FRn	(float) FPUL → FRn	1111nnnn00101101	—	—	—
FMAC	FR0,FRm,FRn	FR0*FRm + FRn → FRn	1111nnnnmmmm1110	—	—	—
FMUL	FRm,FRn	FRn*FRm → FRn	1111nnnnmmmm0010	—	—	—
FNEG	FRn	FRn ∧ H'8000 0000 → FRn	1111nnnn01001101	—	—	—
FSQRT	FRn	√FRn → FRn	1111nnnn01101101	—	—	—
FSUB	FRm,FRn	FRn – FRm → FRn	1111nnnnmmmm0001	—	—	—
FTRC	FRm,FPUL	(long) FRm → FPUL	1111mmmm00111101	—	—	—

Table 3.11 Floating-Point Double-Precision Instructions

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

Table 3.12 Floating-Point Control Instructions

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

Table 3.13 Floating-Point Graphics Acceleration Instructions

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

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

Section 4 Pipelining

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

4.1 Pipelines

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

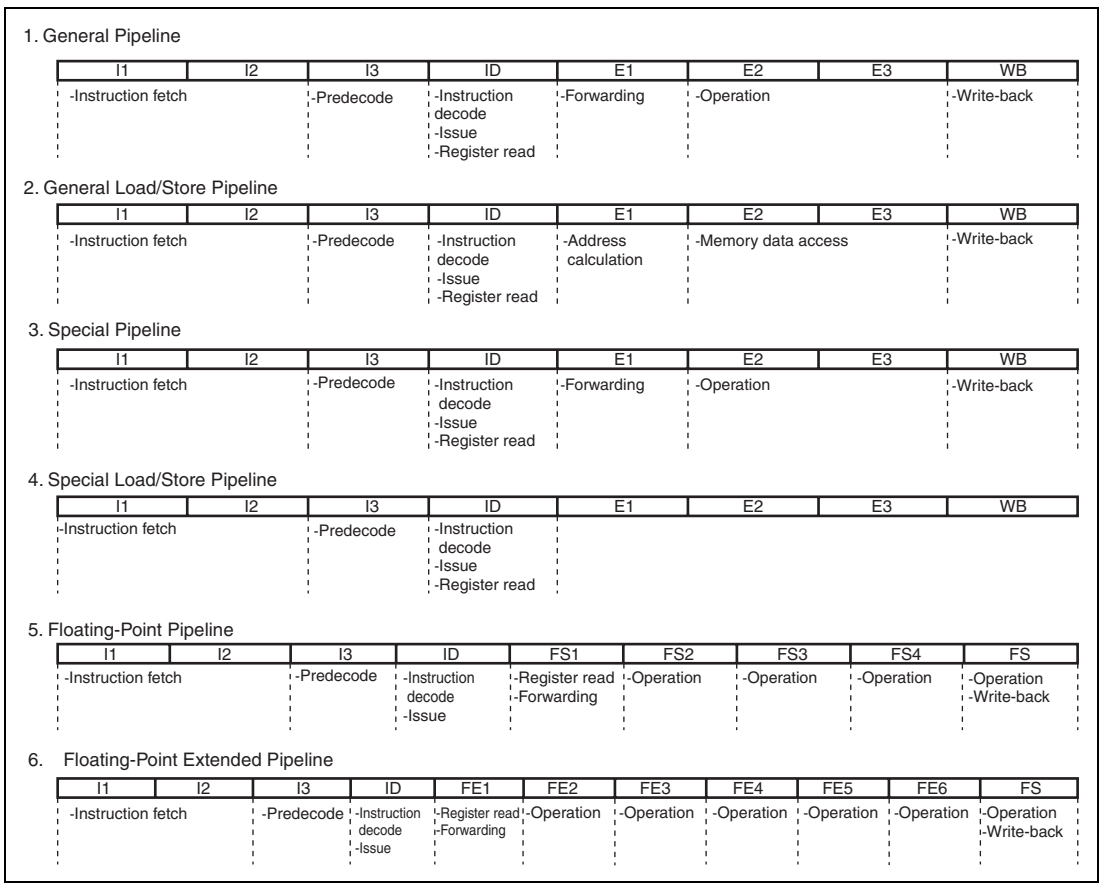


Figure 4.1 Basic Pipelines

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

Table 4.1 Representations of Instruction Execution Patterns

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

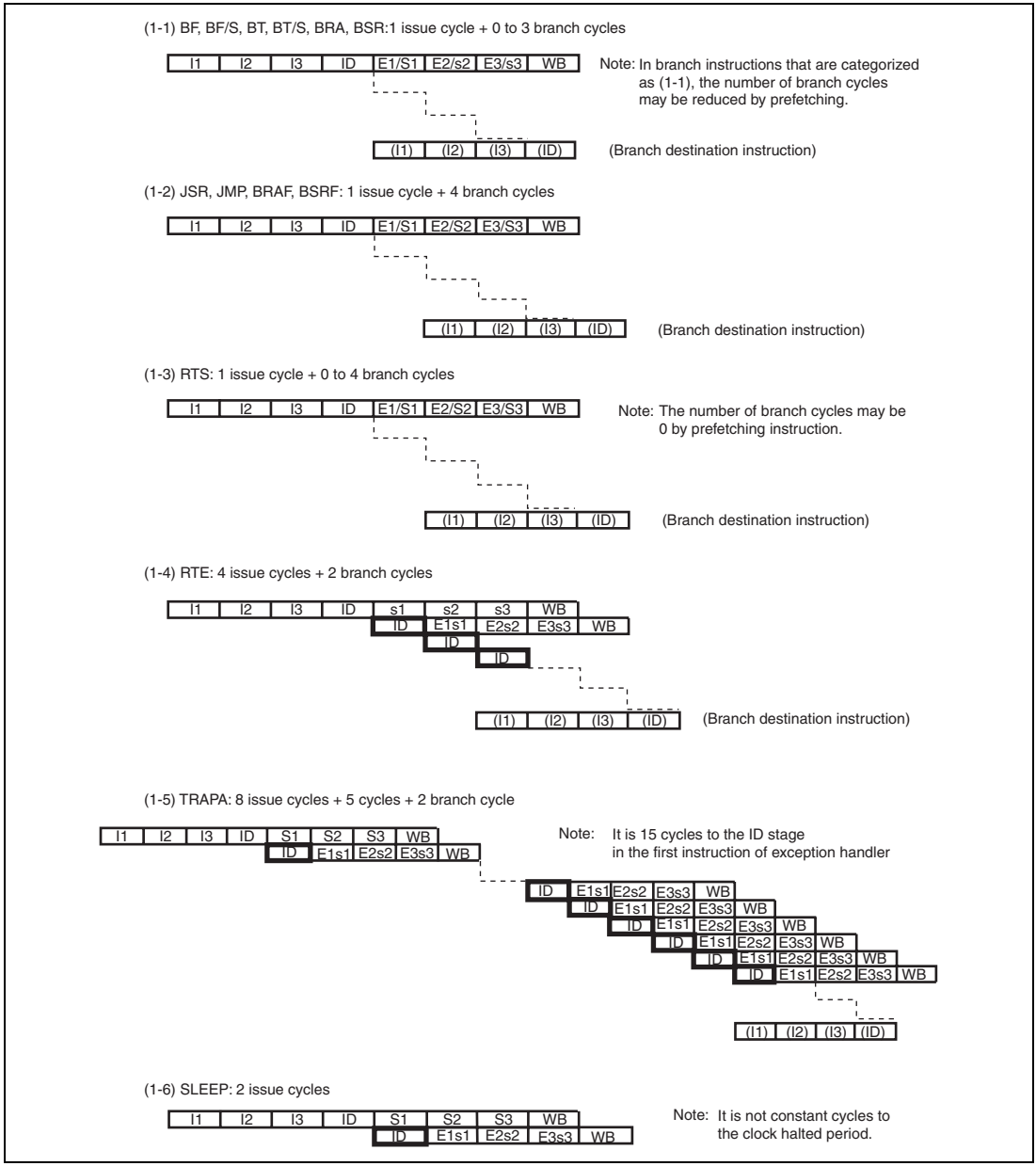


Figure 4.2 Instruction Execution Patterns (1)

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

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

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

I1	I2	I3	ID	E1	E2	E3	WB

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

MOVA

I1	I2	I3	ID	s1	s2	s3	WB

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

MOV#, NOP

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

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

MOV

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

Figure 4.2 Instruction Execution Patterns (2)

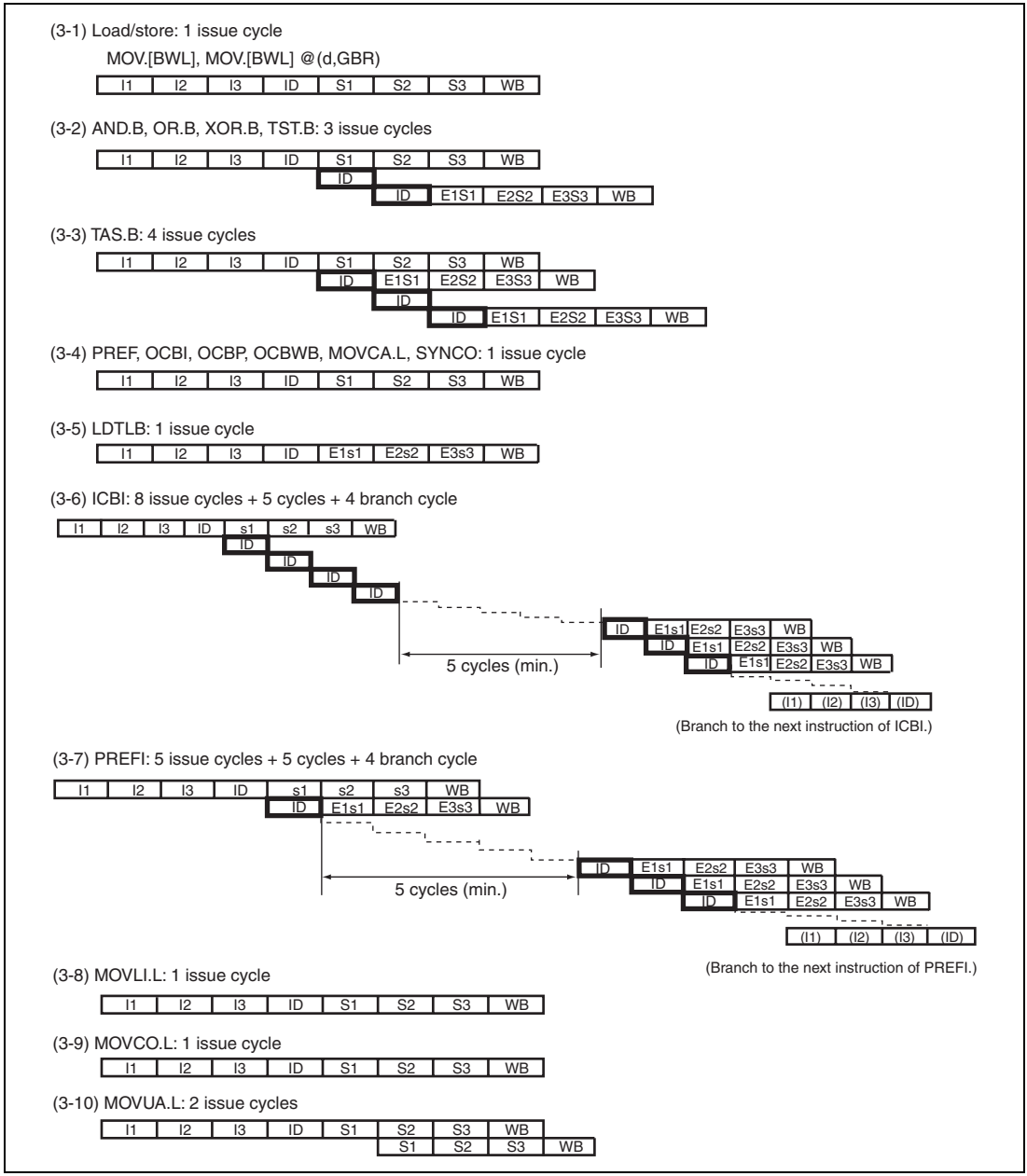


Figure 4.2 Instruction Execution Patterns (3)

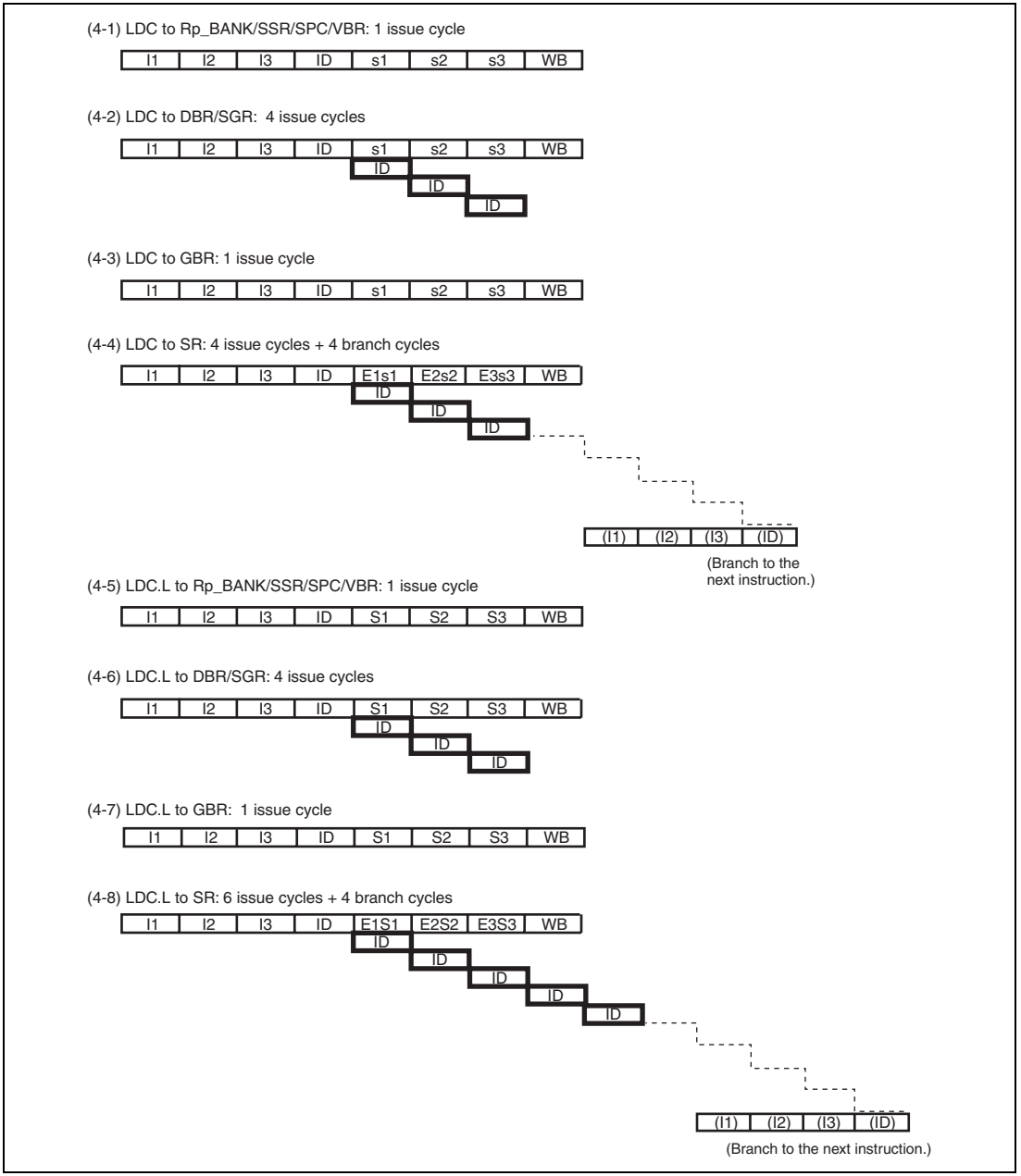


Figure 4.2 Instruction Execution Patterns (4)

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

I1	I2	I3	ID	s1	s2	s3	WB

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

I1	I2	I3	ID	E1s1	E2s2	E3s3	WB

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

I1	I2	I3	ID	S1	S2	S3	WB

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

I1	I2	I3	ID	E1S1	E2S2	E3S3	WB

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

I1	I2	I3	ID	s1	s2	s3	WB

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

I1	I2	I3	ID	S1	S2	S3	WB

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

I1	I2	I3	ID	s1	s2	s3	WB

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

I1	I2	I3	ID	S1	S2	S3	WB

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

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

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

Figure 4.2 Instruction Execution Patterns (5)

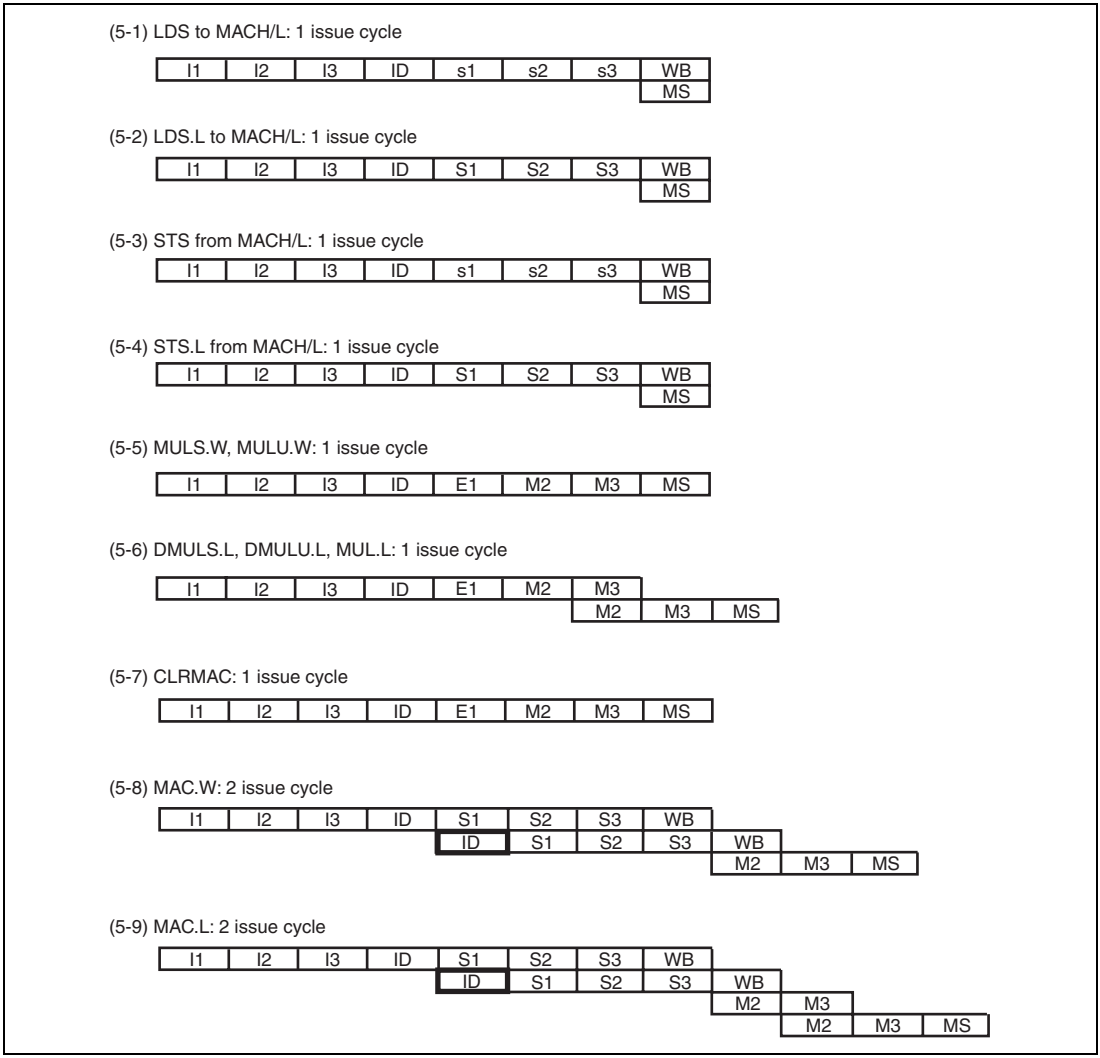


Figure 4.2 Instruction Execution Patterns (6)

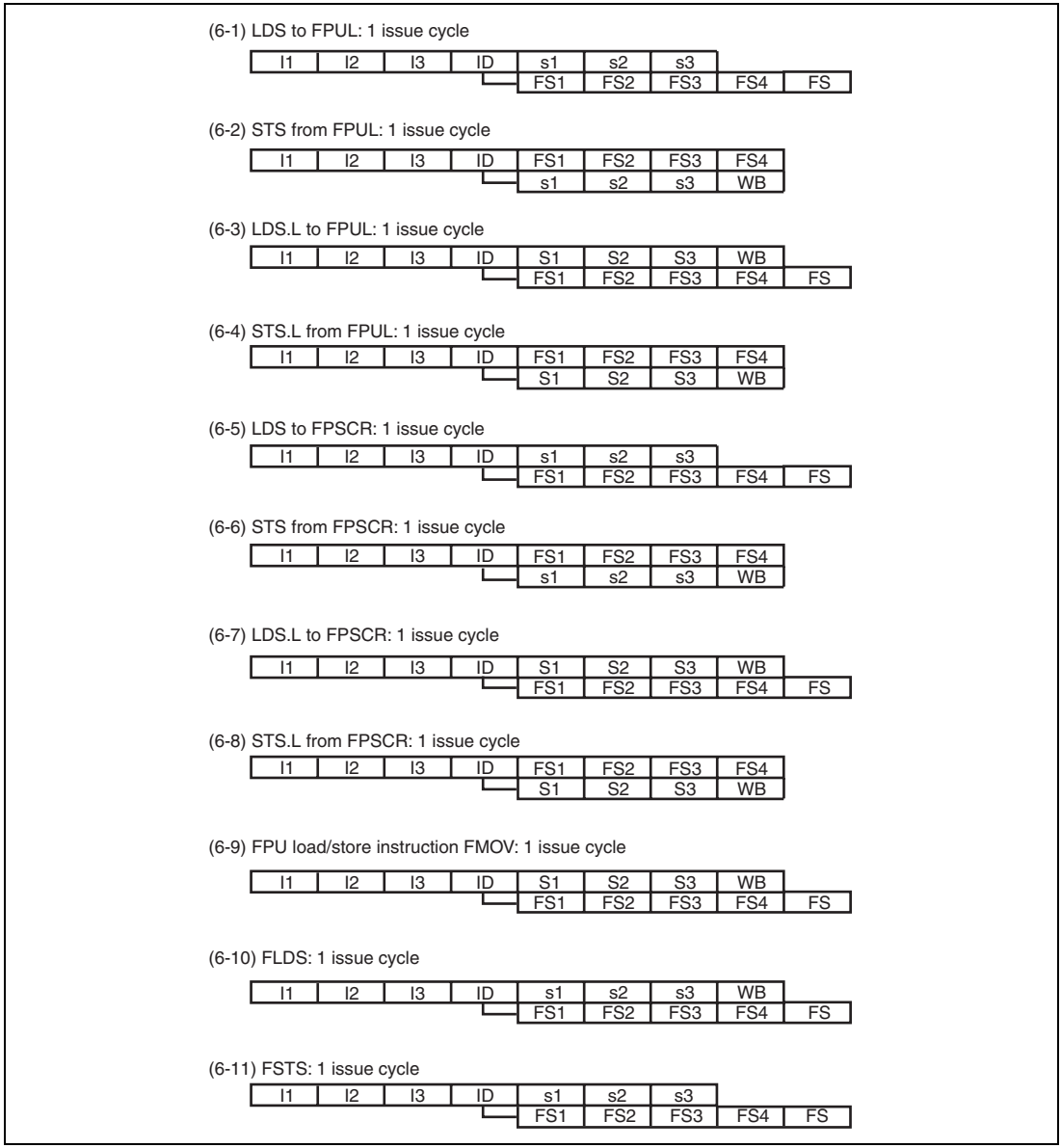
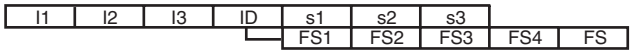
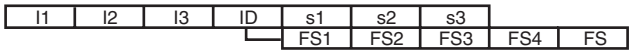


Figure 4.2 Instruction Execution Patterns (7)

(6-12) Single-precision FABS, FNEG/double-precision FABS, FNEG: 1 issue cycle



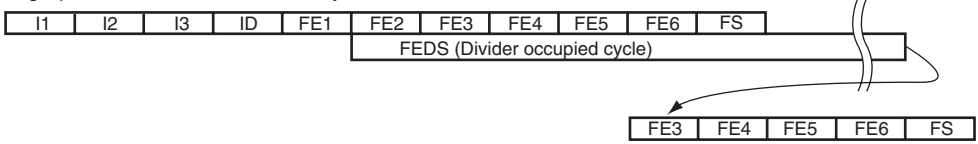
(6-13) FLDI0, FLDI1: 1 issue cycle



(6-14) Single-precision floating-point computation: 1 issue cycle

FCMP/EQ, FCMP/GT, FADD, FLOAT, FMAC, FMUL, FSUB, FTRC, FRCHG, FSCHG, FPCHG

(6-15) Single-precision FDIV/FSQRT: 1 issue cycle

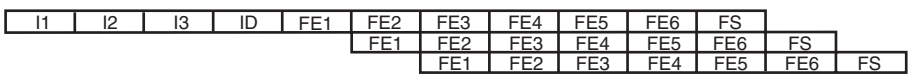


(6-16) Double-precision floating-point computation: 1 issue cycle

FCMP/EQ, FCMP/GT, FADD, FLOAT, FSUB, FTRC, FCNVSD, FCNVDS

(6-17) Double-precision floating-point computation: 1 issue cycle

FMUL



(6-18) Double-precision FDIV/FSQRT: 1 issue cycle

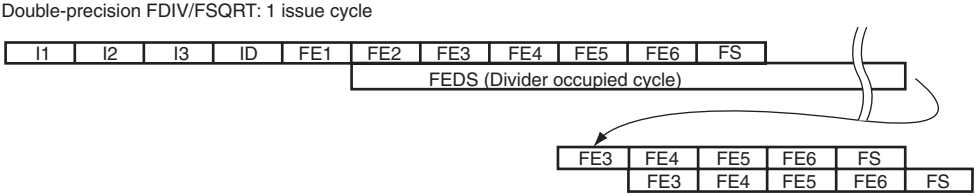


Figure 4.2 Instruction Execution Patterns (8)

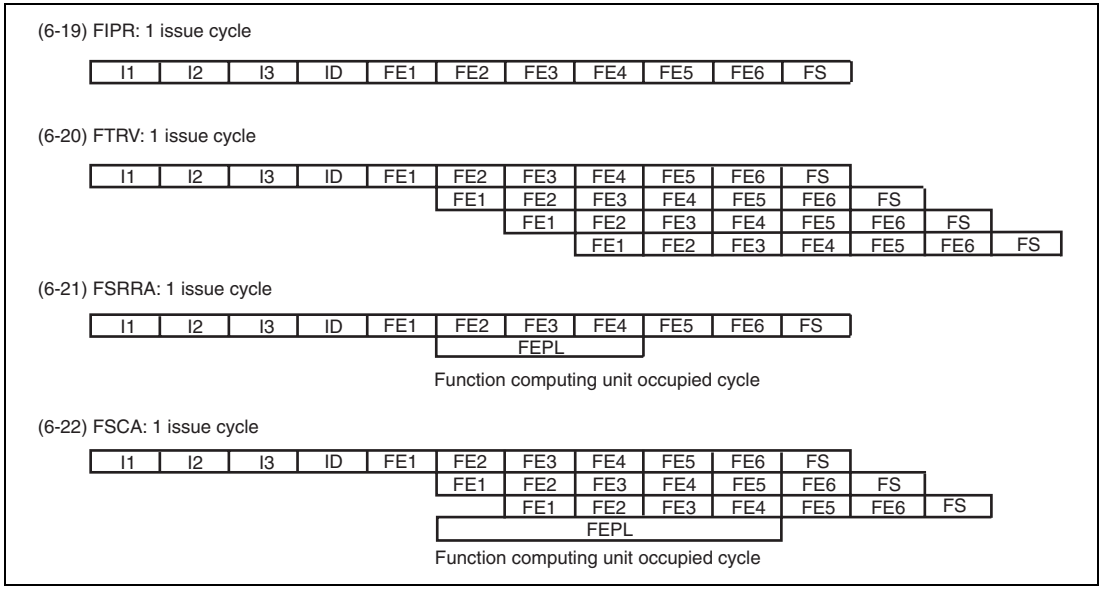


Figure 4.2 Instruction Execution Patterns (9)

4.2 Parallel-Executability

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

Table 4.2 Instruction Groups

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

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

Legend:

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

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

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

Table 4.3 Combination of Preceding and Following Instructions

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

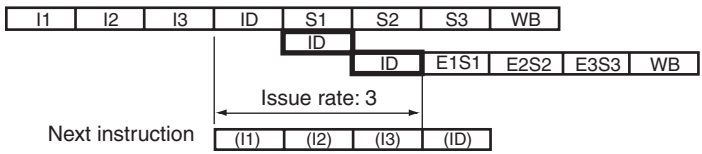
4.3 Issue Rates and Execution Cycles

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

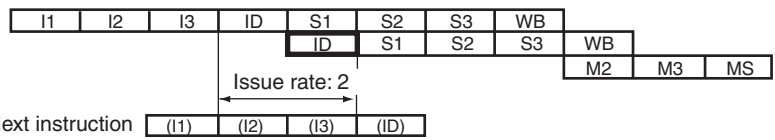
1. Issue Rate

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

E.g. AND.B instruction



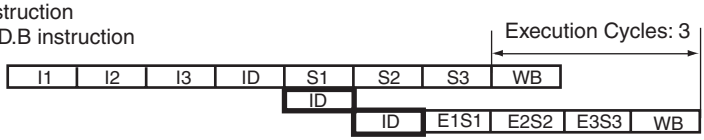
E.g. MAC.W instruction



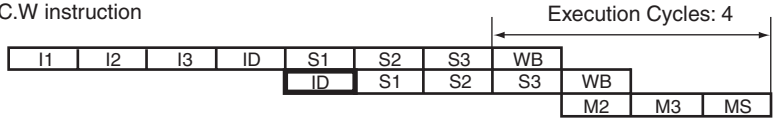
2. Execution Cycles

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

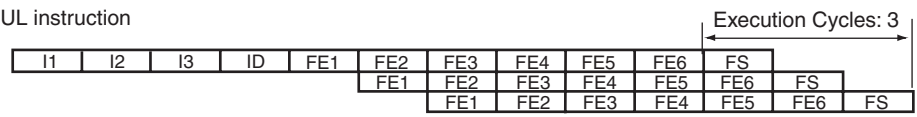
CPU instruction
E.g. AND.B instruction



E.g. MAC.W instruction



FPU instruction
E.g. FMUL instruction



E.g. FDIV instruction

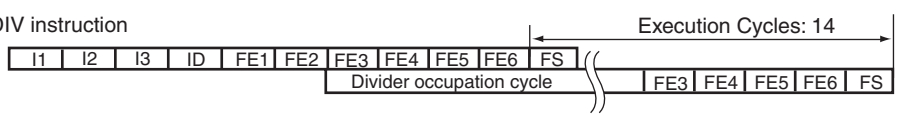


Table 4.4 Issue Rates and Execution Cycles

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

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

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

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

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

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

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

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

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

Section 5 Exception Handling

5.1 Summary of Exception Handling

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

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

5.2 Register Descriptions

Table 5.1 lists the configuration of registers related exception handling.

Table 5.1 Register Configuration

Register Name	Abbr.	R/W	P4 Address*	Area 7 Address*	Access Size
TRAPA exception register	TRA	R/W	H'FF00 0020	H'1F00 0020	32
Exception event register	EXPEVT	R/W	H'FF00 0024	H'1F00 0024	32
Interrupt event register	INTEVT	R/W	H'FF00 0028	H'1F00 0028	32
CPU operation mode register	CPUOPM	R/W	H'FF2F 0000	H'1F2F 0000	32
Non-support detection exception register	EXPMASK	R/W	H'FF2F 0004	H'1F2F 0004	32

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

Table 5.2 States of Register in Each Operating Mode

Register Name	Abbr.	Power-on Reset	Manual Reset	Sleep/ Light Sleep
TRAPA exception register	TRA	Undefined	Undefined	Retained
Exception event register	EXPEVT	H'0000 0000	H'0000 0020	Retained
Interrupt event register	INTEVT	Undefined	Undefined	Retained
CPU operation mode register	CPUOPM	H'0000 03C0	H'0000 03C0	Retained
Non-support detection exception register	EXPMASK	H'0000 0013	H'0000 0013	Retained

5.2.1 TRAPA Exception Register (TRA)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 2	TRACODE	Undefined	R/W	TRAPA Code 8-bit immediate data of TRAPA instruction is set
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

5.2.2 Exception Event Register (EXPEVT)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	EXPCODE	H'000 or H'020	R/W	Exception Code The exception code for a reset or general exception is set. For details, see table 5.3.

5.2.3 Interrupt Event Register (INTEVT)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	INTCODE	Undefined	R/W	Exception Code The exception code for an interrupt is set. For details, see table 5.3.

5.2.4 CPU Operation Mode Register (CPUOPM)

The CPUOPM is used to control the CPU operation mode. This register can be read from or written to the address H'FF2F0000 in P4 area or H'1F2F0000in area 7 as 32-bit size. The write value to the reserved bits should be the initial value. The operation is not guaranteed if the write value is not the initial value.

The CPUOPM register should be updated by the CPU store instruction not the access from SuperHyway bus master except CPU.

After the CPUOPM is updated, read CPUOPM once, and execute one of the following two methods.

1. Execute a branch using the RTE instruction.
2. Execute the ICBI instruction for any address (including non-cacheable area).

After one of these methods is executed, it is guaranteed that the CPU runs under the updated CPUOPM value.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved
9 to 6	—	1111		The write value must be the initial value.
5	RABD	0	R/W	Speculative execution bit for subroutine return 0: Instruction fetch for subroutine return is issued speculatively.* 1: Instruction fetch for subroutine return is not issued speculatively.
4	—	0	R	Reserved The write value must be the initial value.

Bit	Bit Name	Initial Value	R/W	Description
3	INTMU	0	R/W	Interrupt mode switch bit 0: SR.IMASK is not changed when an interrupt is accepted. 1: SR.IMASK is changed to the accepted interrupt level.
2 to 0	—	All 0	R	Reserved The write value must be the initial value.

Note: * Speculative instruction fetch may issue the access to the address that should not be accessed from the program. Therefore a bus access to an unexpected area or an internal instruction address error may cause a problem. As for the effect of this bus access to unexpected memory area, refer to Section 2.7.2, Instruction Prefetching and Its Side Effects. When the speculative execution for subroutine return is enabled, the RTS instruction should be used to return to the address set in PR by the JSR, BSR, or BSRF Instructions. It can prevent the access to unexpected address and avoid the problem.

5.2.5 Non-Support Detection Exception Register (EXPMASK)

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

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

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

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

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

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

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

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

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

5.3 Exception Handling Functions

5.3.1 Exception Handling Flow

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

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

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

5.3.2 Exception Handling Vector Addresses

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

5.4 Exception Types and Priorities

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

Table 5.3 Exceptions

Exception Category	Execution Mode	Exception	Priority Level*2	Priority Order*2	Exception Transition Direction*3		Exception Code*4
					Vector Address	Offset	
Reset	Abort type	Power-on reset	1	1	CPU0: H'A000 0000 CPU1: Module stop	—	H'000
		Manual reset	1	2	Address set in CPUnRESETVEC register	—	H'020
		H-UDI reset	1	1	CPU0: H'A000 0000 CPU1: Module stop	—	H'000
		Instruction TLB multiple-hit exception	1	3	Address set in CPUnRESETVEC registe	—	H'140
		Data TLB multiple-hit exception	1	4	Address set in CPUnRESETVEC registe	—	H'140
General exception	Re-execution type	User break before instruction execution*1	2	0	(VBR/DBR)	H'100/—	H'1E0
		Instruction address error	2	1	(VBR)	H'100	H'0E0
		Instruction TLB miss exception	2	2	(VBR)	H'400	H'040
		Instruction TLB protection violation exception	2	3	(VBR)	H'100	H'0A0
		General illegal instruction exception	2	4	(VBR)	H'100	H'180
		Slot illegal instruction exception	2	4	(VBR)	H'100	H'1A0
		General FPU disable exception	2	4	(VBR)	H'100	H'800
		Slot FPU disable exception	2	4	(VBR)	H'100	H'820
		Data address error (read)	2	5	(VBR)	H'100	H'0E0
		Data address error (write)	2	5	(VBR)	H'100	H'100

Exception Category	Execution Mode	Exception	Priority Level*[2]	Priority Order*[2]	Exception Transition Direction*[3]		Exception Code*[4]
					Vector Address	Offset	
General exception	Re-execution type	Data TLB miss exception (read)	2	6	(VBR)	H'400	H'040
		Data TLB miss exception (write)	2	6	(VBR)	H'400	H'060
		Data TLB protection violation exception (read)	2	7	(VBR)	H'100	H'0A0
		Data TLB protection violation exception (write)	2	7	(VBR)	H'100	H'0C0
		FPU exception	2	8	(VBR)	H'100	H'120
		Initial page write exception	2	9	(VBR)	H'100	H'080
	Completion type	Unconditional trap (TRAPA)	2	4	(VBR)	H'100	H'160
		User break after instruction execution*	2	10	(VBR/DBR)	H'100/—	H'1E0
Interrupt	Completion type	Nonmaskable interrupt	3	—	(VBR)	H'600	H'1C0
		General interrupt request	4	—	(VBR)	H'600	—

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

5.5 Exception Flow

5.5.1 Exception Flow

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

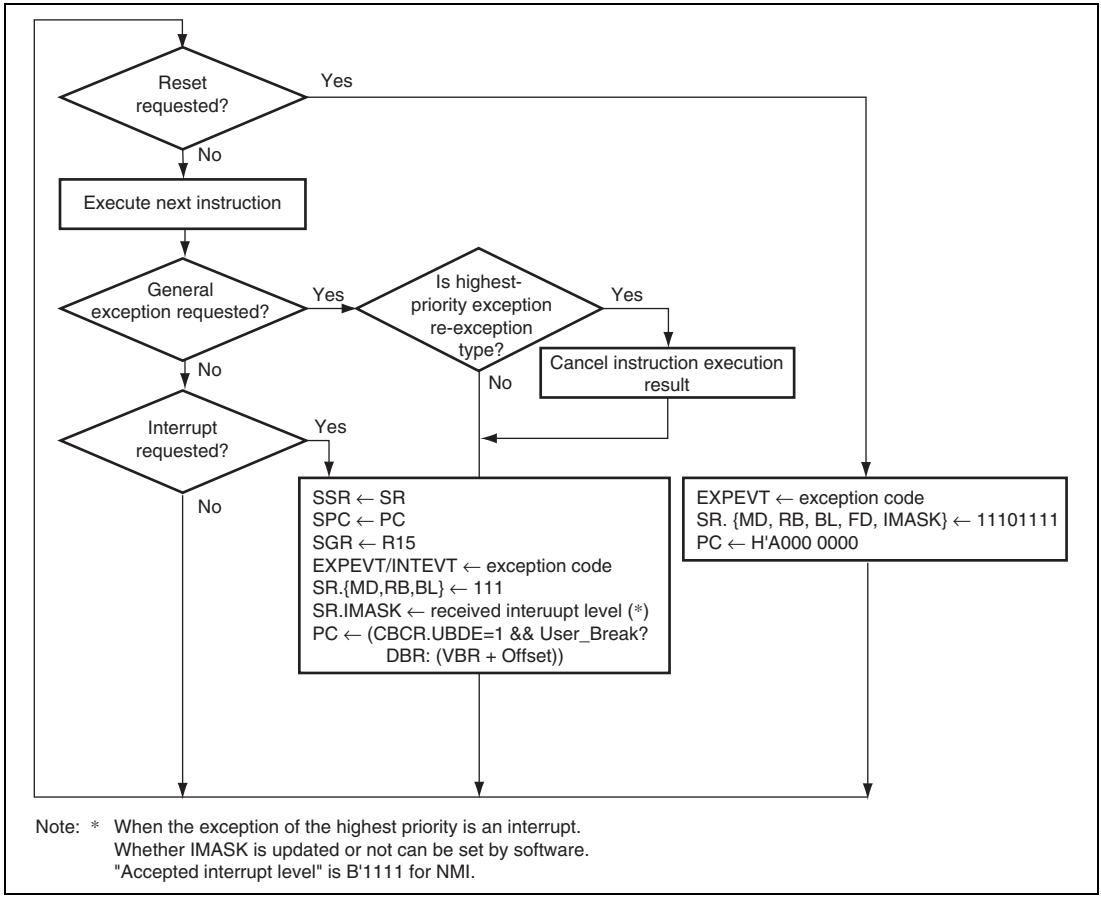


Figure 5.1 Instruction Execution and Exception Handling

5.5.2 Exception Source Acceptance

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

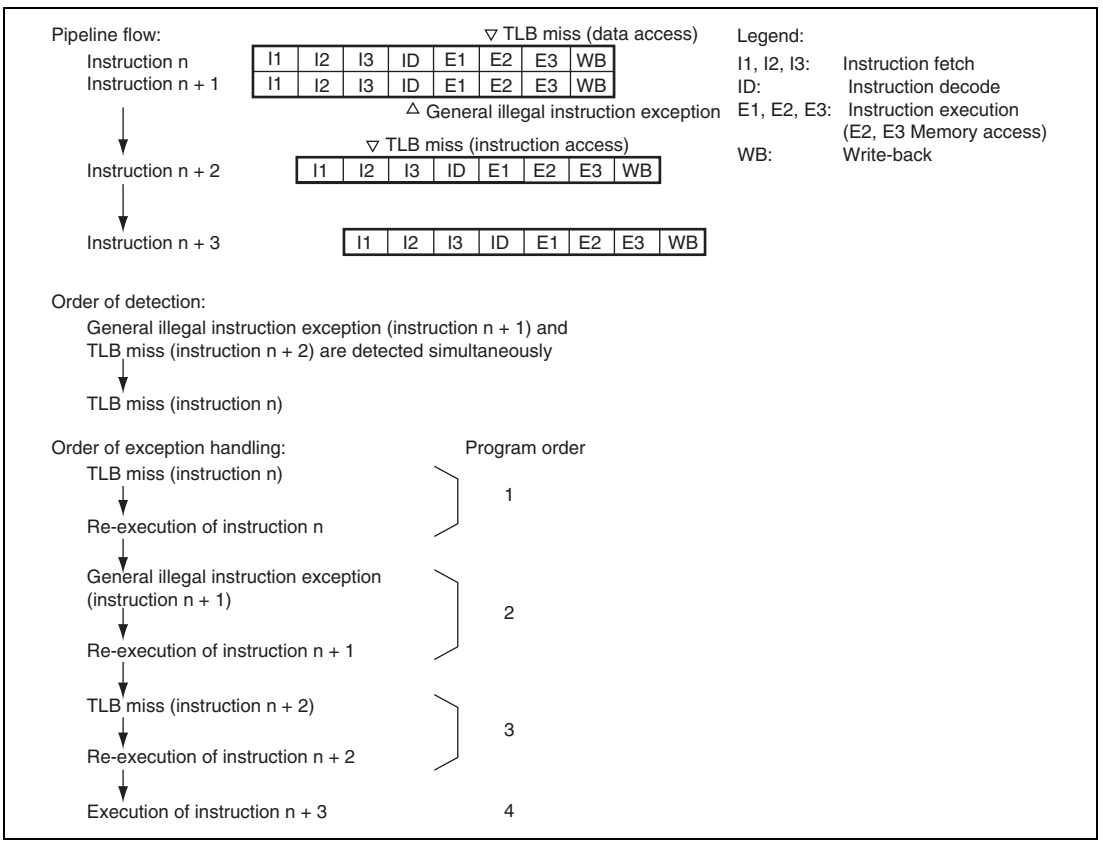


Figure 5.2 Example of General Exception Acceptance Order

5.5.3 Exception Requests and BL Bit

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

When the BL bit in SR is 1 and an general exception other than a user break is generated, the CPU's internal registers and the registers of the other modules are set to their states following a manual reset, and the CPU branches to the same address as in a reset (H'A0000000). For the operation in the event of a user break, see section 31, User Break Controller (UBC). If an ordinary interrupt occurs, the interrupt request is held pending and is accepted after the BL bit has been cleared to 0 by software. If a nonmaskable interrupt (NMI) occurs, it can be held pending or accepted according to the setting made by software. For further details, refer to section 10, Interrupt Controller (INTC).

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

5.5.4 Return from Exception Handling

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

5.6 Description of Exceptions

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

5.6.1 Resets

(1) Power-On Reset

- Condition:

  Power-on reset request
- Operations:

  Exception code H'000 is set in EXPEVT, initialization of CPU0 and its on-chip peripheral modules is carried out, and then a branch is made to the reset vector (H'A0000000). CPU1 is initialized and then enters the module stop state. For details, see the register descriptions in the relevant sections. A power-on reset should be executed when power is supplied.

(2) Manual Reset

- Condition:

  Manual reset request
- Operations:

  Exception code H'020 is set in EXPEVT, initialization of the CPU and on-chip peripheral modules is carried out, and then a branch is made to the reset vector (the address set in the CPUnRESETVEC register). The registers initialized by a power-on reset and manual reset are different. For details, see the register descriptions in the relevant sections.

(3) H-UDI Reset

- Source: SDIR.TI[7:4] = B'0110 (negation) or B'0111 (assertion)
- Transition address: H'A0000000
- Transition operations:

  Exception code H'000 is set in EXPEVT. For CPU0, VBR and SR are initialized, and a branch is made to PC = H'A000 0000. For CPU1, VBR and SR are initialized, and a transition is made to the module stop state.

  CPU and on-chip peripheral module initialization is performed. For details, see section 32, User Debugging Interface (H-UDI), and the register descriptions in the relevant sections.

(4) Instruction TLB Multiple Hit Exception

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

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

  Exception code H'140 is set in EXPEVT. After VBR and SR are initialized, a branch is made to the reset vector (the address set in the CPUnRESETVEC register).

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

(5) Data TLB Multiple-Hit Exception

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

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

  Exception code H'140 is set in EXPEVT. After VBR and SR are initialized, a branch is made to the reset vector (the address set in the CPUnRESETVEC register).

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

5.6.2 General Exceptions

(1) Data TLB Miss Exception

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

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

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

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

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

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

(2) Instruction TLB Miss Exception

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

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

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

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

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

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

(3) Initial Page Write Exception

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

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

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

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

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

(4) Data TLB Protection Violation Exception

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

Table 5.4 UTLB Protection Information (TLB Compatible Mode)

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

Table 5.5 UTLB Protection Information (TLB Extended Mode)

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

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

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

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

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

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

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

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

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

(5) Instruction TLB Protection Violation Exception

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

Table 5.6 ITLB Protection Information (TLB Compatible Mode)

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

Table 5.7 ITLB Protection Information (TLB Extended Mode)

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

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

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

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

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

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

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

(6) Data Address Error

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

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

- Transition operations:

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

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

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

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

(7) Instruction Address Error

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

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

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

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

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

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

(8) Unconditional Trap

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

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

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

(9) General Illegal Instruction Exception

- Sources:
  - — Decoding of an undefined instruction not in a delay slot

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

    Undefined instruction: H'FFFD
  - — Decoding in user mode of a privileged instruction not in a delay slot

    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC instructions that access GBR
- Transition address: VBR + H'00000100
- Transition operations:

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

  Exception code H'180 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined code other than H'FFFD is decoded.

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

(10) Slot Illegal Instruction Exception

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

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

(11) General FPU Disable Exception

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

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

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

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

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

(12) Slot FPU Disable Exception

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

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

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

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

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

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

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

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

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

  For details of PC, etc., when a data break is set, see section 31, User Break Controller (UBC).

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

(14) FPU Exception

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

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

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

5.6.3 Interrupts

(1) NMI (Nonmaskable Interrupt)

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

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

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

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

(2) General Interrupt Request

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

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

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

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

5.6.4 Priority Order with Multiple Exceptions

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

(1) Instructions that Make Two Accesses to Memory

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

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

8. Initial page write exception in second data transfer

(2) Indivisible Delayed Branch Instruction and Delay Slot Instruction

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

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

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

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

5.7 Usage Notes

(1) Return from Exception Handling

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

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

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

A. General exception

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

B. Interrupt

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

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

(3) SPC when an Exception Occurs

A. Re-execution type general exception

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

B. Completion type general exception or interrupt

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

(4) RTE Instruction Delay Slot

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

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

(5) Changing the SR Register Value and Accepting Exception

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

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

Section 6 Floating-Point Unit (FPU)

6.1 Features

The FPU has the following features.

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

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

6.2 Data Formats

6.2.1 Floating-Point Format

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

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

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

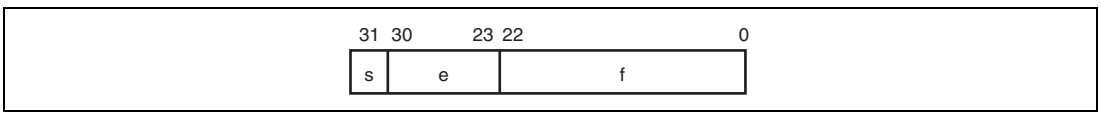


Figure 6.1 Format of Single-Precision Floating-Point Number

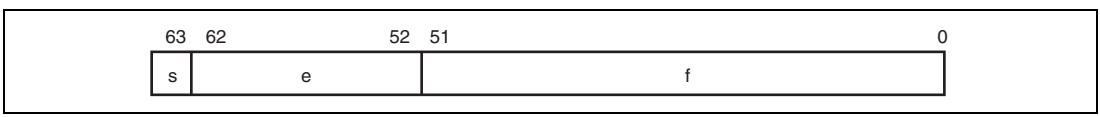


Figure 6.2 Format of Double-Precision Floating-Point Number

The exponent is expressed in biased form, as follows:

$$e = E + bias$$

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

Table 6.1 Floating-Point Number Formats and Parameters

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

Floating-point number value v is determined as follows:

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

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

Table 6.2 Floating-Point Ranges

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

6.2.2 Non-Numbers (NaN)

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

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

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

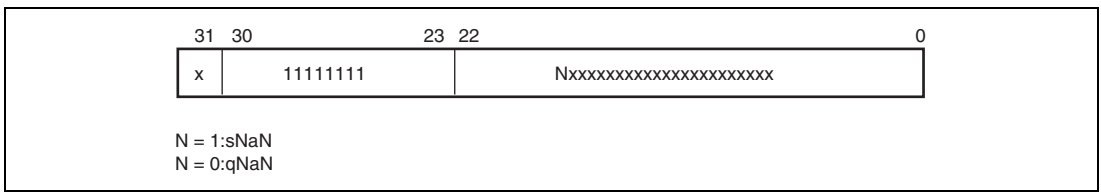


Figure 6.3 Single-Precision NaN Bit Pattern

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

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

Following three instructions are used as transfer instructions between registers.

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

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

The qNAN values as operation results are as follows:

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

See section 10, Instruction Descriptions of the SH-4A Extended Functions Software Manual for details of floating-point operations when a non-number (NaN) is input.

6.2.3 Denormalized Numbers

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

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

When the DN bit in FPSCR is 0, a denormalized number (source operand or operation result) is processed as it is. See section 10, Instruction Descriptions of the SH-4A Extended Functions Software Manual for details of floating-point operations when a denormalized number is input.

6.3 Register Descriptions

6.3.1 Floating-Point Registers

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

1. Floating-point registers, FPRi_BANKj (32 registers)
   FPR0_BANK0 to FPR15_BANK0
   FPR0_BANK1 to FPR15_BANK1
2. Single-precision floating-point registers, FRi (16 registers)
   When FPSCR.FR = 0, FR0 to FR15 are allocated to FPR0_BANK0 to FPR15_BANK0;
   when FPSCR.FR = 1, FR0 to FR15 are allocated to FPR0_BANK1 to FPR15_BANK1.
3. Double-precision floating-point registers, DRi (8 registers): A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14, FR15}
4. Single-precision floating-point vector registers, FVi (4 registers): An FV register comprises four FR registers.
   FV0 = {FR0, FR1, FR2, FR3}, FV4 = {FR4, FR5, FR6, FR7},
   FV8 = {FR8, FR9, FR10, FR11}, FV12 = {FR12, FR13, FR14, FR15}
5. Single-precision floating-point extended registers, XFi (16 registers)
   When FPSCR.FR = 0, XF0 to XF15 are allocated to FPR0_BANK1 to FPR15_BANK1;
   when FPSCR.FR = 1, XF0 to XF15 are allocated to FPR0_BANK0 to FPR15_BANK0.
6. Double-precision floating-point extended registers, XDi (8 registers): An XD register comprises two XF registers.
   XD0 = {XF0, XF1}, XD2 = {XF2, XF3}, XD4 = {XF4, XF5}, XD6 = {XF6, XF7},
   XD8 = {XF8, XF9}, XD10 = {XF10, XF11}, XD12 = {XF12, XF13}, XD14 = {XF14, XF15}

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

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

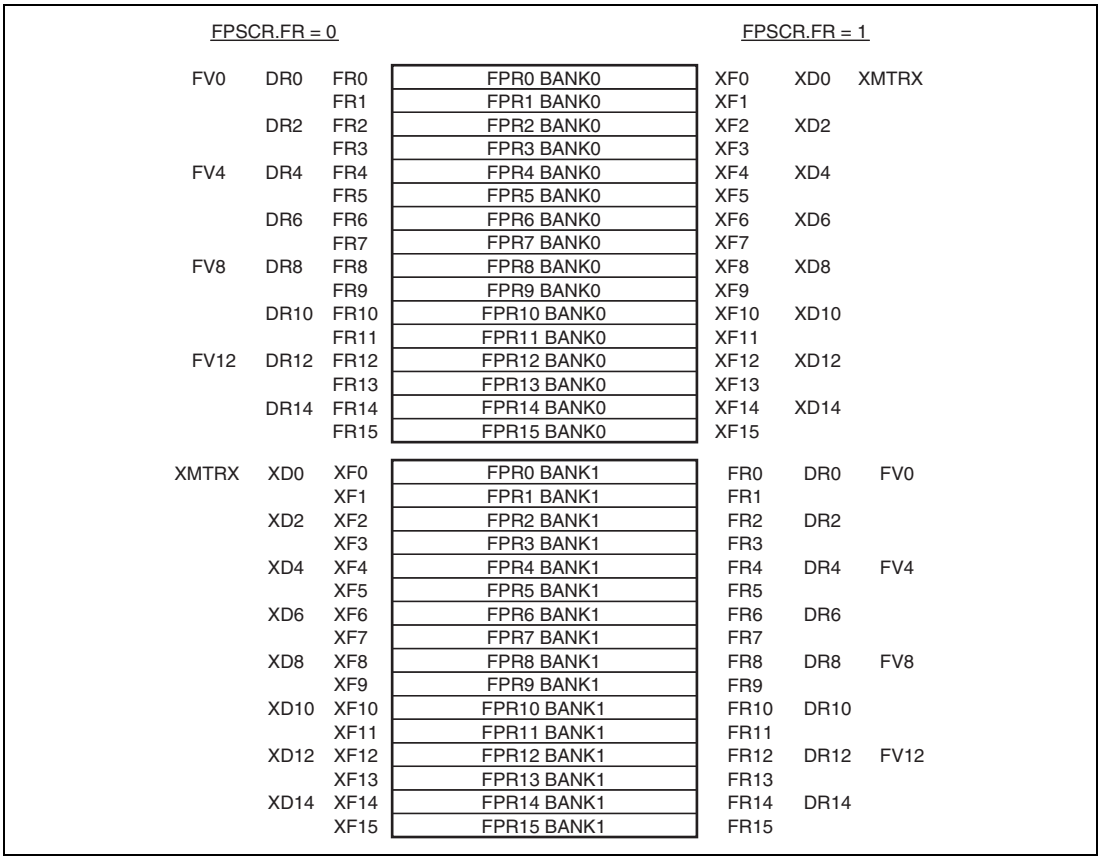


Figure 6.4 Floating-Point Registers

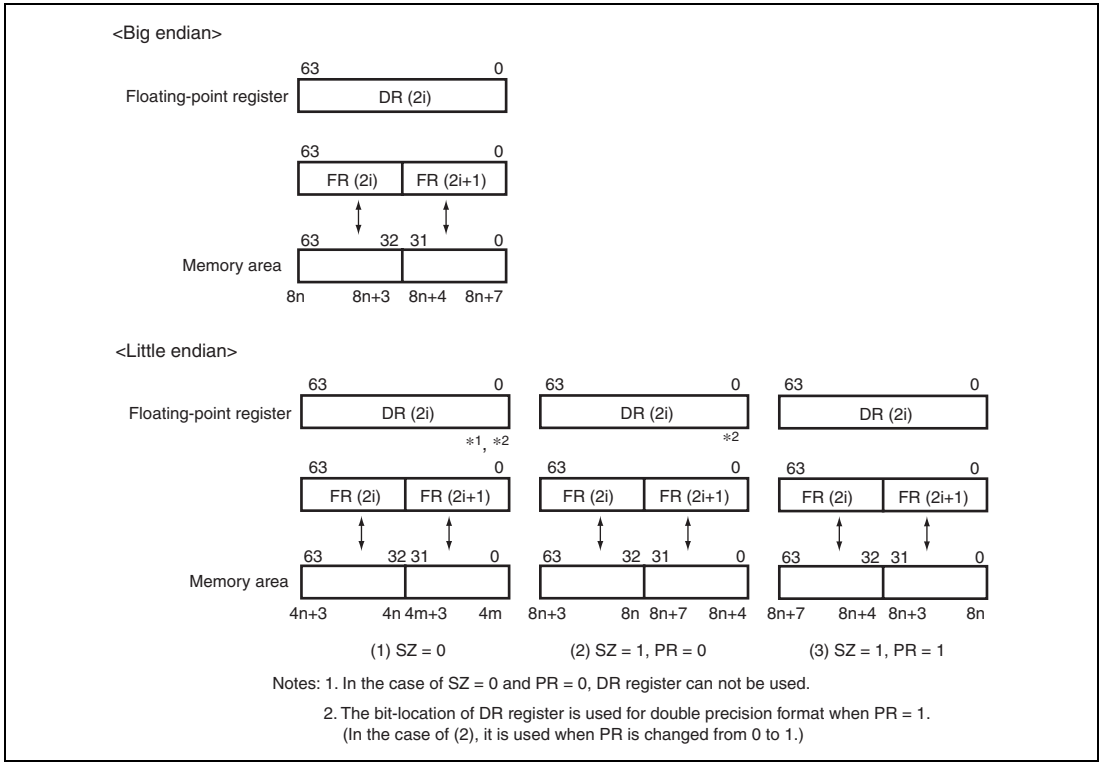
6.3.2 Floating-Point Status/Control Register (FPSCR)

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

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

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

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



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

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

Figure 6.5 Relation between SZ Bit and Endian

Table 6.3 Bit Allocation for FPU Exception Handling

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

6.3.3 Floating-Point Communication Register (FPUL)

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

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

6.4 Rounding

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

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

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

(1) Round to Nearest

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

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

(2) Round to Zero

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

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

6.5 Floating-Point Exceptions

6.5.1 General FPU Disable Exceptions and Slot FPU Disable Exceptions

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

6.5.2 FPU Exception Sources

The exception sources are as follows:

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

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

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

6.5.3 FPU Exception Handling

FPU exception handling is initiated in the following cases:

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

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

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

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

- Invalid operation (V): qNaN is generated as the result.
- Division by zero (Z): Infinity with the same sign as the unrounded value is generated.
- Overflow (O):

  When rounding mode = RZ, the maximum normalized number, with the same sign as the unrounded value, is generated.
  When rounding mode = RN, infinity with the same sign as the unrounded value is generated.
- Underflow (U):

  When FPSCR.DN = 0, a denormalized number with the same sign as the unrounded value, or zero with the same sign as the unrounded value, is generated.
  When FPSCR.DN = 1, zero with the same sign as the unrounded value, is generated.
- Inexact exception (I): An inexact result is generated.

6.6 Graphics Support Functions

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

6.6.1 Geometric Operation Instructions

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

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

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

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

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

This instruction is basically used for the following purposes:

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

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

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

This instruction is basically used for the following purposes:

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

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

(3) FRCHG

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

6.6.2 Pair Single-Precision Data Transfer

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

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

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

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

- FSCHG

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

Section 7 Memory Management Unit (MMU)

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

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

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

Selection between 29-bit address mode and 32-bit address extended mode is made by setting the relevant control register (bit SE in the PASCR register) by software. This LSI supports 32-bit boot mode (the system starts up in 32-bit address extended mode at power-on reset), which is specified through external pins.

Selection between TLB compatible mode and TLB extended mode is made by setting the relevant control register (bit ME in the MMUCR register) by software. The range of physical address mapping is explained through sections 7.1, Overview of MMU, to 7.7, Memory-Mapped TLB Configuration, for the case of 29-bit address mode, which is followed by section 7.8, 32-Bit Address Extended Mode, where differences from 29-bit address mode are explained.

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

7.1 Overview of MMU

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

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

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

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

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

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

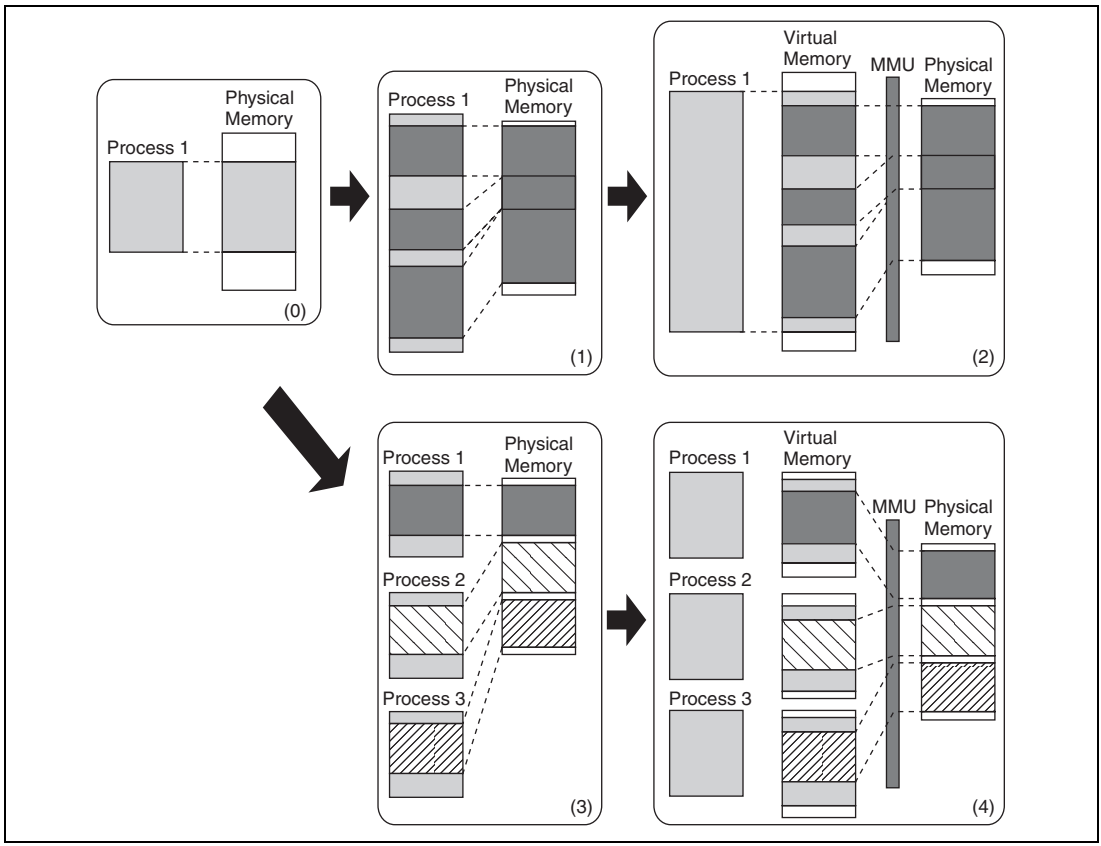


Figure 7.1 Role of MMU

7.1.1 Address Spaces

(1) Virtual Address Space

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

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

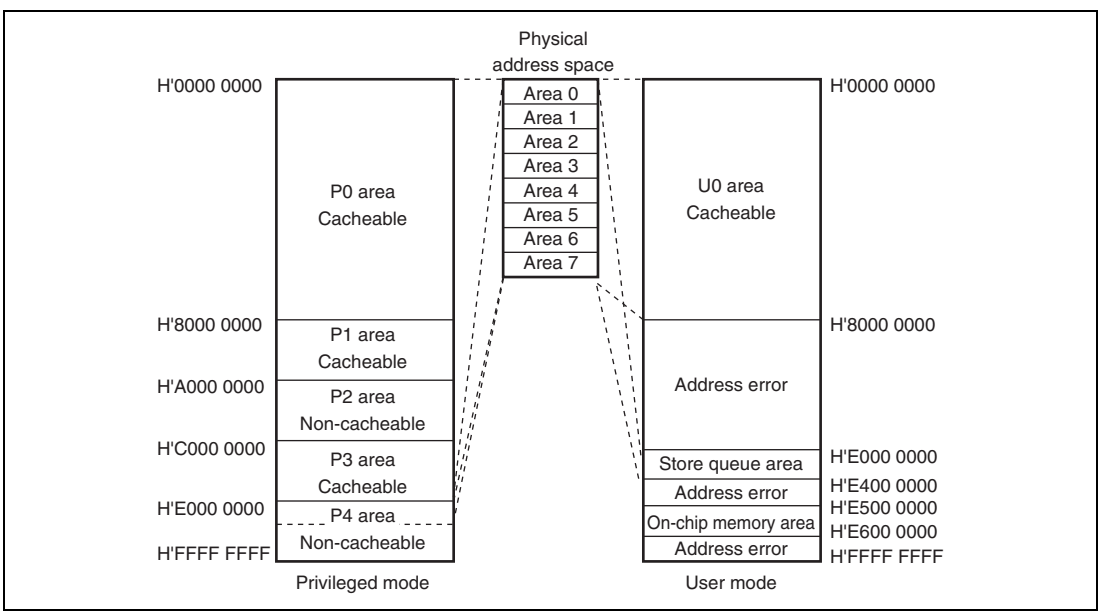


Figure 7.2 Virtual Address Space (AT in MMUCR = 0)

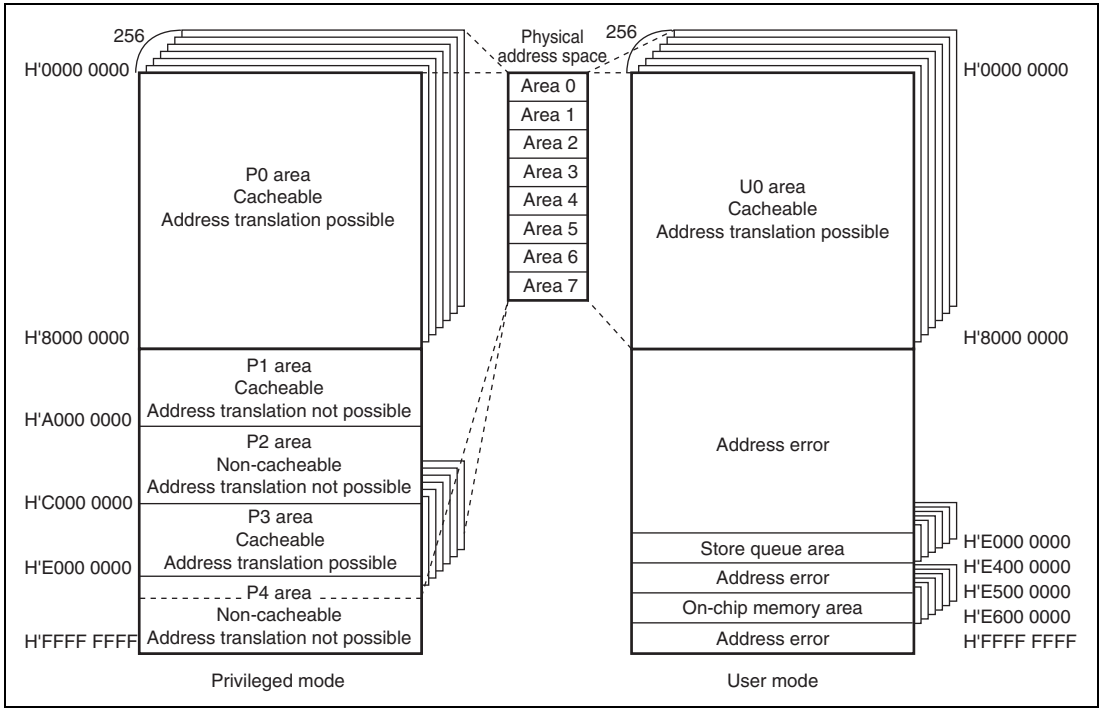


Figure 7.3 Virtual Address Space (AT in MMUCR = 1)

(a) P0, P3, and U0 Areas

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

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

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

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

(b) P1 Area

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

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

(c) P2 Area

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

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

(d) P4 Area

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

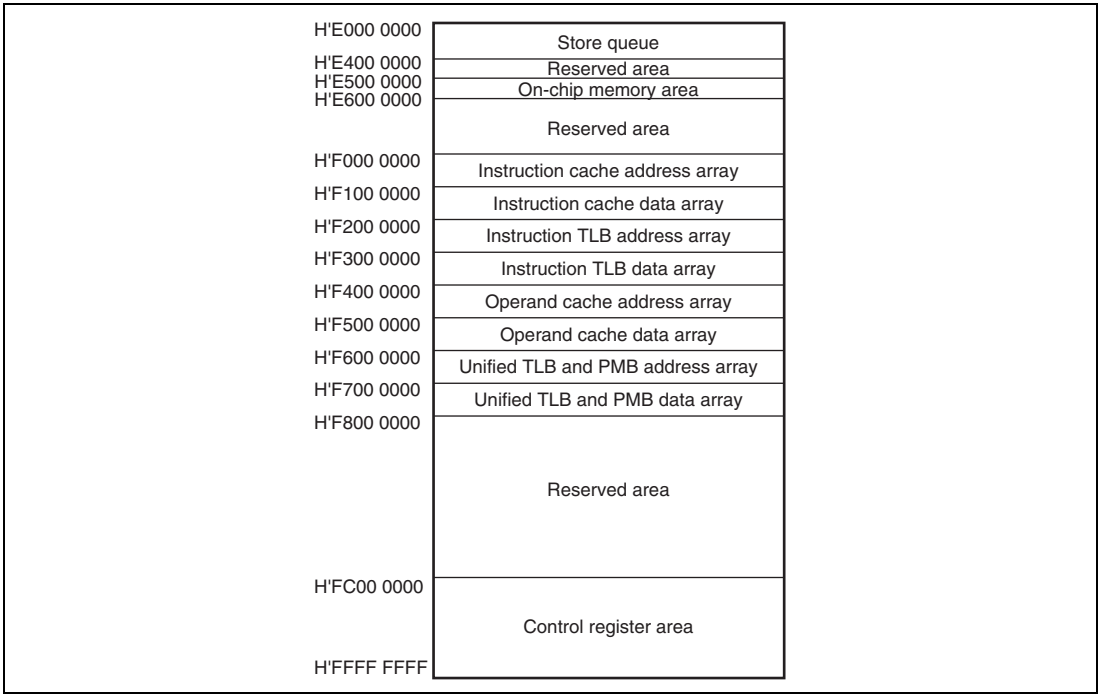


Figure 7.4 P4 Area

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

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

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

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

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

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

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

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

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

The area from H'F610 0000 to H'F61F FFFF is used for direct access to the PMB address array. For details, see section 7.8.5, Memory-Mapped PMB Configuration.

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

The area from H'F710 0000 to H'F71F FFFF is used for direct access to the PMB data array. For details, see section 7.8.5, Memory-Mapped PMB Configuration.

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

(2) Physical Address Space

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

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

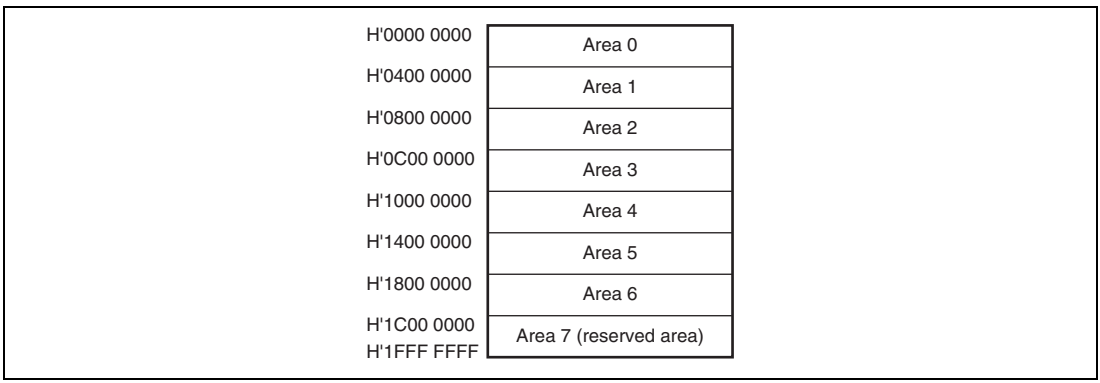


Figure 7.5 Physical Address Space

(3) Address Translation

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

the return from the exception handling routine, the instruction which caused the TLB miss exception is re-executed.

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

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

(5) Address Space Identifier (ASID)

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

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

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

7.2 Register Descriptions

The following registers are related to MMU processing.

Table 7.1 Register Configuration

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

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

Table 7.2 Register States in Each Processing State

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

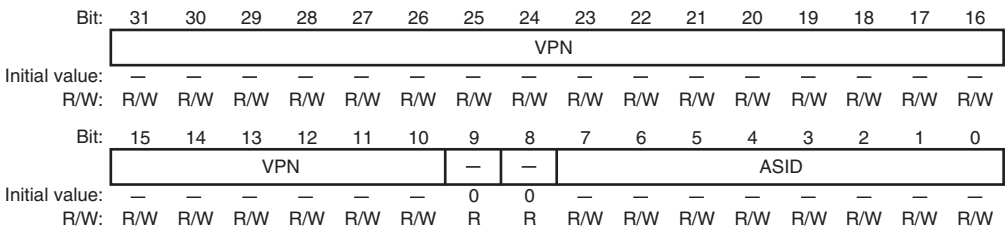
7.2.1 Page Table Entry High Register (PTEH)

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

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

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

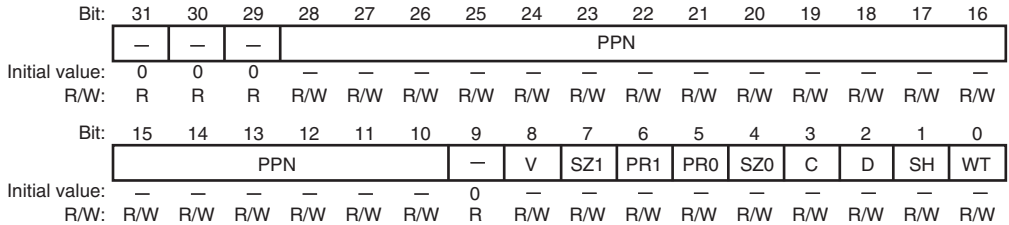
Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.



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

7.2.2 Page Table Entry Low Register (PTEL)

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



Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 10	PPN	Undefined	R/W	Physical Page Number
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	V	Undefined	R/W	Page Management Information The meaning of each bit is same as that of corresponding bit in Common TLB (UTLB). For details, see section 7.3, TLB Functions (TLB Compatible Mode; MMUCR.ME = 0) and section 7.4, TLB Functions (TLB Extended Mode; MMUCR.ME = 1). Note: SZ1, PR1, SZ0, and PR0 bits are valid only in TLB compatible mode.
7	SZ1	Undefined	R/W	
6	PR1	Undefined	R/W	
5	PR0	Undefined	R/W	
4	SZ0	Undefined	R/W	
3	C	Undefined	R/W	
2	D	Undefined	R/W	
1	SH	Undefined	R/W	
0	WT	Undefined	R/W	

7.2.3 Translation Table Base Register (TTB)

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

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

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

7.2.4 TLB Exception Address Register (TEA)

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEA								Virtual address at which MMU exception or address error occurred							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEA								Virtual address at which MMU exception or address error occurred							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.2.5 MMU Control Register (MMUCR)

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

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

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

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

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	LRUI	000000	R/W	Least Recently Used ITLB These bits indicate the ITLB entry to be replaced. The LRU (least recently used) method is used to decide the ITLB entry to be replaced in the event of an ITLB miss. The entry to be purged from the ITLB can be confirmed using the LRUI bits. LRUI is updated by means of the algorithm shown below. x means that updating is not performed. 000xxx: ITLB entry 0 is used 1xx00x: ITLB entry 1 is used x1x1x0: ITLB entry 2 is used xx1x11: ITLB entry 3 is used xxxxxx: Other than above When the LRUI bit settings are as shown below, the corresponding ITLB entry is updated by an ITLB miss. Ensure that values for which "Setting prohibited" is indicated below are not set at the discretion of software. After a power-on or manual reset, the LRUI bits are initialized to 0, and therefore a prohibited setting is never made by a hardware update. x means "don't care". 111xxx: ITLB entry 0 is updated 0xx11x: ITLB entry 1 is updated x0x0x1: ITLB entry 2 is updated xx0x00: ITLB entry 3 is updated Other than above: Setting prohibited
25, 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 18	URB	000000	R/W	UTLB Replace Boundary These bits indicate the UTLB entry boundary at which replacement is to be performed. Valid only when URB ≠ 0.
17, 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	URC	000000	R/W	UTLB Replace Counter These bits serve as a random counter for indicating the UTLB entry for which replacement is to be performed with an LDTLB instruction. This bit is incremented each time the UTLB is accessed. If URB > 0, URC is cleared to 0 when the condition URC = URB is satisfied. Also note that if a value is written to URC by software which results in the condition of URC > URB, incrementing is first performed in excess of URB until URC = H'3F. URC is not incremented by an LDTLB instruction.
9	SQMD	0	R/W	Store Queue Mode Specifies the right of access to the store queues. 0: User/privileged access possible 1: Privileged access possible (address error exception in case of user access)
8	SV	0	R/W	Single Virtual Memory Mode/Multiple Virtual Memory Mode Switching When this bit is changed, ensure that 1 is also written to the TI bit. 0: Multiple virtual memory mode 1: Single virtual memory mode
7	ME	0	R/W	TLB Extended Mode Switching 0: TLB compatible mode 1: TLB extended mode For modifying the ME bit value, always set the TI bit to 1 to invalidate the contents of ITLB and UTLB. The selection of TLB operating mode made by the ME bit does not affect the functionality or operation of the PMB.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TI	0	R/W	TLB Invalidate Bit Writing 1 to this bit invalidates (clears to 0) all valid UTLB/ITLB bits. This bit is always read as 0.
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	AT	0	R/W	Address Translation Enable Bit These bits enable or disable the MMU. 0: MMU disabled 1: MMU enabled MMU exceptions are not generated when the AT bit is 0. In the case of software that does not use the MMU, the AT bit should be cleared to 0.

7.2.6 Page Table Entry Assistance Register (PTEA)

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 8	EPR	Undefined	R/W	Page Control Information
7 to 4	ESZ	Undefined	R/W	Each bit has the same function as the corresponding bit of the unified TLB (UTLB). For details, see section 7.4, TLB Functions (TLB Extended Mode; MMUCR.ME = 1)
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

7.2.7 Physical Address Space Control Register (PASCR)

PASCR controls the operation in the physical address space.

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

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

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

7.2.8 Instruction Re-Fetch Inhibit Control Register (IRMCR)

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

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

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	R2	0	R/W	Re-Fetch Inhibit 2 after Register Change When MMUCR, PASCR, CCR, PTEH, or RAMCR is changed, this bit controls whether re-fetch is performed for the next instruction. 0: Re-fetch is performed 1: Re-fetch is not performed
3	R1	0	R/W	Re-Fetch Inhibit 1 after Register Change When a register allocated in addresses H'FF200000 to H'FF2FFFFF is changed, this bit controls whether re-fetch is performed for the next instruction. 0: Re-fetch is performed 1: Re-fetch is not performed

Bit	Bit Name	Initial Value	R/W	Description
2	LT	0	R/W	Re-Fetch Inhibit after LDTLB Execution This bit controls whether re-fetch is performed for the next instruction after the LDTLB instruction has been executed. 0: Re-fetch is performed 1: Re-fetch is not performed
1	MT	0	R/W	Re-Fetch Inhibit after Writing Memory-Mapped TLB This bit controls whether re-fetch is performed for the next instruction after writing memory-mapped ITLB/UTLB while the AT bit in MMUCR is set to 1. 0: Re-fetch is performed 1: Re-fetch is not performed
0	MC	0	R/W	Re-Fetch Inhibit after Writing Memory-Mapped IC This bit controls whether re-fetch is performed for the next instruction after writing memory-mapped IC while the ICE bit in CCR is set to 1. 0: Re-fetch is performed 1: Re-fetch is not performed

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

7.3.1 Unified TLB (UTLB) Configuration

The UTLB is used for the following two purposes:

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

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

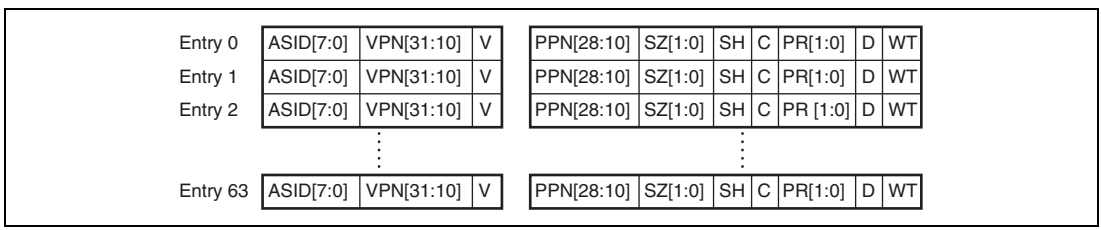


Figure 7.6 UTLB Configuration (TLB Compatible Mode)

Legend:

- VPN: Virtual page number
  For 1-Kbyte page: Upper 22 bits of virtual address
  For 4-Kbyte page: Upper 20 bits of virtual address
  For 64-Kbyte page: Upper 16 bits of virtual address
  For 1-Mbyte page: Upper 12 bits of virtual address
- ASID: Address space identifier
  Indicates the process that can access a virtual page.
  In single virtual memory mode and user mode, or in multiple virtual memory mode, if the SH bit is 0, this identifier is compared with the ASID in PTEH when address comparison is performed.

- SH: Share status bit
  When 0, pages are not shared by processes.
  When 1, pages are shared by processes.
- SZ[1:0]: Page size bits
  Specify the page size.
  00: 1-Kbyte page
  01: 4-Kbyte page
  10: 64-Kbyte page
  11: 1-Mbyte page
- V: Validity bit
  Indicates whether the entry is valid.
  0: Invalid
  1: Valid
  Cleared to 0 by a power-on reset.
  Not affected by a manual reset.
- PPN: Physical page number
  Upper 22 bits of the physical address of the physical page number.
  With a 1-Kbyte page, PPN[28:10] are valid.
  With a 4-Kbyte page, PPN[28:12] are valid.
  With a 64-Kbyte page, PPN[28:16] are valid.
  With a 1-Mbyte page, PPN[28:20] are valid.
  The synonym problem must be taken into account when setting the PPN (see section 7.5.5, Avoiding Synonym Problems).
- PR[1:0]: Protection key data
  2-bit data expressing the page access right as a code.
  00: Can be read from only in privileged mode
  01: Can be read from and written to in privileged mode
  10: Can be read from only in privileged or user mode
  11: Can be read from and written to in privileged mode or user mode
- C: Cacheability bit
  Indicates whether a page is cacheable.
  0: Not cacheable
  1: Cacheable
  When the control register area is mapped, this bit must be cleared to 0.

- D: Dirty bit
  Indicates whether a write has been performed to a page.
  0: Write has not been performed
  1: Write has been performed
- WT: Write-through bit
  Specifies the cache write mode.
  0: Copy-back mode
  1: Write-through mode

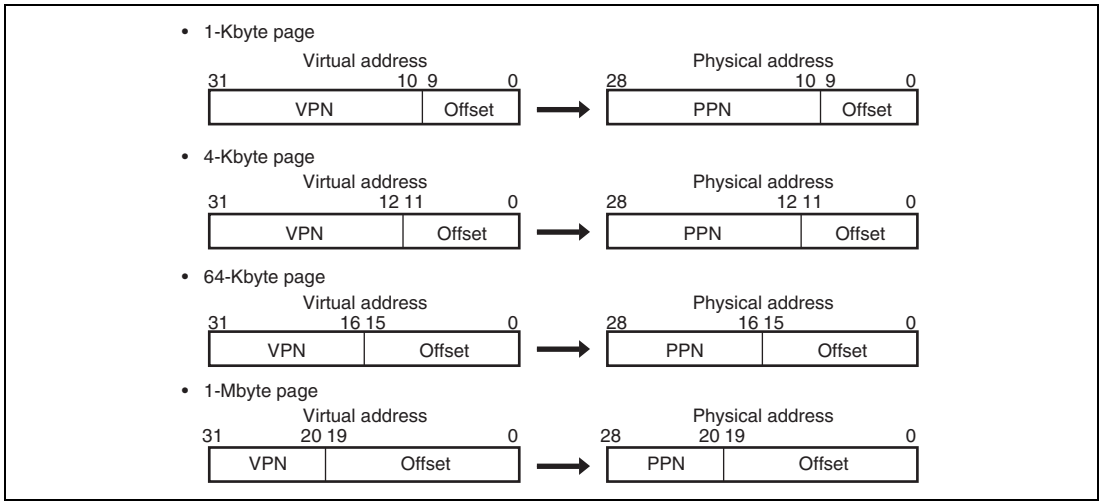


Figure 7.7 Relationship between Page Size and Address Format (TLB Compatible Mode)

7.3.2 Instruction TLB (ITLB) Configuration

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

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

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

Figure 7.8 ITLB Configuration (TLB Compatible Mode)

7.3.3 Address Translation Method

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

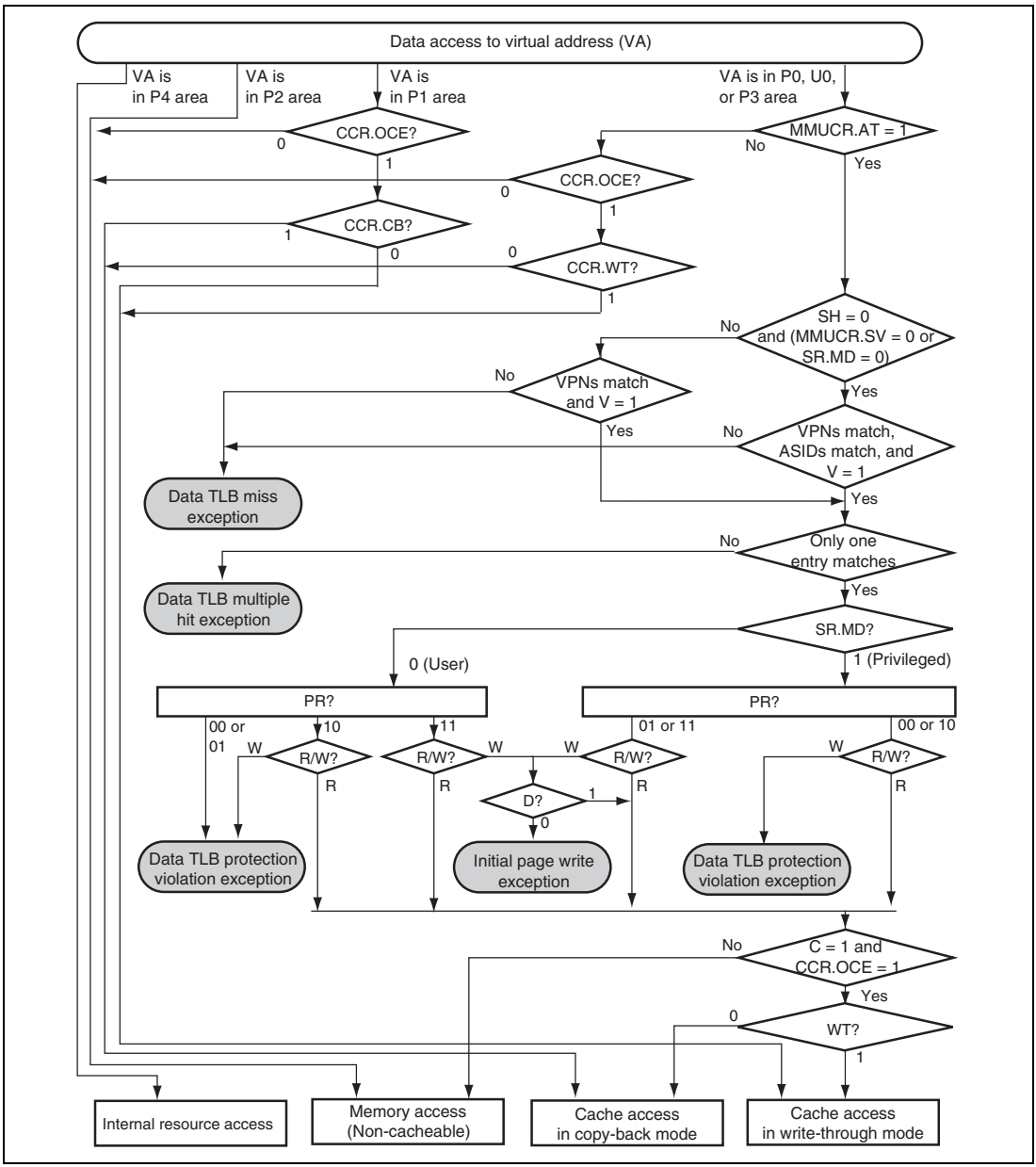


Figure 7.9 Flowchart of Memory Access Using UTLB (TLB Compatible Mode)

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

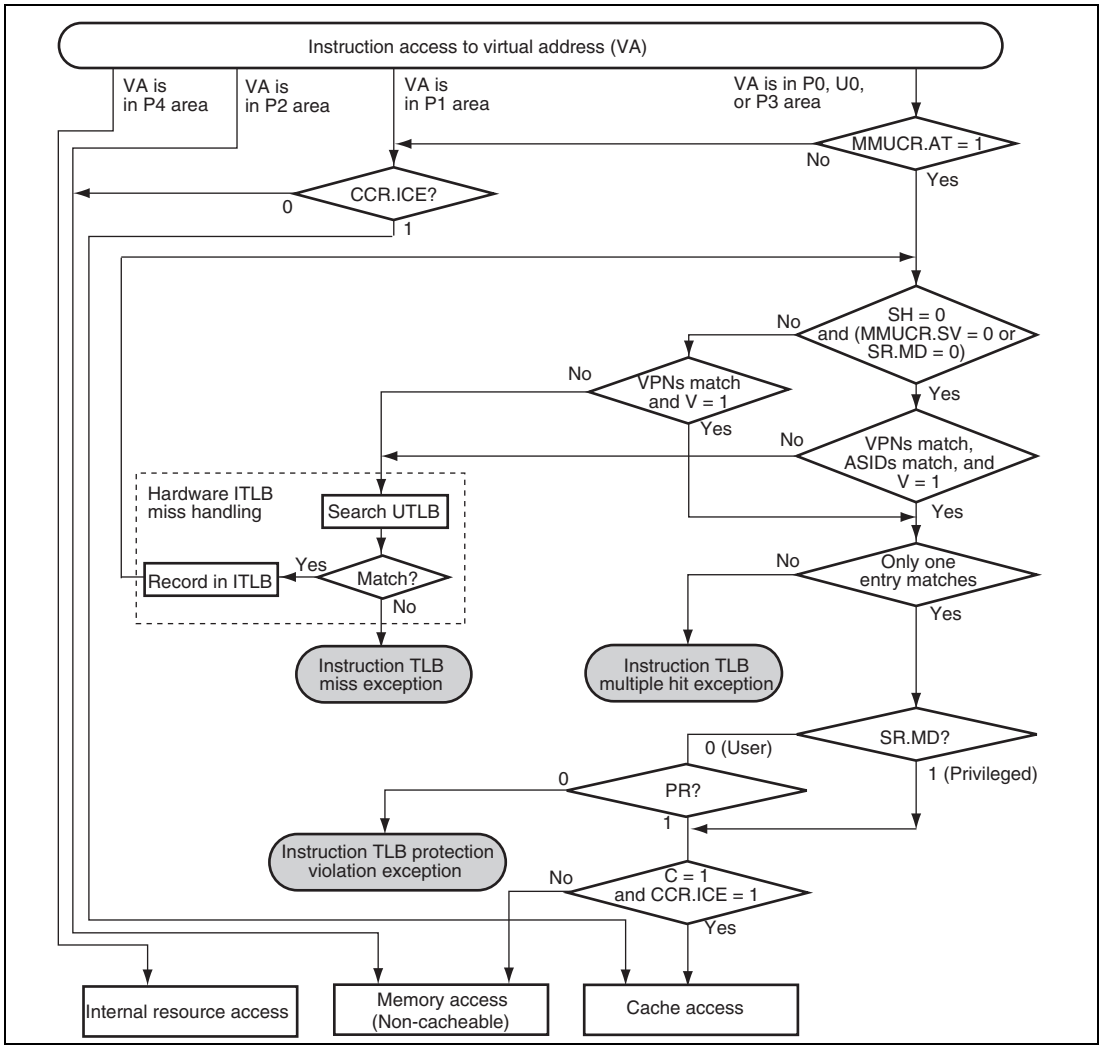


Figure 7.10 Flowchart of Memory Access Using ITLB (TLB Compatible Mode)

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

7.4.1 Unified TLB (UTLB) Configuration

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

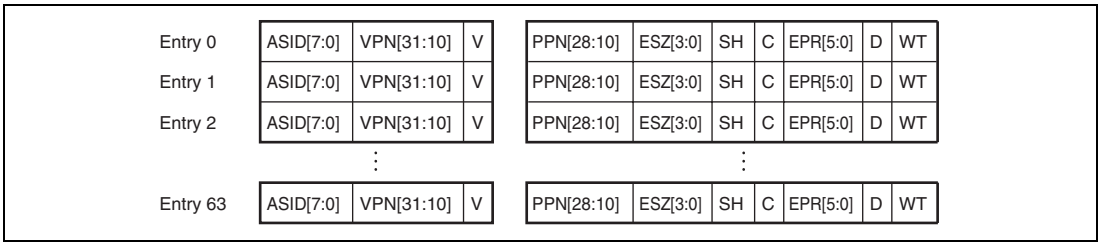


Figure 7.11 UTLB Configuration (TLB Extended Mode)

Legend:

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

- ESZ: Page size bits
  Specify the page size.
  0000: 1-Kbyte page
  0001: 4-Kbyte page
  0010: 8-Kbyte page
  0100: 64-Kbyte page
  0101: 256-Kbyte page
  0111: 1-Mbyte page
  1000: 4-Mbyte page
  1100: 64-Mbyte page

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

- V: Validity bit
  Indicates whether the entry is valid.
  0: Invalid
  1: Valid
  Cleared to 0 by a power-on reset.
  Not affected by a manual reset.
- PPN: Physical page number
  Upper 19 bits of the physical address.
  With a 1-Kbyte page, PPN[28:10] are valid.
  With a 4-Kbyte page, PPN[28:12] are valid.
  With a 8-Kbyte page, PPN[28:13] are valid.
  With a 64-Kbyte page, PPN[28:16] are valid.
  With a 256-Kbyte page, PPN[28:18] are valid.
  With a 1-Mbyte page, PPN[28:20] are valid.
  With a 4-Mbyte page, PPN[28:22] are valid.
  With a 64-Mbyte page, PPN[28:26] are valid.
  The synonym problem must be taken into account when setting the PPN (see section 7.5.5, Avoiding Synonym Problems).

- EPR: Protection key data

  6-bit data expressing the page access right as a code.

  Reading, writing, and execution (instruction fetch) in privileged mode and reading, writing, and execution (instruction fetch) in user mode can be set independently. Each bit is disabled by 0 and enabled by 1.

  EPR[5]: Reading in privileged mode

  EPR[4]: Writing in privileged mode

  EPR[3]: Execution in privileged mode (instruction fetch)

  EPR[2]: Reading in user mode

  EPR[1]: Writing in user mode

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

- C: Cacheability bit

  Indicates whether a page is cacheable.

  0: Not cacheable

  1: Cacheable

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

- D: Dirty bit

  Indicates whether a write has been performed to a page.

  0: Write has not been performed.

  1: Write has been performed.

- WT: Write-through bit

  Specifies the cache write mode.

  0: Copy-back mode

  1: Write-through mode

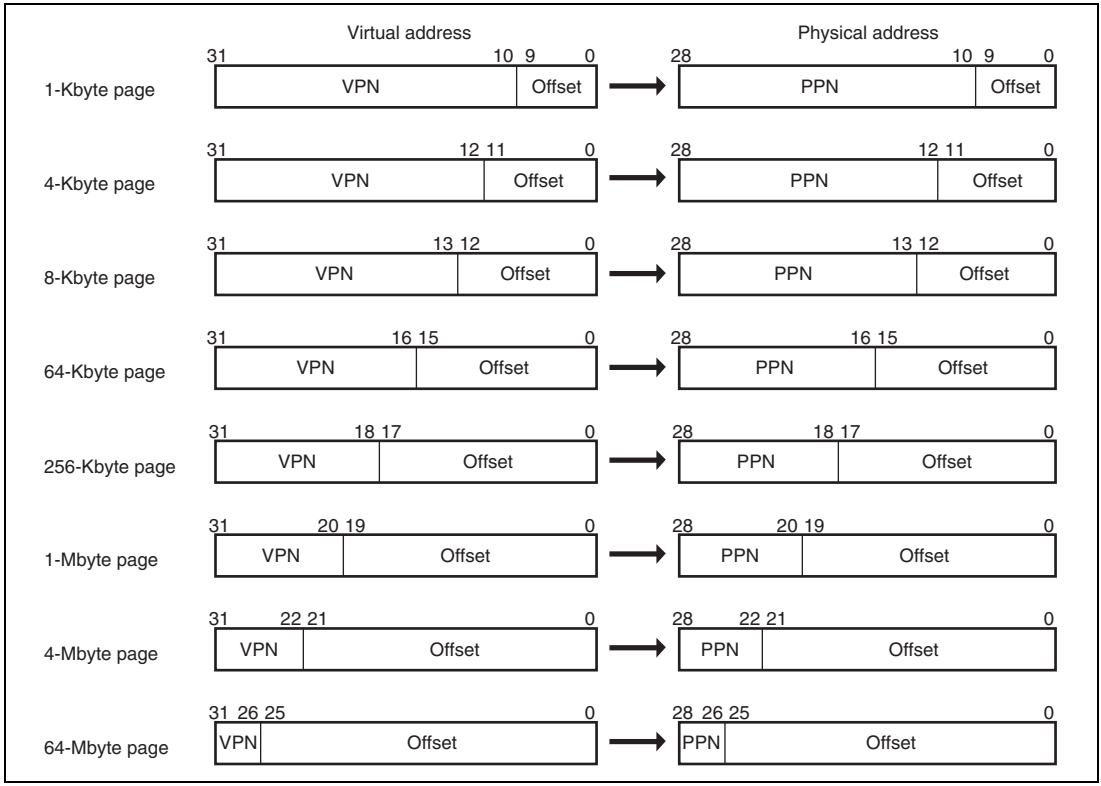


Figure 7.12 Relationship between Page Size and Address Format (TLB Extended Mode)

7.4.2 Instruction TLB (ITLB) Configuration

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

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

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

Figure 7.13 ITLB Configuration (TLB Extended Mode)

7.4.3 Address Translation Method

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

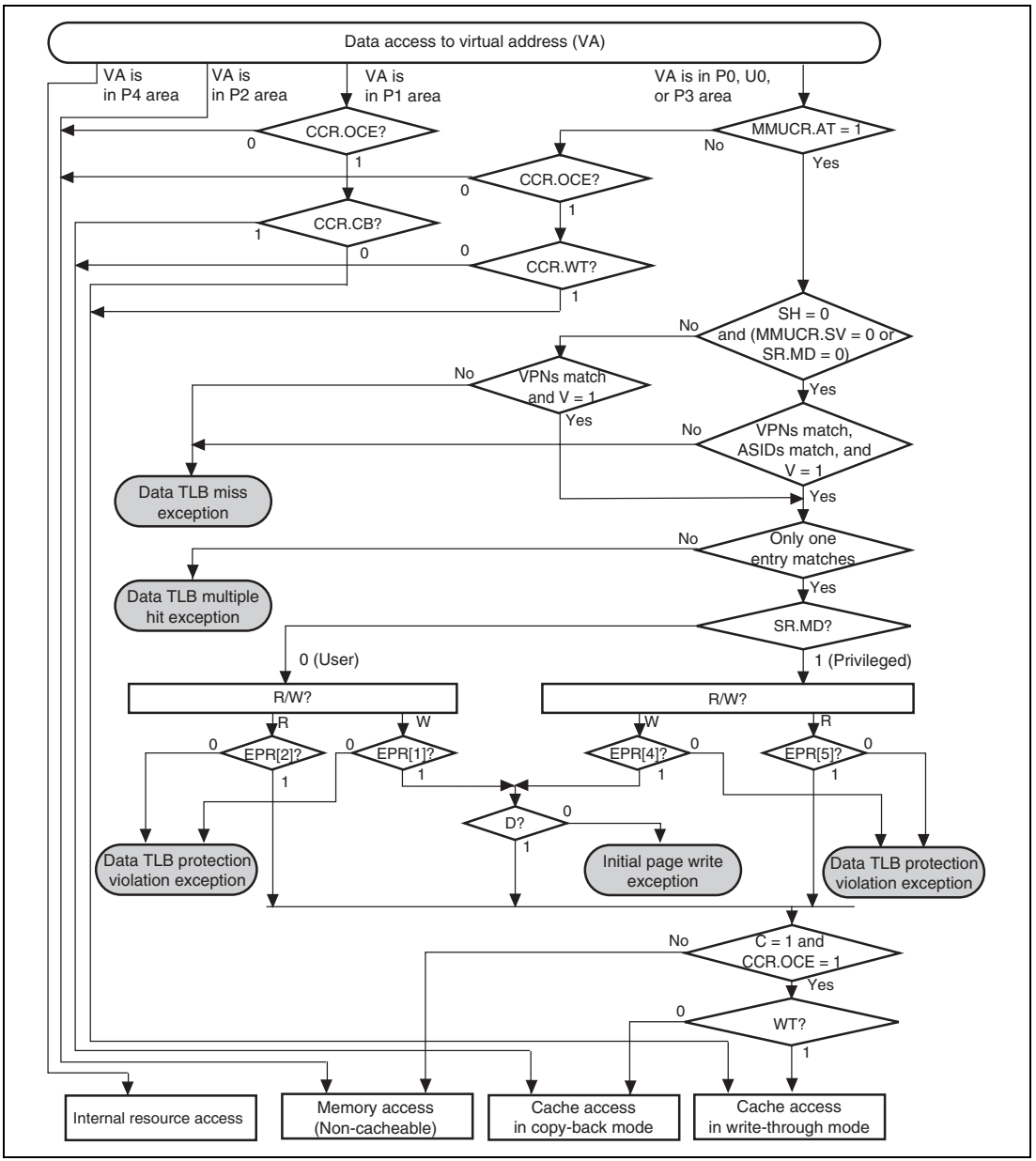


Figure 7.14 Flowchart of Memory Access Using UTLB (TLB Extended Mode)

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

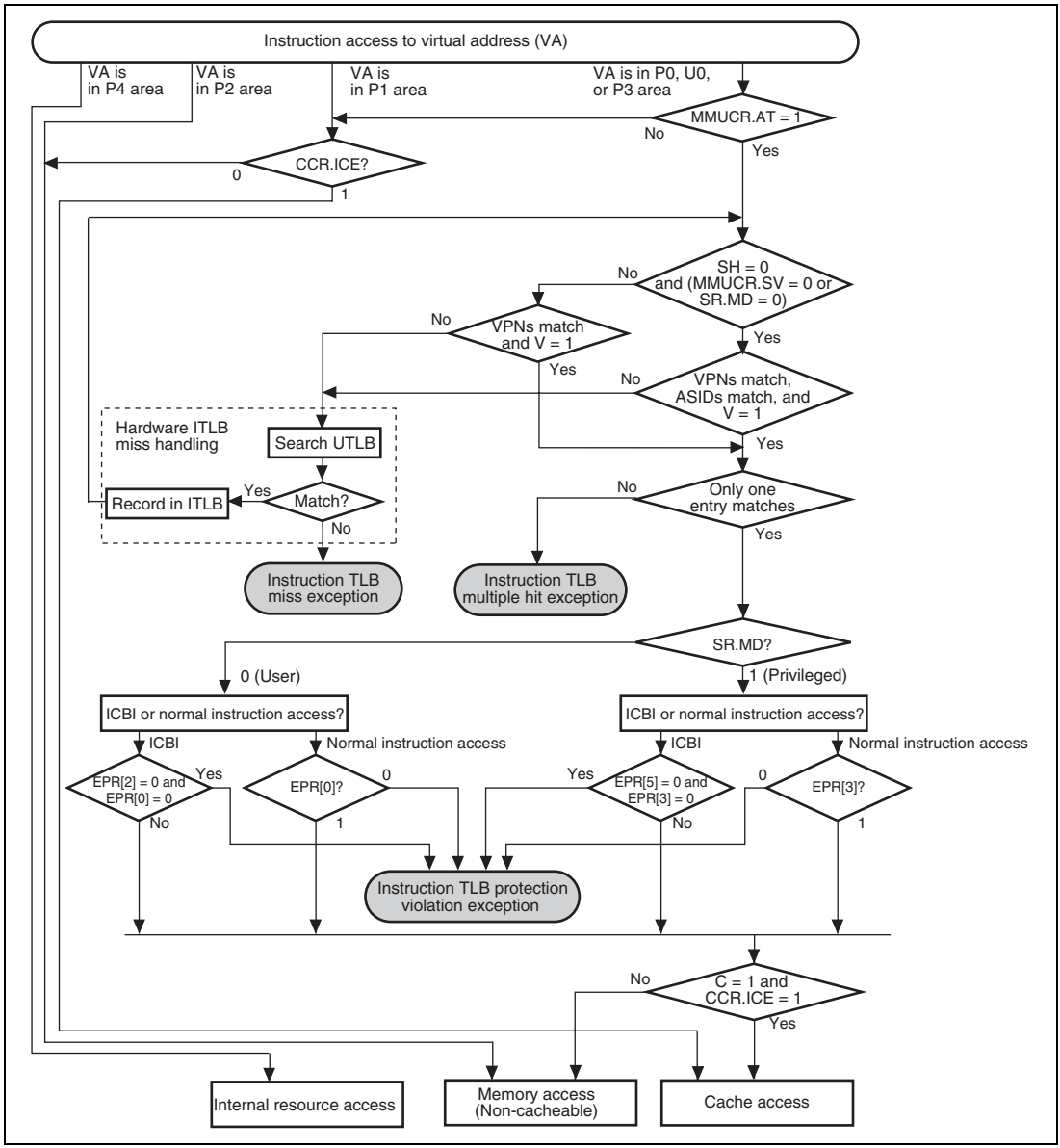


Figure 7.15 Flowchart of Memory Access Using ITLB (TLB Extended Mode)

7.5 MMU Functions

7.5.1 MMU Hardware Management

This LSI supports the following MMU functions.

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

7.5.2 MMU Software Management

Software processing for the MMU consists of the following:

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

7.5.3 MMU Instruction (LDTLB)

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

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

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

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

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

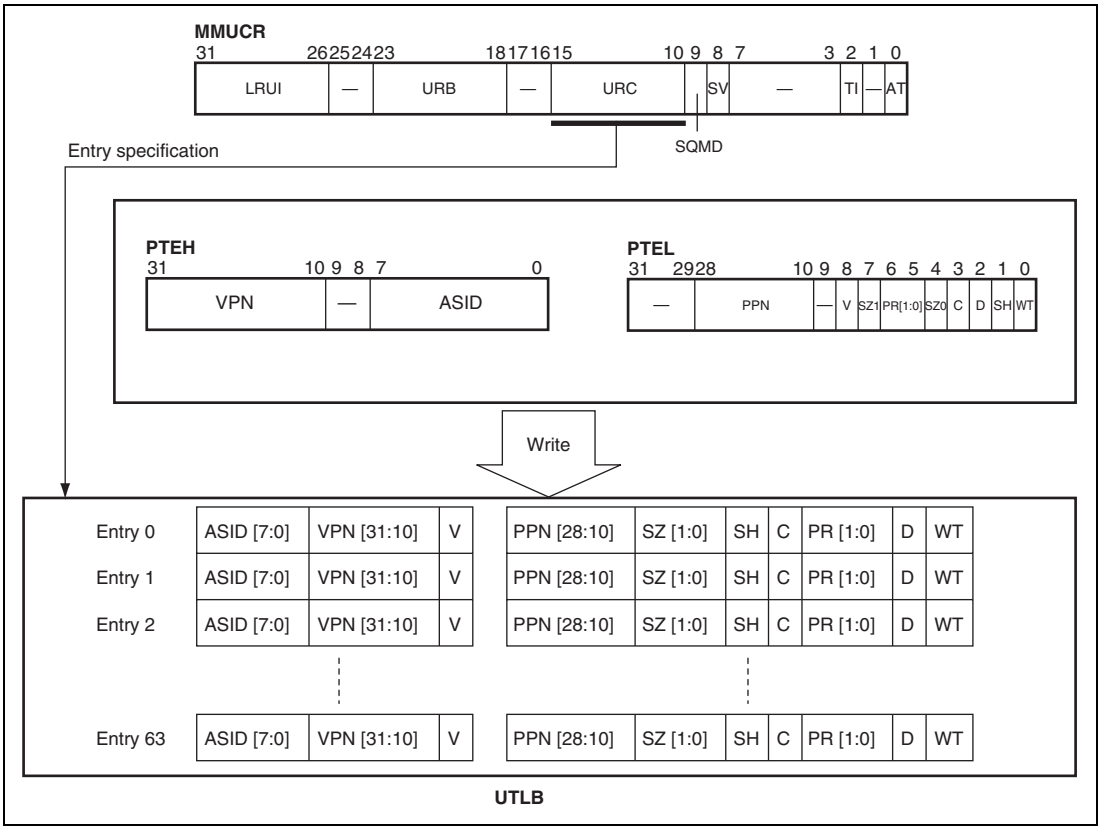


Figure 7.16 Operation of LDTLB Instruction (TLB Compatible Mode)

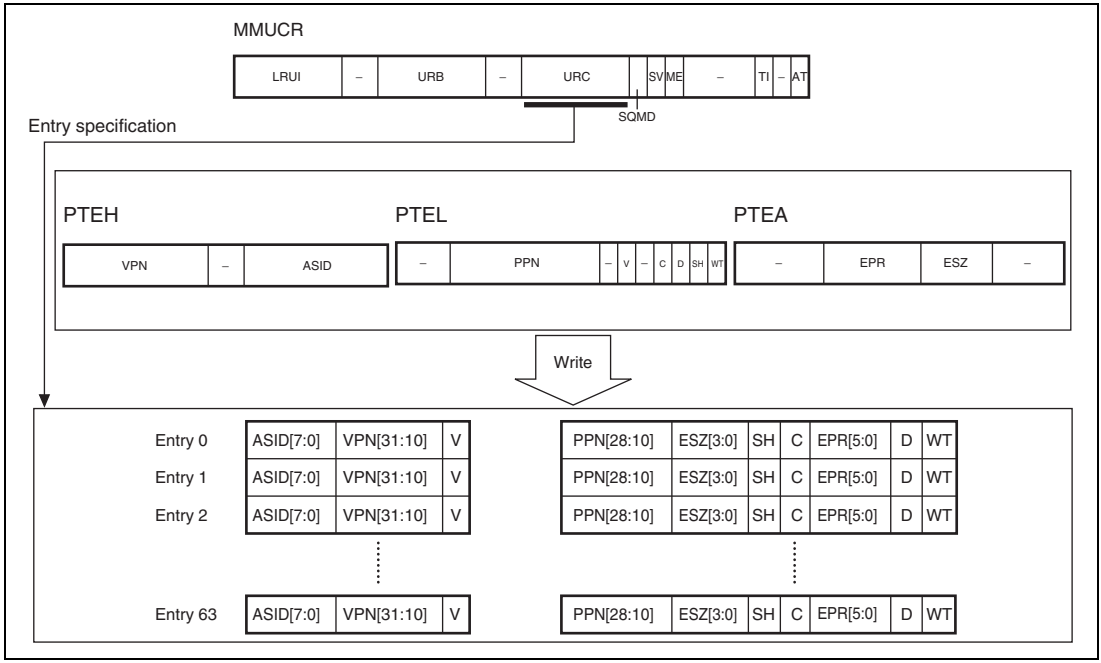


Figure 7.17 Operation of LDTLB Instruction (TLB Extended Mode)

7.5.4 Hardware ITLB Miss Handling

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

7.5.5 Avoiding Synonym Problems

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

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

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

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

7.6 MMU Exceptions

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

7.6.1 Instruction TLB Multiple Hit Exception

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

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

(1) Hardware Processing

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

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

(2) Software Processing (Reset Routine)

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

7.6.2 Instruction TLB Miss Exception

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

(1) Hardware Processing

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

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

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

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

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

For the execution of the LDTLB instruction, see section 7.10.1, Note on Using LDTLB Instruction.

7.6.3 Instruction TLB Protection Violation Exception

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

(1) Hardware Processing

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

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

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

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

7.6.4 Data TLB Multiple Hit Exception

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

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

(1) Hardware Processing

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

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

(2) Software Processing (Reset Routine)

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

7.6.5 Data TLB Miss Exception

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

(1) Hardware Processing

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

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'040 in the case of a read, or H'060 in the case of a write in EXPEVT (OCBP, OCBWB: read; OCBI, MOVCA.L: write).
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.

6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0400 to the contents of VBR, and starts the data TLB miss exception handling routine.

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

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

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

For the execution of the LDTLB instruction, see section 7.10.1, Note on Using LDTLB Instruction.

7.6.6 Data TLB Protection Violation Exception

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

(1) Hardware Processing

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

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

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

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

7.6.7 Initial Page Write Exception

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

(1) Hardware Processing

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

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

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

Software is responsible for the following processing:

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

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

7.7 Memory-Mapped TLB Configuration

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

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

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

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

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

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

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

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

7.7.1 ITLB Address Array

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

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

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

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

1. ITLB address array read

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

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

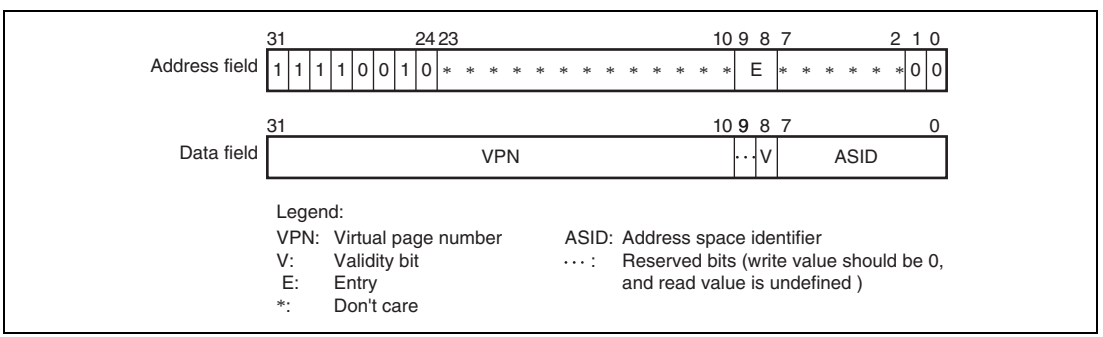


Figure 7.18 Memory-Mapped ITLB Address Array

7.7.2 ITLB Data Array (TLB Compatible Mode)

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

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

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

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

1. ITLB data array read

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

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

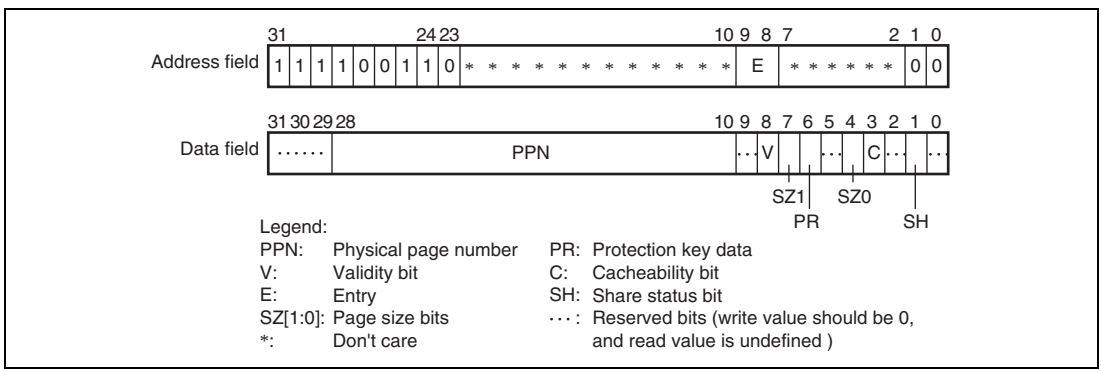


Figure 7.19 Memory-Mapped ITLB Data Array (TLB Compatible Mode)

7.7.3 ITLB Data Array (TLB Extended Mode)

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

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

(1) ITLB Data Array 1

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

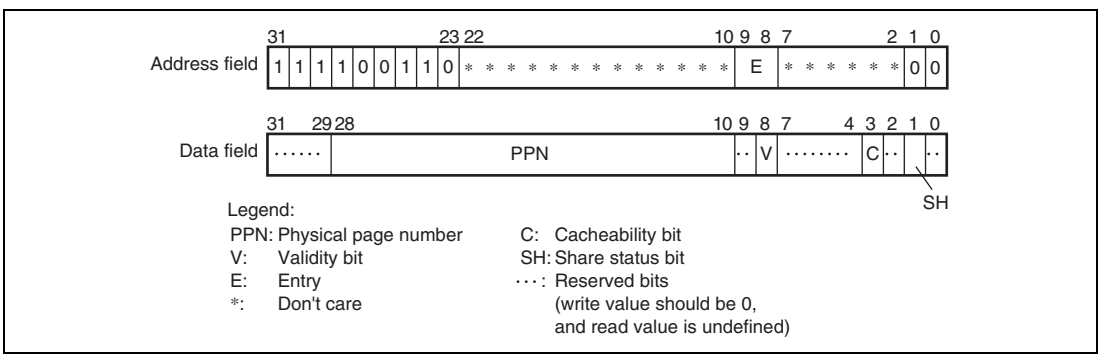


Figure 7.20 Memory-Mapped ITLB Data Array 1 (TLB Extended Mode)

(2) ITLB Data Array 2

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

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

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

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

1. ITLB data array 2 read

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

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

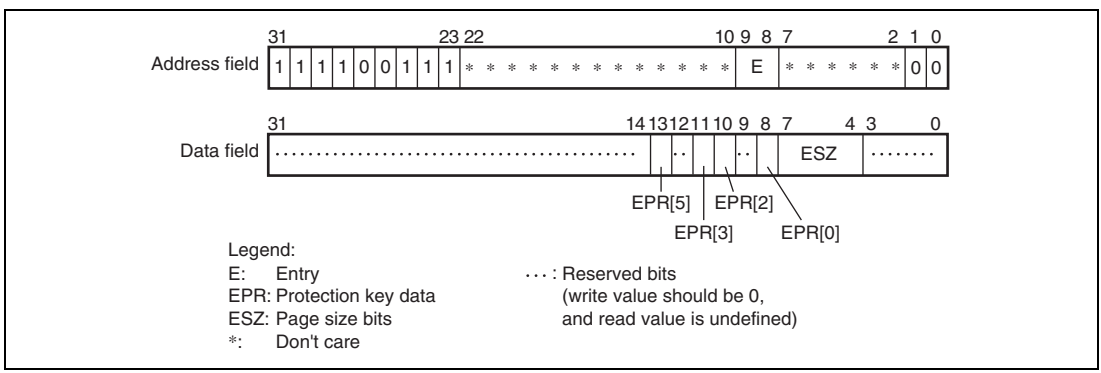


Figure 7.21 Memory-Mapped ITLB Data Array 2 (TLB Extended Mode)

7.7.4 UTLB Address Array

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

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

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

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

1. UTLB address array read

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

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

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

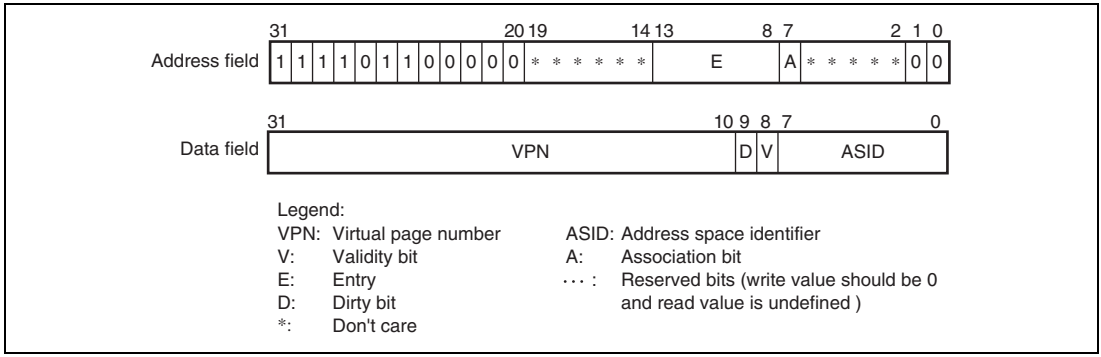


Figure 7.22 Memory-Mapped UTLB Address Array

7.7.5 UTLB Data Array (TLB Compatible Mode)

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

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

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

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

1. UTLB data array read

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

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

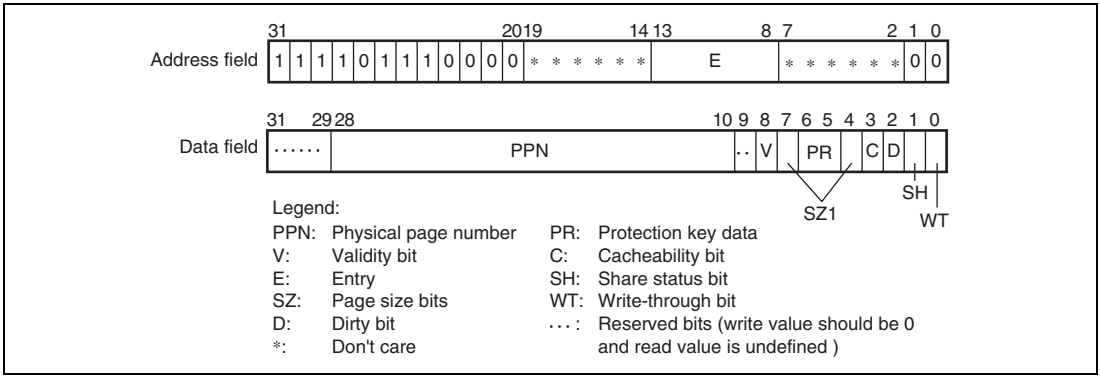


Figure 7.23 Memory-Mapped UTLB Data Array (TLB Compatible Mode)

7.7.6 UTLB Data Array (TLB Extended Mode)

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

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

(1) UTLB Data Array 1

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

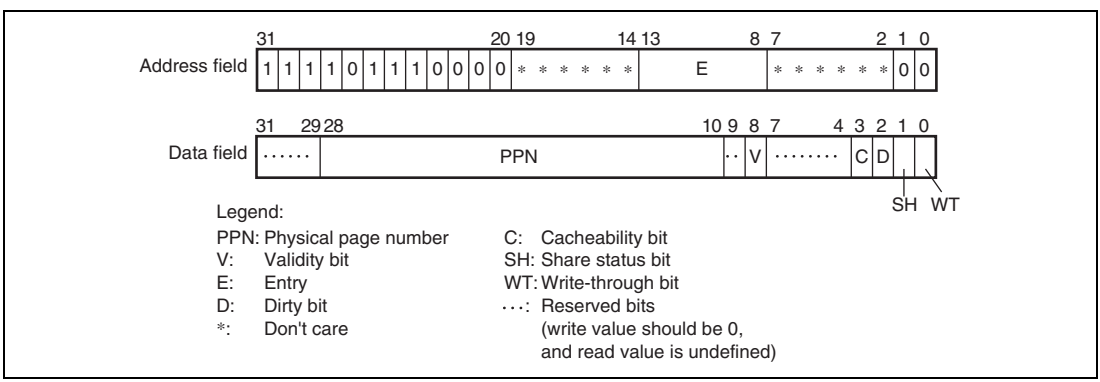


Figure 7.24 Memory-Mapped UTLB Data Array 1 (TLB Extended Mode)

(2) UTLB Data Array 2

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

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

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

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

1. UTLB data array 2 read

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

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

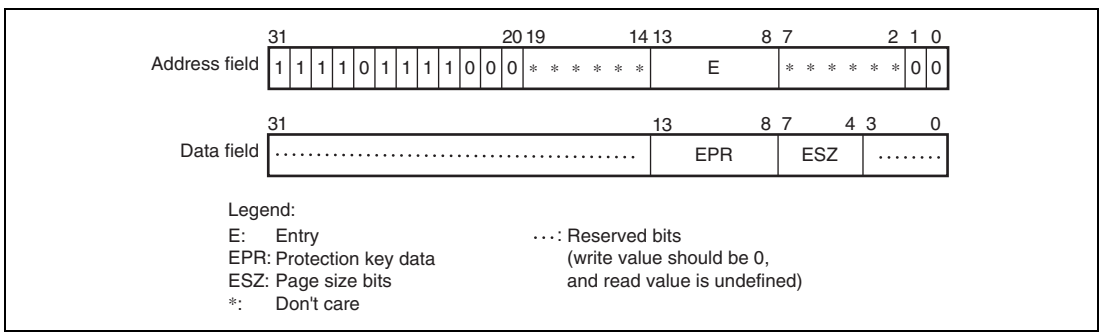


Figure 7.25 Memory-Mapped UTLB Data Array 2 (TLB Extended Mode)

7.8 32-Bit Address Extended Mode

Setting the SE bit in PASCR to 1 changes mode from 29-bit address mode which handles the 29-bit physical address space to 32-bit address extended mode which handles the 32-bit physical address space.

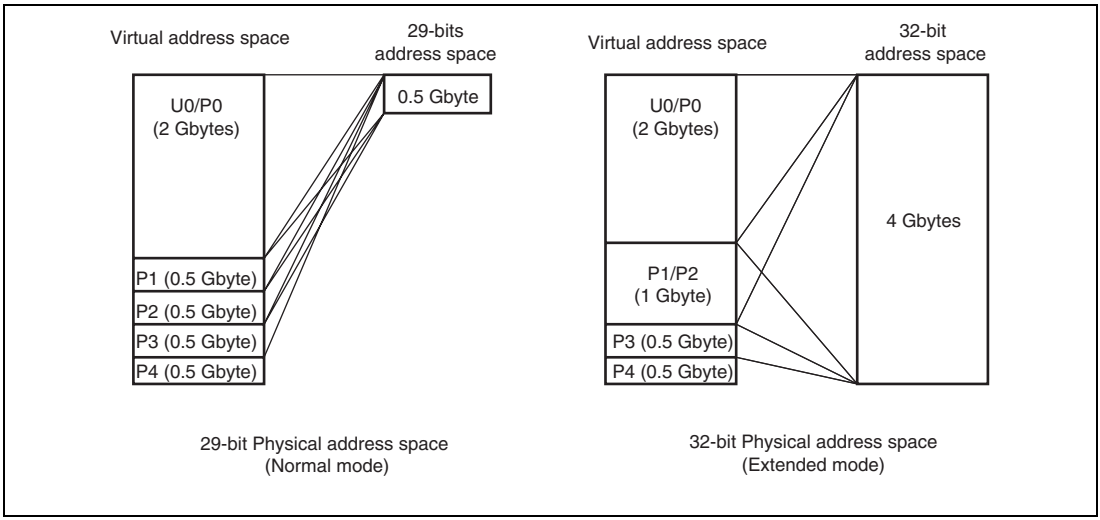


Figure 7.26 Physical Address Space (32-Bit Address Extended Mode)

7.8.1 Overview of 32-Bit Address Extended Mode

In 32-bit address extended mode, the privileged space mapping buffer (PMB) is introduced. The PMB maps virtual addresses in the P1 or P2 area which are not translated in 29-bit address mode to the 32-bit physical address space. In areas which are target for address translation of the TLB (UTLB/ITLB), upper three bits in the PPN field of the UTLB or ITLB are extended and then addresses after the TLB translation can handle the 32-bit physical addresses.

As for the cache operation, P1 area is cacheable and P2 area is non-cacheable in the case of 29-bit address mode, but the cache operation of both P1 and P2 area are determined by the C bit and WT bit in the PMB in the case of 32-bit address mode.

7.8.2 Transition to 32-Bit Address Extended Mode

This LSI enters 29-bit address mode after a power-on reset. Transition is made to 32-bit address extended mode by setting the SE bit in PASCR to 1. In 32-bit address extended mode, the MMU operates as follows.

1. When the AT bit in MMUCR is 0, virtual addresses in the U0, P0, or P3 area become 32-bit physical addresses. Addresses in the P1 or P2 area are translated according to the PMB mapping information. B'10 should be set to the upper 2 bits of virtual page number (VPN[31:30]) in the PMB in order to indicate P1 or P2 area. The operation is not guaranteed when the value except B'10 is set to these bits.
2. When the AT bit in MMUCR is 1, virtual addresses in the U0, P0, or P3 area are translated to 32-bit physical addresses according to the TLB conversion information. Addresses in the P1 or P2 area are translated according to the PMB mapping information. B'10 should be set to the upper 2 bits of virtual page number (VPN[31:30]) in the PMB in order to indicate P1 or P2 area. The operation is not guaranteed when the value except B'10 is set to these bits.
3. Regardless of the setting of the AT bit in MMUCR, bits 31 to 29 in physical addresses become B'111 in the control register area (addresses H'FC00 0000 to H'FFFF FFFF). When the control register area is recorded in the UTLB and accessed, B'111 should be set to PPN[31:29].

7.8.3 Privileged Space Mapping Buffer (PMB) Configuration

In 32-bit address extended mode, virtual addresses in the P1 or P2 area are translated according to the PMB mapping information. The PMB has 16 entries and configuration of each entry is as follows.

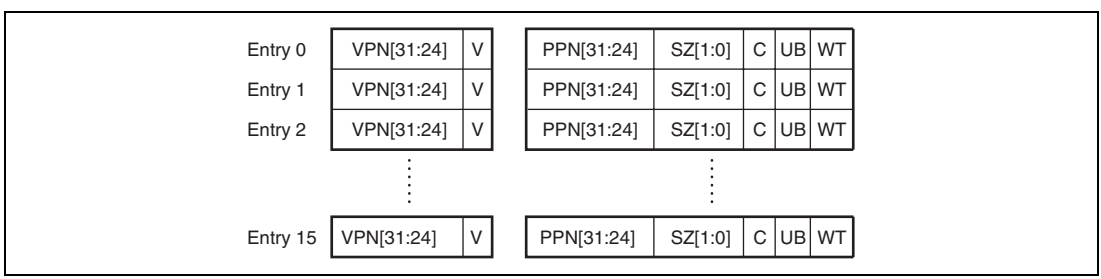


Figure 7.27 PMB Configuration

Legend:

- VPN: Virtual page number
  For 16-Mbyte page: Upper 8 bits of virtual address
  For 64-Mbyte page: Upper 6 bits of virtual address
  For 128-Mbyte page: Upper 5 bits of virtual address
  For 512-Mbyte page: Upper 3 bits of virtual address

  Note: B'10 should be set to the upper 2 bits of VPN in order to indicate P1 or P2 area.

- SZ: Page size bits
  Specify the page size.
  00: 16-Mbyte page
  01: 64-Mbyte page
  10: 128-Mbyte page
  11: 512-Mbyte page
- V: Validity bit
  Indicates whether the entry is valid.
  0: Invalid
  1: Valid
  Cleared to 0 by a power-on reset.
  Not affected by a manual reset.
- PPN: Physical page number
  Upper 8 bits of the physical address of the physical page number.
  With a 16-Mbyte page, PPN[31:24] are valid.
  With a 64-Mbyte page, PPN[31:26] are valid.
  With a 128-Mbyte page, PPN[31:27] are valid.
  With a 512-Mbyte page, PPN[31:29] are valid.
- C: Cacheability bit
  Indicates whether a page is cacheable.
  0: Not cacheable
  1: Cacheable
- WT: Write-through bit
  Specifies the cache write mode.
  0: Copy-back mode
  1: Write-through mode

- UB: Buffered write bit

  Specifies whether a buffered write is performed.

  0: Buffered write (Data access of subsequent processing proceeds without waiting for the write to complete.)

  1: Unbuffered write (Data access of subsequent processing is stalled until the write has completed.)

7.8.4 PMB Function

This LSI supports the following PMB functions.

1. Only memory-mapped write can be used for writing to the PMB. The LDTLB instruction cannot be used to write to the PMB.
2. Software must ensure that every accessed P1 or P2 address has a corresponding PMB entry before the access occurs. When an access to an address in the P1 or P2 area which is not recorded in the PMB is made, this LSI is reset by the TLB. In this case, the accessed address in the P1 or P2 area which causes the TLB reset is stored in the TEA and code H'140 in the EXPEVT.
3. The SH-4A does not guarantee the operation when multiple hit occurs in the PMB. Special care should be taken when the PMB mapping information is recorded by software.
4. The PMB does not have an associative write function.
5. Since there is no PR field in the PMB, read/write protection cannot be preformed. The address translation target of the PMB is the P1 or P2 address. In user mode access, an address error exception occurs.
6. Both entries from the UTLB and PMB are mixed and recorded in the ITLB by means of the hardware ITLB miss handling. However, these entries can be identified by checking whether VPN[31:30] is 10 or not. When an entry from the PMB is recorded in the ITLB, H'00, 01, and 1 are recorded in the ASID, PR, and SH fields which do not exist in the PMB, respectively.

7.8.5 Memory-Mapped PMB Configuration

To enable the PMB to be managed by software, its contents are allowed to be read from and written to by a P1 or P2 area program with a MOV instruction in privileged mode. The PMB address array is allocated to addresses H'F610 0000 to H'F61F FFFF in the P4 area and the PMB data array to addresses H'F710 0000 to H'F71F FFFF in the P4 area. VPN and V in the PMB can be accessed as an address array, PPN, V, SZ, C, WT, and UB as a data array. V can be accessed from both the address array side and the data array side. A program which executes a PMB memory-mapped access should be placed in the page area at which the C bit in PMB is cleared to 0.

1. PMB address array read

   When memory reading is performed while bits 31 to 20 in the address field are specified as H'F61 which indicates the PMB address array and bits 11 to 8 in the address field as an entry, bits 31 to 24 in the data field are read as VPN and bit 8 in the data field as V.

2. PMB address array write

   When memory writing is performed while bits 31 to 20 in the address field are specified as H'F61 which indicates the PMB address array and bits 11 to 8 in the address field as an entry, and bits 31 to 24 in the data field are specified as VPN and bit 8 in the data field as V, data is written to the specified entry.

3. PMB data array read

   When memory reading is performed while bits 31 to 20 in the address field are specified as H'F71 which indicates the PMB data array and bits 11 to 8 in the address field as an entry, bits 31 to 24 in the data field are read as PPN, bit 9 in the data field as UB, bit 8 in the data field as V, bits 7 and 4 in the data field as SZ, bit 3 in the data field as C, and bit 0 in the data field as WT.

4. PMB data array write

   When memory writing is performed while bits 31 to 20 in the address field are specified as H'F71 which indicates the PMB data array and bits 11 to 8 in the address field as an entry, and bits 31 to 24 in the data field are specified as PPN, bit 9 in the data field as UB, bit 8 in the data field as V, bits 7 and 4 in the data field as SZ, bit 3 in the data field as C, and bit 0 in the data field as WT, data is written to the specified entry.

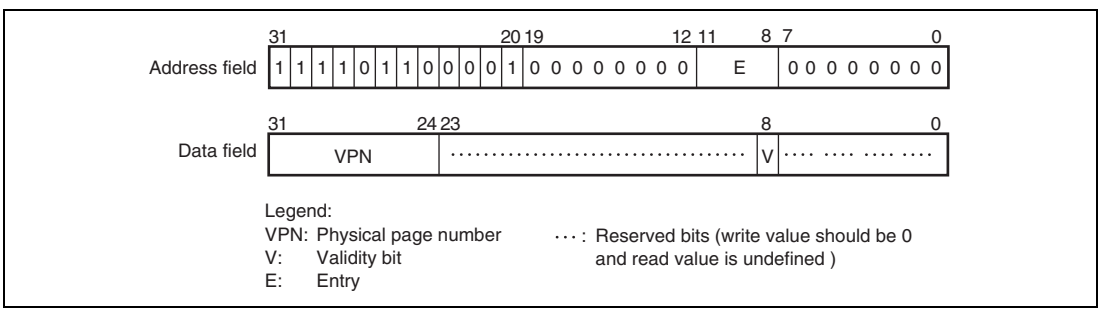


Figure 7.28 Memory-Mapped PMB Address Array

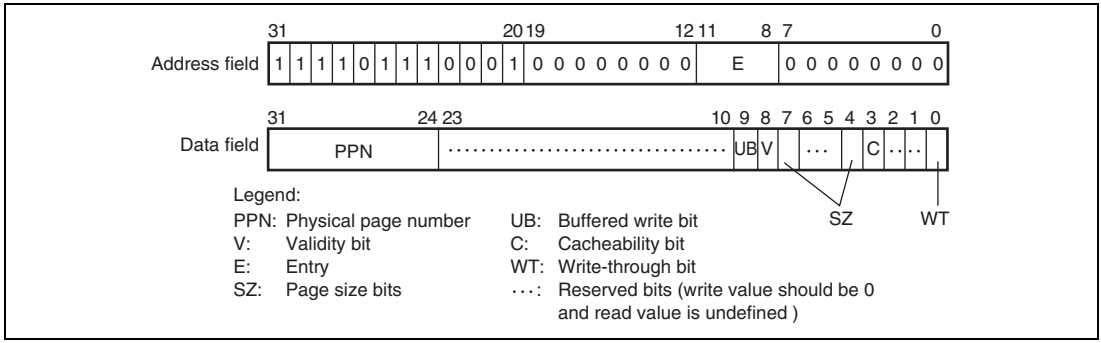


Figure 7.29 Memory-Mapped PMB Data Array

7.8.6 Notes on Using 32-Bit Address Extended Mode

When using 32-bit address extended mode, note that the items described in this section are extended or changed as follows.

(1) PASCR

The SE bit is added in bit 31 in the control register (PASCR). The bits 6 to 0 of the UB in the PASCR are invalid (Note that the bit 7 of the UB is still valid). When writing to the P1 or P2 area, the UB bit in the PMB controls whether a buffered write is performed or not. When the MMU is enabled, the UB bit in the TLB controls writing to the P0, P3, or U0 area. When the MMU is disabled, writing to the P0, P3, or U0 area is always performed as a buffered write.

Bit	Bit Name	Initial Value	R/W	Description
31	SE	0	R/W	0: 29-bit address mode 1: 32-bit address extended mode
30 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	UB	All 0	R/W	Buffered Write Control for Each Area (64 Mbytes) When writing is performed without using the cache or in the cache write-through mode, these bits specify whether the CPU waits for the end of writing for each area. 0: The CPU does not wait for the end of writing 1: The CPU stalls and waits for the end of writing UB[7]: Corresponding to the control register area UB[6:0]: These bits are invalid in 32-bit address extended mode.

(2) ITLB

The PPN field in the ITLB is extended to bits 31 to 10.

(3) UTLB

The PPN field in the UTLB is extended to bits 31 to 10. The same UB bit as that in the PMB is added in each entry of the UTLB.

- UB: Buffered write bit
  Specifies whether a buffered write is performed.
  0: Buffered write (Subsequent processing proceeds without waiting for the write to complete.)
  1: Unbuffered write (Subsequent processing is stalled until the write has completed.)

In a memory-mapped TLB access, the UB bit can be read from or written to by bit 9 in the data array.

(4) PTEL

The same UB bit as that in the PMB is added in bit 9 in PTEL. This UB bit is written to the UB bit in the UTLB by the LDTLB instruction. The PPN field is extended to bits 31 to 10.

(5) CCR.CB

The CB bit in CCR is invalid. Whether a cacheable write for the P1 area is performed in copy-back mode or write-though mode is determined by the WT bit in the PMB.

(6) IRMCR.MT

The MT bit in IRMCR is valid for a memory-mapped PMB write.

(7) QACR0, QACR1

AREA0[4:2]/AREA1[4:2] fields of QACR0/QACR1 are extended to AREA0[7:2]/AREA1[7:2] corresponding to physical address [31:26].

(8) LSA0, LSA1, LDA0, LDA1

L0SADR, L1SADR, L0DADR, and L1DADR fields are extended to bits 31 to 10.

When using 32-bit address mode, the following notes should be applied to software.

1. For the SE bit switching, switching from 0 to 1 is only supported in a boot routine which is allocated in an area where caching and TLB-based address translation are not allowed and runs after a power-on reset or manual reset.
2. After switching the SE bit, an area in which the program is allocated becomes the target of the PMB address translation. Therefore, the area should be recorded in the PMB before switching the SE bit. An address which may be accessed in the P1 or P2 area such as the exception handler should also be recorded in the PMB.
3. When an external memory access occurs by an operand memory access located before the MOV.L instruction which switches the SE bit, external memory space addresses accessed in both address modes should be the same.
4. Note that the V bit is mapped to both address array and data array in PMB registration. That is, first write 0 to the V bit in one of arrays and then write 1 to the V bit in another array.

7.9 32-Bit Boot Function

The address mode of this LSI after a power-on reset or manual reset can be switched between 29-bit address mode and 32-bit address extended mode by specifying external pins. The following changes apply when this LSI is booted up in 32-bit address extended mode.

7.9.1 Initial Entries to PMB

When 32-bit address extended mode is specified by external pins, the following initial entries are recorded in the PMB after a power-on reset or manual reset, and the SE bit in the PASCR register is initialized to 1. For entries 2 to 15, only the V bit is initialized to 0.

Entry	VPN[31:24]	PPN[31:24]	V	SZ[1:0]	C	UB	WT
0	10000000	00000000	1	11	1	0	1
1	10100000	00000000	1	11	0	0	0

7.9.2 Notes on 32-Bit Boot

Immediately after a power-on or manual reset, the P1 or P2 area is mapped to the PMB. Therefore, when an area other than that indicated by the initial entry needs to be mapped, follow the procedures below to modify the PMB, taking care not to generate PMB misses and multiple PMB hits. The procedure should be set up within the boot routine and should be executed before activation of the caches and TLB (CCR.ICE = 1, CCR.OCE = 1, and MMUCR.AT = 1). Do not use routines other than the boot routine to change the value recorded in the PMB.

(1) When the Program Modifying the PMB is in the P1 or P2 Area

1. Read the initial entry, change only the SZ bits to reduce the page size, and save the new value over the previous entry. The program that changes the PMB should be allocated within 1 Mbyte of the top of the page with the reduced size.
2. Invalidate the entry remaining in the ITLB that corresponds to the PMB by writing 1 to the TI bit in the MMUCR register.
3. In the memory-mapped PMB, record PMB entries to fill the P1 or P2 area in which the PMB translation information is evicted by step 1.
4. Execute one of the following steps, A, B, and C. Do not execute a branch or operand access for the P1 or P2 area in which the PMB translation information is evicted by step 1.
   A. Perform a branch using the RTE instruction.
   B. Execute the ICBI instruction for any address (including non-cacheable area).

C. If the MT bit in IRMCR is set to 0 (initial value) before accessing the memory-mapped PMB, no specific sequence is required.

However, correct operation with method C may no longer be guaranteed in future SuperH-family products. Selection of step A or B is recommended to ensure compatibility with future SuperH-family products.

(2) When the Program Modifying the PMB is in Areas Other than the P1 or P2 Area

1. Invalidate the entry remaining in the ITLB by writing 1 to the TI bit in MMUCR.
2. In the memory-mapped PMB, change PMB entries.
3. Execute one of the following steps, A, B, and C. Do not execute a branch or operand access for the P1 or P2 area before this execution.
   A. Perform a branch using the RTE instruction.
   B. Execute the ICBI instruction for any address (including non-cacheable area).
   C. If the MT bit in IRMCR is set to 0 (initial value) before accessing the memory-mapped PMB, no specific sequence is required.

   However, correct operation with method C may no longer be guaranteed in future SuperH-family products. Selection of step A or B is recommended to ensure compatibility with future SuperH-family products.

7.10 Usage Notes

7.10.1 Note on Using LDTLB Instruction

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

1. In 29-bit address mode, follow A. and B. below. In 32-bit address mode, follow A. through D. below.
   A. Place the TLB miss exception handling routine[*1] only in the P1, P2 area ,or the on-chip memory so that all the instruction accesses[*3] in the TLB miss exception handling routine should occur solely in the P1, P2 area, or the on-chip memory. Clear the RP bit in the RAMCR register to 0 (initial value), when the TLB miss exception handling routine is placed in the on-chip memory.
   B. Use only one page of the PMB for instruction accesses[*3] in the TLB miss exception handling routine[*1]. In 32-bit address mode, do not place them in the last 64 bytes of a page of the PMB.
   C. In 32-bit address mode, obey 1 and 2 below when recording information in the UTLB in the MMU-related exception[*2] handling routine.
      a. When the TLB miss exception occurs, and recording the information of a page with the access right in the UTLB, do not record the page, in which the exception has occurred, in the UTLB using the following two operations.
         — Specifies the protection key data that causes a protection violation exception upon re-execution of the instruction that has caused the TLB miss exception and records the page, in which the TLB miss exception has occurred, in the UTLB.
         — Specifies the protection key data that does not cause a protection violation exception in the protection violation exception handling routine to record the page in the UTLB and re-executes the instruction that has caused the protection violation exception.
      b. When an initial page write exception occurs and the TLB entry in the UTLB of which the dirty bit is 1 is replaced, before the write instruction for the page corresponding to this replaced TLB entry is completed, register the TLB entry of which the dirty bit is 1.
   D. Do not make an attempt to execute the FDIV or FSQRT instruction in the TLB miss exception handling routine.
2. If a TLB miss exception occurs, add 1 to MMUCR.URC before executing an LDTLB instruction.

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

2. MMU-related exceptions are: instruction TLB miss exception, instruction TLB miss protection violation exception, data TLB miss exception, data TLB protection violation exception, and initial page write exception.

3. Instruction accesses include the PREFI and ICBI instructions.

Section 8 Caches

This LSI has an on-chip 32-Kbyte instruction cache (IC) for instructions and an on-chip 32-Kbyte operand cache (OC) for data. For details of the extended function, refer to appendix A corresponding section.

8.1 Features

The features of the cache are given in table 8.1.

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

Table 8.1 Cache Features

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

Table 8.2 Store Queue Features

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

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

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

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

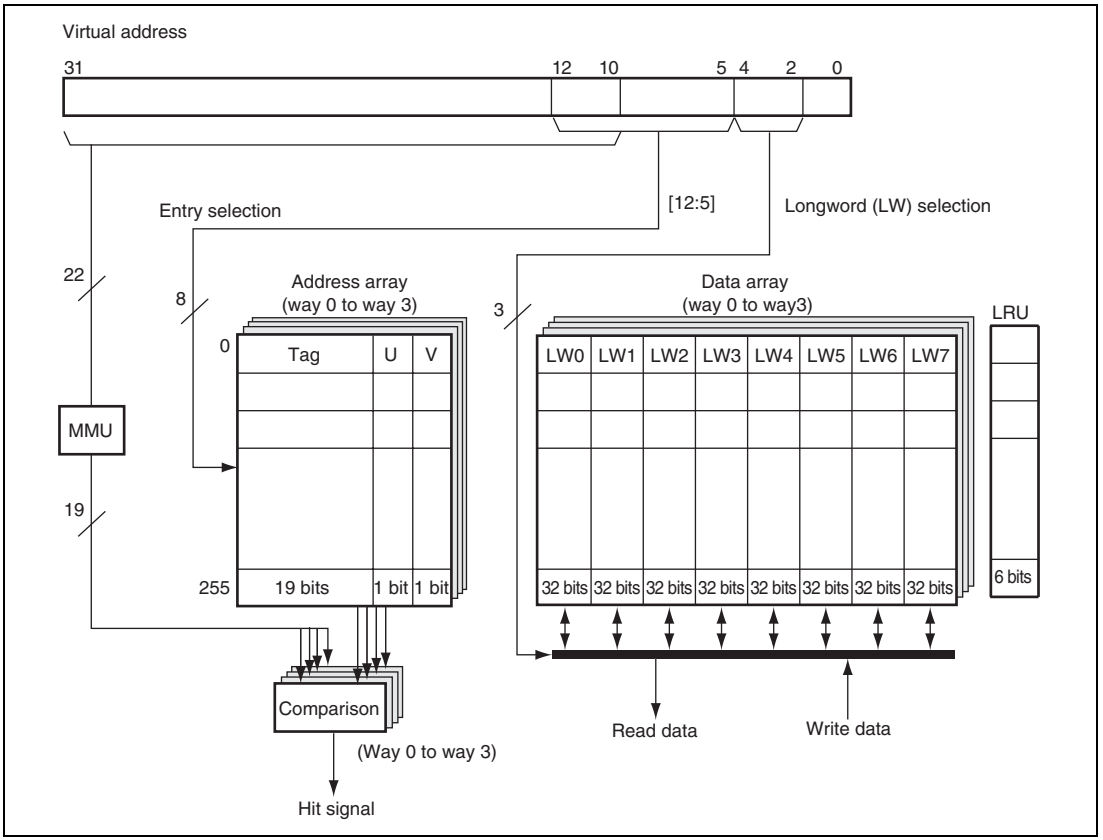


Figure 8.1 Configuration of Operand Cache (Cache size = 32 Kbytes)

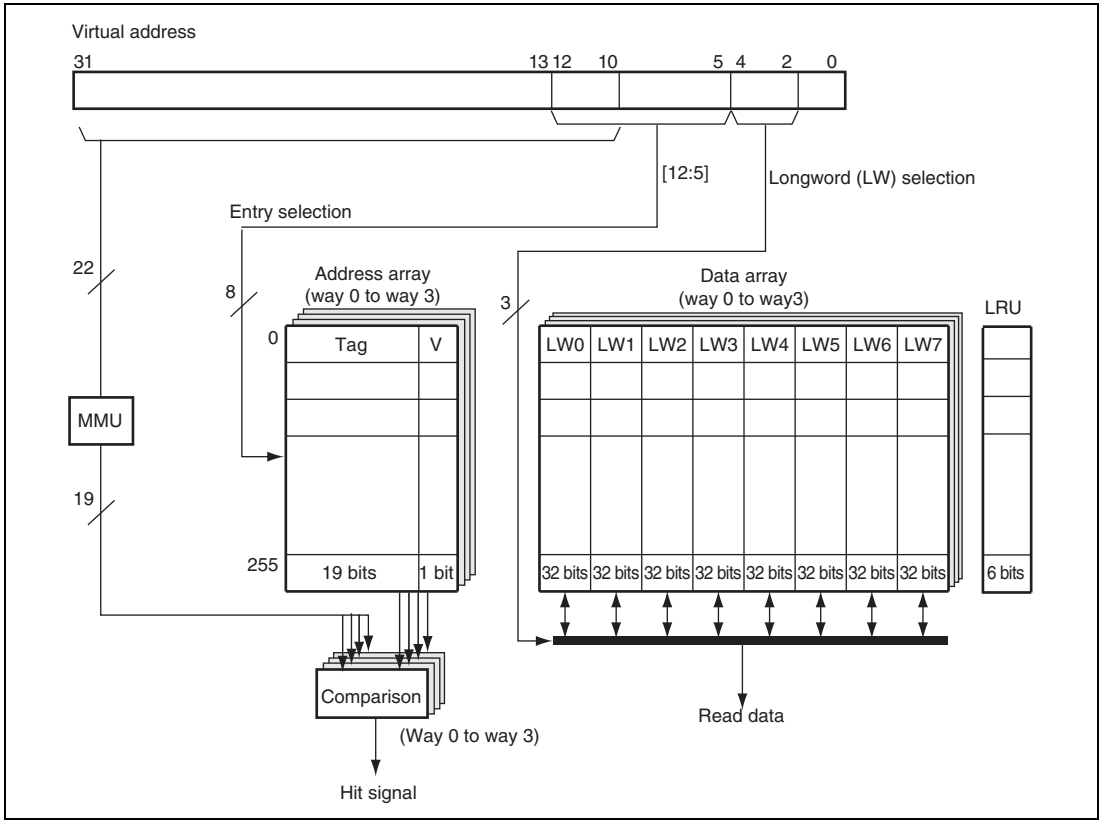


Figure 8.2 Configuration of Instruction Cache (Cache size = 32 Kbytes)

- Tag

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

- V bit (validity bit)

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

- U bit (dirty bit)

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

- Data array

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

- LRU

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

8.2 Register Descriptions

The following registers are related to cache.

Table 8.3 Register Configuration

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

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

Table 8.4 Register States in Each Processing State

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

8.2.1 Cache Control Register (CCR)

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

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

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

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	ICI	0	R/W	IC Invalidation Bit When 1 is written to this bit, the V bits of all IC entries are cleared to 0. This bit is always read as 0.

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

8.2.2 Queue Address Control Register 0 (QACR0)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 2	AREA0	Undefined	R/W	When the MMU is disabled, these bits generate physical address bits [28:26] for SQ0.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

8.2.3 Queue Address Control Register 1 (QACR1)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 2	AREA1	Undefined	R/W	When the MMU is disabled, these bits generate physical address bits [28:26] for SQ1.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

8.2.4 On-Chip Memory Control Register (RAMCR)

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

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

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

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	RMD	0	R/W	On-Chip Memory Access Mode Bit For details, see section 9.4, On-Chip Memory Protective Functions.
8	RP	0	R/W	On-Chip Memory Protection Enable Bit For details, see section 9.4, On-Chip Memory Protective Functions.
7	IC2W	0	R/W	IC Two-Way Mode bit 0: IC is a four-way operation 1: IC is a two-way operation For details, see section 8.4.3, IC Two-Way Mode.
6	OC2W	0	R/W	OC Two-Way Mode bit 0: OC is a four-way operation 1: OC is a two-way operation For details, see section 8.3.6, OC Two-Way Mode.
5	ICWPD	0	R/W	IC Way Prediction Stop Selects whether the IC way prediction is used. 0: Instruction cache performs way prediction. 1: Instruction cache does not perform way prediction.
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

8.3 Operand Cache Operation

8.3.1 Read Operation

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

1. The tag, V bit, U bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tags read from the each way is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU:
   — If there is a way whose tag matches and its V bit is 1, see No. 3.
   — If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 0, see No. 4.
   — If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 1, see No. 5.
3. Cache hit

   The data indexed by virtual address bits [4:0] is read from the data field of the cache line on the hitted way in accordance with the access size. Then the LRU bits are updated to indicate the hitted way is the latest one.
4. Cache miss (no write-back)

   Data is read into the cache line on the way, which is selected to replace, from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data(8 bytes) including the cache-missed data. When the corresponding data arrives in the cache, the read data is returned to the CPU. While the remaining data on the cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and 0 is written to the U bit on the way. Then the LRU bit is updated to indicate the way is latest one.
5. Cache miss (with write-back)

   The tag and data field of the cache line on the way which is selected to replace are saved in the write-back buffer. Then data is read into the cache line on the way which is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data, and when the corresponding data arrives in the cache, the read data is returned to the CPU. While the remaining one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit, and 0 to the U bit. And the LRU bits are updated to indicate the way is latest one. The data in the

write-back buffer is then written back to external memory.

8.3.2 Prefetch Operation

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

1. The tag, V bit, U bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU:
   - If there is a way whose tag matches and its V bit is 1, see No. 3.
   - If there is no way whose tag matches and the V bit is 1, and the U bit of the way which is selected to replace using the LRU bits is 0, see No. 4.
   - If there is no way whose tag matches and the V bit is 1, and the U bit of the way which is selected to replace using the LRU bits is 1, see No. 5.
3. Cache hit

   Then the LRU bits are updated to indicate the hitted way is the latest one.
4. Cache miss (no write-back)

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

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

8.3.3 Write Operation

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

1. The tag, V bit, U bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU:
   - If there is a way whose tag matches and its V bit is 1, see No. 3 for copy-back and No. 4 for write-through.
   - I If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 0, see No. 5 for copy-back and No. 7 for write-through.
   - If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 1, see No. 6 for copy-back and No. 7 for write-through.
3. Cache hit (copy-back)

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

   A data write in accordance with the access size is performed for the data field on the hit way which is indexed by virtual address bits [4:0]. A write is also performed to external memory corresponding to the virtual address. Then the LRU bits are updated to indicate the way is the latest one. In this case, the U bit isn't updated.
5. Cache miss (copy-back, no write-back)

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

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

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

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

7. Cache miss (write-through)

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

8.3.4 Write-Back Buffer

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

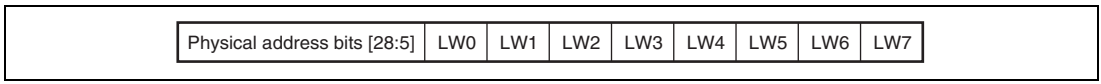


Figure 8.3 Configuration of Write-Back Buffer

8.3.5 Write-Through Buffer

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

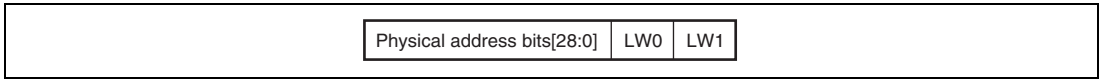


Figure 8.4 Configuration of Write-Through Buffer

8.3.6 OC Two-Way Mode

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

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

8.4 Instruction Cache Operation

8.4.1 Read Operation

When the IC is enabled (ICE = 1 in CCR) and instruction fetches are performed from a cacheable area, the instruction cache operates as follows:

1. The tag, V bit, U bit and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU:
   — If there is a way whose tag matches and the V bit is 1, see No. 3.
   — If there is no way whose tag matches and the V bit is 1, see No. 4.
3. Cache hit

   The data indexed by virtual address bits [4:2] is read as an instruction from the data field on the hit way. The LRU bits are updated to indicate the way is the latest one.
4. Cache miss

   Data is read into the cache line on the way which selected using LRU bits to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data, and when the corresponding data arrives in the cache, the read data is returned to the CPU as an instruction. While the remaining one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, and 1 is written to the V bit, the LRU bits are updated to indicate the way is the latest one.

8.4.2 Prefetch Operation

When the IC is enabled (ICE = 1 in CCR) and instruction prefetches are performed from a cacheable area, the instruction cache operates as follows:

1. The tag, V bit, Ubit and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU:
   — If there is a way whose tag matches and the V bit is 1, see No. 3.
   — If there is no way whose tag matches and the V bit is 1, see No. 4.
3. Cache hit

   The LRU bits is updated to indicate the way is the latest one.

4. Cache miss

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

8.4.3 IC Two-Way Mode

When the IC2W bit in RAMCR is set to 1, IC two-way mode which only uses way 0 and way 1 in the IC is entered. Thus, power consumption can be reduced. In this mode, only way 0 and way 1 are used even if a memory-mapped IC access is made.

The IC2W bit should be modified by a program in the P2 area. At that time, if the valid line has already been recorded in the IC, 1 should be written to the ICI bit in CCR and all entries in the IC should be invalid before modifying the IC2W bit.

8.4.4 Instruction Cache Way Prediction Operation

This LSI incorporates an instruction cache (IC) way prediction scheme to reduce power consumption. This is achieved by activating only the data array that corresponds to a predicted way. When way prediction misses occur, data must be re-read from the right way, which may lead to lower performance in instruction fetching. Setting the ICWPD bit to 1 disables the IC way prediction scheme. Since way prediction misses do not occur in this mode, there is no loss of performance in instruction fetching but the IC consumes more power. The ICWPD bit should be modified by a program in the non-cacheable P2 area. If a valid line has already been recorded in the IC at this time, invalidate all entries in the IC by writing 1 to the ICI bit in CCR before modifying the ICWPD bit.

8.5 Cache Operation Instruction

8.5.1 Coherency between Cache and External Memory

(1) Cache Operation Instruction

Coherency between cache and external memory should be assured by software. In this LSI, the following six instructions are supported for cache operations. Details of these instructions are given in section 11, Instruction Descriptions of the SH-4A Extended Functions Software Manual.

- Operand cache invalidate instruction: OCBI @Rn
  Operand cache invalidation (no write-back)
- Operand cache purge instruction: OCBP @Rn
  Operand cache invalidation (with write-back)
- Operand cache write-back instruction: OCBWB @Rn
  Operand cache write-back
- Operand cache allocate instruction: MOVCA.L R0,@Rn
  Operand cache allocation
- Instruction cache invalidate instruction: ICBI @Rn
  Instruction cache invalidation
- Operand access synchronization instruction: SYNCO
  Wait for data transfer completion

(2) Coherency Control

The operand cache can receive "PURGE" and "FLUSH" transaction from SuperHyway bus to control the cache coherency. Since the address used by the PURGE and FLUSH transaction is a physical address, do not use the 1 Kbyte page size to avoid cache synonym problem in MMU enable mode.

- PURGE transaction
  When the operand cache is enabled, the PURGE transaction checks the operand cache and invalidates the hit entry. If the invalidated entry is dirty, the data is written back to the external memory. If the transaction is not hit to the cache, it is no-operation.
- FLUSH transaction
  When the operand cache is enabled, the FLUSH transaction checks the operand cache and if the hit line is dirty, then the data is written back to the external memory. If the transaction is not hit to the cache or the hit entry is not dirty, it is no-operation.

(3) Changes in Instruction Specifications Regarding Coherency Control

Of the operand cache operating instructions, the coherency control-related specifications of OCBI, OCBP, and OCBWB have been changed from those of the SH-4A with H'20-valued VER bits in the processor version register (PVR).

- Changes in the invalidate instruction OCBI@Rn

  When Rn is designating an address in a non-cacheable area, this instruction is executed as NOP in the SH-4A with H'20-valued VER bits in the processor version register (PVR). In this LSI, this instruction invalidates the operand cache line designated by way = Rn[14:13] and entry = Rn[12:5] provided that Rn[31:24] = H'F4 (OC address array area). In this process, writing back of the line does not take place even if the line to be invalidated is dirty. This operation is only executable in privileged mode, and an address error exception occurs in user mode. TLB-related exceptions do not occur.

  Do not execute this instruction to invalidate the memory-mapped array areas and control register areas for which Rn[31:24] is not H'F4, and their reserved areas (H'F0 to H'F3, H'F5 to H'FF).

- Changes in the purge instruction OCBP@Rn

  When Rn is designating an address in a non-cacheable area, this instruction is executed as NOP in the SH-4A with H'20-valued VER bits in the processor version register (PVR). In this LSI, this instruction invalidates the operand cache line designated by way = Rn[14:13] and entry = Rn[12:5] provided that Rn[31:24] = H'F4 (OC address array area). In this process, writing back of the line takes place when the line to be invalidated is dirty. This operation is only executable in privileged mode, and an address error exception occurs in user mode. TLB-related exceptions do not occur.

  Do not execute this instruction to invalidate the memory-mapped array areas and control register areas for which Rn[31:24] is not H'F4, and their reserved areas (H'F0 to H'F3, H'F5 to H'FF).

- Changes in the write-back instruction OCBWB@Rn

  When Rn is designating an address in a non-cacheable area, this instruction is executed as NOP in the SH-4A with H'20-valued VER bits in the processor version register (PVR). In this LSI, provided that Rn[31:24] = H'F4 (OC address array area), this instruction writes back the operand cache line designated by way = Rn[14:13] and entry = Rn[12:5] if it is dirty and clears the dirty bit to 0. This operation is only executable in privileged mode, and an address error exception occurs in user mode. TLB-related exceptions do not occur.

  Do not execute this instruction to invalidate the memory-mapped array areas and control register areas for which Rn[31:24] is not H'F4, and their reserved areas (H'F0 to H'F3, H'F5 to H'FF).

8.5.2 Prefetch Operation

This LSI supports a prefetch instruction to reduce the cache fill penalty incurred as the result of a cache miss. If it is known that a cache miss will result from a read or write operation, it is possible to fill the cache with data beforehand by means of the prefetch instruction to prevent a cache miss due to the read or write operation, and so improve software performance. If a prefetch instruction is executed for data already held in the cache, or if the prefetch address results in a UTLB miss or a protection violation, the result is no operation, and an exception is not generated. Details of the prefetch instruction are given in section 11, Instruction Descriptions of the SH-4A Extended Functions Software Manual.

- Prefetch instruction (OC) : PREF @Rn
- Prefetch instruction (IC) : PREFI @Rn

8.6 Memory-Mapped Cache Configuration

The IC and OC can be managed by software. The contents of IC data array can be read from or written to by a program in the P2 area by means of a MOV instruction in privileged mode. The contents of IC address array can also be read from or written to in privileged mode by a program in the P2 area or the IL memory area by means of a MOV instruction. Operation is not guaranteed if access is made from a program in another area. In this case, execute one of the following three methods for executing a branch to the P0, U0, P1, or P3 area.

1. Execute a branch using the RTE instruction.
2. Execute a branch to the P0, U0, P1, or P3 area after executing the ICBI instruction for any address (including non-cacheable area).
3. If the MC bit in IRMCR is 0 (initial value) before making an access to the memory-mapped IC, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after making an access to the memory-mapped IC.

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

In privileged mode, the OC contents can be read from or written to by a program in the P1 or P2 area by means of a MOV instruction. Operation is not guaranteed if access is made from a program in another area. The IC and OC are allocated to the P4 area in the virtual address space. Only data accesses can be used on both the IC address array and data array and the OC address array and data array, and accesses are always longword-size. Instruction fetches cannot be performed in these areas. For reserved bits, a write value of 0 should be specified and the read value is undefined.

8.6.1 IC Address Array

The IC address array is allocated to addresses H'F000 0000 to H'F0FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the write tag and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F0 indicating the IC address array, and the way is specified by bits [14:13] and the entry by bits [12:5]. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the IC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag is indicated by bits [31:10], and the V bit by bit [0]. As the IC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address specification only in the case of a write in which association is performed.

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

1. IC address array read

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

   The tag and V bit specified in the data field are written to the IC entry corresponding to the way and entry set in the address field. The A bit in the address field should be cleared to 0.
3. IC address array write (associative)

   When a write is performed with the A bit in the address field set to 1, the tag in each way stored in the entry specified in the address field is compared with the tag specified in the data field. The way numbers of bits [14:13] in the address field are not used. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the ITLB. If the addresses match and the V bit in the way is 1, the V bit specified in the data field is written into the IC entry. In other cases, no operation is performed. This operation is used to invalidate a specific IC entry. If an ITLB miss occurs during address translation, or the comparison shows a mismatch, an exception is not generated, no operation is performed, and the write is not executed.

Note: IC address array associative writing function may not be supported in the future SuperH Series. Therefore, it is recommended that the ICBI instruction should be used to operate the IC definitely by handling ITLB miss and reporting ITLB miss exception.

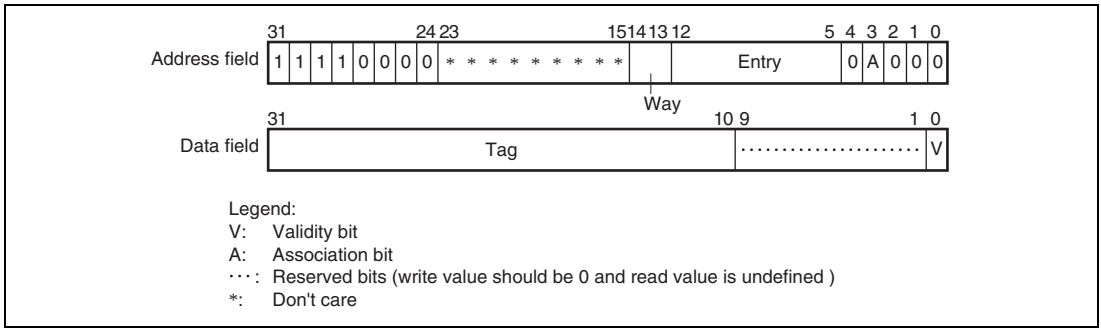


Figure 8.5 Memory-Mapped IC Address Array (Cache size = 32 Kbytes)

8.6.2 IC Data Array

The IC data array is allocated to addresses H'F100 0000 to H'F1FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F1 indicating the IC data array, and the way is specified by bits [14:13] and the entry by bits [12:5]. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

The data field is used for the longword data specification.

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

1. IC data array read

   Longword data is read into the data field from the data specified by the longword specification bits in the address field in the IC entry corresponding to the way and entry set in the address field.
2. IC data array write

   The longword data specified in the data field is written for the data specified by the longword specification bits in the address field in the IC entry corresponding to the way and entry set in the address field.

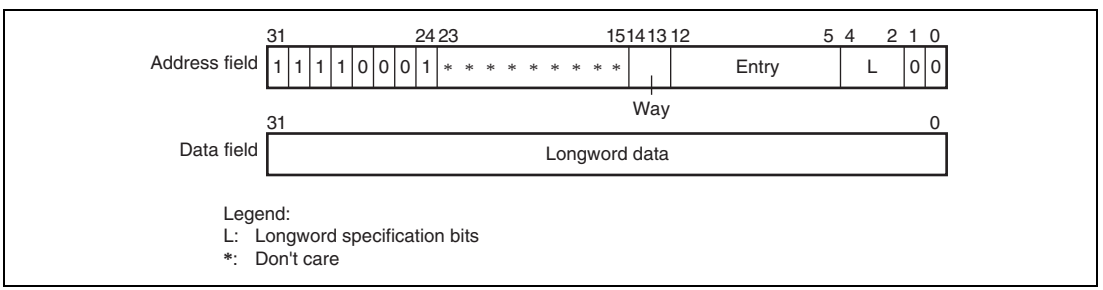


Figure 8.6 Memory-Mapped IC Data Array (Cache size = 32 Kbytes)

8.6.3 OC Address Array

The OC address array is allocated to addresses H'F400 0000 to H'F4FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the write tag, U bit, and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F4 indicating the OC address array, and the way is specified by bits [14:13] and the entry by bits [12:5]. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the OC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag is indicated by bits [31:10], the U bit by bit [1], and the V bit by bit [0]. As the OC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address specification only in the case of a write in which association is performed.

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

1. OC address array read

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

   The tag, U bit, and V bit specified in the data field are written to the OC entry corresponding to the way and entry set in the address field. The A bit in the address field should be cleared to 0.

   When a write is performed to a cache line for which the U bit and V bit are both 1, after write-back of that cache line, the tag, U bit, and V bit specified in the data field are written.
3. OC address array write (associative)

   When a write is performed with the A bit in the address field set to 1, the tag in each way stored in the entry specified in the address field is compared with the tag specified in the data field. The way numbers of bits [14:13] in the address field are not used. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the UTLB. If the addresses match and the V bit in the way is 1, the U bit and V bit specified in the data field are written into the OC entry. In other cases, no operation is performed. This operation is used to invalidate a specific OC entry. If the OC entry U bit is 1, and 0 is written to the V bit or to the U bit, write-back is performed. If a UTLB miss occurs during address translation, or the comparison shows a mismatch, an exception is not generated, no operation is performed, and the write is not executed.

Note: OC address array associative writing function may not be supported in the future SuperH Series. Therefore, it is recommended that the OCBI, OCBP, or OCBWB instruction should be used to operate the OC definitely by reporting data TLB miss exception.

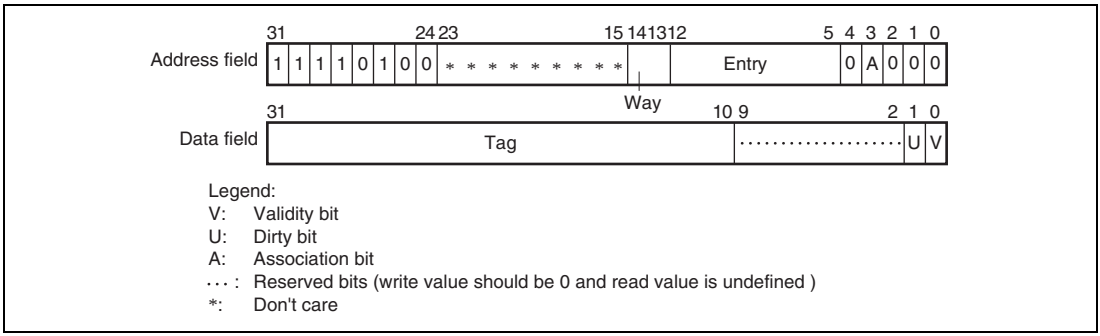


Figure 8.7 Memory-Mapped OC Address Array (Cache size = 32 Kbytes)

8.6.4 OC Data Array

The OC data array is allocated to addresses H'F500 0000 to H'F5FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F5 indicating the OC data array, and the way is specified by bits [14:13] and the entry by bits [12:5]. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

The data field is used for the longword data specification.

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

1. OC data array read

   Longword data is read into the data field from the data specified by the longword specification bits in the address field in the OC entry corresponding to the way and entry set in the address field.
2. OC data array write

   The longword data specified in the data field is written for the data specified by the longword specification bits in the address field in the OC entry corresponding to the way and entry set in the address field. This write does not set the U bit to 1 on the address array side.

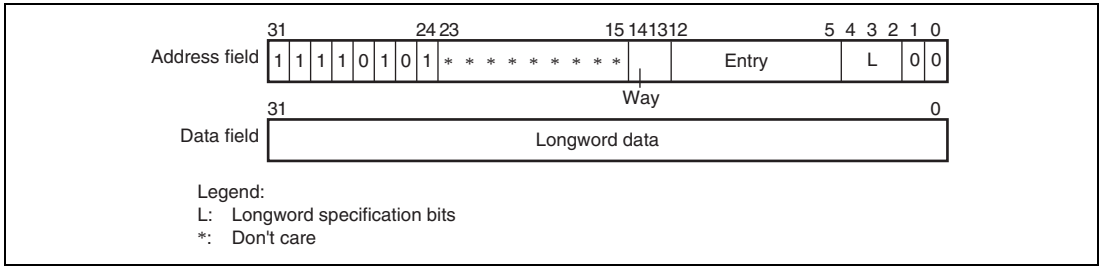


Figure 8.8 Memory-Mapped OC Data Array (Cache size = 32 Kbytes)

8.6.5 Memory-Mapped Cache Associative Write Operation

Associative writing to the IC and OC address arrays may not be supported in future SuperH-family products. The use of instructions ICBI, OCBI, OCBP, and OCBWB is recommended. These instructions handle ITLB misses, and notify instruction TLB miss exceptions and data TLB miss exceptions, thus providing a sure way of controlling the IC and OC. As a transitional measure, this LSI generates address errors when this function is used. If compatibility with previous products is a crucial consideration, on the other hand, the MMCAW bit in EXPMASK (H'FF2F 0004) can be set to 1 to enable this function. However, instructions ICBI, OCBI, OCBP, and OCBWB should be used to guarantee compatibility with future SuperH-family products.

8.7 Store Queues

This LSI supports two 32-byte store queues (SQs) to perform high-speed writes to external memory.

8.7.1 SQ Configuration

There are two 32-byte store queues, SQ0 and SQ1, as shown in figure 8.9. These two store queues can be set independently.

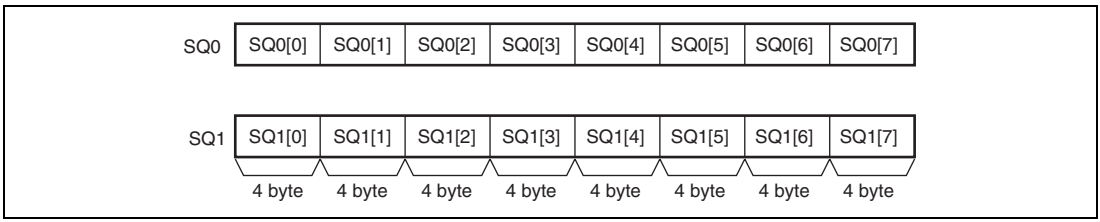


Figure 8.9 Store Queue Configuration

8.7.2 Writing to SQ

A write to the SQs can be performed using a store instruction for addresses H'E000 0000 to H'E3FF FFFC in the P4 area. A longword or quadword access size can be used. The meanings of the address bits are as follows:

[31:26]	: 111000	Store queue specification
[25:6]	: Don't care	Used for external memory transfer/access right
[5]	: 0/1	0: SQ0 specification
		1: SQ1 specification
[4:2]	: LW specification	Specifies longword position in SQ0/SQ1
[1:0]	: 00	Fixed at 0

8.7.3 Transfer to External Memory

Transfer from the SQs to external memory can be performed with a prefetch instruction (PREF). Issuing a PREF instruction for addresses H'E000 0000 to H'E3FF FFFC in the P4 area starts a transfer from the SQs to external memory. The transfer length is fixed at 32 bytes, and the start address is always at a 32-byte boundary. While the contents of one SQ are being transferred to external memory, the other SQ can be written to without a penalty cycle. However, writing to the SQ involved in the transfer to external memory is kept waiting until the transfer is completed.

The physical address bits [28:0] of the SQ transfer destination are specified as shown below, according to whether the MMU is enabled or disabled.

- When MMU is enabled (AT = 1 in MMUCR)

  The SQ area (H'E000 0000 to H'E3FF FFFF) is set in VPN of the UTLB, and the transfer destination physical address in PPN. The ASID, V, SZ, SH, PR, and D bits have the same meaning as for normal address translation, but the C and WT bits have no meaning with regard to this page. When a prefetch instruction is issued for the SQ area, address translation is performed and physical address bits [28:10] are generated in accordance with the SZ bit specification. For physical address bits [9:5], the address prior to address translation is generated in the same way as when the MMU is disabled. Physical address bits [4:0] are fixed at 0. Transfer from the SQs to external memory is performed to this address.
- When MMU is disabled (AT = 0 in MMUCR)

  The SQ area (H'E000 0000 to H'E3FF FFFF) is specified as the address at which a PREF instruction is issued. The meanings of address bits [31:0] are as follows:

[31:26]	: 111000	Store queue specification
[25:6]	: Address	Transfer destination physical address bits [25:6]
[5]	: 0/1	0: SQ0 specification
		1: SQ1 specification and transfer destination physical address bit [5]
[4:2]	: Don't care	No meaning in a prefetch
[1:0]	: 00	Fixed at 0

  Physical address bits [28:26], which cannot be generated from the above address, are generated from QACR0 and QACR1.

  QACR0[4:2] : Physical address bits [28:26] corresponding to SQ0
  QACR1[4:2] : Physical address bits [28:26] corresponding to SQ1

  Physical address bits [4:0] are always fixed at 0 since burst transfer starts at a 32-byte boundary.

8.7.4 Determination of SQ Access Exception

Determination of an exception in a write to an SQ or transfer to external memory (PREF instruction) is performed as follows according to whether the MMU is enabled or disabled. If an exception occurs during a write to an SQ, the SQ contents before the write are retained. If an exception occurs in a data transfer from an SQ to external memory, the transfer to external memory will be aborted.

- When MMU is enabled (AT = 1 in MMUCR)

  Operation is in accordance with the address translation information recorded in the UTLB, and the SQMD bit in MMUCR. Write type exception judgment is performed for writes to the SQs, and read type exception judgment for transfer from the SQs to external memory (using a PREF instruction). As a result, a TLB miss exception or protection violation exception is generated as required. However, if SQ access is enabled in privileged mode only by the SQMD bit in MMUCR, an address error will occur even if address translation is successful in user mode.
- When MMU is disabled (AT = 0 in MMUCR)

  Operation is in accordance with the SQMD bit in MMUCR.

  0: Privileged/user mode access possible

  1: Privileged mode access possible

  If the SQ area is accessed in user mode when the SQMD bit in MMUCR is set to 1, an address error will occur.

8.7.5 Reading from SQ

In privileged mode in this LSI, reading the contents of the SQs may be performed by means of a load instruction for addresses H'FF00 1000 to H'FF00 103C in the P4 area. Only longword access is possible.

[31:6]	: H'FF00 1000	Store queue specification
[5]	: 0/1	0: SQ0 specification
		1: SQ1 specification
[4:2]	: LW specification	Specifies longword position in SQ0/SQ1
[1:0]	: 00	Fixed at 0

8.8 Notes on Using 32-Bit Address Extended Mode

In 32-bit address extended mode, the items described in this section are extended as follows.

1. The tag bits [28:10] (19 bits) in the IC and OC are extended to bits [31:10] (22 bits).
2. An instruction which operates the IC (a memory-mapped IC access and writing to the ICI bit in CCR) should be located in the P1 or P2 area. The cacheable bit (C bit) in the corresponding entry in the PMB should be 0.
3. Bits [4:2] (3 bits) for the AREA0 bit in QACR0 and the AREA1 bit in QACR1 are extended to bits [7:2] (6 bits).

Section 9 On-Chip Memory

This LSI includes three types of memory modules for storage of instructions and data: OL memory, and IL memory. The OL memory is suitable for data storage while the IL memory is suitable for instruction storage. For details of the extended function, refer to appendix A corresponding section.

9.1 Features

(1) OL Memory

- Capacity
  The OL memory in this LSI is 16 Kbytes.
- Page
  The OL memory is divided into two pages (pages 0 and 1).
- Memory map
  The OL memory is allocated in the addresses shown in table 9.1.

Table 9.1 OL memory Addresses

			Memory Size
Address Space		**Page**	**16 Kbytes**
Virtual address		Page 0	H'E500E000 to H'E500FFFF
		Page 1	H'E5010000 to H'E5011FFF
Physical address	CPU 0	Page 0	H'1400E000 to H'1400FFFF
		Page 1	H'14010000 to H'14011FFF
	CPU 1	Page 0	H'1480E000 to H'1480FFFF
		Page 1	H'14810000 to H'14811FFF

- Ports
  Each page has three independent read/write ports and is connected to the SuperHyway bus, the cache/RAM internal bus, and operand bus. The operand bus is used when the OL memory is accessed through operand access. The cache/RAM internal bus is used when the OL memory is accessed through instruction fetch. The SuperHyway bus is used for OL memory access from the SuperHyway bus master module.

- Priority

  In the event of simultaneous accesses to the same page from different buses, the access requests are processed according to priority. The priority order is: SuperHyway bus > Cache/RAM internal bus > operand bus.

(2) IL Memory

- Capacity

  The IL memory in this LSI is 8 Kbytes.
- Page

  The IL memory is divided into two pages (pages 0 and 1).
- Memory map

  The IL memory is allocated to the addresses shown in table 9.2.

Table 9.2 IL Memory Addresses

		Memory Size
Address Space		**8 Kbytes**
Virtual address	Page 0	H'E5200000 to H'E5200FFF
	Page 1	H'E5201000 to H'E5201FFF
Physical address	CPU 0	H'14200000 to H'14201FFF
	CPU 1	H'14A00000 to H'14A01FFF

- Ports

  The page has three independent read/write ports and is connected to the SuperHyway bus, the cache/RAM internal bus, and the instruction bus. The instruction bus is used when the IL memory is accessed through instruction fetch. The cache/RAM internal bus is used when the IL memory is accessed through operand access. The SuperHyway bus is used for IL memory access from the SuperHyway bus master module.
- Priority

  In the event of simultaneous accesses to the same page from different buses, the access requests are processed according to priority. The priority order is: SuperHyway bus > cache/RAM internal bus > instruction bus.

9.2 Register Descriptions

The following registers are related to the on-chip memory.

Table 9.3 Register Configuration

Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
On-chip memory control register	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32
OL memory transfer source address register 0	LSA0	R/W	H'FF00 0050	H'1F00 0050	32
OL memory transfer source address register 1	LSA1	R/W	H'FF00 0054	H'1F00 0054	32
OL memory transfer destination address register 0	LDA0	R/W	H'FF00 0058	H'1F00 0058	32
OL memory transfer destination address register 1	LDA1	R/W	H'FF00 005C	H'1F00 005C	32

Note: * The P4 address is the address used when using P4 area in the virtual address space. The area 7 address is the address used when accessing from area 7 in the physical address space using the TLB.

Table 9.4 Register States in Each Processing Mode

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep/ Light Sleep
On-chip memory control register	RAMCR	H'0000 0000	H'0000 0000	Retained
OL memory transfer source address register 0	LSA0	Undefined	Undefined	Retained
OL memory transfer source address register 1	LSA1	Undefined	Undefined	Retained
OL memory transfer destination address register 0	LDA0	Undefined	Undefined	Retained
OL memory transfer destination address register 1	LDA1	Undefined	Undefined	Retained

9.2.1 On-Chip Memory Control Register (RAMCR)

RAMCR controls the protective functions in the on-chip memory.

When updating RAMCR, please follow limitation described at section 8.2.4, On-Chip Memory Control Register (RAMCR).

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	RMD	0	R/W	On-Chip Memory Access Mode Specifies the right of access to the on-chip memory from the virtual address space. 0: An access in privileged mode is allowed. (An address error exception occurs in user mode.) 1: An access in user/ privileged mode is allowed.
8	RP	0	R/W	On-Chip Memory Protection Enable Selects whether or not to use the protective functions using ITLB and UTLB for accessing the on-chip memory from the virtual address space. 0: Protective functions are not used. 1: Protective functions are used. For further details, refer to section 9.4, On-Chip Memory Protective Functions.

Bit	Bit Name	Initial Value	R/W	Description
7	IC2W	0	R/W	IC Two-Way Mode For further details, refer to section 8.4.3, IC Two-Way Mode.
6	OC2W	0	R/W	OC Two-Way Mode For further details, refer to section 8.3.6, OC Two-Way Mode.
5	ICWPD	0	R/W	IC Way Prediction Disable For further details, refer to section 8.4.4, Instruction Cache Way Prediction Operation.
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.2.2 OL memory Transfer Source Address Register 0 (LSA0)

When MMUCR.AT = 0 or RAMCR.RP = 0, the LSA0 specifies the transfer source physical address for block transfer to page 0A or 0B of the OL memory.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	L0SADR												
Initial value :	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L0SADR						—	—	—	—	L0SSZ					
Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 10	L0SADR	Undefined	R/W	OL memory Page 0 Block Transfer Source Address When MMUCR.AT = 0 or RAMCR.RP = 0, these bits specify the transfer source physical address for block transfer to page 0A or 0B in the OL memory.
9 to 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 to 0	L0SSZ	Undefined	R/W	OL memory Page 0 Block Transfer Source Address Select When MMUCR.AT = 0 or RAMCR.RP = 0, these bits select whether the operand addresses or L0SADR values are used as bits 15 to 10 of the transfer source physical address for block transfer to page 0A or 0B in the OL memory. L0SSZ[5:0] correspond to the transfer source physical addresses [15:10]. 0: The operand address is used as the transfer source physical address. 1: The L0SADR value is used as the transfer source physical address. Settable values: 111111: Transfer source physical address is specified in 1-Kbyte units. 111110: Transfer source physical address is specified in 2-Kbyte units. 111100: Transfer source physical address is specified in 4-Kbyte units. 111000: Transfer source physical address is specified in 8-Kbyte units. 110000: Transfer source physical address is specified in 16-Kbyte units. 100000: Transfer source physical address is specified in 32-Kbyte units. 000000: Transfer source physical address is specified in 64-Kbyte units. Settings other than the ones given above are prohibited.

9.2.3 OL memory Transfer Source Address Register 1 (LSA1)

When MMUCR.AT = 0 or RAMCR.RP = 0, the LSA1 specifies the transfer source physical address for block transfer to page 1A or 1B in the OL memory.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	L1SADR												
Initial value :	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L1SADR						—	—	—	—	L1SSZ					
Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 10	L1SADR	Undefined	R/W	OL memory Page 1 Block Transfer Source Address When MMUCR.AT = 0 or RAMCR.RP = 0, these bits specify transfer source physical address for block transfer to page 1A or 1B in the OL memory.
9 to 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 to 0	L1SSZ	Undefined	R/W	OL memory Page 1 Block Transfer Source Address Select When MMUCR.AT = 0 or RAMCR.RP = 0, these bits select whether the operand addresses or L1SADR values are used as bits 15 to 10 of the transfer source physical address for block transfer to page 1A or 1B in the OL memory. L1SSZ bits [5:0] correspond to the transfer source physical addresses [15:10]. 0: The operand address is used as the transfer source physical address. 1: The L1SADR value is used as the transfer source physical address. Settable values: 111111: Transfer source physical address is specified in 1-Kbyte units. 111110: Transfer source physical address is specified in 2-Kbyte units. 111100: Transfer source physical address is specified in 4-Kbyte units. 111000: Transfer source physical address is specified in 8-Kbyte units. 110000: Transfer source physical address is specified in 16-Kbyte units. 100000: Transfer source physical address is specified in 32-Kbyte units. 000000: Transfer source physical address is specified in 64-Kbyte units. Settings other than the ones given above are prohibited.

9.2.4 OL memory Transfer Destination Address Register 0 (LDA0)

When MMUCR.AT = 0 or RAMCR.RP = 0, LDA0 specifies the transfer destination physical address for block transfer to page 0A or 0B of the OL memory.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	L0DADR												
Initial value :	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L0DADR						—	—	—	—	L0DSZ					
Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 10	L0DADR	Undefined	R/W	OL memory Page 0 Block Transfer Destination Address When MMUCR.AT = 0 or RAMCR.RP = 0, these bits specify transfer destination physical address for block transfer to page 0A or 0B in the OL memory.
9 to 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 to 0	L0DSZ	Undefined	R/W	OL memory Page 0 Block Transfer Destination Address Select When MMUCR.AT = 0 or RAMCR.RP = 0, these bits select whether the operand addresses or L0DADR values are used as bits 15 to 10 of the transfer destination physical address for block transfer to page 0A or 0B in the OL memory. L0DSZ bits [5:0] correspond to the transfer destination physical address bits [15:10]. 0: The operand address is used as the transfer destination physical address. 1: The L0DADR value is used as the transfer destination physical address. Settable values: 111111: Transfer destination physical address is specified in 1-Kbyte units. 111110: Transfer destination physical address is specified in 2-Kbyte units. 111100: Transfer destination physical address is specified in 4-Kbyte units. 111000: Transfer destination physical address is specified in 8-Kbyte units. 110000: Transfer destination physical address is specified in 16-Kbyte units. 100000: Transfer destination physical address is specified in 32-Kbyte units. 000000: Transfer destination physical address is specified in 64-Kbyte units. Settings other than the ones given above are prohibited.

9.2.5 OL memory Transfer Destination Address Register 1 (LDA1)

When MMUCR.AT = 0 or RAMCR.RP = 0, LDA1 specifies the transfer destination physical address for block transfer to page 1A or 1B in the OL memory.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	L1DADR												
Initial value :	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L1DADR						—	—	—	—	L1DSZ					
Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 10	L1DADR	Undefined	R/W	OL memory Page 1 Block Transfer Destination Address When MMUCR.AT = 0 or RAMCR.RP = 0, these bits specify transfer destination physical address for block transfer to page 1A or 1B in the OL memory.
9 to 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 to 0	L1DSZ	Undefined	R/W	OL memory Page 1 Block Transfer Destination Address Select When MMUCR.AT = 0 or RAMCR.RP = 0, these bits select whether the operand addresses or L1DADR values are used as bits 15 to 10 of the transfer destination physical address for block transfer to page 1A or 1B in the OL memory. L1DSZ bits [5:0] correspond to the transfer destination physical addresses [15:10]. 0: The operand address is used as the transfer destination physical address. 1: The L1DADR value is used as the transfer destination physical address. Settable values: 111111: Transfer destination physical address is specified in 1-Kbyte units. 111110: Transfer destination physical address is specified in 2-Kbyte units. 111100: Transfer destination physical address is specified in 4-Kbyte units. 111000: Transfer destination physical address is specified in 8-Kbyte units. 110000: Transfer destination physical address is specified in 16-Kbyte units. 100000: Transfer destination physical address is specified in 32-Kbyte units. 000000: Transfer destination physical address is specified in 64-Kbyte units. Settings other than the ones given above are prohibited.

9.3 Operation

9.3.1 Instruction Fetch Access from the CPU

(1) OL Memory

Instruction fetch access from the CPU is performed via the cache/RAM internal bus. This access takes more than one cycle.

(2) IL Memory

Instruction fetch access from the CPU is performed directly via the instruction bus for a given virtual address. In the case of successive accesses to the same page of IL memory and as long as no page conflict occurs, the access takes one cycle.

9.3.2 Operand Access from the CPU and Access from the FPU

Note: Operand access is applied for PC relative access (@(disp,pc)).

(1) OL Memory

Access from the CPU or FPU is performed via the operand bus for a given virtual address. Read access from the operand bus by virtual address takes one cycle if the access is made successively to the same page of OL memory and as long as no page conflict occurs. Write access from the operand bus by virtual address takes one cycle as long as no page conflict occurs.

(2) IL Memory

Operand access from the CPU and access from the FPU are performed via the cache/RAM internal bus. Access via the cache/RAM internal bus takes more than one cycle.

9.3.3 Access from the SuperHyway Bus Master Module

On-chip memory is always accessed by the SuperHyway bus master module, such as DMAC, via the SuperHyway bus which is a physical address bus. The same addresses as for the virtual addresses must be used.

9.3.4 OL Memory Block Transfer

High-speed data transfer can be performed through block transfer between the OL memory and external memory without cache utilization.

Data can be transferred from the external memory to the OL memory through a prefetch instruction (PREF). Block transfer from the external memory to the OL memory begins when the PREF instruction is issued to the address in the OL memory area in the virtual address space.

Data can be transferred from the OL memory to the external memory through a write-back instruction (OCBWB). Block transfer from the OL memory to the external memory begins when the OCBWB instruction is issued to the address in the OL memory area in the virtual address space.

In either case, transfer rate is fixed to 32 bytes. Since the start address is always limited to a 32-byte boundary, the lower five bits of the address indicated by Rn are ignored, and are always dealt with as all 0s. In either case, other pages and cache can be accessed during block transfer, but the CPU will stall if the page which is being transferred is accessed before data transfer ends.

The physical addresses [28:0] of the external memory performing data transfers with the OL memory are specified as follows according to whether the MMU is enabled or disabled.

(1) When MMU is Enabled (MMUCR.AT = 1) and RAMCR.RP = 1

An address of the OL memory area is specified to the UTLB VPN field, and to the physical address of the transfer source (in the case of the PREF instruction) or the transfer destination (in the case of the OCBWB instruction) to the PPN field. The ASID, V, SZ, SH, PR, and D bits have the same meaning as normal address conversion; however, the C and WT bits have no meaning in this page.

When the PREF instruction is issued to the OL memory area, address conversion is performed in order to generate the physical address bits [28:10] in accordance with the SZ bit specification. The physical address bits [9:5] are generated from the virtual address prior to address conversion. The physical address bits [4:0] are fixed to 0. Block transfer is performed to the OL memory from the external memory which is specified by these physical addresses.

When the OCBWB instruction is issued to the OL memory area, address conversion is performed in order to generate the physical address bits [28:10] in accordance with the SZ bit specification. The physical address bits [9:5] are generated from the virtual address prior to address conversion. The physical address bits [4:0] are fixed to 0. Block transfer is performed from the OL memory to the external memory specified by these physical addresses.

In PREF or OCBWB instruction execution, an MMU exception is checked as read type. After the MMU execution check, a TLB miss exception or protection error exception occurs if necessary. If an exception occurs, the block transfer is inhibited.

(2) When MMU is Disabled (MMUCR.AT = 0) or RAMCR.RP = 0

The transfer source physical address in block transfer to page 0A or 0B in the OL memory is set in the L0SADR bits of the LSA0 register. And the L0SSZ bits in the LSA0 register choose either the virtual addresses specified through the PRFF instruction or the L0SADR values as bits 15 to 10 of the transfer source physical address. In other words, the transfer source area can be specified in units of 1 Kbyte to 64 Kbytes.

The transfer destination physical address in block transfer from page 0A or 0B in the OL memory is set in the L0DADR bits of the LDA0 register. And the L0DSZ bits in the LDA0 register choose either the virtual addresses specified through the OCBWB instruction or the L0DADR values as bits 15 to 10 of the transfer destination physical address. In other words, the transfer source area can be specified in units of 1 Kbyte to 64 Kbytes.

Block transfer to page 1A or 1B in the OL memory is set to LSA1 and LDA1 as with page 0A or 0B in the OL memory.

When the PREF instruction is issued to the OL memory area, the physical address bits [28:10] are generated in accordance with the LSA0 or LSA1 specification. The physical address bits [9:5] are generated from the virtual address. The physical address bits [4:0] are fixed to 0. Block transfer is performed from the external memory specified by these physical addresses to the OL memory.

When the OCBWB instruction is issued to the OL memory area, the physical address bits [28:10] are generated in accordance with the LDA0 or LDA1 specification. The physical address bits [9:5] are generated from the virtual address. The physical address bits [4:0] are fixed to 0. Block transfer is performed from the OL memory to the external memory specified by these physical addresses.

9.4 On-Chip Memory Protective Functions

This LSI implements the following protective functions to the on-chip memory by using the on-chip memory access mode bit (RMD) and the on-chip memory protection enable bit (RP) in the on-chip memory control register (RAMCR).

- Protective functions for access from the CPU and FPU

  When RAMCR.RMD = 0, and the on-chip memory is accessed in user mode, it is determined to be an address error exception.

  When MMUCR.AT = 1 and RAMCR.RP = 1, MMU exception and address error exception are checked in the on-chip memory area which is a part of area P4 as with the area P0/P3/U0.

The above descriptions are summarized in table 9.5.

Table 9.5 Protective Function Exceptions to Access On-Chip Memory

MMUCR.AT	RAMCR.RP	SR.MD	RAMCR. RMD	Always Occurring Exceptions	Possibly Occurring Exceptions
0	x	0	0	Address error exception	—
			1	—	—
		1	x	—	—
1	0	0	0	Address error exception	—
			1	—	—
		1	x	—	—
	1	0	0	Address error exception	—
			1	—	MMU exception
		1	x	—	MMU exception

Legend:

x: Don't care

9.5 Usage Notes

9.5.1 Page Conflict

In the event of simultaneous access to the same page from different buses, page conflict occurs. Although each access is completed correctly, this kind of conflict tends to lower OL memory accessibility. Therefore it is advisable to provide all possible preventative software measures. For example, conflicts will not occur if each bus accesses different pages.

9.5.2 Access Across Different Pages

(1) OL Memory

Read access from the operand bus is performed in one cycle when the access is made successively to the same page but takes multiple cycles (a maximum of two wait cycles may be required) when the access is made across pages or the previous access was made to memory other than OL memory. For this reason, from the viewpoint of performance optimization, it is recommended to design the software such that the page corresponding to the address for read access from the operand bus does not change so often.

(2) IL Memory

Access from the instruction bus is performed in one cycle when the access is made successively to the same page but takes multiple cycles (a maximum of two wait cycles may be required) when the access is made across pages or the previous access was made to memory other than IL memory. For this reason, from the viewpoint of performance optimization, it is recommended to design the software such that the target page does not change so often in access from the instruction bus. For example, allocating a separate program for each page will deliver better efficiency.

9.5.3 On-Chip Memory Coherency

(1) OL Memory

In order to allocate instructions in the OL memory, write an instruction to the OL memory, execute the following sequence, then branch to the rewritten instruction.

- SYNCO
- ICBI @Rn

In this case, the target for the ICBI instruction can be any address (OL memory address may be possible) within the range where no address error exception occurs, and cache hit/miss is possible.

(2) IL Memory

In order to allocate instructions in the IL memory, write an instruction to the IL memory, execute the following sequence, then branch to the rewritten instruction.

- SYNCO
- ICBI @Rn

In this case, the target for the ICBI instruction can be any address (IL memory address may be possible) within the range where no address error exception occurs, and cache hit/miss is possible.

9.5.4 Sleep Mode

(1) OL Memory, IL Memory

The SuperHyway bus master module, such as DMAC, cannot access OL memory and IL memory in sleep mode.

9.6 Note on Using 32-Bit Address Extended Mode

In 32-bit address extended mode, L0SADR fields in LSA0, L1SADR fields in LSA1, L0DADR fields in LDA0, and L1DADR fields in LDA1 are extended from 19-bit [28:10] to 22-bit [31:10].

Section 10 Interrupt Controller (INTC)

The interrupt controller (INTC) determines the priority of interrupt sources and controls the flow of interrupt requests to the CPU0/1 (SH-4A). The INTC has registers for setting the priority of each of the interrupts and processing of interrupt requests follows the priority order set in these registers by the user.

10.1 Features

The INTC has the following features:

- The external interrupts and the on-chip peripheral module can be set to the fixed distribution interrupt mode or automatic distribution interrupt mode.

  The interrupt distribution set-up register or peripheral interrupt distribution set-up register is used to set the external interrupt or on-chip peripheral module interrupt to the fixed distribution interrupt mode or automatic distribution interrupt mode.
- Fifteen levels of external interrupt priority can be set

  By setting the interrupt priority registers, the priorities of external interrupts can be selected from 15
- NMI noise canceller function

  An NMI input-level bit indicates the NMI pin state. The bit can be read within the interrupt exception handling routine to confirm the pin state and thus achieve a form of noise cancellation.
- NMI request masking when the block bit (BL) in the status register (SR) is set to 1

  Masking or non-masking of NMI requests when the BL bit in SR is set to 1 can be selected.
- Automatically updates the IMASK bit in SR according to the accepted interrupt level
- Thirty priority levels for interrupts from on-chip peripheral modules

  By setting the ten interrupt priority registers for the on-chip peripheral module interrupts, any of 30 priority levels can be assigned to the individual requesting sources.
- User-mode interrupt disabling function

  An interrupt mask level in the user interrupt mask level register (USERIMASK) can be specified to disable interrupts which do not have higher priority than the specified mask level. This setting can be made in user mode.

Figure 10.1 shows a block diagram of the INTC.

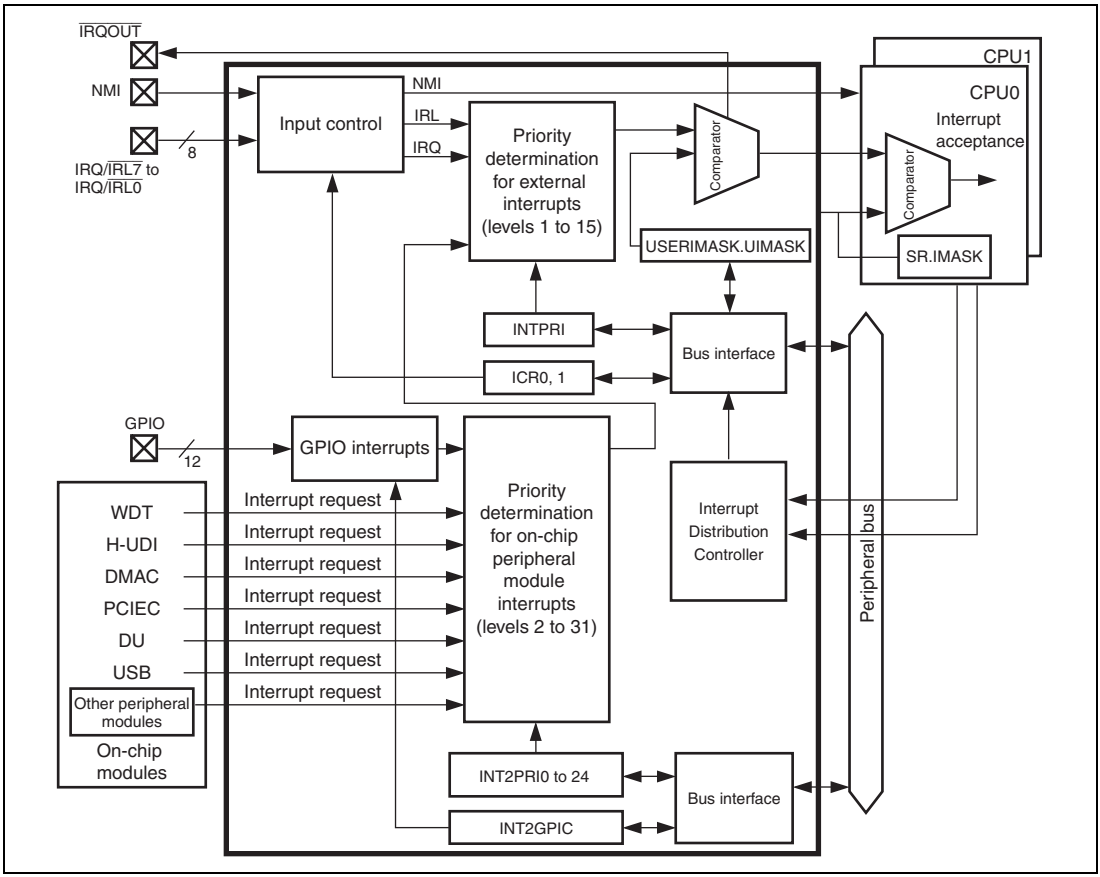


Figure 10.1 Block Diagram of INTC

10.1.1 Interrupt Method

The basic flow of exception handling for interrupts is as follows.

In interrupt exception handling, the contents of the program counter (PC), status register (SR), and general register 15 (R15) are saved in the saved program counter (SPC), saved status register (SSR), and saved general register15 (SGR), and the CPU starts execution of the interrupt exception handling routine at the corresponding vector address. An interrupt exception handling routine is a program written by the user to handle a specific exception. The interrupt exception handling routine is terminated and control returned to the original program by executing a return-from-exception instruction (RTE). This instruction restores the contents of PC and SR and returns control to the normal processing routine at the point at which the exception occurred. The contents of SGR are not written back to R15 by the RTE instruction.

1. The contents of the PC, SR and R15 are saved in SPC, SSR and SGR, respectively.
2. The block (BL) bit in SR is set to 1.
3. The mode (MD) bit in SR is set to 1.
   The register bank (RB) bit in SR is set to 1.
   In a reset, the FPU disable (FD) bit in SR is cleared to 0.
6. The exception code is written to bits 13 to 0 of the interrupt event register (INTEVT).
7. Processing jumps to the start address of the interrupt exception handling routine, vector base register (VBR) + H'600.
8. The processing branches to the corresponding exception handling vector address and the exception handling routine starts.

10.1.2 Interrupt Sources

Table 10.1 shows an example of the interrupt types. The INTC supports both external interrupts and on-chip peripheral module interrupts.

External interrupts refer to the interrupts input through the external NMI, IRL, and IRQ pins.

The IRQ and IRL interrupts are assigned to the same pins in the SH7786. The pin functions are selected to suit the system configuration.

Ether level-sense, or the rising or falling edge, can be selected for the detection of IRQ input.

Table 10.1 Interrupt Sources

Source		Inverted values of pin input L : Low input H: High input (See table 10.8)	Number of Sources (max)	INTEVT	Corresponding registers					Default Priority
					Priority Setting (INTPRI)	Mask/Clear 0,1 (CnINTMSK0, 1 CnINTMSKCLR0, 1)	Mask/Clear 2 (INTMSK2 INTMSKCLR2)	Distribution Mode Setting (INTDISTCR0, INTDISTCR1)	Interrupt Source Indicate (INTREQ)	
External Interrupts	NMI	—	1	H'1C0	—	—		—	—	High
	IRL	$\overline{IRL}$[3:0] = LLLL (H'0)	2	H'200	—	MSK1[31] MSKCLR1[31]	MSK2[31] MSKCLR2[31]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = LLLL (H'0)		H'900	—	MSK1[30] MSKCLR1[30]	MSK2[15] MSKCLR2[15]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = LLLH (H'1)		H'220	—	MSK1[31] MSKCLR1[31]	MSK2[30] MSKCLR2[30]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = LLLH (H'1)		H'920	—	MSK1[30] MSKCLR1[30]	MSK2[14] MSKCLR2[14]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = LLHL (H'2)		H'240	—	MSK1[31] MSKCLR1[31]	MSK2[29] MSKCLR2[29]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = LLHL (H'2)		H'940	—	MSK1[30] MSKCLR1[30]	MSK2[13] MSKCLR2[13]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = LLHH (H'3)		H'260	—	MSK1[31] MSKCLR1[31]	MSK2[28] MSKCLR2[28]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = LLHH (H'3)		H'960	—	MSK1[30] MSKCLR1[30]	MSK2[12] MSKCLR2[12]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = LHLL (H'4)		H'280	—	MSK1[31] MSKCLR1[31]	MSK2[27] MSKCLR2[27]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = LHLL (H'4)		H'980	—	MSK1[30] MSKCLR1[30]	MSK2[11] MSKCLR2[11]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = LHLH (H'5)		H'2A0	—	MSK1[31] MSKCLR1[31]	MSK2[26] MSKCLR2[26]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = LHLH (H'5)		H'9A0	—	MSK1[30] MSKCLR1[30]	MSK2[10] MSKCLR2[10]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = LHHL (H'6)		H'2C0	—	MSK1[31] MSKCLR1[31]	MSK2[25] MSKCLR2[25]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = LHHL (H'6)		H'9C0	—	MSK1[30] MSKCLR1[30]	MSK2[9] MSKCLR2[9]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = LHHH (H'7)		H'2E0	—	MSK1[31] MSKCLR1[31]	MSK2[24] MSKCLR2[24]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = LHHH (H'7)		H'9E0	—	MSK1[30] MSKCLR1[30]	MSK2[8] MSKCLR2[8]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = HLLL (H'8)		H'300	—	MSK1[31] MSKCLR1[31]	MSK2[23] MSKCLR2[23]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = HLLL (H'8)		H'A00	—	MSK1[30] MSKCLR1[30]	MSK2[7] MSKCLR2[7]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = HLLH (H'9)		H'320	—	MSK1[31] MSKCLR1[31]	MSK2[22] MSKCLR2[22]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = HLLH (H'9)		H'A20	—	MSK1[30] MSKCLR1[30]	MSK2[6] MSKCLR2[6]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = HLHL (H'A)		H'340	—	MSK1[31] MSKCLR1[31]	MSK2[21] MSKCLR2[21]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = HLHL (H'A)		H'A40	—	MSK1[30] MSKCLR1[30]	MSK2[5] MSKCLR2[5]	DISTCR1[30]	—	
		$\overline{IRL}$[3:0] = HLHH (H'B)		H'360	—	MSK1[31] MSKCLR1[31]	MSK2[20] MSKCLR2[20]	DISTCR1[31]	—	
		$\overline{IRL}$[7:4] = HLHH (H'B)		H'A60	—	MSK1[30] MSKCLR1[30]	MSK2[4] MSKCLR2[4]	DISTCR1[30]	—	Low

Source			Number of Sources (max)	INTEVT	Corresponding registers					Default Priority
		Inverted values of pin input L : Low input H: High input (See table 10.8)			Priority Setting (INTPRI)	Mask/Clear 0,1 (CnINTMSK0, 1 CnINTMSKCLR0, 1)	Mask/Clear 2 (INTMSK2 INTMSKCLR2)	Distribution Mode Setting (INTDISTCR0, INTDISTCR1)	Interrupt Source Indicate (INTREQ)	
External Interrupts	IRL	$\overline{\text{IRL}}$[3:0] = HHLL (H'C)	2	H'380	—	MSK1[31] MSKCLR1[31]	MSK2[19] MSKCLR2[19]	DISTCR1[31]	—	High
		$\overline{\text{IRL}}$[7:4] = HHLL (H'C)		H'A80	—	MSK1[30] MSKCLR1[30]	MSK2[3] MSKCLR2[3]	DISTCR1[30]	—	
		$\overline{\text{IRL}}$[3:0] = HHLH (H'D)		H'3A0	—	MSK1[31] MSKCLR1[31]	MSK2[18] MSKCLR2[18]	DISTCR1[31]	—	
		$\overline{\text{IRL}}$[7:4] = HHLH (H'D)		H'AA0	—	MSK1[30] MSKCLR1[30]	MSK2[2] MSKCLR2[2]	DISTCR1[30]	—	
		$\overline{\text{IRL}}$[3:0] = HHHL (H'E)		H'3C0	—	MSK1[31] MSKCLR1[31]	MSK2[17] MSKCLR2[17]	DISTCR1[31]	—	
		$\overline{\text{IRL}}$[7:4] = HHHL (H'E)		H'AC0	—	MSK1[30] MSKCLR1[30]	MSK2[1] MSKCLR2[1]	DISTCR1[30]	—	Low

Source			Number of Sources (max)	INTEVT	Corresponding registers					Default Priority
					Priority Setting (INTPRI)	Mask/Clear 0,1 (CnINTMSK0, 1 CnINTMSKCLR0, 1)	Mask/Clear 2 (INTMSK2 INTMSKCLR2)	Distribution Mode Setting (INTDISTCR0, 1)	Interrupt Source Indicate (INTREQ)	
External Interrupts	IRQ	IRQ[0]	8	H'200	INTPRI[31:28]	MSK0[31] MSKCLR0[31]	—	DISTCR0[24]	INTREQ[31]	High
		IRQ[1]		H'240	INTPRI[27:24]	MSK0[30] MSKCLR0[30]	—	DISTCR0[25]	INTREQ[30]	
		IRQ[2]		H'280	INTPRI[23:20]	MSK0[29] MSKCLR0[29]	—	DISTCR0[26]	INTREQ[29]	
		IRQ[3]		H'2C0	INTPRI[19:16]	MSK0[28] MSKCLR0[28]	—	DISTCR0[27]	INTREQ[28]	
		IRQ[4]		H'300	INTPRI[15:12]	MSK0[27] MSKCLR0[27]	—	DISTCR0[28]	INTREQ[27]	
		IRQ[5]		H'340	INTPRI[11:8]	MSK0[26] MSKCLR0[26]	—	DISTCR0[29]	INTREQ[26]	
		IRQ[6]		H'380	INTPRI[7:4]	MSK0[25] MSKCLR0[25]	—	DISTCR0[30]	INTREQ[25]	
		IRQ[7]		H'3C0	INTPRI[3:0]	MSK0[24] MSKCLR0[24]	—	DISTCR0[31]	INTREQ[24]	Low

Source		Number of Sources (max)	INTEVT	Corresponding registers						Default Priority
				Priority Setting (INT2PRI00 to 24)	Mask/Clear 0,1 (CnINT2MSK0 to 3 CnINT2MSKCLR0 to 3)	Interrupt Mask (PERIACKMSK 0 to 3)	Distribution Mode Setting (INT2DISTCR 0 to 3)	Interrupt Source Indicate (CnINT2Ax_0 to 3)*[1]	Interrupt Detail Source Indicate (INT2B00 to 44)	
Internal Peripheral Module Interrupts	WDT	1	H'3E0	PRI00[4:0]	MSK0[0] MSKCLR0[0]	MSK0[0]	DISTCR0[0]	2Ax_0[0]	2B00[0]	High
	TMU-ch0 to 2	4	H'400	PRI01[28:24]	MSK1[31] MSKCLR1[31]	MSK1[31]	DISTCR1[31]	2Ax_1[31]	2B01[0]	
			H'420	PRI01[20:16]	MSK1[30] MSKCLR1[30]	MSK1[30]	DISTCR1[30]	2Ax_1[30]	2B01[1]	
			H'440	PRI01[12:8]	MSK1[29] MSKCLR1[29]	MSK1[29]	DISTCR1[29]	2Ax_1[29]	2B01[2]	
			H'460	PRI01[4:0]	MSK1[28] MSKCLR1[28]	MSK1[28]	DISTCR1[28]	2Ax_1[28]	2B01[3]	
	TMU-ch3 to 5	3	H'480	PRI02[28:24]	MSK1[27] MSKCLR1[27]	MSK1[27]	DISTCR1[27]	2Ax_1[27]	2B02[0]	
			H'4A0	PRI02[20:16]	MSK1[26] MSKCLR1[26]	MSK1[26]	DISTCR1[26]	2Ax_1[26]	2B02[1]	
			H'4C0	PRI02[12:8]	MSK1[25] MSKCLR1[25]	MSK1[25]	DISTCR1[25]	2Ax_1[25]	2B02[2]	
	—	—	H'4E0	—	—	—	—	—	—	
	DMAC (0)	7	H'500	PRI03[28:24]	MSK1[23] MSKCLR1[23]	MSK1[23]	DISTCR1[23]	2Ax_1[23]	2B03[0]	
			H'520	PRI03[20:16]	MSK1[22] MSKCLR1[22]	MSK1[22]	DISTCR1[22]	2Ax_1[22]	2B03[1]	
			H'540	PRI03[12:8]	MSK1[21] MSKCLR1[21]	MSK1[21]	DISTCR1[21]	2Ax_1[21]	2B03[2]	
			H'560	PRI03[4:0]	MSK1[20] MSKCLR1[20]	MSK1[20]	DISTCR1[20]	2Ax_1[20]	2B03[3]	
			H'580	PRI04[28:24]	MSK1[19] MSKCLR1[19]	MSK1[19]	DISTCR1[19]	2Ax_1[19]	2B03[4]	
			H'5A0	PRI04[20:16]	MSK1[18] MSKCLR1[18]	MSK1[18]	DISTCR1[18]	2Ax_1[18]	2B03[5]	
			H'5C0	PRI04[12:8]	MSK1[17] MSKCLR1[17]	MSK1[17]	DISTCR1[17]	2Ax_1[17]	2B03[6]	
	H-UDI1	1	H'5E0	PRI04[4:0]	MSK1[16] MSKCLR1[16]	MSK1[16]	DISTCR1[16]	2Ax_1[16]	2B04[1]	
	H-UDI0	1	H'600	PRI05[28:24]	MSK1[15] MSKCLR1[15]	MSK1[15]	DISTCR1[15]	2Ax_1[15]	2B04[0]	
	DMAC (1)	12	H'620	PRI05[20:16]	MSK1[14] MSKCLR1[14]	MSK1[14]	DISTCR1[14]	2Ax_1[14]	2B05 [7],[5],[0]	
			H'640	PRI05[12:8]	MSK1[13] MSKCLR1[13]	MSK1[13]	DISTCR1[13]	2Ax_1[13]	2B05 [15],[13],[8]	
			H'660	PRI05[4:0]	MSK1[12] MSKCLR1[12]	MSK1[12]	DISTCR1[12]	2Ax_1[12]	2B05 [23],[21],[16]	
			H'680	PRI06[28:24]	MSK1[11] MSKCLR1[11]	MSK1[11]	DISTCR1[11]	2Ax_1[11]	2B05 [31],[29],[24]	
	HPB	14	H'6A0	PRI06[20:16]	MSK1[10] MSKCLR1[10]	MSK1[10]	DISTCR1[10]	2Ax_1[10]	2B06[6:0]	
			H'6C0	PRI06[12:8]	MSK1[9] MSKCLR1[9]	MSK1[9]	DISTCR1[9]	2Ax_1[9]	2B06[11:7]	
			H'6E0	PRI06[4:0]	MSK1[8] MSKCLR1[8]	MSK1[8]	DISTCR1[8]	2Ax_1[8]	2B06[13:12]	Low

Source		Number of Sources (max)	INTEVT	Corresponding registers						Default Priority
				Priority Setting (INT2PRI00 to 24)	Mask/Clear 0,1 (CnINT2MSK0 to 3 CnINT2MSKCLR0 to 3)	Interrupt Mask (PERIACKMSK 0 to 3)	Distribution Mode Setting (INT2DISTCR 0 to 3)	Interrupt Source Indicate (CnINT2Ax_0 to 3)*1	Interrupt Detail Source Indicate (INT2B00 to 44)	
Internal Peripheral Module Interrupts	SCIF-ch0	4	H'700	PRI07[28:24]	MSK1[7] MSKCLR1[7]	MSK1[7]	DISTCR1[7]	2Ax_1[7]	2B07[0]	High
			H'720	PRI07[20:16]	MSK1[6] MSKCLR1[6]	MSK1[6]	DISTCR1[6]	2Ax_1[6]	2B07[1]	
			H'740	PRI07[12:8]	MSK1[5] MSKCLR1[5]	MSK1[5]	DISTCR1[5]	2Ax_1[5]	2B07[2]	
			H'760	PRI07[4:0]	MSK1[4] MSKCLR1[4]	MSK1[4]	DISTCR1[4]	2Ax_1[4]	2B07[3]	
	SCIF-ch1	4	H'780	PRI08[28:24]	MSK1[3] MSKCLR1[3]	MSK1[3]	DISTCR1[3]	2Ax_1[3]	2B08[3:0]	
	TMU-ch6 to 8	3	H'7A0	PRI08[20:16]	MSK1[2] MSKCLR1[2]	MSK1[2]	DISTCR1[2]	2Ax_1[2]	2B09[2:0]	
	TMU-ch9 to 11	3	H'7C0	PRI08[12:8]	MSK1[1] MSKCLR1[1]	MSK1[1]	DISTCR1[1]	2Ax_1[1]	2B10[2:0]	
	—	—	H'7E0	—	—	—	—	—	—	
	SCIF-ch2	4	H'840	PRI09[12:8]	MSK2[29] MSKCLR2[29]	MSK2[29]	DISTCR2[29]	2Ax_2[29]	2B12[3:0]	
	SCIF-ch3	4	H'860	PRI09[4:0]	MSK2[28] MSKCLR2[28]	MSK2[28]	DISTCR2[28]	2Ax_2[28]	2B13[3:0]	
	SCIF-ch4	4	H'880	PRI10[28:24]	MSK2[27] MSKCLR2[27]	MSK2[27]	DISTCR2[27]	2Ax_2[27]	2B14[3:0]	
	SCIF-ch5	4	H'8A0	PRI10[20:16]	MSK2[26] MSKCLR2[26]	MSK2[26]	DISTCR2[26]	2Ax_2[26]	2B15[3:0]	
	Ether	33	H'8C0	PRI10[12:8]	MSK2[25] MSKCLR2[25]	MSK2[25]	DISTCR2[25]	2Ax_2[25]	2B16 [5:4],[2:0]	
			H'8E0	PRI10[4:0]	MSK2[24] MSKCLR2[24]	MSK2[24]	DISTCR2[24]	2Ax_2[24]	2B17 [30],[27: 0]	
	PCIEC0	68	H'AE0	PRI14[4:0]	MSK2[8] MSKCLR2[8]	MSK2[8]	DISTCR2[8]	2Ax_2[8]	2B18[21:0]	
			H'B00	PRI15[28:24]	MSK2[7] MSKCLR2[7]	MSK2[7]	DISTCR2[7]	2Ax_2[7]	2B19[21:0]	
			H'B20	PRI15[20:16]	MSK2[6] MSKCLR2[6]	MSK2[6]	DISTCR2[6]	2Ax_2[6]	2B20[23:0]	
	PCIEC1	68	H'B40	PRI15[12:8]	MSK2[5] MSKCLR2[5]	MSK2[5]	DISTCR2[5]	2Ax_2[5]	2B21[21:0]	
			H'B60	PRI15[4:0]	MSK2[4] MSKCLR2[4]	MSK2[4]	DISTCR2[4]	2Ax_2[4]	2B22[21:0]	
			H'B80	PRI16[28:24]	MSK2[3] MSKCLR2[3]	MSK2[3]	DISTCR2[3]	2Ax_2[3]	2B23[23:0]	
	USB	4	H'BA0	PRI16[20:16]	MSK2[2] MSKCLR2[2]	MSK2[2]	DISTCR2[2]	2Ax_2[2]	2B24[3:0]	
	SDI0	4	H'BC0	PRI16[12:8]	MSK2[1] MSKCLR2[1]	MSK2[1]	DISTCR2[1]	2Ax_2[1]	2B25[3:0]	
	SDI1	4	H'BE0	PRI16[4:0]	MSK2[0] MSKCLR2[0]	MSK2[0]	DISTCR2[0]	2Ax_2[0]	2B26[3:0]	
	—	—	H'C00	—	—	—	—	—	—	
			H'C20	—	—	—	—	—	—	
			H'C40	—	—	—	—	—	—	
	—	—	H'C60	—	—	—	—	—	—	
			H'C80	—	—	—	—	—	—	
			H'CA0	—	—	—	—	—	—	
	I2C-ch0	12	H'CC0	PRI18[12:8]	MSK3[25] MSKCLR3[25]	MSK3[25]	DISTCR3[25]	2Ax_3[25]	2B29 [14:8],[4:0]	Low

Source		Number of Sources (max)	INTEVT	Corresponding registers						Default Priority
				Priority Setting (INT2PRI00 to 24)	Mask/Clear 0,1 (CnINT2MSK0 to 3 CnINT2MSKCLR0 to 3)	Interrupt Mask (PERIACKMSK 0 to 3)	Distribution Mode Setting (INT2DISTCR 0 to 3)	Interrupt Source Indicate (CnINT2Ax_0 to 3)*[1]	Interrupt Detail Source Indicate (INT2B00 to 44)	
Internal Peripheral Module Interrupts	I2C-ch1	12	H'CE0	PRI18[4:0]	MSK3[24] MSKCLR3[24]	MSK3[24]	DISTCR3[24]	2Ax_3[24]	2B30 [14:8],[4:0]	High
	DU	13	H'D00	PRI19[28:24]	MSK3[23] MSKCLR3[23]	MSK3[23]	DISTCR3[23]	2Ax_3[23]	2B31[15:14], [11][9:0]	
	SSI-ch0	1	H'D20	PRI19[20:16]	MSK3[22] MSKCLR3[22]	MSK3[22]	DISTCR3[22]	2Ax_3[22]	2B32[0]	
	SSI-ch1	1	H'D40	PRI19[12:8]	MSK3[21] MSKCLR3[21]	MSK3[21]	DISTCR3[21]	2Ax_3[21]	2B32[1]	
	SSI-ch2	1	H'D60	PRI19[4:0]	MSK3[20] MSKCLR3[20]	MSK3[20]	DISTCR3[20]	2Ax_3[20]	2B32[2]	
	SSI-ch3	1	H'D80	PRI20[28:24]	MSK3[19] MSKCLR3[19]	MSK3[19]	DISTCR3[19]	2Ax_3[19]	2B32[3]	
	PCIEC2	68	H'DA0	PRI20[20:16]	MSK3[18] MSKCLR3[18]	MSK3[18]	DISTCR3[18]	2Ax_3[18]	2B33[21:0]	
			H'DC0	PRI20[12:8]	MSK3[17] MSKCLR3[17]	MSK3[17]	DISTCR3[17]	2Ax_3[17]	2B34[21:0]	
			H'DE0	PRI20[4:0]	MSK3[16] MSKCLR3[16]	MSK3[16]	DISTCR3[16]	2Ax_3[16]	2B35[23:0]	
	HAC-ch0	1	H'E00	PRI21[28:24]	MSK3[15] MSKCLR3[15]	MSK3[15]	DISTCR3[15]	2Ax_3[15]	2B36[0]	
	HAC-ch1	1	H'E20	PRI21[20:16]	MSK3[14] MSKCLR3[14]	MSK3[14]	DISTCR3[14]	2Ax_3[14]	2B36[1]	
	FLCTL	4	H'E40	PRI21[12:8]	MSK3[13] MSKCLR3[13]	MSK3[13]	DISTCR3[13]	2Ax_3[13]	2B37[3:0]	
	—	—	H'E60	—	—	—	—	—	—	
	HSPI	1	H'E80	PRI22[28:24]	MSK3[11] MSKCLR3[11]	MSK3[11]	DISTCR3[11]	2Ax_3[11]	2B39[0]	
	GPIO0	4	H'EA0	PRI22[20:16]	MSK3[10] MSKCLR3[10]	MSK3[10]	DISTCR3[10]	2Ax_3[10]	2B40[7:4]	
	GPIO1	4	H'EC0	PRI22[12:8]	MSK3[9] MSKCLR3[9]	MSK3[9]	DISTCR3[9]	2Ax_3[9]	2B40[3:0]	
	Thermal Sensor	4	H'EE0	PRI22[4:0]	MSK3[8] MSKCLR3[8]	MSK3[8]	DISTCR3[8]	2Ax_3[8]	2B41[3:0]	
	INITCI0	1	H'F00	*[2]	—	—	—	—	—	
	INITCI1	1	H'F20	*[2]	—	—	—	—	—	
	INITCI2	1	H'F40	*[2]	—	—	—	—	—	
	INITCI3	1	H'F60	*[2]	—	—	—	—	—	
	INITCI4	1	H'F80	*[2]	—	—	—	—	—	
	INITCI5	1	H'FA0	*[2]	—	—	—	—	—	
	INITCI6	1	H'FC0	*[2]	—	—	—	—	—	
	INITCI7	1	H'FE0	*[2]	—	—	—	—	—	Low

Notes: 1. CnINT2Ax_0 to 3 means CnINT2A0_0 to 3 or CnINT2A1_0 to 3.

2. It should be set by the Setting Interrupt Priority Order among CPUs (CnICIPRI, n=0, 1).

10.2 Input/Output Pins

Table 10.2 shows the pin configuration.

Table 10.2 INTC Pin Configuration

Pin Name	Function	I/O	Description
NMI	Nonmaskable interrupt input pin	Input	Nonmaskable interrupt request signal input
IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL0}}$	External interrupt input pins	Input	Interrupt request signal input of IRQ7 to IRQ0 or $\overline{\text{IRL}}$[7:4] and $\overline{\text{IRL}}$[3:0] The IRQ/$\overline{\text{IRL7}}$ pin is multiplexed with SCIF0_RTS (SCIF0 I/O), SDIF0D3 (SDIF0 I/O), MODE3 (mode control input), and port H3 (GPIO I/O) pin, the IRQ/$\overline{\text{IRL6}}$ pin is multiplexed with SCIF0_SCK (SCIF0 I/O), SDIF0D2 (SDIF0 I/O), MODE2 (mode control input), and port H2 (GPIO I/O) pin, the IRQ/$\overline{\text{IRL5}}$ pin is multiplexed with SCIF0_RXD (SCIF0 Input), SDIF0D1 (SDIF0 I/O), MODE1 (mode control input), and port H1 (GPIO I/O) pin, and the IRQ/$\overline{\text{IRL4}}$ pin is multiplexed with SCIF0_TXD (SCIF0 Output), SDIF0D0 (SDIF0 I/O), MODE0 (mode control input), and port H0 (GPIO I/O) pin.
$\overline{\text{IRQOUT}}$	Interrupt request output pin	Output	Informs an external device of the generation of an interrupt request. The $\overline{\text{IRQOUT}}$ pin is multiplexed with I2C_SDA1 (I2C I/O) pin. $\overline{\text{IRQOUT}}$ outputs the low level even if the interrupt request is not accepted by the CPU because of the priority of a generated interrupt request being lower than SR.IMASK. However, $\overline{\text{IRQOUT}}$ is not asserted in the following cases: (1) IRL interrupt • The IRL interrupt is masked by INTMASK1. • The IRL interrupt is masked by INTMASK2. (2) $\overline{\text{IRQ}}$ interrupt • The IRQ interrupt is masked by INTMASK1. • The priority of the IRQ interrupt is set to H'0 by INTPRI (3) On-chip module interrupt • The on-chip module interrupt is masked by INT2MSKR. • The priority is set to H'00 or H'01 by INT2PRI0 to INT2PRI24.

10.3 Register Descriptions

Table 10.3 shows the INTC register configuration. Table 10.4 shows the register states in each operating mode.

Table 10.3 INTC Register Configuration

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync. Clock
Interrupt control register 0	ICR0	R/W	H'FE41 0000	H'1E41 0000	32	SHck
Interrupt control register 1	ICR1	R/W	H'FE41 001C	H'1E41 001C	32	SHck
Interrupt priority register	INTPRI	R/W	H'FE41 0010	H'1E41 0010	32	SHck
Interrupt source register	INTREQ	R/(W)*1	H'FE41 0024	H'1E41 0024	32	SHck
CPU0 Interrupt mask register 0*1	C0INTMSK0*1	R/W	H'FE41 0030	H'1E41 0030	32	SHck
CPU1 Interrupt mask register 0*1	C1INTMSK0*1	R/W	H'FE41 0034	H'1E41 0034	32	SHck
CPU0 Interrupt mask register 1*1	C0INTMSK1*1	R/W	H'FE41 0040	H'1E41 0040	32	SHck
CPU1 Interrupt mask register 1*1	C1INTMSK1*1	R/W	H'FE41 0044	H'1E41 0044	32	SHck
CPU0 Interrupt mask clear register 0*1	C0INTMSKCLR 0*1	R/W	H'FE41 0050	H'1E41 0050	32	SHck
CPU1 Interrupt mask clear register 0*1	C1INTMSKCLR 0*1	R/W	H'FE41 0054	H'1E41 0054	32	SHck
CPU0 Interrupt mask clear register 1*1	C0INTMSKCLR 1*1	R/W	H'FE41 0060	H'1E41 0060	32	SHck
CPU1 Interrupt mask clear register 1*1	C1INTMSKCLR 1*1	R/W	H'FE41 0064	H'1E41 0064	32	SHck
Interrupt mask register 2	INTMSK2	R/W	H'FE41 0068	H'1E41 0068	32	SHck
Interrupt mask clear register 2	INTMSKCLR2	R/W	H'FE41 006C	H'1E41 006C	32	SHck
Register for Interrupt among CPU (CPU0)*4	C0INTICI*3	R/W	H'FE41 0070	H'1E41 0070	32	SHck
Register for Interrupt among CPU (CPU1)*4	C1INTICI*3	R/W	H'FE41 0074	H'1E41 0074	32	SHck
Register for Clearing Interrupt among CPU (CPU0)*4	C0INTICICLR*3	R/W	H'FE41 0080	H'1E41 0080	32	SHck
Register for Clearing Interrupt among CPU (CPU1)*4	C1INTICICLR*3	R/W	H'FE41 0084	H'1E41 0084	32	SHck
Register for Setting Interrupt priority order among CPU (CPU0)*4	C0ICIPRI*3	R/W	H'FE41 0090	H'1E41 0090	32	SHck
Register for Setting Interrupt priority order among CPU (CPU1)*4	C1ICIPRI*3	R/W	H'FE41 0094	H'1E41 0094	32	SHck
Register for Clearing Interrupt Priority among CPU (CPU0)*4	C0ICIPRICLR*3	R/W	H'FE41 00A0	H'1E41 00A0	32	SHck
Register for Clearing Interrupt Priority among CPU (CPU1)*4	C1ICIPRICLR*3	R/W	H'FE41 00A4	H'1E41 00A4	32	SHck

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync. Clock
Interrupt Distribution register 0*[5]	INTDISTCR0*[4]	R/W	H'FE41 00B0	H'1E41 00B0	32	SHck
Interrupt Distribution register 1*[5]	INTDISTCR1*[4]	R/W	H'FE41 00B4	H'1E41 00B4	32	SHck
Interrupt Acknowledgement register*[5]	INTACK*[4]	R	H'FE41 00B8	H'1E41 00B8	32	SHck
Interrupt Acknowledgement Clear register*[5]	INTACKCLR*[4]	W	H'FE41 00BC	H'1E41 00BC	32	SHck
NMI flag control register	NMIFCR	R/(W)*[2]	H'FE41 00C0	H'1E41 00C0	32	SHck
NMI set register	NMISET	R/(W)*[3]	H'FE41 00C4	H'1E41 00C4	32	SHck
IRQ Interrupt acknowledgement mask register*[6]	IRQACKMASK*[4]	R	H'FE41 00D0	H'1E41 00D0	32	SHck
IRL Interrupt acknowledgement mask register*[6]	IRLACKMASK*[4]	R	H'FE41 00D4	H'1E41 00D4	32	SHck
Peripheral Interrupt acknowledgement mask register 0*[6]	PERIACKMASK0*[4]	R	H'FE41 00D8	H'1E41 00D8	32	SHck
Peripheral Interrupt acknowledgement mask register 1*[6]	PERIACKMASK1*[4]	R	H'FE41 00DC	H'1E41 00DC	32	SHck
Peripheral Interrupt acknowledgement mask register 2*[6]	PERIACKMASK2*[4]	R	H'FE41 00E0	H'1E41 00E0	32	SHck
Peripheral Interrupt acknowledgement mask register 3*[6]	PERIACKMASK3*[4]	R	H'FE41 00E4	H'1E41 00E4	32	SHck
User interrupt mask level register	USERIMASK	R/W	H'FE41 1000	H'1E41 1000	32	SHck
Interrupt priority registers	INT2PRI0	R/W	H'FE41 0800	H'1E41 0800	32	SHck
	INT2PRI1	R/W	H'FE41 0804	H'1E41 0804	32	SHck
	INT2PRI2	R/W	H'FE41 0808	H'1E41 0808	32	SHck
	INT2PRI3	R/W	H'FE41 080C	H'1E41 080C	32	SHck
	INT2PRI4	R/W	H'FE41 0810	H'1E41 0810	32	SHck
	INT2PRI5	R/W	H'FE41 0814	H'1E41 0814	32	SHck
	INT2PRI6	R/W	H'FE41 0818	H'1E41 0818	32	SHck
	INT2PRI7	R/W	H'FE41 081C	H'1E41 081C	32	SHck
	INT2PRI8	R/W	H'FE41 0820	H'1E41 0820	32	SHck
	INT2PRI9	R/W	H'FE41 0824	H'1E41 0824	32	SHck
	INT2PRI10	R/W	H'FE41 0828	H'1E41 0828	32	SHck
	INT2PRI11	R/W	H'FE41 082C	H'1E41 082C	32	SHck
	INT2PRI12	R/W	H'FE41 0830	H'1E41 0830	32	SHck
	INT2PRI13	R/W	H'FE41 0834	H'1E41 0834	32	SHck
	INT2PRI14	R/W	H'FE41 0838	H'1E41 0838	32	SHck
	INT2PRI15	R/W	H'FE41 083C	H'1E41 083C	32	SHck
	INT2PRI16	R/W	H'FE41 0840	H'1E41 0840	32	SHck

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync. Clock
Interrupt priority registers	INT2PRI17	R/W	H'FE41 0844	H'1E41 0844	32	SHck
	INT2PRI18	R/W	H'FE41 0848	H'1E41 0848	32	SHck
	INT2PRI19	R/W	H'FE41 084C	H'1E41 084C	32	SHck
	INT2PRI20	R/W	H'FE41 0850	H'1E41 0850	32	SHck
	INT2PRI21	R/W	H'FE41 0854	H'1E41 0854	32	SHck
	INT2PRI22	R/W	H'FE41 0858	H'1E41 0858	32	SHck
	INT2PRI23	R/W	H'FE41 085C	H'1E41 085C	32	SHck
	INT2PRI24	R/W	H'FE41 0860	H'1E41 0860	32	SHck
Peripheral Interrupt Distribution register 0*[5]	INT2DISTCR0 *[4]	R/W	H'FE41 0900	H'1E41 0900	32	SHck
Peripheral Interrupt Distribution register 1*[5]	INT2DISTCR1 *[4]	R/W	H'FE41 0904	H'1E41 0904	32	SHck
Peripheral Interrupt Distribution register 2*[5]	INT2DISTCR2 *[4]	R/W	H'FE41 0908	H'1E41 0908	32	SHck
Peripheral Interrupt Distribution register 3*[5]	INT2DISTCR3 *[4]	R/W	H'FE41 090C	H'1E41 090C	32	SHck
Interrupt source register (not affected by the mask state) (CPU0)	C0INT2A0_0	R	H'FE41 0A00	H'1E41 0A00	32	SHck
Interrupt source register (not affected by the mask state) (CPU0)	C0INT2A0_1	R	H'FE41 0A04	H'1E41 0A04	32	SHck
Interrupt source register (not affected by the mask state) (CPU0)	C0INT2A0_2	R	H'FE41 0A08	H'1E41 0A08	32	SHck
Interrupt source register (not affected by the mask state) (CPU0)	C0INT2A0_3	R	H'FE41 0A0C	H'1E41 0A0C	32	SHck
Interrupt source register (affected by the mask state) (CPU0)	C0INT2A1_0	R	H'FE41 0A10	H'1E41 0A10	32	SHck
Interrupt source register (affected by the mask state) (CPU0)	C0INT2A1_1	R	H'FE41 0A14	H'1E41 0A14	32	SHck
Interrupt source register (affected by the mask state) (CPU0)	C0INT2A1_2	R	H'FE41 0A18	H'1E41 0A18	32	SHck
Interrupt source register (affected by the mask state) (CPU0)	C0INT2A1_3	R	H'FE41 0A1C	H'1E41 0A1C	32	SHck
Interrupt mask register 0 (CPU0)	C0INT2MSK0*[1]	R/W	H'FE41 0A20	H'1E41 0A20	32	SHck
Interrupt mask register 1 (CPU0)	C0INT2MSK1*[1]	R/W	H'FE41 0A24	H'1E41 0A24	32	SHck
Interrupt mask register 2 (CPU0)	C0INT2MSK2*[1]	R/W	H'FE41 0A28	H'1E41 0A28	32	SHck
Interrupt mask register 3 (CPU0)	C0INT2MSK3*[1]	R/W	H'FE41 0A2C	H'1E41 0A2C	32	SHck
Peripheral Interrupt mask clear register 0 (CPU0)	C0INT2MSKCLR0*[1]	R/W	H'FE41 0A30	H'1E41 0A30	32	SHck
Interrupt mask clear register 1 (CPU0)	C0INT2MSKCLR1*[1]	R/W	H'FE41 0A34	H'1E41 0A34	32	SHck
Interrupt mask clear register 2 (CPU0)	C0INT2MSKCLR2*[1]	R/W	H'FE41 0A38	H'1E41 0A38	32	SHck

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync. Clock
Interrupt mask clear register 3 (CPU0)	C0INT2MSKCLR3*[1]	R/W	H'FE41 0A3C	H'1E41 0A3C	32	SHck
Interrupt source register (not affected by the mask state) (CPU1)	C1INT2A0_0	R	H'FE41 0B00	H'1E41 0B00	32	SHck
Interrupt source register (not affected by the mask state) (CPU1)	C1INT2A0_1	R	H'FE41 0B04	H'1E41 0B04	32	SHck
Interrupt source register (not affected by the mask state) (CPU1)	C1INT2A0_2	R	H'FE41 0B08	H'1E41 0B08	32	SHck
Interrupt source register (not affected by the mask state) (CPU1)	C1INT2A0_3	R	H'FE41 0B0C	H'1E41 0B0C	32	SHck
Interrupt source register (affected by the mask state) (CPU1)	C1INT2A1_0	R	H'FE41 0B10	H'1E41 0B10	32	SHck
Interrupt source register (affected by the mask state) (CPU1)	C1INT2A1_1	R	H'FE41 0B14	H'1E41 0B14	32	SHck
Interrupt source register (affected by the mask state) (CPU1)	C1INT2A1_2	R	H'FE41 0B18	H'1E41 0B18	32	SHck
Interrupt source register (affected by the mask state) (CPU1)	C1INT2A1_3	R	H'FE41 0B1C	H'1E41 0B1C	32	SHck
Interrupt mask register 0 (CPU1)	C1INT2MSK0*[1]	R/W	H'FE41 0B20	H'1E41 0B20	32	SHck
Interrupt mask register 1 (CPU1)	C1INT2MSK1*[1]	R/W	H'FE41 0B24	H'1E41 0B24	32	SHck
Interrupt mask register 2 (CPU1)	C1INT2MSK2*[1]	R/W	H'FE41 0B28	H'1E41 0B28	32	SHck
Interrupt mask register 3 (CPU1)	C0INT2MSK3*[1]	R/W	H'FE41 0B2C	H'1E41 0B2C	32	SHck
Peripheral Interrupt mask clear register 0 (CPU1)	C1INT2MSKCLR0*[1]	R/W	H'FE41 0B30	H'1E41 0B30	32	SHck
Interrupt mask clear register 1 (CPU1)	C1INT2MSKCLR1*[1]	R/W	H'FE41 0B34	H'1E41 0B34	32	SHck
Interrupt mask clear register 2 (CPU1)	C1INT2MSKCLR2*[1]	R/W	H'FE41 0B38	H'1E41 0B38	32	SHck
Interrupt mask clear register 3 (CPU1)	C1INT2MSKCLR3*[1]	R/W	H'FE41 0B3C	H'1E41 0B3C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 00	INT2B00	R	H'FE41 0C00	H'1E41 0C00	32	SHck
Peripheral Interrupt Detail Source Indicate Register 01	INT2B01	R	H'FE41 0C04	H'1E41 0C04	32	SHck
Peripheral Interrupt Detail Source Indicate Register 02	INT2B02	R	H'FE41 0C08	H'1E41 0C08	32	SHck
Peripheral Interrupt Detail Source Indicate Register 03	INT2B03	R	H'FE41 0C0C	H'1E41 0C0C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 04	INT2B04	R	H'FE41 0C10	H'1E41 0C10	32	SHck
Peripheral Interrupt Detail Source Indicate Register 05	INT2B05	R	H'FE41 0C14	H'1E41 0C14	32	SHck
Peripheral Interrupt Detail Source Indicate Register 06	INT2B06	R	H'FE41 0C18	H'1E41 0C18	32	SHck

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync. Clock
Peripheral Interrupt Detail Source Indicate Register 07	INT2B07	R	H'FE41 0C1C	H'1E41 0C1C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 08	INT2B08	R	H'FE41 0C20	H'1E41 0C20	32	SHck
Peripheral Interrupt Detail Source Indicate Register 09	INT2B09	R	H'FE41 0C24	H'1E41 0C24	32	SHck
Peripheral Interrupt Detail Source Indicate Register 10	INT2B10	R	H'FE41 0C28	H'1E41 0C28	32	SHck
Peripheral Interrupt Detail Source Indicate Register 11	INT2B11	R	H'FE41 0C2C	H'1E41 0C2C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 12	INT2B12	R	H'FE41 0C30	H'1E41 0C30	32	SHck
Peripheral Interrupt Detail Source Indicate Register 13	INT2B13	R	H'FE41 0C34	H'1E41 0C34	32	SHck
Peripheral Interrupt Detail Source Indicate Register 14	INT2B14	R	H'FE41 0C38	H'1E41 0C38	32	SHck
Peripheral Interrupt Detail Source Indicate Register 15	INT2B15	R	H'FE41 0C3C	H'1E41 0C3C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 16	INT2B16	R	H'FE41 0C40	H'1E41 0C40	32	SHck
Peripheral Interrupt Detail Source Indicate Register 17	INT2B17	R	H'FE41 0C44	H'1E41 0C44	32	SHck
Peripheral Interrupt Detail Source Indicate Register 18	INT2B18	R	H'FE41 0C48	H'1E41 0C48	32	SHck
Peripheral Interrupt Detail Source Indicate Register 19	INT2B19	R	H'FE41 0C4C	H'1E41 0C4C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 20	INT2B20	R	H'FE41 0C50	H'1E41 0C50	32	SHck
Peripheral Interrupt Detail Source Indicate Register 21	INT2B21	R	H'FE41 0C54	H'1E41 0C54	32	SHck
Peripheral Interrupt Detail Source Indicate Register 22	INT2B22	R	H'FE41 0C58	H'1E41 0C58	32	SHck
Peripheral Interrupt Detail Source Indicate Register 23	INT2B23	R	H'FE41 0C5C	H'1E41 0C5C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 24	INT2B24	R	H'FE41 0C60	H'1E41 0C60	32	SHck
Peripheral Interrupt Detail Source Indicate Register 25	INT2B25	R	H'FE41 0C64	H'1E41 0C64	32	SHck
Peripheral Interrupt Detail Source Indicate Register 26	INT2B26	R	H'FE41 0C68	H'1E41 0C68	32	SHck
Peripheral Interrupt Detail Source Indicate Register 27	INT2B27	R	H'FE41 0C6C	H'1E41 0C6C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 28	INT2B28	R	H'FE41 0C70	H'1E41 0C70	32	SHck

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync. Clock
Peripheral Interrupt Detail Source Indicate Register 29	INT2B29	R	H'FE41 0C74	H'1E41 0C74	32	SHck
Peripheral Interrupt Detail Source Indicate Register 30	INT2B30	R	H'FE41 0C78	H'1E41 0C78	32	SHck
Peripheral Interrupt Detail Source Indicate Register 31	INT2B31	R	H'FE41 0C7C	H'1E41 0C7C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 32	INT2B32	R	H'FE41 0C80	H'1E41 0C80	32	SHck
Peripheral Interrupt Detail Source Indicate Register 33	INT2B33	R	H'FE41 0C84	H'1E41 0C84	32	SHck
Peripheral Interrupt Detail Source Indicate Register 34	INT2B34	R	H'FE41 0C88	H'1E41 0C88	32	SHck
Peripheral Interrupt Detail Source Indicate Register 35	INT2B35	R	H'FE41 0C8C	H'1E41 0C8C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 36	INT2B36	R	H'FE41 0C90	H'1E41 0C90	32	SHck
Peripheral Interrupt Detail Source Indicate Register 37	INT2B37	R	H'FE41 0C94	H'1E41 0C94	32	SHck
Peripheral Interrupt Detail Source Indicate Register 38	INT2B38	R	H'FE41 0C98	H'1E41 0C98	32	SHck
Peripheral Interrupt Detail Source Indicate Register 39	INT2B39	R	H'FE41 0C9C	H'1E41 0C9C	32	SHck
Peripheral Interrupt Detail Source Indicate Register 40	INT2B40	R	H'FE41 0CA0	H'1E41 0CA0	32	SHck
Peripheral Interrupt Detail Source Indicate Register 41	INT2B41	R	H'FE41 0CA4	H'1E41 0CA4	32	SHck
Peripheral Interrupt Detail Source Indicate Register 42	INT2B42	R	H'FE41 0CA8	H'1E41 0CA8	32	SHck
Peripheral Interrupt Detail Source Indicate Register 43	INT2B43	R	H'FE41 0CAC	H'1E41 0CAC	32	SHck
Peripheral Interrupt Detail Source Indicate Register 44	INT2B44	R	H'FE41 0CB0	H'1E41 0CB0	32	SHck
GPIO interrupt set register	INT2GPIC	R/W	H'FE41 0CC0	H'1E41 0CC0	32	SHck
Thermal Sensor interrupt set register	INT2THSC	R/W	H'FE41 0CC4	H'1E41 0CC4	32	SHck

Notes: 1. The interrupt source registers (INTREQ) are readable and conditionally writable registers. For details, refer to section 10.3.1 (4).
2. The NMI flag control register (NMIFCR) is readable and conditionally writable register. For details, refer to section 10.3.1 (11).
3. NMI setting register
4. Registers due to the interrupt among CPUs.
5. Registers due to the SMP support.
6. Registers for verifications.

Table 10.4 Register States in Each Operating Mode

Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction
Interrupt control register 0	ICR0	H'x000 0000*	H'x000 0000*	Retained
Interrupt control register 1	ICR1	H'0000 0000	H'0000 0000	Retained
Interrupt priority register	INTPRI	H'0000 0000	H'0000 0000	Retained
Interrupt source register	INTREQ	H'0000 0000	H'0000 0000	Retained
CPU0 Interrupt mask register 0	C0INTMSK0	H'FF00 0000	H'FF00 0000	Retained
CPU1 Interrupt mask register 0	C1INTMSK0	H'FF00 0000	H'FF00 0000	Retained
CPU0 Interrupt mask register 1	C0INTMSK1	H'C000 0000	H'C000 0000	Retained
CPU1 Interrupt mask register 1	C1INTMSK1	H'C000 0000	H'C000 0000	Retained
CPU0 Interrupt mask clear register 0	C0INTMSKCLR0	H'0000 0000	H'0000 0000	Retained
CPU1 Interrupt mask clear register 0	C1INTMSKCLR0	H'0000 0000	H'0000 0000	Retained
CPU0 Interrupt mask clear register 1	C0INTMSKCLR1	H'0000 0000	H'0000 0000	Retained
CPU1 Interrupt mask clear register 1	C1INTMSKCLR1	H'0000 0000	H'0000 0000	Retained
Interrupt mask register 2	INTMSK2	H'0000 0000	H'0000 0000	Retained
Interrupt mask clear register 2	INTMSKCLR2	H'0000 0000	H'0000 0000	Retained
Register for Interrupt among CPU (CPU0)	C0INTICI	H'0000 0000	H'0000 0000	Retained
Register for Interrupt among CPU (CPU1)	C1INTICI	H'0000 0000	H'0000 0000	Retained
Register for Clearing Interrupt among CPU (CPU0)	C0INTICICLR	H'0000 0000	H'0000 0000	Retained
Register for Clearing Interrupt among CPU (CPU1)	C1INTICICLR	H'0000 0000	H'0000 0000	Retained
Register for Settng Interrupt priority order among CPU (CPU0)	C0ICIPRI	H'0000 0000	H'0000 0000	Retained
Register for Settng Interrupt priority order among CPU (CPU1)	C1ICIPRI	H'0000 0000	H'0000 0000	Retained
Register for Clearing Interrupt Priority among CPU (CPU0)	C0ICIPRICLR	H'0000 0000	H'0000 0000	Retained
Register for Clearing Interrupt Priority among CPU (CPU1)	C1ICIPRICLR	H'0000 0000	H'0000 0000	Retained
Interrupt Distribution register 0	INTDISTCR0	H'0000 0000	H'0000 0000	Retained
Interrupt Distribution register 1	INTDISTCR1	H'0000 0000	H'0000 0000	Retained
Interrupt Acknowledgement register	INTACK	H'0000 0000	H'0000 0000	Retained
Interrupt Acknowledgement Clear register	INTACKCLR	H'0000 0000	H'0000 0000	Retained
NMI flag control register	NMIFCR	H'x000 0000*	H'x000 0000*	Retained
NMI set register	NMISET	H'0000 0000	H'0000 0000	Retained
IRQ Interrupt acknowledgement mask register	IRQACKMASK	H'0000 0000	H'0000 0000	Retained
IRL Interrupt acknowledgement mask register	IRLACKMASK	H'0000 0000	H'0000 0000	Retained

Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction
Peripheral Interrupt acknowledgement mask register 0	PERIACKMASK0	H'0000 0000	H'0000 0000	Retained
Peripheral Interrupt acknowledgement mask register 1	PERIACKMASK1	H'0000 0000	H'0000 0000	Retained
Peripheral Interrupt acknowledgement mask register 2	PERIACKMASK2	H'0000 0000	H'0000 0000	Retained
Peripheral Interrupt acknowledgement mask register 3	PERIACKMASK3	H'0000 0000	H'0000 0000	Retained
User interrupt mask level register	USERIMASK	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 0	INT2PRI0	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 1	INT2PRI1	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 2	INT2PRI2	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 3	INT2PRI3	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 4	INT2PRI4	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 5	INT2PRI5	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 6	INT2PRI6	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 7	INT2PRI7	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 8	INT2PRI8	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 9	INT2PRI9	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 10	INT2PRI10	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 11	INT2PRI11	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 12	INT2PRI12	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 13	INT2PRI13	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 14	INT2PRI14	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 15	INT2PRI15	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 16	INT2PRI16	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 17	INT2PRI17	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 18	INT2PRI18	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 19	INT2PRI19	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 20	INT2PRI20	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 21	INT2PRI21	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 22	INT2PRI22	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 23	INT2PRI23	H'0000 0000	H'0000 0000	Retained
Interrupt priority register 24	INT2PRI24	H'0000 0000	H'0000 0000	Retained
Peripheral Interrupt Distribution register 0	INT2DISTCR0	H'0000 0000	H'0000 0000	Retained
Peripheral Interrupt Distribution register 1	INT2DISTCR1	H'0000 0000	H'0000 0000	Retained
Peripheral Interrupt Distribution register 2	INT2DISTCR2	H'0000 0000	H'0000 0000	Retained
Peripheral Interrupt Distribution register 3	INT2DISTCR3	H'0000 0000	H'0000 0000	Retained

Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction
Interrupt source register (not affected by the mask state) (CPU0)	C0INT2A0_0	H'xxxx xxxx	H'xxxx xxxx	Retained
Interrupt source register (not affected by the mask state) (CPU0)	C0INT2A0_1	H'xxxx xxxx	H'xxxx xxxx	Retained
Interrupt source register (not affected by the mask state) (CPU0)	C0INT2A0_2	H'xxxx xxxx	H'xxxx xxxx	Retained
Interrupt source register (not affected by the mask state) (CPU0)	C0INT2A0_3	H'xxxx xxxx	H'xxxx xxxx	Retained
Interrupt source register (affected by the mask state) (CPU0)	C0INT2A1_0	H'0000 0000	H'0000 0000	Retained
Interrupt source register (affected by the mask state) (CPU0)	C0INT2A1_1	H'0000 0000	H'0000 0000	Retained
Interrupt source register (affected by the mask state) (CPU0)	C0INT2A1_2	H'0000 0000	H'0000 0000	Retained
Interrupt source register (affected by the mask state) (CPU0)	C0INT2A1_3	H'0000 0000	H'0000 0000	Retained
Interrupt mask register 0 (CPU0)	C0INT2MSK0	H'FFFF FFFF	H'FFFF FFFF	Retained
Interrupt mask register 1 (CPU0)	C0INT2MSK1	H'FFFF FFFF	H'FFFF FFFF	Retained
Interrupt mask register 2 (CPU0)	C0INT2MSK2	H'FFFF FFFF	H'FFFF FFFF	Retained
Interrupt mask register 3 (CPU0)	C0INT2MSK3	H'FFFF FFFF	H'FFFF FFFF	Retained
Interrupt mask clear register 0 (CPU0)	C0INT2MSKCLR0	H'0000 0000	H'0000 0000	Retained
Interrupt mask clear register 1 (CPU0)	C0INT2MSKCLR1	H'0000 0000	H'0000 0000	Retained
Interrupt mask clear register 2 (CPU0)	C0INT2MSKCLR2	H'0000 0000	H'0000 0000	Retained
Interrupt mask clear register 3 (CPU0)	C0INT2MSKCLR3	H'0000 0000	H'0000 0000	Retained
Interrupt source register (not affected by the mask state) (CPU1)	C1INT2A0_0	H'xxxx xxxx	H'xxxx xxxx	Retained
Interrupt source register (not affected by the mask state) (CPU1)	C1INT2A0_1	H'xxxx xxxx	H'xxxx xxxx	Retained
Interrupt source register (not affected by the mask state) (CPU1)	C1INT2A0_2	H'xxxx xxxx	H'xxxx xxxx	Retained
Interrupt source register (not affected by the mask state) (CPU1)	C1INT2A0_3	H'xxxx xxxx	H'xxxx xxxx	Retained
Interrupt source register (affected by the mask state) (CPU1)	C1INT2A1_0	H'0000 0000	H'0000 0000	Retained
Interrupt source register (affected by the mask state) (CPU1)	C1INT2A1_1	H'0000 0000	H'0000 0000	Retained
Interrupt source register (affected by the mask state) (CPU1)	C1INT2A1_2	H'0000 0000	H'0000 0000	Retained
Interrupt source register (affected by the mask state) (CPU1)	C1INT2A1_3	H'0000 0000	H'0000 0000	Retained
Interrupt mask register 0 (CPU1)	C1INT2MSK0	H'FFFF FFFF	H'FFFF FFFF	Retained

Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction
Interrupt mask register 1 (CPU1)	C1INT2MSK1	H'FFFF FFFF	H'FFFF FFFF	Retained
Interrupt mask register 2 (CPU1)	C1INT2MSK2	H'FFFF FFFF	H'FFFF FFFF	Retained
Interrupt mask register 3 (CPU1)	C1INT2MSK3	H'FFFF FFFF	H'FFFF FFFF	Retained
Interrupt mask clear register 0 (CPU1)	C1INT2MSKCLR0	H'0000 0000	H'0000 0000	Retained
Interrupt mask clear register 1 (CPU1)	C1INT2MSKCLR1	H'0000 0000	H'0000 0000	Retained
Interrupt mask clear register 2 (CPU1)	C1INT2MSKCLR2	H'0000 0000	H'0000 0000	Retained
Interrupt mask clear register 3 (CPU1)	C1INT2MSKCLR3	H'0000 0000	H'0000 0000	Retained
Peripheral Interrupt Detail Source Indicate Register 0	INT2B00	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 1	INT2B01	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 2	INT2B02	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 3	INT2B03	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 4	INT2B04	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 5	INT2B05	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 6	INT2B06	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 7	INT2B07	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 8	INT2B08	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 9	INT2B09	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 10	INT2B10	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 11	INT2B11	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 12	INT2B12	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 13	INT2B13	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 14	INT2B14	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 15	INT2B15	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 16	INT2B16	H'xxxx xxxx	H'xxxx xxxx	Retained

Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction
Peripheral Interrupt Detail Source Indicate Register 17	INT2B17	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 18	INT2B18	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 19	INT2B19	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 20	INT2B20	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 21	INT2B21	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 22	INT2B22	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 23	INT2B23	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 24	INT2B24	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 25	INT2B25	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 26	INT2B26	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 27	INT2B27	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 28	INT2B28	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 29	INT2B29	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 30	INT2B30	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 31	INT2B31	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 32	INT2B32	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 33	INT2B33	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 34	INT2B34	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 35	INT2B35	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 36	INT2B36	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 37	INT2B37	H'xxxx xxxx	H'xxxx xxxx	Retained

Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction
Peripheral Interrupt Detail Source Indicate Register 38	INT2B38	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 39	INT2B39	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 40	INT2B40	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 41	INT2B41	H'xxxx xxxx	H'xxxx xxxx	Retained
Peripheral Interrupt Detail Source Indicate Register 42	INT2B42	H'0000 0000	H'0000 0000	Retained
Peripheral Interrupt Detail Source Indicate Register 43	INT2B43	H'0000 0000	H'0000 0000	Retained
Peripheral Interrupt Detail Source Indicate Register 44	INT2B44	H'0000 0000	H'0000 0000	Retained
GPIO interrupt set register	INT2GPIC	H'0000 0000	H'0000 0000	Retained
Thermal Sensor Interrupt register	INT2THSC	H'0000 0000	H'0000 0000	Retained

Note: * Initial values of ICR0.NMIL and NMIFCR.NMIL depend on the level input to the NMI pin.

10.3.1 External Interrupt Request Registers

(1) Interrupt Control Register 0 (ICR0)

ICR0 is a 32-bit readable and partially writable register that sets the input signal detection mode for the external interrupt input pins and NMI pin, and indicates the level being input on the NMI pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NMIL	MAI	—	—	—	—	NMIB	NMIE	IRLM0	IRLM1	—	—	—	—	—	—
Initial value:	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	NMIL	Undefined	R	NMI Input Level Indicates the signal level being input on the NMI pin. Reading this bit allows the user to know the NMI pin level, and writing is invalid. 0: Low level is being input on the NMI pin 1: High level is being input on the NMI pin
30	MAI	0	R/W	MAI (mask all interrupts) Interrupt Mask Specifies whether all interrupts are masked while the NMI pin is at the low level regardless of the setting of the BL bit in SR of the CPU. 0: Interrupts remain enabled even when the NMI pin goes low 1: Interrupts are disabled when the NMI pin goes low
29 to 26	—	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
25	NMIB	0	R/W	NMI Block Mode Selects whether an NMI interrupt is held until the BL bit in SR is cleared to 0 or detected immediately when the BL bit in SR of the CPU is set to 1. 0: An NMI interrupt is held when the BL bit in SR is set to 1 (initial value) 1: An NMI interrupt is not held when the BL bit in SR is set to 1 Note: If interrupts are accepted with the BL bit in SR set to 1, information saved for any previous exception (SSR, SPC, SGR, and INTEVT) is lost.
24	NMIE	0	R/W	NMI Edge Select Selects whether an interrupt request signal to the NMI pin is detected at the rising edge or the falling edge. 0: An interrupt request is detected at the falling edge of NMI input (initial value) 1: An interrupt request is detected at the rising edge of NMI input
23	IRLM0*[1]*[2]	0	R/W	IRL Pin Mode 0 Selects whether IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ are used as encoded interrupt requests (IRL3 to IRL0) or as four independent interrupts (IRQ3 to IRQ0 interrupts). 0: IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ are used as the encoded interrupt requests (initial value) 1: IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ are used as four independent interrupt requests
22	IRLM1*[1]*[2]	0	R/W	IRL Pin Mode 1 Selects whether IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ are used as 4-bit level-encoded interrupt requests (IRL7 to IRL4 interrupts) or as four independent interrupts (IRQ7 to IRQ4 interrupts). 0: IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ are used as the 4-bit level-encoded interrupt requests (initial value) 1: IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ are used as four independent interrupt requests
21 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. When IRLM0 and IRLM1 are changed from 0 to 1, the IRL interrupt source that has been detected or held is cleared. When IRLM0 and IRLM1 are changed from 1 to 0, the IRL interrupt source that has been detected or held is not cleared.

2. When using the IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ pins or IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ pins as encoded IRL interrupt inputs, set IM00 to IM03 or IM04 to IM07 of the interrupt mask register 0 to 1, respectively.

(2) Interrupt Control Register 1 (ICR1)

ICR1 is a 32-bit readable/writable register that specifies the individual input signal detection modes for the respective external interrupt input pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL0}}$. These settings are only valid when IRLM0 or IRLM1 of ICR0 is set to 1 so that IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ or IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ pins are used as individual interrupts (IRQ7 to IRQ0 interrupts) inputs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IRQ0S		IRQ1S		IRQ2S		IRQ3S		IRQ4S		IRQ5S		IRQ6S		IRQ7S	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31, 30	IRQ0S	00	R/W	IRQn Sense Select Selects whether the corresponding individual pin interrupt signal on the IRQ/IRL7 to IRQ/IRL0 pins is detected on rising or falling edges, or at the high or low level. 00: The interrupt request is detected on falling edges of the IRQn input. 01: The interrupt request is detected on rising edges of the IRQn input. 10: The interrupt request is detected when the IRQn input is at the low level. 11: The interrupt request is detected when the IRQn input is at the high level. Note: n = 0 to 7
29, 28	IRQ1S	00	R/W	
27, 26	IRQ2S	00	R/W	
25, 24	IRQ3S	00	R/W	
23, 22	IRQ4S	00	R/W	
21, 20	IRQ5S	00	R/W	
19, 18	IRQ6S	00	R/W	
17, 16	IRQ7S	00	R/W	
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: When the IRQnS setting is changed from edge sense (IRQnS is 00 or 01) to level sense (IRQnS is 10 or 11), the IRQ interrupt source that has been edge sensed is cleared.When the IRQnS setting is changed from level sense (IRQnS is 10 or 11) to edge sense (IRQnS is 00 or 01), the IRQ interrupt source that has been sensed or held is cleared. When the IRQnS setting is changed form falling-edge sense (IRQnS is 00) to rising edge sense (IRQnS is 01), or changed from rising edge sense (IRQnS is 01) to the falling edge sense (IRQnS is 00), the IRQ interrupt source that has been sensed before changing the setting is not cleared. Likewise, when IRQnS setting is changed from low-level sense (IRQnS is 10) to high-level sense (IRQnS is 11), or changed from high-level sense (IRQnS is 11) to the low-level sense (IRQnS is 10), the IRQ interrupt source that has been sensed before changing the setting is not cleared.

(3) Interrupt Priority Register (INTPRI)

INTPRI is a 32-bit readable/writable register used to set the priorities of IRQ[7:0] (as levels from 15 to 0). These settings are only valid for IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ or IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ when set up as individual IRQ interrupts (IRQ7 to IRQ0 interrupts) by setting the IRLM0 or IRLM1 bit in ICR0 to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IP0				IP1				IP2				IP3			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IP4				IP5				IP6				IP7			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	IP0	H'0	R/W	Set the priority of IRQ0 as an individual pin interrupt request.
27 to 24	IP1	H'0	R/W	Set the priority of IRQ1 as an individual pin interrupt request.
23 to 20	IP2	H'0	R/W	Set the priority of IRQ2 as an individual pin interrupt request.
19 to 16	IP3	H'0	R/W	Set the priority of IRQ3 as an individual pin interrupt request..
15 to 12	IP4	H'0	R/W	Set the priority of IRQ4 as an individual pin interrupt request.
11 to 8	IP5	H'0	R/W	Set the priority of IRQ5 as an individual pin interrupt request.
7 to 4	IP6	H'0	R/W	Set the priority of IRQ6 as an individual pin interrupt request.
3 to 0	IP7	H'0	R/W	Set the priority of IRQ7 as an individual pin interrupt request.

Note: Interrupt priorities are established by setting values from H'F to H'1 in each of the 4-bit fields. A larger value corresponds to a higher priority. When the value H'0 is set in a field, the corresponding interrupt request is masked (initial value).

(4) Interrupt Source Register (INTREQ)

INTREQ is a 32-bit readable and conditionally writable register that indicates which of the IRQ [n] (n = 0 to 7) interrupts is currently asserting a request for the INTC.

Even if an interrupt is masked by the setting in INTPRI or INTMSK0, operation of the corresponding INTREQ bit is not affected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IR0	IR1	IR2	IR3	IR4	IR5	IR6	IR7	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description Edge Detection (IRQnS = 00 or 01)*[1]	Description Level Detection (IRQnS = 10 or 11)*[1]
31	IR0	0	R/(W)	[When read] 0: The corresponding IRQ interrupt request has not been detected. 1: The corresponding IRQ interrupt request has been detected. [When written]*[2] When clearing each bit, write a 0 after having read a 1 from it. Writing 1 to the bit is ignored.	0: The corresponding IRQ interrupt pin is not asserted. 1: The corresponding IRQ interrupt pin is asserted, but the CPU has not accepted the interrupt request yet. Writing have no effect.*[3]
30	IR1	0	R/(W)		
29	IR2	0	R/(W)		
28	IR3	0	R/(W)		
27	IR4	0	R/(W)		
26	IR5	0	R/(W)		
25	IR6	0	R/(W)		
24	IR7	0	R/(W)		
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	

Notes: Values of IR0 to IR3 have no meaning when the IRLM0 bit in ICR0 register is set to 1 (IRL pin mode 0).

Values of IR4 to IR7 have no meaning when the IRLM1 bit in ICR0 register is set to 1 (IRL pin mode 1).

1. n = 0 to 7
2. Write 1 to the bit if it should not be cleared yet.
3. For the method of clearing the IRQ interrupt request that has been detected by level sensing, refer to section 10.7.2.

(5) Interrupt Mask Register 0 (CnINTMSK0, n = 0 to 1)

CnINTMSK0 is a 32-bit readable and conditionally writable register that sets masking for each of the interrupt requests IRQn (n = 0 to 7). To clear the mask setting for an interrupt, write 1 to the corresponding bit in CnINTMSKCLR0. Writing 0 to the bits in CnINTMSK0 has no effect. By reading this register once after writing to this register or after clearing the mask by setting CnIMTMSKCLR0, the time length necessary for reflecting the register value can be assured (the value read is reflected to the mask status).

When using IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ pins or IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ pins for encoded IRL interrupt inputs, write 1 to IM00 to IM03 or IM04 to IM07, respectively.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM00	IM01	IM02	IM03	IM04	IM05	IM06	IM07	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	IM00	1	R/W	Sets masking of individual pin interrupt source on IRQ0.
30	IM01	1	R/W	Sets masking of individual pin interrupt source on IRQ1.
29	IM02	1	R/W	Sets masking of individual pin interrupt source on IRQ2.
28	IM03	1	R/W	Sets masking of individual pin interrupt source on IRQ3.
27	IM04	1	R/W	Sets masking of individual pin interrupt source on IRQ4.
26	IM05	1	R/W	Sets masking of individual pin interrupt source on IRQ5.
25	IM06	1	R/W	Sets masking of individual pin interrupt source on IRQ6.
24	IM07	1	R/W	Sets masking of individual pin interrupt source on IRQ7.
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

[When read]
0: The interrupts are accepted.
1: The interrupts are masked.
[When written]
0: No effect
1: Masks the interrupt

(6) Interrupt Mask Register 1 (CnINTMSK1, n = 0 to 1)

CnINTMSK1 is a 32-bit readable and conditionally writable register that sets masking for IRL interrupt requests. To clear the mask setting for the interrupt, write 1 to the corresponding bit in CnINTMSKCLR1. Writing 0 to the bits in CnINTMSK1 has no effect. By reading this register once after writing to this register or after clearing the mask by setting CnIMTMSKCLR1, the time length necessary for reflecting the register value can be assured (the value read is reflected to the mask status).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM10	IM11	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	IM10	1	R/W	Mask setting for all $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ interrupt sources when pins IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ operate as an encoded interrupt input. [When read] 0: The interrupt is accepted. 1: The interrupt is masked. [When written] 0: No effect 1: Masks the interrupt
30	IM11	1	R/W	Mask setting for all $\overline{\text{IRL7}}$ to $\overline{\text{IRL4}}$ interrupt sources when pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ operate as an encoded interrupt input.
29 to 0	—	All 0	R	Reserved These bits are always read as 1. The write value should always be 1.

(7) Interrupt Mask Clear Register 0 (CnINTMSKCLR0, n = 0 to 1)

CnINTMSKCLR0 is a 32-bit write-only register that clears the mask settings for each of the interrupt requests IRQn (n = 0 to 7). Undefined values are read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IC00	IC01	IC02	IC03	IC04	IC05	IC06	IC07	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	IC00	0	R/W	Clears masking of IRQ0 interrupt.
30	IC01	0	R/W	Clears masking of IRQ1 interrupt.
29	IC02	0	R/W	Clears masking of IRQ2 interrupt.
28	IC03	0	R/W	Clears masking of IRQ3 interrupt.
27	IC04	0	R/W	Clears masking of IRQ4 interrupt.
26	IC05	0	R/W	Clears masking of IRQ5 interrupt.
25	IC06	0	R/W	Clears masking of IRQ6 interrupt.
24	IC07	0	R/W	Clears masking of IRQ7 interrupt.
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

[When read]
Undefined values are read.
[When written]
0: No effect
1: Clears the corresponding interrupt mask (enables the interrupt)

(8) Interrupt Mask Clear Register 1 (CnINTMSKCLR1, n = 0 to 1)

CnINTMSKCLR1 is a 32-bit write-only register that clears the mask settings for the IRL interrupt requests. Undefined values are read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IC10	IC11	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	IC10	0	R/W	Clears masking of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ interrupt sources when $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ operate as an encoded interrupt input. [When read] Undefined values are read. [When written] 0: No effect 1: Clears the corresponding interrupt mask (enables the interrupt)
30	IC11	0	R/W	Clears masking of $\overline{\text{IRL7}}$ to $\overline{\text{IRL4}}$ interrupt sources when $\overline{\text{IRL7}}$ to $\overline{\text{IRL4}}$ operate as an encoded interrupt input.
29 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(9) Interrupt Mask Register 2 (INTMSK2)

INTMSK2 is a 32-bit readable and conditionally writable register that sets masking for IRL interrupt requests for input level pattern on the $\overline{\text{IRL}}$ pins. To clear the mask setting for the interrupt, write 1 to the corresponding bit in INTMSKCLR2. Writing 0 to the bits in INTMSK2 has no effect. By reading this register once after writing to this register or after clearing the mask by setting IMTMSKCLR2, the time length necessary for reflecting the register value can be assured (the value read is reflected to the mask status).

INTMSK2 settings are valid when the IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ pins or IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ pins are used for encoded IRL interrupt inputs, and the corresponding IRL interrupt is not masked by INTMSK1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM015	IM014	IM013	IM012	IM011	IM010	IM009	IM008	IM007	IM006	IM005	IM004	IM003	IM002	IM001	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IM115	IM114	IM113	IM112	IM111	IM110	IM109	IM108	IM107	IM106	IM105	IM104	IM103	IM102	IM101	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31	IM015	0	R/W	Masks the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LLLL (H'0).
30	IM014	0	R/W	Masks the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LLLH (H'1).
29	IM013	0	R/W	Masks the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LLHL (H'2).
28	IM012	0	R/W	Masks the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LLHH (H'3).
27	IM011	0	R/W	Masks the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LHLL (H'4).
26	IM010	0	R/W	Masks the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LHLH (H'5).
25	IM009	0	R/W	Masks the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LHHL (H'6).
24	IM008	0	R/W	Masks the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LHHH (H'7).
23	IM007	0	R/W	Masks the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = HLLL (H'8).

[When read]
0: The interrupt is accepted.
1: The interrupt is masked.
[When written]
0: No effect
1: Masks the interrupt

Bit	Bit Name	Initial Value	R/W	Description
22	IM006	0	R/W	Masks the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HLLH (H'9).
21	IM005	0	R/W	Masks the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HLHL (H'A).
20	IM004	0	R/W	Masks the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HLHH (H'B).
19	IM003	0	R/W	Masks the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HHLL (H'C).
18	IM002	0	R/W	Masks the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HHLH (H'D).
17	IM001	0	R/W	Masks the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HHHL (H'E).
16	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
15	IM115	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LLLL (H'0).
14	IM114	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LLLH (H'1).
13	IM113	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LLHL (H'2).
12	IM112	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LLHH (H'3).
11	IM111	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LHLL (H'4).
10	IM110	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LHLH (H'5).
9	IM109	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LHHL (H'6).
8	IM108	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LHHH (H'7).
7	IM107	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HLLL (H'8).
6	IM106	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HLLH (H'9).
5	IM105	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HLHL (H'A).

[When read]
0: The interrupt is accepted.
1: The interrupt is masked.
[When written]
0: No effect
1: Masks the interrupt

Bit	Bit Name	Initial Value	R/W	Description
4	IM104	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HLHH (H'B). [When read] 0: The interrupt is accepted. 1: The interrupt is masked. [When written] 0: No effect 1: Masks the interrupt
3	IM103	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HHLL (H'C).
2	IM102	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HHLH (H'D).
1	IM101	0	R/W	Masks the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HHHL (H'E).
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

(10) Interrupt Mask Clear Register 2 (INTMSKCLR2)

INTMSKCLR2 is a 32-bit write-only register that clears the mask settings for the IRL interrupt requests for each input level pattern on the $\overline{\text{IRL}}$ pins. Undefined values are read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IC015	IC014	IC013	IC012	IC011	IC010	IC009	IC008	IC007	IC006	IC005	IC004	IC003	IC002	IC001	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC115	IC114	IC113	IC112	IC111	IC110	IC109	IC108	IC107	IC106	IC105	IC104	IC103	IC102	IC101	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31	IC015	—	R/W	Clears masking of the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LLLL (H'0).
30	IC014	—	R/W	Clears masking of the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LLLH (H'1).
29	IC013	—	R/W	Clears masking of the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LLHL (H'2).
28	IC012	—	R/W	Clears masking of the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LLHH (H'3).
27	IC011	—	R/W	Clears masking of the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LHLL (H'4).
26	IC010	—	R/W	Clears masking of the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LHLH (H'5).
25	IC009	—	R/W	Clears masking of the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LHHL (H'6).
24	IC008	—	R/W	Clears masking of the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = LHHH (H'7).
23	IC007	—	R/W	Clears masking of the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = HLLL (H'8).
22	IC006	—	R/W	Clears masking of the interrupt source of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ = HLLH (H'9).

[When read]
Undefined values are read.

[When written]
0: No effect
1: Clears the corresponding interrupt mask (enables the interrupt)

Bit	Bit Name	Initial Value	R/W	Description
21	IC005	—	R/W	Clears masking of the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HLHL (H'A).
20	IC004	—	R/W	Clears masking of the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HLHH (H'B).
19	IC003	—	R/W	Clears masking of the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HHLL (H'C).
18	IC002	—	R/W	Clears masking of the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HHLH (H'D).
17	IC001	—	R/W	Clears masking of the interrupt source of $\overline{IRL3}$ to $\overline{IRL0}$ = HHHL (H'E).
16	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
15	IC115	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LLLL (H'0).
14	IC114	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LLLH (H'1).
13	IC113	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LLHL (H'2).
12	IC112	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LLHH (H'3).
11	IC111	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LHLL (H'4).
10	IC110	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LHLH (H'5).
9	IC109	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LHHL (H'6).
8	IC108	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = LHHH (H'7).
7	IC107	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HLLL (H'8).

[When read]
Undefined values are read.

[When written]
0: No effect
1: Clears the corresponding interrupt mask (enables the interrupt)

Bit	Bit Name	Initial Value	R/W	Description	
6	IC106	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HLLH (H'9).	[When read] Undefined values are read. [When written] 0: No effect 1: Clears the corresponding interrupt mask (enables the interrupt)
5	IC105	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HLHL (H'A).	
4	IC104	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HLHH (H'B).	
3	IC103	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HHLL (H'C).	
2	IC102	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HHLH (H'D).	
1	IC101	—	R/W	Clears masking of the interrupt source of $\overline{IRL7}$ to $\overline{IRL4}$ = HHHL (H'E).	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.	

(11) NMI Flag Control Register (NMIFCR)

NMIFCR is a 32-bit readable and conditionally writable register that has an NMI flag (NMIFL bit). The NMIFL bit is automatically set to 1 when an NMI interrupt is detected by the INTC. Writing 0 to the NMIFL bit clears it.

The value of the NMIFL bit does not affect acceptance of the NMI by the CPU. Although an NMI request detected by the INTC is cleared when the CPU accepts the NMI, the NMIFL bit is not cleared automatically. Even if 0 is written to the NMIFL bit before the NMI request is accepted by the CPU, the NMI request is not canceled.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31	NMIL	x	R	NMI Input Level Indicates the level of the signal input to the NMI pin; that is, this bit is read to determine the level on the NMI pin. This bit cannot be modified. 0: The low level is being input to the NMI pin 1: The high level is being input to the NMI pin
30 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	NMIFL	0	R/(W)	NMI Flag (NMI Interrupt Request Detection) Indicates whether an NMI interrupt request signal has been detected. This bit is automatically set to 1 when the INTC detects an NMI interrupt request. Write 0 to clear the bit. Writing 1 to this bit has no effect. [When read] 1: NMI has been detected 0: NMI has not been detected [When written] 0: Clears the NMI flag 1: No effect
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(12) NMI Set control Register (NMISET)

By setting the NMI set bit (NMISET1) to 1, INTC outputs the NMI request to CPU1. When CPU acknowledges NMISET1, it is automatically cleared to 0.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NMI SET1	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	NMISET1	0	R/W	NMI set 1 0: INTC does not output NMI request to CPU1. 1: INTC outputs NMI request to CPU1.
0	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.

10.3.2 User Mode Interrupt Disable Function

(1) User Interrupt Mask Level Setting Register (USERIMASK)

USERIMASK is a 32-bit readable and conditionally writable register that sets the acceptable interrupt level. This register is allocated to the 4-Kbyte page that the other registers in the INTC are not allocated. Therefore, only this register can be set to be accessible in user mode by changing the address to area 7 address through the MMU.

The interrupts that the level is lower than the level set in the UIMASK bits are masked. When H'F is set in the UIMASK bit, all interrupts other than the NMI are masked.

The interrupts that the level is higher than the level set in the UIMASK bits are accepted under the following conditions. The corresponding interrupt mask bit in the interrupt mask register is cleared to 0 (the interrupt is enabled). The IMASK bit in SR is set lower than its interrupt level.

The value of the UIMASK bit does not change even if an interrupt is accepted.

USERIMASK is initialized to H'0000 0000 (all interrupts are enabled) by a power-on reset or manual reset.

To prevent incorrect writing, this register should not be written to unless bits 31 to 24 are set to H'A5.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Code for writing (H'A5)								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Code for writing)	H'00	R/W	Reserved Code for writing (H'A5) These bits are always read as 0. Set these bits to H'A5 when writing to the UIMASK bits (Write to the UIMASK bits with these bits set to H'A5).
23 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 4	UIMASK	H'0	R/W	Interrupt Mask Level The interrupts whose level is equal to or lower than the value set in the UIMASK bits are masked.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(2) Procedure for Using the User Interrupt Mask Level Register

By setting the interrupt mask level in USERIMASK, the interrupts whose level is equal to or lower than the value set in USERIMASK are disabled. This function is used to disable less urgent interrupts when more urgent processing is performed by the tasks such as device drivers operating in user mode to reduce the processing time.

USERIMASK is allocated in a different 64-Kbyte space apart from the one where other INTC registers are allocated. Access to this register in user mode involves address translation by the MMU. In a multitasking OS, the memory-protection functions of the MMU should be used to control the processes that can access USERIMASK. Clear USERIMASK to 0 before completing a task or switching to another task. If a task is completed with the UIMASK bits set to a value other than 0, the interrupts whose level is equal to or lower than UIMASK remain disabled. This can lead to problems, for example, the OS may not be able to switch between tasks.

An example procedure for using USERIMASK is described below.

1. Classify interrupts into A and B, as described below. Then, set the interrupt level of A-type interrupts higher than that of the B-priority interrupts.
   A. Interrupts to be accepted by device drivers (interrupts used in the OS, such as a timer interrupt)
   B. Interrupts that should not be accepted by device drivers
2. Set the MMU so that the access to the address space containing USERIMASK is only allowed for the device drivers that need to disable the interrupts.
3. Branch to the device driver.
4. In the device driver operating in user mode, set the UIMASK bits to mask the B-type interrupts.
5. Process more urgent interrupts in the device driver.
6. Clear the UIMASK bit to 0 and return from the processing by the device driver.

10.3.3 On-chip Module Interrupt Priority Registers

(1) Interrupt Priority Registers (INT2PRI0 to INT2PRI24)

INT2PRI0 to INT2PRI24 are 32-bit readable/writable registers that set priority levels (31 to 0) of the on-chip peripheral module interrupts. These registers are initialized to H'0000 0000 by a reset.

These registers can set the priority of each interrupt source in 30 levels (H'00 and H'01 mask the interrupt request) by the 5-bit field.

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

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

Table 10.5 shows the correspondence between interrupt request sources and bits in INT2PRI0 to INT2PRI24.

Table 10.5 Interrupt Request Sources and INT2PRI0 to INT2PRI24

Register	Bits			
	28 to 24	20 to 16	12 to 8	4 to 0
INT2PRI0	Reserved*	Reserved*	Reserved*	WDT (H'3E0)
INT2PRI1	TMU_ch0 to 2 (H'400)	TMU_ch0 to 2 (H'420)	TMU_ch0 to 2 (H'440)	TMU_ch0 to 2 (H'460)
INT2PRI2	TMU_ch3 to 5 (H'480)	TMU_ch3 to 5 (H'4A0)	TMU_ch3 to 5 (H'4C0)	(H'4E0)
INT2PRI3	DMAC(0) (H'500)	DMAC(0) (H'520)	DMAC(0) (H'540)	DMAC(0) (H'560)
INT2PRI4	DMAC(0) (H'580)	DMAC(0) (H'5A0)	DMAC(0) (H'5C0)	H-UDI1 (H'5E0)
INT2PRI5	H-UDI0 (H'600)	DMAC(1) (H'620)	DMAC(1) (H'640)	DMAC(1) (H'660)
INT2PRI6	DMAC(1) (H'680)	HPB (H'6A0)	HPB (H'6C0)	HPB (H'6E0)
INT2PRI7	SCIF_ch0 (H'700)	SCIF_ch0 (H'720)	SCIF_ch0 (H'740)	SCIF_ch0 (H'760)
INT2PRI8	SCIF_ch1 (H'780)	TMU_ch6 to 8 (H'7A0)	TMU_ch9 to 11 (H'7C0)	(H'7E0)
INT2PRI9	Reserved*	Reserved*	SCIF_ch2 (H'840)	SCIF_ch3 (H'860)
INT2PRI10	SCIF_ch4 (H'880)	SCIF_ch5 (H'8A0)	Ether (H'8C0)	Ether (H'8E0)
INT2PRI11	Reserved*	Reserved*	Reserved*	Reserved*
INT2PRI12	Reserved*	Reserved*	Reserved*	Reserved*
INT2PRI13	Reserved*	Reserved*	Reserved*	Reserved*
INT2PRI14	Reserved*	Reserved*	Reserved*	PCIEC0 (H'AE0)
INT2PRI15	PCIEC0 (H'B00)	PCIEC0 (H'B20)	PCIEC1 (H'B40)	PCIEC1 (H'B60)
INT2PRI16	PCIEC1 (H'B80)	USB (H'BA0)	SDI0 (H'BC0)	SDI1 (H'BE0)
INT2PRI17	(H'C00)	(H'C20)	(H'C40)	(H'C60)
INT2PRI18	(H'C80)	(H'CA0)	I2C_ch0 (H'CC0)	I2C_ch1 (H'CE0)
INT2PRI19	DU (H'D00)	SSI_ch0 (H'D20)	SSI_ch1 (H'D40)	SSI_ch2 (H'D60)
INT2PRI20	SSI_ch3 (H'D80)	PCIEC2 (H'DA0)	PCIEC2 (H'DC0)	PCIEC2 (H'DE0)
INT2PRI21	HAC-ch0 (H'E00)	HAC-ch1 (H'E20)	FLCTL (H'E40)	(H'E60)
INT2PRI22	HSPI (H'E80)	GPIO0 (H'EA0)	GPIO1 (H'EC0)	Thermal sensor (H'EE0)
INT2PRI23	Reserved*	Reserved*	Reserved*	Reserved*
INT2PRI24	Reserved*	Reserved*	Reserved*	Reserved*

(Each number indicates a value of EventCode)

Note: The larger the value is, the higher the priority is. If the value is set to H'00 or H'01, the request is masked. For details, see table 10.1, Interrupt Sources.

* Reserved bits are always read as 0. The write value should always be 0.

(2) Interrupt Source Register (Not affected by Mask Setting) (CnINT2A0_0 to 3, n=0 to 1)

CnINT2A0_0 to CnINT2A0_3 are 32-bit read-only registers that indicate the interrupt sources of on-chip peripheral modules by EventCode units. Even if an interrupt is masked by the interrupt mask register, the corresponding bit in CnINT2A0 is set (further interrupt operation is not performed for the corresponding bit). Use CnINT2A1 instead if the bits for the interrupt sources masked by the interrupt mask registers should not be set.

CnINT2A0_0

Bit	Source (EventCode)	Initial Value	R/W	Description
31 to 1	Reserved	—	R	These bits are always read as 0.
0	WDT (H'3E0)	—	R	[When read] 0: No interrupt 1: An interrupt occurred [When written] 0/1: Invalid

CnINT2A0_1

Bit	Source (EventCode)	Initial Value	R/W	Description
31	TMU-ch0 to 2 (H'400)	—	R	[When read] 0: No interrupt 1: An interrupt occurred [When written] 0/1: Invalid
30	TMU-ch0 to 2 (H'420)	—	R	
29	TMU-ch0 to 2 (H'440)	—	R	
28	TMU-ch0 to 2 (H'460)	—	R	
27	TMU-ch3 to 5 (H'480)	—	R	
26	TMU-ch3 to 5 (H'4A0)	—	R	
25	TMU-ch3 to 5 (H'4C0)	—	R	
24	(H'4E0)	—	R	
23	DMAC0 (H'500)	—	R	
22	DMAC0 (H'520)	—	R	
21	DMAC0 (H'540)	—	R	
20	DMAC0 (H'560)	—	R	
19	DMAC0 (H'580)	—	R	
18	DMAC0 (H'5A0)	—	R	
17	DMAC0 (H'5C0)	—	R	
16	H-UDI1 (H'5E0)	—	R	
15	H-UDI0 (H'600)	—	R	
14	DMAC1 (H'620)	—	R	
13	DMAC1 (H'640)	—	R	
12	DMAC1 (H'660)	—	R	
11	DMAC1 (H'680)	—	R	
10	HPB (H'6A0)	—	R	
9	HPB (H'6C0)	—	R	
8	HPB (H'6E0)	—	R	
7	SCIF0 (H'700)	—	R	
6	SCIF0 (H'720)	—	R	
5	SCIF0 (H'740)	—	R	
4	SCIF0 (H'760)	—	R	
3	SCIF1 (H'780)	—	R	
2	TMU-ch6 to 8 (H'7A0)	—	R	
1	TMU-ch9 to 11 (H'7C0)	—	R	
0	(H'7E0)	—	R	

CnINT2A0_2

Bit	Source (EventCode)	Initial Value	R/W	Description
31, 30	Reserved	—	R	These bits are always read as 0.
29	SCIF2 (H'840)	—	R	[When read] 0: No interrupt 1: An interrupt occurred [When written] 0/1: Invalid
28	SCIF3 (H'860)	—	R	
27	SCIF4 (H'880)	—	R	
26	SCIF5 (H'8A0)	—	R	
25	Ether (H'8C0)	—	R	
24	Ether (H'8E0)	—	R	
23 to 9	Reserved	—	R	These bits are always read as 0.
8	PCIEC0 (H'AE0)	—	R	[When read] 0: No interrupt 1: An interrupt occurred [When written] 0/1: Invalid
7	PCIEC0 (H'B00)	—	R	
6	PCIEC0 (H'B20)	—	R	
5	PCIEC1 (H'B40)	—	R	
4	PCIEC1 (H'B60)	—	R	
3	PCIEC1 (H'B80)	—	R	
2	USB (H'BA0)	—	R	
1	SDI0 (H'BC0)	—	R	
0	SDI1 (H'BE0)	—	R	

CnINT2A0_3

Bit	Source (EventCode)	Initial Value	R/W	Description
31	(H'C00)	—	R	[When read] 0: No interrupt 1: An interrupt occurred [When written] 0/1: Invalid
30	(H'C20)	—	R	
29	(H'C40)	—	R	
28	(H'C60)	—	R	
27	(H'C80)	—	R	
26	(H'CA0)	—	R	
25	I2C0 (H'CC0)	—	R	
24	I2C1 (H'CE0)	—	R	
23	DU (H'D00)	—	R	
22	SSI0 (H'D20)	—	R	
21	SSI1 (H'D40)	—	R	
20	SSI2 (H'D60)	—	R	
19	SSI3 (H'D80)	—	R	
18	PCIEC2 (H'DA0)	—	R	
17	PCIEC2 (H'DC0)	—	R	
16	PCIEC2 (H'DE0)	—	R	
15	HAC0 (H'E00)	—	R	
14	HAC1 (H'E20)	—	R	
13	FLCTL (H'E40)	—	R	
12	(H'E60)	—	R	
11	HSPI (H'E80)	—	R	
10	GPIO0 (H'EA0)	—	R	
9	GPIO1 (H'EC0)	—	R	
8	Thermal sensor (H'EE0)	—	R	
7 to 0	Reserved	—	R	These bits are always read as 0.

(3) Interrupt Source Register (Affected by Mask States) (CnINT2A1_0 to 3, n=0 to 1)

CnINT2A1_0 to CnINT2A1_3 are 32-bit read-only registers that indicate the interrupt sources of on-chip peripheral modules. If an interrupt is masked by the interrupt mask register or interrupt acknowledgement mask register, the corresponding bit in CnINT2A1 is not set to 1. Use CnINT2A0 to check whether interrupts have been generated, regardless of the state of the interrupt mask register.

CnINT2A1_0

Bit	Source (EventCode)	Initial Value	R/W	Description
31 to 1	Reserved	—	R	These bits are always read as 0.
0	WDT (H'3E0)	—	R	[When read] 0: No interrupt 1: An interrupt occurred [When written] 0/1: Invalid

CnINT2A1_1

Bit	Source (EventCode)	Initial Value	R/W	Description
31	TMU-ch0 to 2 (H'400)	—	R	[When read] 0: No interrupt 1: An interrupt occurred [When written] 0/1: Invalid
30	TMU-ch0 to 2 (H'420)	—	R	
29	TMU-ch0 to 2 (H'440)	—	R	
28	TMU-ch0 to 2 (H'460)	—	R	
27	TMU-ch3 to 5 (H'480)	—	R	
26	TMU-ch3 to 5 (H'4A0)	—	R	
25	TMU-ch3 to 5 (H'4C0)	—	R	
24	(H'4E0)	—	R	
23	DMAC0 (H'500)	—	R	
22	DMAC0 (H'520)	—	R	
21	DMAC0 (H'540)	—	R	
20	DMAC0 (H'560)	—	R	
19	DMAC0 (H'580)	—	R	
18	DMAC0 (H'5A0)	—	R	
17	DMAC0 (H'5C0)	—	R	
16	H-UDI1 (H'5E0)	—	R	
15	H-UDI0 (H'600)	—	R	
14	DMAC1 (H'620)	—	R	
13	DMAC1 (H'640)	—	R	
12	DMAC1 (H'660)	—	R	
11	DMAC1 (H'680)	—	R	
10	HPB (H'6A0)	—	R	
9	HPB (H'6C0)	—	R	
8	HPB (H'6E0)	—	R	
7	SCIF0 (H'700)	—	R	
6	SCIF0 (H'720)	—	R	
5	SCIF0 (H'740)	—	R	
4	SCIF0 (H'760)	—	R	
3	SCIF1 (H'780)	—	R	
2	TMU-ch6 to 8 (H'7A0)	—	R	
1	TMU-ch9 to 11 (H'7C0)	—	R	
0	(H'7E0)	—	R	

CnINT2A1_2

Bit	Source (EventCode)	Initial Value	R/W	Description
31, 30	Reserved	—	R	These bits are always read as 0.
29	SCIF2 (H'840)	—	R	[When read] 0: No interrupt 1: An interrupt occurred [When written] 0/1: Invalid
28	SCIF3 (H'860)	—	R	
27	SCIF4 (H'880)	—	R	
26	SCIF5 (H'8A0)	—	R	
25	Ether (H'8C0)	—	R	
24	Ether (H'8E0)	—	R	
23 to 9	Reserved	—	R	These bits are always read as 0.
8	PCIEC0 (H'AE0)	—	R	[When read] 0: No interrupt 1: An interrupt occurred [When written] 0/1: Invalid
7	PCIEC0 (H'B00)	—	R	
6	PCIEC0 (H'B20)	—	R	
5	PCIEC1 (H'B40)	—	R	
4	PCIEC1 (H'B60)	—	R	
3	PCIEC1 (H'B80)	—	R	
2	USB (H'BA0)	—	R	
1	SDI0 (H'BC0)	—	R	
0	SDI1 (H'BE0)	—	R	

CnINT2A1_3

Bit	Source (EventCode)	Initial Value	R/W	Description
31	(H'C00)	—	R	[When read] 0: No interrupt 1: An interrupt occurred [When written] 0/1: Invalid
30	(H'C20)	—	R	
29	(H'C40)	—	R	
28	(H'C60)	—	R	
27	(H'C80)	—	R	
26	(H'CA0)	—	R	
25	I2C0 (H'CC0)	—	R	
24	I2C1 (H'CE0)	—	R	
23	DU (H'D00)	—	R	
22	SSI0 (H'D20)	—	R	
21	SSI1 (H'D40)	—	R	
20	SSI2 (H'D60)	—	R	
19	SSI3 (H'D80)	—	R	
18	PCIEC2 (H'DA0)	—	R	
17	PCIEC2 (H'DC0)	—	R	
16	PCIEC2 (H'DE0)	—	R	
15	HAC0 (H'E00)	—	R	
14	HAC1 (H'E20)	—	R	
13	FLCTL (H'E40)	—	R	
12	(H'E60)	—	R	
11	HSPI (H'E80)	—	R	
10	GPIO0 (H'EA0)	—	R	
9	GPIO1 (H'EC0)	—	R	
8	Thermal sensor (H'EE0)	—	R	
7 to 0	Reserved	—	R	These bits are always read as 0.

If the interrupt masking is set by CnINT2MSKR or the interrupt masking by CnINT2MSKR is cleared by CnINT2MSKCLR, the reflection time required for CnINT2A1 is guaranteed by hardware. Therefore, after the interrupt mask is set or cleared, the contents that reflect the setting of CnINT2MSKR can be read.

(4) Interrupt Mask Register 0 to 3 (CnINT2MSKR0 to 3, n=0 to 1)

CnINT2MSKR0 to CnINT2MSKR3 is a 32-bit readable/writable registers that can mask interrupts for sources indicated in the interrupt source register. When a bit in this register is set to 1, the interrupt in the corresponding bit is not notified. CnINT2MSKR is initialized to H'FFFF FFFF (all masked) by a reset.

After this register is written to or the masking is cleared by writing to CnINT2MSKCLR0 to 3, the timing required to reflect the register value is guaranteed by reading from this register once.

CnINT2MASK0

Bit	Source (EventCode)	Initial Value	R/W	Description
31 to 1	Reserved	All 1	R/W	These bits are always read as 0. The write value should always be 0.
0	WDT (H'3E0)	1	R/W	[When read] 0: No mask 1: Mask [When written] 0: Invalid 1: Mask

CnINT2MASK1

Bit	Source (EventCode)	Initial Value	R/W	Description
31	TMU-ch0 to 2 (H'400)	1	R/W	[When read] 0: No mask 1: Mask [When written] 0: Invalid 1: Mask
30	TMU-ch0 to 2 (H'420)	1	R/W	
29	TMU-ch0 to 2 (H'440)	1	R/W	
28	TMU-ch0 to 2 (H'460)	1	R/W	
27	TMU-ch3 to 5 (H'480)	1	R/W	
26	TMU-ch3 to 5 (H'4A0)	1	R/W	
25	TMU-ch3 to 5 (H'4C0)	1	R/W	
24	(H'4E0)	1	R/W	
23	DMAC0 (H'500)	1	R/W	
22	DMAC0 (H'520)	1	R/W	
21	DMAC0 (H'540)	1	R/W	
20	DMAC0 (H'560)	1	R/W	
19	DMAC0 (H'580)	1	R/W	
18	DMAC0 (H'5A0)	1	R/W	
17	DMAC0 (H'5C0)	1	R/W	
16	H-UDI1 (H'5E0)	1	R/W	
15	H-UDI0 (H'600)	1	R/W	
14	DMAC1 (H'620)	1	R/W	
13	DMAC1 (H'640)	1	R/W	
12	DMAC1 (H'660)	1	R/W	
11	DMAC1 (H'680)	1	R/W	
10	HPB (H'6A0)	1	R/W	
9	HPB (H'6C0)	1	R/W	
8	HPB (H'6E0)	1	R/W	
7	SCIF0 (H'700)	1	R/W	
6	SCIF0 (H'720)	1	R/W	
5	SCIF0 (H'740)	1	R/W	
4	SCIF0 (H'760)	1	R/W	
3	SCIF1 (H'780)	1	R/W	
2	TMU-ch6 to 8 (H'7A0)	1	R/W	
1	TMU-ch9 to 11 (H'7C0)	1	R/W	
0	(H'7E0)	1	R/W	

CnINT2MASK2

Bit	Source (EventCode)	Initial Value	R/W	Description
31, 30	Reserved	All 1	R/W	These bits are always read as 0. The write value should always be 0.
29	SCIF2 (H'840)	1	R/W	[When read] 0: No mask 1: Mask [When written] 0: Invalid 1: Mask
28	SCIF3 (H'860)	1	R/W	
27	SCIF4 (H'880)	1	R/W	
26	SCIF5 (H'8A0)	1	R/W	
25	Ether (H'8C0)	1	R/W	
24	Ether (H'8E0)	1	R/W	
23 to 9	Reserved	All 1	R/W	These bits are always read as 0. The write value should always be 0.
8	PCIEC0 (H'AE0)	1	R/W	[When read] 0: No mask 1: Mask [When written] 0: Invalid 1: Mask
7	PCIEC0 (H'B00)	1	R/W	
6	PCIEC0 (H'B20)	1	R/W	
5	PCIEC1 (H'B40)	1	R/W	
4	PCIEC1 (H'B60)	1	R/W	
3	PCIEC1 (H'B80)	1	R/W	
2	USB (H'BA0)	1	R/W	
1	SDI0 (H'BC0)	1	R/W	
0	SDI1 (H'BE0)	1	R/W	

CnINT2MASK3

Bit	Source (EventCode)	Initial Value	R/W	Description
31	(H'C00)	1	R/W	[When read] 0: No mask 1: Mask [When written] 0: Invalid 1: Mask
30	(H'C20)	1	R/W	
29	(H'C40)	1	R/W	
28	(H'C60)	1	R/W	
27	(H'C80)	1	R/W	
26	(H'CA0)	1	R/W	
25	I2C0 (H'CC0)	1	R/W	
24	I2C1 (H'CE0)	1	R/W	
23	DU (H'D00)	1	R/W	
22	SSI0 (H'D20)	1	R/W	
21	SSI1 (H'D40)	1	R/W	
20	SSI2 (H'D60)	1	R/W	
19	SSI3 (H'D80)	1	R/W	
18	PCIEC2 (H'DA0)	1	R/W	
17	PCIEC2 (H'DC0)	1	R/W	
16	PCIEC2 (H'DE0)	1	R/W	
15	HAC0 (H'E00)	1	R/W	
14	HAC1 (H'E20)	1	R/W	
13	FLCTL (H'E40)	1	R/W	
12	(H'E60)	1	R/W	
11	HSPI (H'E80)	1	R/W	
10	GPIO0 (H'EA0)	1	R/W	
9	GPIO1 (H'EC0)	1	R/W	
8	Thermal sensor (H'EE0)	1	R/W	
7 to 0	Reserved	All 1	R/W	These bits are always read as 0. The write value should always be 0.

Note: Mask setting or clearing for the detail source of each module is impossible at the INTC. To mask the detail source, set the mask bit of each module register.

(5) Interrupt Mask Clear Register 0 to 3 (CnINT2MSKCR0 to 3, n=0 to 1)

CnINT2MSKCR is a 32-bit write-only register that clears the masking set in the interrupt mask register. When the corresponding bit in this register is set to 1, the interrupt source masking is cleared. These bits are always read as 0.

CnINT2MSKCLR0

Bit	Source (EventCode)	Initial Value	R/W	Description
31 to 1	Reserved	All 0	W	The value of reserved bits is undefined. The write value should always be 0.
0	WDT (H'3E0)	0	W	[When read] 0/1: Don't care [When written] 0: Invalid 1: Clear mask

CnINT2MSKCLR1

Bit	Source (EventCode)	Initial Value	R/W	Description
31	TMU-ch0 to 2 (H'400)	0	W	[When read]
30	TMU-ch0 to 2 (H'420)	0	W	0/1: Don't care
29	TMU-ch0 to 2 (H'440)	0	W	[When written]
28	TMU-ch0 to 2 (H'460)	0	W	0: Invalid
27	TMU-ch3 to 5 (H'480)	0	W	1: Clear mask
26	TMU-ch3 to 5 (H'4A0)	0	W	
25	TMU-ch3 to 5 (H'4C0)	0	W	
24	(H'4E0)	0	W	
23	DMAC0 (H'500)	0	W	
22	DMAC0 (H'520)	0	W	
21	DMAC0 (H'540)	0	W	
20	DMAC0 (H'560)	0	W	
19	DMAC0 (H'580)	0	W	
18	DMAC0 (H'5A0)	0	W	
17	DMAC0 (H'5C0)	0	W	
16	H-UDI1 (H'5E0)	0	W	
15	H-UDI0 (H'600)	0	W	
14	DMAC1 (H'620)	0	W	
13	DMAC1 (H'640)	0	W	
12	DMAC1 (H'660)	0	W	
11	DMAC1 (H'680)	0	W	
10	HPB (H'6A0)	0	W	
9	HPB (H'6C0)	0	W	
8	HPB (H'6E0)	0	W	
7	SCIF0 (H'700)	0	W	
6	SCIF0 (H'720)	0	W	
5	SCIF0 (H'740)	0	W	
4	SCIF0 (H'760)	0	W	
3	SCIF1 (H'780)	0	W	
2	TMU-ch6 to 8 (H'7A0)	0	W	
1	TMU-ch9 to 11 (H'7C0)	0	W	
0	(H'7E0)	0	W	

CnINT2MSKCLR2

Bit	Source (EventCode)	Initial Value	R/W	Description
31, 30	Reserved	All 0	W	The value of reserved bits is undefined. The write value should always be 0.
29	SCIF2 (H'840)	0	W	[When read] 0/1: Don't care [When written] 0: Invalid 1: Clear mask
28	SCIF3 (H'860)	0	W	
27	SCIF4 (H'880)	0	W	
26	SCIF5 (H'8A0)	0	W	
25	Ether (H'8C0)	0	W	
24	Ether (H'8E0)	0	W	
23 to 9	Reserved	All 0	W	The value of reserved bits is undefined. The write value should always be 0.
8	PCIEC0 (H'AE0)	0	W	[When read] 0/1: Don't care [When written] 0: Invalid 1: Clear mask
7	PCIEC0 (H'B00)	0	W	
6	PCIEC0 (H'B20)	0	W	
5	PCIEC1 (H'B40)	0	W	
4	PCIEC1 (H'B60)	0	W	
3	PCIEC1 (H'B80)	0	W	
2	USB (H'BA0)	0	W	
1	SDI0 (H'BC0)	0	W	
0	SDI1 (H'BE0)	0	W	

CnINT2MSKCLR3

Bit	Source (EventCode)	Initial Value	R/W	Description
31	(H'C00)	0	W	[When read] 0/1: Don't care [When written] 0: Invalid 1: Clear mask
30	(H'C20)	0	W	
29	(H'C40)	0	W	
28	(H'C60)	0	W	
27	(H'C80)	0	W	
26	(H'CA0)	0	W	
25	I2C0 (H'CC0)	0	W	
24	I2C1 (H'CE0)	0	W	
23	DU (H'D00)	0	W	
22	SSI0 (H'D20)	0	W	
21	SSI1 (H'D40)	0	W	
20	SSI2 (H'D60)	0	W	
19	SSI3 (H'D80)	0	W	
18	PCIEC2 (H'DA0)	0	W	
17	PCIEC2 (H'DC0)	0	W	
16	PCIEC2 (H'DE0)	0	W	
15	HAC0 (H'E00)	0	W	
14	HAC1 (H'E20)	0	W	
13	FLCTL (H'E40)	0	W	
12	(H'E60)	0	W	
11	HSPI (H'E80)	0	W	
10	GPIO0 (H'EA0)	0	W	
9	GPIO1 (H'EC0)	0	W	
8	Thermal sensor (H'EE0)	0	W	
7 to 0	Reserved	All 0	W	The value of reserved bits is undefined. The write value should always be 0.

Note: Mask setting or clearing for the detail source of each module is impossible at the INTC. To mask the detail source, set the mask bit of each module register.

(6) Detail Interrupt Source Indicate Register (INT2B00 to 44)

INT2B00 to INT2B44 are registers that indicate details of each interrupt source module indicated at peripheral interrupt source register. These registers are not affected by the mask state of interrupt mask register, so any bit in them can be read out (when an interrupt output mask is set in the source module, a detail of the interrupt is not indicated to the these registers). When using the mask of each detail source individually, set the interrupt mask register or interrupt enable register which the corresponding module has is needed. About the meaning of the detail source in each register, show the specification of the peripheral module that generated the interrupt.

0) INT2B00 register: WDT INTEVT=H'3E0

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

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

1) INT2B01 register: TMU-ch0 to 2 INTEVT=H'400,420,440,460

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TICPI2	TUNI2	TUNI1	TUNI0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

2) INT2B02 register: TMU-ch3 to 5 INTEVT=H'480,4A0,4C0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														TUNI5	TUNI4	TUNI3
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

3) INT2B03 register: SuperHyway DMAC0 INTEVT=H'500,520,540,560,580,5A0,5C0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										DMA0AE	DMA0INT5	DMA0INT4	DMA0INT3	DMA0INT2	DMA0INT1	DMA0INT0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

4) INT2B04 register: HUDI0,1 INTEVT=H'5E0,600

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															H-UDI1	H-UDI0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

5) INT2B05 register: SuperHyway DMAC1 INTEVT=H'620,640,660,680

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

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

6) INT2B06 register: HPB-DMAC0 to 13 INTEVT=H'6A0,6C0,6E0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			DTE13	DTE12	DTE11	DTE10	DTE9	DTE8	DTE7	DTE6	DTE5	DTE4	DTE3	DTE2	DTE1	DTE0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

7) INT2B07 register: SCIF0 INTEVT=H'700,720,740,760

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TXI0	BRI0	RXI0	ERI0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

8) INT2B08 register: SCIF1 INTEVT=H'780

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TXI1	BRI1	RXI1	ERI1
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

9) INT2B09 register: TMU-ch6 to 8 INTEVT=H'7A0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														TUNI8	TUNI7	TUNI6
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

10) INT2B10 register: TMU-ch9 to 11 INTEVT=H'7C0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														TUNI11	TUNI10	TUNI9
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

11) INT2B11 register: Reserve

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

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

12) INT2B12 register: SCIF2 INTEVT=H'840

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TXI2	BRI2	RXI2	ERI2
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

13) INT2B13 register: SCIF3 INTEVT=H'860

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TXI3	BRI3	RXI3	ERI3
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

14) INT2B14 register: SCIF4 INTEVT=H'880

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TXI4	BRI4	RXI4	ERI4
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15) INT2B15 register: SCIF5 INTEVT=H'8A0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TXI5	BRI5	RXI5	ERI5
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

16) INT2B16 register: Ether INTEVT=H'8C0

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

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

17) INT2B17 register: Ether_DMAC INTEVT=H'8E0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		TWB			LKON	TABT	RABT	RFRMER	BER	MINT	FTC	TDE	TFE	FRC	RDE	RFE
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TINT8	TINT7	TINT6	TINT5	TINT4	TINT3	TINT2	TINT1	RINT8	RINT7	RINT6	RINT5	RINT4	RINT3	RINT2	RINT1
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

18) INT2B18 register: PCIEC0-3 INTEVT=H'AE0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PCI0ASTINTD	PCI0ASTINTC	PCI0ASTINTB	PCI0ASTINTA	PCI0MSI31	PCI0MSI30
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCI0MSI29	PCI0MSI28	PCI0MSI27	PCI0MSI26	PCI0MSI25	PCI0MSI24	PCI0MSI23	PCI0MSI22	PCI0MSI21	PCI0MSI20	PCI0MSI19	PCI0MSI18	PCI0MSI17	PCI0MSI16	PCI0MSI15	PCI0MSI14
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

19) INT2B19 register: PCIEC0-2 INTEVT=H'B00

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PCI0MSI13	PCI0MSI12	PCI0MSI11	PCI0MSI10	PCI0MSI9	PCI0MSI8
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCI0MSI7	PCI0MSI6	PCI0MSI5	PCI0MSI4	PCI0MSI3	PCI0MSI2	PCI0MSI1	PCI0MSI0	PCI0DMAC3	PCI0DMAC2	PCI0DMAC1	PCI0DMAC0	PCI0DMACE	PCI0RXERP	PCI0RXCTRL	PCI0TXCTRL
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

20) INT2B20 register: PCIEC0-1 INTEVT=H'B20

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									PCI0RXVCX2	PCI0RXVCX1	PCI0RXVCX0	PCI0TXVCX2	PCI0TXVCX1	PCI0TXVCX0	PCI0RXVCXERR	PCI0RXVC0ERR
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCI0INTTL	PCI0INTDL	PCI0INTMAC	PCI0INTPM	PCI0INTRXVC02	PCI0INTRXVC01	PCI0INTRXVC00	PCI0INTTXVC02	PCI0INTTXVC01	PCI0INTTXVC00	PCI0INTPCIMES	PCI0INTPCIPOWER	PCI0INTPCICERR	PCI0INTPCINFERR	PCI0INTPCIFERR	PCI0INTPCISERR
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

21) INT2B21 register: PCIEC1-3 INTEVT=H'B40

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PCI1ASTINTD	PCI1ASTINTC	PCI1ASTINTB	PCI1ASTINTA	PCI1MSI31	PCI1MSI30
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCI1MSI29	PCI1MSI28	PCI1MSI27	PCI1MSI26	PCI1MSI25	PCI1MSI24	PCI1MSI23	PCI1MSI22	PCI1MSI21	PCI1MSI20	PCI1MSI19	PCI1MSI18	PCI1MSI17	PCI1MSI16	PCI1MSI15	PCI1MSI14
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

22) INT2B22 register: PCIEC1-2 INTEVT=H'B60

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PCI1MSI13	PCI1MSI12	PCI1MSI11	PCI1MSI10	PCI1MSI9	PCI1MSI8
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCI1MSI7	PCI1MSI6	PCI1MSI5	PCI1MSI4	PCI1MSI3	PCI1MSI2	PCI1MSI1	PCI1MSI0	PCI1DMAC3	PCI1DMAC2	PCI1DMAC1	PCI1DMAC0	PCI1DMACE	PCI1RXERP	PCI1RXCTRL	PCI1TXCTRL
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

23) INT2B23 register: PCIEC1-1 INTEVT=H'B80

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									PCI1RXVCX2	PCI1RXVCX1	PCI1RXVCX0	PCI1TXVCX2	PCI1TXVCX1	PCI1TXVCX0	PCI1RXVCXERR	PCI1RXVC0ERR
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCI1INTTL	PCI1INTDL	PCI1INTMAC	PCI1INTPM	PCI1INTRXVC02	PCI1INTRXVC01	PCI1INTRXVC00	PCI1INTTXVC02	PCI1INTTXVC01	PCI1INTTXVC00	PCI1INTPCIMES	PCI1INTPCIPOWER	PCI1INTPCICERR	PCI1INTPCINFERR	PCI1INTPCIFERR	PCI1INTPCISERR
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

24) INT2B24 register: USB INTEVT=H'BA0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													USB-F1	USB-F0	USB-EHCI	USB-OHCI
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

25) INT2B25 register: SDI0 INTEVT=H'BC0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													SD3	SD2	SD1	SD0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

26) INT2B26 register: SDI1 INTEVT=H'BE0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													SD3	SD2	SD1	SD0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

27) INT2B27 register: Reserved

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

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

28) INT2B28 register: Reserved

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

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

29) INT2B29 register: I2C0 INTEVT=H'CC0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		MNR	MAL	MST	MDE	MDT	MDR	MAT				SSR	SDE	SDT	SDR	SAR
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

30) INT2B30 register: I2C1 INTEVT=H'CE0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		MNR	MAL	MST	MDE	MDT	MDR	MAT				SSR	SDE	SDT	SDR	SAR
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

31) INT2B31 register: DU INTEVT=H'D00

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

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

32) INT2B32 register: SSI0,1,2,3 INTEVT=H'D20,D40,D60,D80

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													SSI3	SSI2	SSI1	SSI0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

33) INT2B33 register: PCIEC2-3 INTEVT=H'DA0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PCI2ASTINTD	PCI2ASTINTC	PCI2ASTINTB	PCI2ASTINTA	PCI2MSI31	PCI2MSI30
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCI2MSI29	PCI2MSI28	PCI2MSI27	PCI2MSI26	PCI2MSI25	PCI2MSI24	PCI2MSI23	PCI2MSI22	PCI2MSI21	PCI2MSI20	PCI2MSI19	PCI2MSI18	PCI2MSI17	PCI2MSI16	PCI2MSI15	PCI2MSI14
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

34) INT2B34 register: PCIEC2-2 INTEVT=H'DC0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PCI2MSI13	PCI2MSI12	PCI2MSI11	PCI2MSI10	PCI2MSI9	PCI2MSI8
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCI2MSI7	PCI2MSI6	PCI2MSI5	PCI2MSI4	PCI2MSI3	PCI2MSI2	PCI2MSI1	PCI2MSI0	PCI2DMAC3	PCI2DMAC2	PCI2DMAC1	PCI2DMAC0	PCI2DMACE	PCI2RXERP	PCI2RXCTRL	PCI2TXCTRL
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

35) INT2B35 register: PCIEC2-1 INTEVT=H'DE0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									PCI2RXVCX2	PCI2RXVCX1	PCI2RXVCX0	PCI2TXVCX2	PCI2TXVCX1	PCI2TXVCX0	PCI2RXVCXERR	PCI2RXVC0ERR
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCI2INTTL	PCI2INTDL	PCI2INTMAC	PCI2INTPM	PCI2INTRXVC02	PCI2INTRXVC01	PCI2INTRXVC00	PCI2INTTXVC02	PCI2INTTXVC01	PCI2INTTXVC00	PCI2INTPCIMES	PCI2INTPCIPOWER	PCI2INTPCICERR	PCI2INTPCINFERR	PCI2INTPCIFERR	PCI2INTPCISERR
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

36) INT2B36 register: HAC0, 1 INTEVT=H'E00, E20

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															HAC1	HAC0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

37) INT2B37 register: FLCTL INTEVT=H'E40

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													FLTRQ1	FLTRQ0	FLTEND	FLSTE
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

38) INT2B38 register: Reserved INTEVT=H'E60

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

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

39) INT2B39 register: HSPI INTEVT=H'E80

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

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

40) INT2B40 register: GPIO INTEVT=H'EA0,EC0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

41) INT2B41 register: Thermal Sensor INTEVT=H'EE0

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ths3	ths2	ths1	ths0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

42) INT2B42, 43, 44 register: Reserved

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

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

10.3.4 GPIO Interrupt Set Register (INT2GPIC)

INT2GPIC enables interrupt requests input from the pins J2, J1, H1, H0, F1, F0, A1 and A0, as GPIO interrupts. In this case, the mask setting by the register at the corresponding module is needed.

A GPIO interrupt is a low-active interrupt. Enable interrupt requests after setting the pins corresponding to the port control register (J, H, F and A) used for GPIO interrupts to be input pins from ports For the port control registers, see section 30, General Input/Output Port (GPIO).

The timing required to reflect the register value is guaranteed by writing to this register, and then, reading from this register once (the interrupt request is reflected).

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	GPIO 07E	GPIO 06E	GPIO 05E	GPIO 04E	GPIO 03E	GPIO 02E	GPIO 01E	GPIO 00E
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10.6 Correspondence of each bit and INT2GPIC register

Bit	Name	Initial Value	R/W	Function	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	Enables GPIO interrupt request for each pin. 0: Disable the corresponding interrupt request 1: Enable the corresponding interrupt request
7	GPIO07E	0	R/W	Enables GPIO07 interrupt request (from pin J2).	
6	GPIO06E	0	R/W	Enables GPIO06 interrupt request (from pin J1).	
5	GPIO05E	0	R/W	Enables GPIO05 interrupt request (from pin H1).	
4	GPIO04E	0	R/W	Enables GPIO04 interrupt request (from pin H0).	
3	GPIO03E	0	R/W	Enables GPIO03 interrupt request (from pin F1).	
2	GPIO02E	0	R/W	Enables GPIO02 interrupt request (from pin F0).	
1	GPIO01E	0	R/W	Enables GPIO01 interrupt request (from pin A1).	
0	GPIO00E	0	R/W	Enables GPIO00 interrupt request (from pin A0).	

When a GPIO port pin is used as an interrupt pin, the GPIO notifies the INTC of the interrupt when the GPIO detects an interrupt. The INTC indicates the interrupt as a 1-bit source in the [10,9] bit in CnINT2A0_3 or CnINT2A1_3.

10.3.5 Thermal Sensor Interrupt Set Register (INT2THSC)

INT2THSC enables interrupt requests input from the thermal sensor. In this case, the mask setting by the register at the corresponding module is needed.

The timing required to reflect the register value is guaranteed by writing to this register, and then, reading from this register once (the interrupt request is reflected).

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

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

Table 10.7 Correspondence of each bit and INT2THSC register

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	Enables THS interrupt request for each pin. 0: Disable the corresponding interrupt request 1: Enable the corresponding interrupt request
3	THS3E	0	R/W	Enables THS3 interrupt request.	
2	THS2E	0	R/W	Enables THS2 interrupt request.	
1	THS1E	0	R/W	Enables THS1 interrupt request.	
0	THS0E	0	R/W	Enables THS0 interrupt request.	

10.3.6 Registers for Among CPUs

(1) Register for Controlling Interrupt among CPUs (CnINTICI, n= 0 to 1)

CnINTICI is a 32-bit readable/writable register that set interrupt request to CPUn (n = 0 to 1). These registers are initialized to H'0000 0000 by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICI7				ICI6				ICI5				ICI4			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICI3				ICI2				ICI1				ICI0			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	ICI7	0000	R/W	Interrupt request among CPUs This generates the interrupt request to CPUn (n = 0 to 1) that handles logical OR of 4-bit in each field as the interrupt request of 1 source. 0: Ignores 0. 1: Generates an interrupt request
27 to 24	ICI6	0000	R/W	
23 to 20	ICI5	0000	R/W	
19 to 16	ICI4	0000	R/W	
15 to 12	ICI3	0000	R/W	
11 to 8	ICI2	0000	R/W	
7 to 4	ICI1	0000	R/W	
3 to 0	ICI0	0000	R/W	

(2) Register for Clearing Interrupt among CPUs (CnINTICICLR, n= 0 to 1)

CnINTICICLR is a 32-bit write-only register that clears the interrupt request to CPUn (n = 0 to 1). These bits are always read as 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICICLR7				ICICLR6				ICICLR5				ICICLR4			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICICLR3				ICICLR2				ICICLR1				ICICLR0			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	ICICLR7	0000	R/W	Clearance of interrupt request among CPUs It clears the bit corresponding to the register for clearing interrupt priority among CPUs (CnINTICI) and cancels the interrupt request. 0: Ignores 0. 1: It clears the bit corresponding to the CnINIICI register to0 and cancels the interrupt request
27 to 24	ICICLR6	0000	R/W	
23 to 20	ICICLR5	0000	R/W	
19 to 16	ICICLR4	0000	R/W	
15 to 12	ICICLR3	0000	R/W	
11 to 8	ICICLR2	0000	R/W	
7 to 4	ICICLR1	0000	R/W	
3 to 0	ICICLR0	0000	R/W	

(3) Register for Setting Interrupt Priority Order among CPUs (CnICIPRI, n= 0 to 1)

CnICIPRI is a 32-bit readable/writable register that sets the priority order of interrupt request among CPU n(n = 0 to 1). These registers are initialized to H'0000 0000 by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICIPRI7				ICIPRI6				ICIPRI5				ICIPRI4			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICIPRI3				ICIPRI2				ICIPRI1				ICIPRI0			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	ICIPRI7	0000	R/W	Priority order of interrupt among CPUs It sets the priority order of interrupt request among CPUs corresponding to each field for every 4 bit. 0: Ignores 0. 1: Sets the interrupt priority level.
27 to 24	ICIPRI6	0000	R/W	
23 to 20	ICIPRI5	0000	R/W	
19 to 16	ICIPRI4	0000	R/W	
15 to 12	ICIPRI3	0000	R/W	
11 to 8	ICIPRI2	0000	R/W	
7 to 4	ICIPRI1	0000	R/W	
3 to 0	ICIPRI0	0000	R/W	

(4) Register for Clearing Interrupt Priority Order among CPUs (CnICIPRICLR, n= 0 to 1)

CnICIPRICLR is a 32-bit write-only register that clears the bit corresponding to the register for interrupt priority order of interrupt among CPUs (CnINTICIPRI). These bits are always read as 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICIPRICLR7				ICIPRICLR6				ICIPRICLR5				ICIPRICLR4			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICIPRICLR3				ICIPRICLR2				ICIPRICLR1				ICIPRICLR0			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	ICIPRICLR7	0000	R/W	Clearance of priority order for interrupt among CPUs It clears the bit corresponding to the register for interrupt priority order of interrupt among CPUs (CnINTICIPRI) and clears the interrupt priority level. 0: Ignores 0. 1: Clears the bit corresponding to the CnINIICIPRI register to 0 and clears the interrupt priority level.
27 to 24	ICIPRICLR6	0000	R/W	
23 to 20	ICIPRICLR5	0000	R/W	
19 to 16	ICIPRICLR4	0000	R/W	
15 to 12	ICIPRICLR3	0000	R/W	
11 to 8	ICIPRICLR2	0000	R/W	
7 to 4	ICIPRICLR1	0000	R/W	
3 to 0	ICIPRICLR0	0000	R/W	

10.3.7 Interrupt Distribution Register

(1) Interrupt Distribution Register 0 (INTDISTCR0)

INTDISTCR0 is a 32-bit readable/writable register that can set whether or not to distribute the IRQ interrupt to multiple CPUs in automatic distribution mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DIST7	DIST6	DIST5	DIST4	DIST3	DIST2	DIST1	DIST0	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DIST7	0	R/W	It sets distribute each interrupt to multiple CPUs in automatic distribution mode. When the corresponding bit receives the interrupt as set to 1, the bit 0 in interrupt acknowledgement register (INTACK) is exclusively set, and mask the corresponding interrupt source to each CPU. The mask is released by ending the interrupt operation and writing the interrupt event to interrupt acknowledgement clear register (INTACKCLR).
	DIST6	0	R/W	
	DIST5	0	R/W	
	DIST4	0	R/W	
	DIST3	0	R/W	
	DIST2	0	R/W	
	DIST1	0	R/W	
	DIST0	0	R/W	
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(2) Interrupt Distribution Register 1 (INTDISTCR1)

INTDISTCR1 is a 32-bit readable/writable register that can set whether or not to distribute the IRL0 and IRL1 interrupt to multiple CPUs in automatic distribution mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DIST0	DIST1	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	DIST0	0	R/W	It sets distribute each interrupt to multiple CPUs in automatic distribution mode. When the corresponding bit receives the interrupt as set to 1, the bit 0 in interrupt acknowledgement register (INTACK) is exclusively set, and mask the corresponding interrupt source to each CPU. The mask is released by ending the interrupt operation and writing the interrupt event to interrupt acknowledgement clear register (INTACKCLR).
30	DIST1	0	R/W	
29 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(3) Peripheral Interrupt Distribution Register (INT2DISTCR0 to 3)

INT2DISTCR0 to INT2DISTCR3 is a 32-bit readable/writable register that can set whether to each peripheral interrupt to multiple CPUs in automatic distribution mode.

INT2DISTCR0

Bit	Source (EventCode)	Initial Value	R/W	Description
31 to 1	Reserved	All 0	R/W	These bits are always read as 0. The write value should always be 0.
0	WDT (H'3E0)	0	R/W	[When read] 0: Distribution is disable 1: Distribution is enable [When written] 0: Invalid 1: Setting distribution

INT2DISTCR1

Bit	Source (EventCode)	Initial Value	R/W	Description
31	TMU-ch0 to 2 (H'400)	0	R/W	[When read] 0: Distribution is disable 1: Distribution is enable [When written] 0: Invalid 1: Setting distribution
30	TMU-ch0 to 2 (H'420)	0	R/W	
29	TMU-ch0 to 2 (H'440)	0	R/W	
28	TMU-ch0 to 2 (H'460)	0	R/W	
27	TMU-ch3 to 5 (H'480)	0	R/W	
26	TMU-ch3 to 5 (H'4A0)	0	R/W	
25	TMU-ch3 to 5 (H'4C0)	0	R/W	
24	(H'4E0)	0	R/W	
23	DMAC0 (H'500)	0	R/W	
22	DMAC0 (H'520)	0	R/W	
21	DMAC0 (H'540)	0	R/W	
20	DMAC0 (H'560)	0	R/W	
19	DMAC0 (H'580)	0	R/W	
18	DMAC0 (H'5A0)	0	R/W	
17	DMAC0 (H'5C0)	0	R/W	
16	H-UDI1 (H'5E0)	0	R/W	
15	H-UDI0 (H'600)	0	R/W	
14	DMAC1 (H'620)	0	R/W	
13	DMAC1 (H'640)	0	R/W	
12	DMAC1 (H'660)	0	R/W	
11	DMAC1 (H'680)	0	R/W	
10	HPB (H'6A0)	0	R/W	
9	HPB (H'6C0)	0	R/W	
8	HPB (H'6E0)	0	R/W	
7	SCIF0 (H'700)	0	R/W	
6	SCIF0 (H'720)	0	R/W	
5	SCIF0 (H'740)	0	R/W	
4	SCIF0 (H'760)	0	R/W	
3	SCIF1 (H'780)	0	R/W	
2	TMU-ch6 to 8 (H'7A0)	0	R/W	
1	TMU-ch9 to 11 (H'7C0)	0	R/W	
0	(H'7E0)	0	R/W	

INT2DISTCR2

Bit	Source (EventCode)	Initial Value	R/W	Description
31, 30	Reserved	All 0	R/W	These bits are always read as 0. The write value should always be 0.
29	SCIF2 (H'840)	0	R/W	[When read] 0: Distribution is disable 1: Distribution is enable [When written] 0: Invalid 1: Setting distribution
28	SCIF3 (H'860)	0	R/W	
27	SCIF4 (H'880)	0	R/W	
26	SCIF5 (H'8A0)	0	R/W	
25	Ether (H'8C0)	0	R/W	
24	Ether (H'8E0)	0	R/W	
23 to 9	Reserved	All 0	R/W	These bits are always read as 0. The write value should always be 0.
8	PCIEC0 (H'AE0)	0	R/W	[When read] 0: Distribution is disable 1: Distribution is enable [When written] 0: Invalid 1: Setting distribution
7	PCIEC0 (H'B00)	0	R/W	
6	PCIEC0 (H'B20)	0	R/W	
5	PCIEC1 (H'B40)	0	R/W	
4	PCIEC1 (H'B60)	0	R/W	
3	PCIEC1 (H'B80)	0	R/W	
2	USB (H'BA0)	0	R/W	
1	SDI0 (H'BC0)	0	R/W	
0	SDI1 (H'BE0)	0	R/W	

INT2DISTCR3

Bit	Source (EventCode)	Initial Value	R/W	Description
31	(H'C00)	0	R/W	[When read] 0: Distribution is disable 1: Distribution is enable [When written] 0: Invalid 1: Setting distribution
30	(H'C20)	0	R/W	
29	(H'C40)	0	R/W	
28	(H'C60)	0	R/W	
27	(H'C80)	0	R/W	
26	(H'CA0)	0	R/W	
25	I2C0 (H'CC0)	0	R/W	
24	I2C1 (H'CE0)	0	R/W	
23	DU (H'D00)	0	R/W	
22	SSI0 (H'D20)	0	R/W	
21	SSI1 (H'D40)	0	R/W	
20	SSI2 (H'D60)	0	R/W	
19	SSI3 (H'D80)	0	R/W	
18	PCIEC2 (H'DA0)	0	R/W	
17	PCIEC2 (H'DC0)	0	R/W	
16	PCIEC2 (H'DE0)	0	R/W	
15	HAC0 (H'E00)	0	R/W	
14	HAC1 (H'E20)	0	R/W	
13	FLCTL (H'E40)	0	R/W	
12	(H'E60)	0	R/W	
11	HSPI (H'E80)	0	R/W	
10	GPIO0 (H'EA0)	0	R/W	
9	GPIO1 (H'EC0)	0	R/W	
8	Thermal sensor (H'EE0)	0	R/W	
7 to 0	Reserved	All 0	R/W	These bits are always read as 0. The write value should always be 0.

10.3.8 Interrupt Acknowledgement Register

(1) Interrupt Acknowledgement Register (INTACK)

Interrupt acknowledgement register (INTACK) is for conducting an exclusive process under the automatic distribution mode by confirming the value after each CPU acknowledges interrupts.

When a CPU acknowledges an interrupt (except NMI), if it is supposed to be processed by the CPU (an interrupt under the fixed distribution mode, or an interrupt under the automatic distribution mode and the CPU has acknowledged the interrupt first among all other CPUs), 1 is set. Where as if it is not supposed to be processed by the CPU (it already has been acknowledged by another CPU), 0 is set (See section 10.5.3). Although INTACK is the same address for all the CPU for the high speed interrupt processing, each CPU can read the corresponding value.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INTACK	0	R/W	This sets the following values when acknowledging an interrupt. 0: Has not acknowledged the interrupt that should be processed. 1: Has acknowledged the interrupt that should be processed.

(2) Interrupt Acknowledgement Clear Register (INTACKCLR)

This register is for canceling an interrupt mask under the automatic distribution mode. When the users write the event code of the corresponding interrupt source to this register after clearing each interrupt source, it clears the mask for the interrupt source (See section 10.5.3).

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

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

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	INTEVENT	All 0	W	Interrupt mask clear event By writing the interrupt event code to this register under automatic distribution mode, it cancels the interrupt mask which has been exclusively acknowledged.

(3) IRQ Interrupt Acknowledgement Mask Register (IRQACKMASK)

Under the automatic distribution mode, IRQ interrupt acknowledgement mask register functions to mask IRQ interrupts that have been acknowledged and would have been processed by each CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AM07	AM06	AM05	AM04	AM03	AM02	AM01	AM00	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	AM07	0	R	INTACK mask While INTC exclusively sets the INTACK register bit 0 to '1', it masks an interrupt with the corresponding source (set AM00 to '1').
30	AM06	0	R	
29	AM05	0	R	
28	AM04	0	R	
27	AM03	0	R	
26	AM02	0	R	
25	AM01	0	R	
24	AM00	0	R	
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(4) IRL Interrupt Acknowledgement Mask Register 1 (IRLACKMASK)

Under the automatic distribution mode, IRL interrupt acknowledgement mask register functions to mask IRL interrupts that have been acknowledged and would have been processed by each CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AM00	AM01	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	AM00	0	R	INTACK mask
30	AM01	0	R	While INTC exclusively sets the INTACK register bit0 to '1', it masks an interrupt with the corresponding source (set AM00 to '1').
29 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(5) Peripheral Interrupt Acknowledgement Mask Register (PERIACKMASK0 to 3)

Under the automatic distribution mode, peripheral interrupt acknowledgement mask register 0 to 3 functions to mask interrupts that have been acknowledged and would have been processed by each CPU.

PERIACKMASK0

Bit	Source (EventCode)	Initial Value	R/W	Description
31 to 1	Reserved	All 0	R	These bits are always read as 0. The write value should always be 0.
0	WDT (H'3E0)	0	R	[When read] 0: ACK mask is disable 1: ACK mask is enable [When written] 0/1: Invalid

PERIACKMASK1

Bit	Source (EventCode)	Initial Value	R/W	Description
31	TMU-ch0 to 2 (H'400)	0	R	[When read] 0: ACK mask is disable 1: ACK mask is enable [When written] 0/1: Invalid
30	TMU-ch0 to 2 (H'420)	0	R	
29	TMU-ch0 to 2 (H'440)	0	R	
28	TMU-ch0 to 2 (H'460)	0	R	
27	TMU-ch3 to 5 (H'480)	0	R	
26	TMU-ch3 to 5 (H'4A0)	0	R	
25	TMU-ch3 to 5 (H'4C0)	0	R	
24	(H'4E0)	0	R	
23	DMAC0 (H'500)	0	R	
22	DMAC0 (H'520)	0	R	
21	DMAC0 (H'540)	0	R	
20	DMAC0 (H'560)	0	R	
19	DMAC0 (H'580)	0	R	
18	DMAC0 (H'5A0)	0	R	
17	DMAC0 (H'5C0)	0	R	
16	H-UDI1 (H'5E0)	0	R	
15	H-UDI0 (H'600)	0	R	
14	DMAC1 (H'620)	0	R	
13	DMAC1 (H'640)	0	R	
12	DMAC1 (H'660)	0	R	
11	DMAC1 (H'680)	0	R	
10	HPB (H'6A0)	0	R	
9	HPB (H'6C0)	0	R	
8	HPB (H'6E0)	0	R	
7	SCIF0 (H'700)	0	R	
6	SCIF0 (H'720)	0	R	
5	SCIF0 (H'740)	0	R	
4	SCIF0 (H'760)	0	R	
3	SCIF1 (H'780)	0	R	
2	TMU-ch6 to 8 (H'7A0)	0	R	
1	TMU-ch9 to 11 (H'7C0)	0	R	
0	(H'7E0)	0	R	

PERIACKMASK2

Bit	Source (EventCode)	Initial Value	R/W	Description
31, 30	Reserved	All 0	R	These bits are always read as 0. The write value should always be 0.
29	SCIF2 (H'840)	0	R	[When read] 0: ACK mask is disable 1: ACK mask is enable [When written] 0/1: Invalid
28	SCIF3 (H'860)	0	R	
27	SCIF4 (H'880)	0	R	
26	SCIF5 (H'8A0)	0	R	
25	Ether (H'8C0)	0	R	
24	Ether (H'8E0)	0	R	
23 to 9	Reserved	All 0	R	These bits are always read as 0. The write value should always be 0.
8	PCIEC0 (H'AE0)	0	R	[When read] 0: ACK mask is disable 1: ACK mask is enable [When written] 0/1: Invalid
7	PCIEC0 (H'B00)	0	R	
6	PCIEC0 (H'B20)	0	R	
5	PCIEC1 (H'B40)	0	R	
4	PCIEC1 (H'B60)	0	R	
3	PCIEC1 (H'B80)	0	R	
2	USB (H'BA0)	0	R	
1	SDI0 (H'BC0)	0	R	
0	SDI1 (H'BE0)	0	R	

PERIACKMASK3

Bit	Source (EventCode)	Initial Value	R/W	Description
31	(H'C00)	0	R	[When read] 0: ACK mask is disable 1: ACK mask is enable [When written] 0/1: Invalid
30	(H'C20)	0	R	
29	(H'C40)	0	R	
28	(H'C60)	0	R	
27	(H'C80)	0	R	
26	(H'CA0)	0	R	
25	I2C0 (H'CC0)	0	R	
24	I2C1 (H'CE0)	0	R	
23	DU (H'D00)	0	R	
22	SSI0 (H'D20)	0	R	
21	SSI1 (H'D40)	0	R	
20	SSI2 (H'D60)	0	R	
19	SSI3 (H'D80)	0	R	
18	PCIEC2 (H'DA0)	0	R	
17	PCIEC2 (H'DC0)	0	R	
16	PCIEC2 (H'DE0)	0	R	
15	HAC0 (H'E00)	0	R	
14	HAC1 (H'E20)	0	R	
13	FLCTL (H'E40)	0	R	
12	(H'E60)	0	R	
11	HSPI (H'E80)	0	R	
10	GPIO0 (H'EA0)	0	R	
9	GPIO1 (H'EC0)	0	R	
8	Thermal sensor (H'EE0)	0	R	
7 to 0	Reserved	All 0	R	These bits are always read as 0. The write value should always be 0.

10.4 Interrupt Sources

There are five types of interrupt sources, NMI, IRQ, IRL, on-chip module interrupts and interrupts among CPUs (ICI). Each interrupt has a priority level (16 to 0). Level 16 is the highest and level 1 is the lowest. When the level is set to 0, the interrupt is masked and interrupt requests are ignored.

10.4.1 NMI Interrupts

The NMI interrupt has the highest priority of level 16. The interrupt is always accepted unless the BL bit in SR of the CPU is set to 1. In sleep mode, the interrupt is accepted even if the BL bit is set to 1.

According to a setting, the NMI interrupt can be accepted even if the BL bit is set to 1.

Input from the NMI pin is detected at the edge. The NMI edge select bit (NMIE) in ICR0 is used to select from the rising or falling edge. After the NMIE bit in ICR0 is modified, the NMI interrupt is not detected for up to six bus clock cycles.

When the INTMU bit in the CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically set to level 15. When the INTMU bit in CPUOPM is cleared to 0, the IMASK value in SR is not affected by accepting an NMI interrupt.

10.4.2 IRQ Interrupts

The IRQ interrupt is valid when 1 is written to the IRLM0 and IRLM1 bits in ICR0 and pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL0}}$ are used for independent interrupts. The rising edge, falling edge, low level, and high level detections are enabled by setting the IRQnS1 and IRQnS0 bits (n = 0 to 7). The priority of interrupts is set by INTPR1.

If an IRQ interrupt request is detected by the low level or high level detection, the pin state of IRQ interrupt state should be retained until interrupt handling starts after interrupts are accepted.

When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically set to the level of the accepted interrupt. When the INTMU bit in CPUOPM is set to 0, the IMASK value in SR is not affected by accepting an IRQ interrupt.

10.4.3 IRL Interrupts

The IRL interrupt is an interrupt input as level from pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ or pins IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$.

The priority level is indicated by pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ or pins IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$. Pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ or pins IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ indicate the interrupt request with the highest priority (level 15) when these pins are all low. When these pins are all high, these pins indicate no interrupt requests (level 0). Figure 10.2 shows an example of IRL interrupt connection, and table 10.8 shows the correspondence between the levels on the IRL pins and interrupt priority.

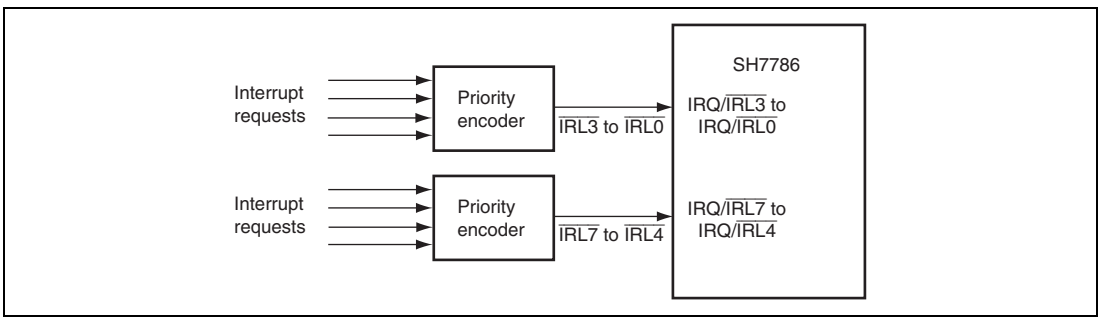


Figure 10.2 Example of IRL Interrupt Connection

Table 10.8 IRL Interrupt Pins (IRL[3:0], IRL[7:4]) and Interrupt Levels

$\overline{\text{IRL3}}$ or $\overline{\text{IRL7}}$	$\overline{\text{IRL2}}$ or $\overline{\text{IRL6}}$	$\overline{\text{IRL1}}$ or $\overline{\text{IRL5}}$	$\overline{\text{IRL0}}$ or $\overline{\text{IRL4}}$	Interrupt Priority Level	Interrupt Request
Low	Low	Low	Low	15	Level 15 interrupt request
Low	Low	Low	High	14	Level 14 interrupt request
Low	Low	High	Low	13	Level 13 interrupt request
Low	Low	High	High	12	Level 12 interrupt request
Low	High	Low	Low	11	Level 11 interrupt request
Low	High	Low	High	10	Level 10 interrupt request
Low	High	High	Low	9	Level 9 interrupt request
Low	High	High	High	8	Level 8 interrupt request
High	Low	Low	Low	7	Level 7 interrupt request
High	Low	Low	High	6	Level 6 interrupt request
High	Low	High	Low	5	Level 5 interrupt request
High	Low	High	High	4	Level 4 interrupt request
High	High	Low	Low	3	Level 3 interrupt request
High	High	Low	High	2	Level 2 interrupt request
High	High	High	Low	1	Level 1 interrupt request
High	High	High	High	0	No interrupt request

IRL interrupt detection requires an on-chip noise-cancellation feature. This detection is performed when the level sampled at every bus clock is the same for three consecutive cycles. This detection can prevent the incorrect level from being taking in when the IRL interrupt pin state changes.

The priority of IRL interrupts should be retained from when an interrupt is accepted to when interrupt handling starts. The level can be changed a higher level.

When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) bit in SR is automatically set to the level of the accepted interrupt. When the INTMU bit in CPUOPM is cleared to 0, the IMASK bit in SR is not affected.

When 1 is written to the IRLM0 and IRLM1 bits in ICR0, pins IRQ/$\overline{\text{IRL0}}$ to IRQ/$\overline{\text{IRL3}}$ or IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ can be used for independent IRQ interrupts. For details, see section 10.4.2, IRQ Interrupts.

10.4.4 On-Chip Peripheral Module Interrupts

On-chip module interrupts are interrupts generated in the on-chip modules.

The interrupt vectors are not allocated to each source, but the source is reflected on INTEVT. Therefore, the source can be easily identified by branching the value of INTEVT as offset during exception handling routine.

A priority ranging from 31 to 0 can be set for each module by INT2PRI0 to INT2PRI24. To notify the CPU of the priority, round down the least 1-bit and change to 4 bits. For details, see section 10.4.6, Priority of On-Chip Peripheral Module Interrupts.

An on-chip peripheral module interrupt source flag or an interrupt enable flag should be updated when the BL bit in SR is set to 1 or when the corresponding interrupt is not generated by the setting of interrupt masking occurs. To prevent the acceptance of incorrect interrupts by the updated interrupt source, read from the on-chip peripheral module register that has the corresponding flag. Then, clear the BL bit to 0 or update the interrupt masking setting so that the corresponding interrupt can be accepted, after waiting for the on-chip peripheral module priority determination time specified by table 10.9, Interrupt Response Time (for example, read from a register operating with the peripheral module clock in the INTC). These procedures can guarantee the required timing. To update multiple flag, read from the register that has the last flag after updating the flag.

If a flag is updated when the BL bit is 0, the processing may skip to interrupt processing routine with the INTEVT value cleared to 0. This is because interrupt processing starts, relative to the timing that the flag is updated and the interrupt request is identified in this LSI. The processing can be continued successfully by executing the RTE instruction.

Note that the GPIO interrupt is a low-active interrupts. Unlike the IRL interrupt and the IRQ interrupt that is detected by the level detection, the GPIO interrupt source cannot be retained by hardware if the pin state is changed and the interrupt request is withdrawn.

10.4.5 Interrupt among CPUs

The interrupt among CPUs occurs by the setting of interrupt control among CPUs register (CnINTICI).

Up to 8 sources (8 fields) can be inputted to 1 CPU and 4-bit logical OR for 1 field generates an interrupt.

2 ways are expected: -

1. Use separate fields depending on the different CPU that made a request.
2. Use separate fields depending on the different source notified among CPUs.

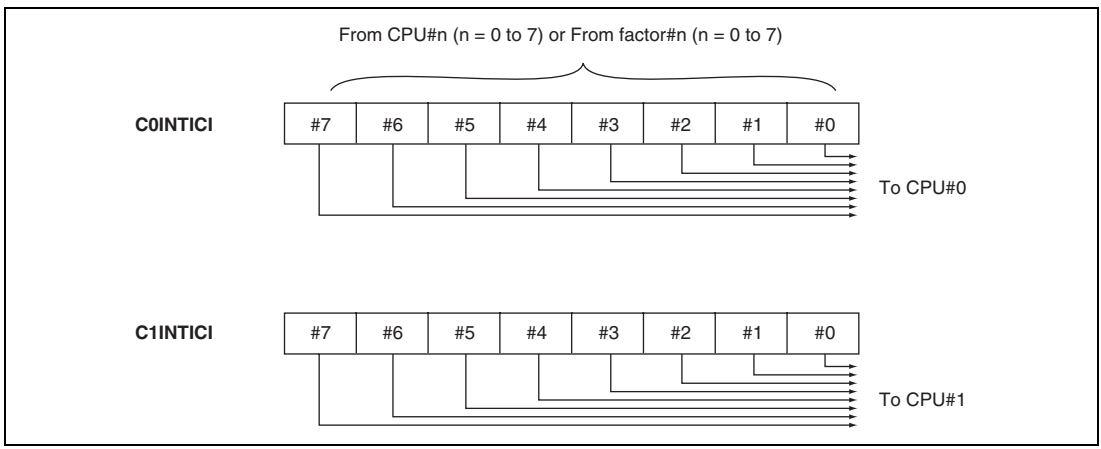


Figure 10.3 Interrupt among CPUs

When there are writings by multiple CPUs to one field, it will be managed by determining the bit position in the field for each CPU.

The CPU, which has acknowledged the interrupt request, clears the interrupt request by making the corresponding bit in the CnINTICI register 0 with the register for clearing interrupt among CPUs (CnINTICICLR).

10.4.6 Priority of On-Chip Peripheral Module

When any interrupt is generated, the on-chip peripheral module interrupt outputs the exception code specific to the source to SH-4A. Accepting the interrupt, SH-4A indicates the corresponding INTEVT code in INTEVT. An interrupt handler can identify the source by reading from INTEVT in SH-4A, without reading the source indicate register in the INTC. For the correspondence between on-chip peripheral module interrupt sources and exception codes, see table 10.1.

As shown in figure 10.4, an on-chip module interrupt source can be set to 30 levels (H'00 and H'01 mask interrupt requests) by the 5-bit field. The interrupt level receive interface in SH-4A can be set to 15 levels (H'0 masks interrupt requests) by the 4-bit field. The priority of on-chip peripheral module interrupts is determined by selecting each interrupt source with 5 bits that are extended 1 bit. Then, change 4 bits that round down the least 1 bit and notifies. For example, two interrupt sources with priority levels set to H'1A and H'1B will both be output to the CPU as the 4-bit priority level H'D. The two interrupts sources have the same priority value. However, although the rounded codes are the same for both interrupt sources, the interrupt with priority level H'1B clearly has priority when we consider the 5-bit data in the priority setting. That is, the 5-bit values in the fields give INTC a way to differentiate between interrupts with the same four-bit priority level. If the interrupts of the same priority coinside, the INTEVT code is notified according to the priority shown in table 10.1.

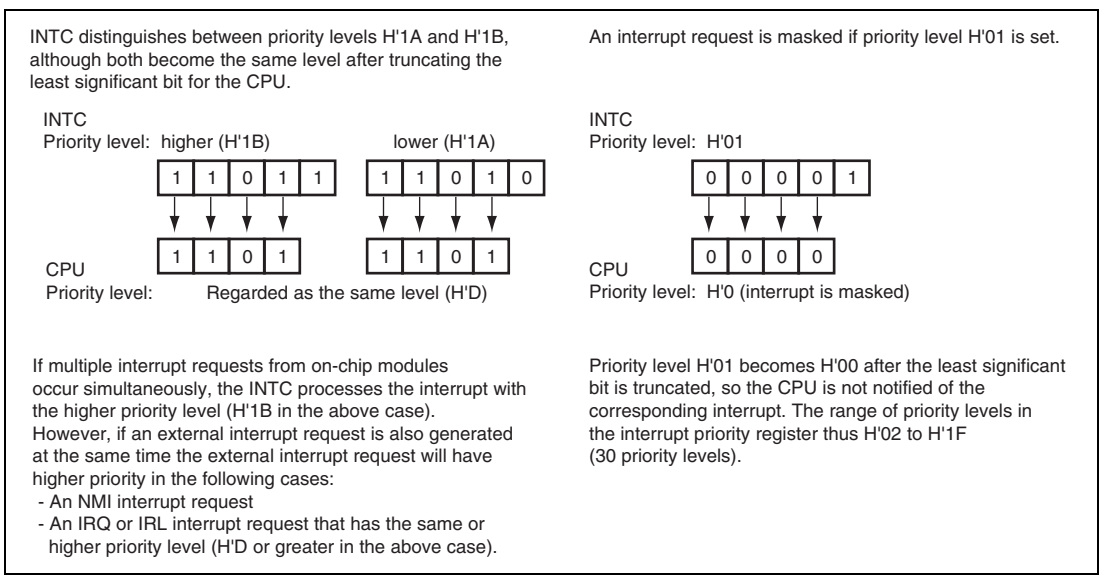


Figure 10.4 Priority of On-Chip Peripheral Module Interrupts

10.4.7 Interrupt Exception Handling and Priority

Table 10.1 shows the interrupt source, codes for INTEVT, and the order of interrupt priority.

A unique INTEVT code is allocated to each interrupt source. The start address of the exception handling routine is the same for all of the interrupt sources. Therefore, the INTEVT value is used to control branching at the start of the exception handling routine. For instance, the INTEVT values are used to branch to offsets.

The priority of the on-chip modules is arbitrarily specified by setting values from 31 to 2 in INT2PRI0 to INT2PRI7. The priority values for the on-chip modules are set to 0 by a reset.

When interrupt sources share the same priority level and are generated simultaneously, they are handled according to the default priority order given in table 10.1.

Values of INTPRI and INT2PRI0 to INT2PRI24 should only be updated when the BL bit in SR is set to 1. To prevent erroneous interrupt acceptance, only clear the BL bit to 0 after having read one of the interrupt priority level-setting registers. This guarantees the necessary timing internally.

10.5 Operation

10.5.1 Interrupt Sequence

The sequence of interrupt operations is described below. Figure 10.5 shows a flowchart of the operations.

1. Interrupt request sources send interrupt request signals to the INTC.
2. The INTC selects the interrupt with the highest priority among the interrupts that have been sent, according to the priority set in INTPRI and INT2PRI0 to INT2PRI24. Lower priority interrupts are held as pending interrupts. If two of the interrupts have the same priority level or multiple interrupts are generated by a single module, the interrupt with the highest priority is selected according to table 10.1.
3. The priority level of the interrupt selected by the INTC is compared with the interrupt mask level (IMASK) set in SR of the CPU. Only the interrupt with a higher priority than the IMASK bit is accepted, and an interrupt request signal is sent to the CPU.
4. The CPU accepts an interrupt at the next break between instructions.
5. The interrupt source code is set in the interrupt event register (INTEVT).
6. The SR and program counter (PC) are saved in SSR and SPC, respectively. At that time, R15 is saved in SGR.
7. The BL, MD, and RB bits in SR are set to 1.
8. Execution jumps to the start address of the interrupt exception handling routine (the sum of the value set in the vector base register (VBR) and H'0000 0600).

In the exception handling routine, the value of INTEVT is branched as an offset. This easily enables to branch the exception handling routine to handling routine for each interrupt source.

Notes: 1. When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically set to the level of the accepted interrupt. When the INTMU bit in CPUOPM is cleared to 0, the value of the IMASK in SR is not affected by the accepted interrupt.

2. The interrupt source flag should be cleared during exception handling routine. To ensure that an interrupt source which should have been cleared is not erroneously accepted again, read the interrupt source flag after it has been cleared, and wait for the time shown in table 10.9. Then, clear the BL bit or execute an RTE instruction.
3. The IRQ interrupts, IRL interrupts, on-chip peripheral module interrupts are initialized to the interrupt masking state by a power-on reset. Therefore, clear the interrupt masking for each interrupt , INTMSK0, INTMSK1, and INT2MSKR by using INTMSKCLR0, INTMSKCLR1, and INT2MSKCLR.

10.5.2 Fixed Distribution Mode

This mode is to distribute interrupt sources to the corresponding CPU regularly.

By setting the interrupt mask register for each CPU, the corresponding interrupt request will be masked and the processing CPU will be regularly selected and used. It is expected that 1 interrupt source is distributed to 1 CPU.

The followings are the examples of expected handler process.

A-1 When acknowledging multiple interrupts,
Set IMASK
Save SSR, SPC and INTEVT
Clear BL
A-2 Analyze the source with and make branch off to the routine for each source (A-3)
A-3 Clear the module interrupt flag
A-4 Process each module
A-5 When there are multiple interrupts, set BL
Restore SSR, SPC and INTEVT
A-6 RTE

10.5.3 Automatic Distribution Mode

This mode is to distribute interrupt sources to CPUs.

The interrupt requests are distributed simultaneously to the target CPUs. As soon as one of the CPUs acknowledges the request, other interrupt requests will be masked.

The CPU, which acknowledges the interrupt, checks the interrupt acknowledgment register (INTACK) and judges whether it should be processed or returned from the handler.

The followings are the examples of expected handler process.

S-1 When the value read from INTACK register is 0 (When the interrupt should not be acknowledged by the corresponding CPU), RTE
S-2 When CPU should acknowledge the interrupt,
S-2-1 Fixed distribution mode process (A-1 to A-3)
S-2-2 Clears the corresponding interrupt source on the interrupt acknowledgment clear register (INTACKCLR)
S-2-3 Fixed distribution mode process (A-4 to A-6)

After one of the CPUs acknowledges the interrupt request, the corresponding interrupt requests will be masked until it is cleared with the interrupt acknowledgment clear register (INTACKCLR). The interrupt requests with other sources will be outputted to CPUs.

Restoration from the sleep mode

If the CPU, which is in sleep mode, can acknowledge the interrupt (when the interrupt that has higher priority than imask occurs), that CPU is restored from the sleep mode.

When multiple CPUs are in sleep modes, 1 CPU that is acknowledgeable is restored from the sleep mode.

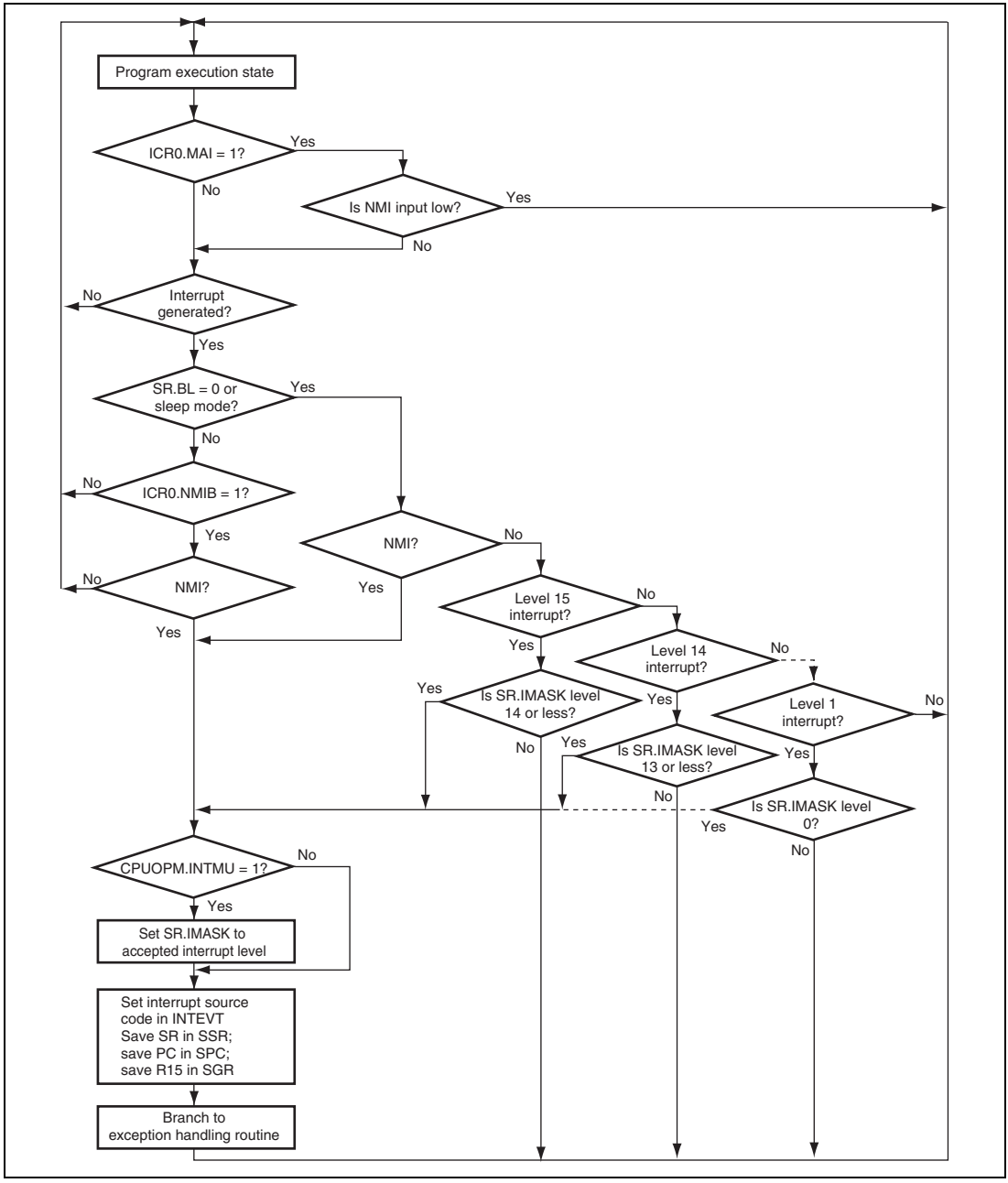


Figure 10.5 Flowchart of Interrupt Operation

10.5.4 Multiple Interrupts

To handle multiple interrupts, the procedure for the interrupt handling routine should be as follows.

1. To identify the interrupt source, set the value of INTECT to an offset and branch it to the interrupt handling routine for each interrupt source.
2. Clear the corresponding interrupt source in the interrupt handling routine.
3. Save SSR and SPC on the stack.
4. Clear the BL bit in SR. When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically set to the level of the accepted interrupt. When the INTMU bit in CPUOPM is cleared to 0, software should be used to set the IMASK bit in SR to the priority level of the accepted interrupt.
5. Execute processing as required in response to the interrupt.
6. Set the BL bit in SR to 1.
7. Release SSR and SPC from the stack.
8. Execute the RTE instruction.

By following the above procedure, if further interrupts are generated right after step 4, an interrupt with higher priority than the one currently being handled can be accepted after step 4. This reduces the interrupt response time for urgent processing.

10.5.5 Interrupt Masking by MAI Bit

When the MAI bit in ICR0 is set to 1, interrupts can be masked whole the NMI pin is low regardless of the settings of the BL and IMASK bits in and SR.

- Normal operation or sleep mode

All other interrupts are masked while the NMI signal is low. Note that only NMI interrupts due to the change of the NMI pin are generated.

10.6 Interrupt Response Time

Table 10.9 shows response time. The response time is the interval from generation of an interrupt request until the start of interrupt exception handling and until fetching of the first instruction of the exception handling routine.

Table 10.9 Interrupt Response Time

	Number of States					
				Peripheral Modules		
Item	**NMI**	**IRL**	**IRQ**	**Other than GPIO/PCIC/ RTC**	**GPIO/PCIC/ RTC**	**Remarks**
Priority determination time	7Bcyc + 1Sacyc	5Bcyc + 1Sacyc + 1Sacyc	3Bcyc + 2Sacyc + 1Sacyc	1Pcyc + 1Sacyc	2Pcyc/3Pcyc + 1Sacyc	
Wait time until the CPU finishes the current sequence			S-1 (≥ 0) × Icyc			
Interval from the start of interrupt exception handling (saving SR and PC) until a SuperHyway bus request is issued to fetch the first instruction of the exception handling routine			11Icyc + 1Scyc			
Respons Total e time	(S + 10) Icyc + 1Scyc + 7Bcyc + 1Sacyc	(S + 10) Icyc + 1Scyc + 5Bcyc + 1Sacyc + 1Sacyc	(S + 10) Icyc + 1Scyc 3Bcyc + 2Sacyc + 1Sacyc	(S + 10) Icyc + 1Scyc + 1Pcyc + 1Sacyc	(S + 10) Icyc + 1Scyc 2Pcyc/3Pcyc + 1Sacyc	

Legend:

Icyc: Period of one CPU clock cycle

Scyc: Period of one SuperHyway clock cycle

Sacyc: Period of 1/2 SuperHyway clock cycle

Bcyc: Period of one CLKOUT cycle

Pcyc: Period of one peripheral clock cycle

S: Number of instruction execution states

Table 10.10 shows response time. The response time is from the interrupt exception handling to the start of fetching the first instruction in exception handling routine. In this case, suppose that the setting values of the following registers that enable or disable interrupt, INTMSK0, INTMSK1, INTMSK2, INT2MSKR, and INT2GPIC, are changed from the interrupt disable state to the interrupt enable state.

Table 10.10 Response Time after Changing the Value of Interrupt Enable/Disable Registers (Interrupt Disabled → Interrupt Enabled)

	Number of States				
	IRL		IRQ	Peripheral Modules	Remarks
Item	INTMSK1	INTMSK2	INTMSK0	INT2MSKR, INT2GPIC	Registers that enable/disable interrupts
Priority determination time*	5Sacyc	8Bcyc + 2Pcyc	4Sacyc	4Sacyc	
Wait time until the CPU finishes the current sequence	S-1 (≥ 0) × Icyc				
Interval from the start of interrupt exception handling (saving SR and PC) until a SuperHyway bus request is issued to fetch the first instruction of the exception handling routine	11Icyc + 1Scyc				
Response time Total	(S + 10) Icyc + 1Scyc + 1Pcyc	(S + 10) Icyc + 1Scyc + 8Bcyc + 2Pcyc	(S + 10) Icyc + 1Scyc + 1Pcyc	(S + 10) Icyc + 1Scyc + 4Pcyc	

Legend:

Icyc: Period of one CPU clock cycle

Scyc: Period of one SuperHyway clock cycle

Sacyc: Period of 1/2 SuperHyway clock cycle

Bcyc: Period of one CLKOUT cycle

Pcyc: Period of one peripheral clock cycle

S: Number of instruction execution states

Note: * When INTMSKCLR0, INTMSKCLR1, INTMSKCLR2, and INT2MSKCLR are written to, INTMSK0, INTMSK1, INTMSK2, and INTMSKR enable an interrupt by clearing the mask bits in INTMSK0, INTMSK1, INTMSK2, and INTMSKR. The priority determination times in table 10.11 are the values after the values of INTMSK0, INTMSK1, INTMSK2, and INT2MSKR are changed.

Table 10.11 shows response time. The response time is the time until when the interrupt request signal from the INTC to the CPU is negated. In this case, suppose that the setting values of the following registers, INTMSK0, INTMSK1, INTMSK2, INT2MSKR, and INT2GPIC, are changed from the interrupt enable state to the interrupt disable state.

Table 10.11 Response Time after Changing the Value of Interrupt Enable/Disable Registers (Interrupt Enabled → Interrupt Disabled)

	Number of States				
	IRL		IRQ	Peripheral Modules	Remarks
Item	INTMSK1	INTMSK2	INTMSK0	INT2MSKR, INT2GPIC	Registers that enable/disable interrupts
Priority determination time	5Sacyc	8Bcyc + 2Pcyc*	4Sacyc	4Sacyc	

Note: * The IRL interrupt source that has been already retained inside cannot cancel the interrupt request signal to the CPU even if the IRL interrupt source is masked.

10.7 Usage Notes

10.7.1 Notes on Setting IRQ/$\overline{\text{IRL[7:0]}}$ Pin Function

When the IRQ/$\overline{\text{IRL[7:0]}}$ pin functions are switched, the INTC may retain an incorrectly detected interrupt request. Therefore, mask the IRL and IRQ interrupt requests before switching the IRQ/$\overline{\text{IRL[7:0]}}$ pin functions.

Table 10.12 Switching Sequence of IRQ/$\overline{\text{IRL[7:0]}}$ Pin Function

Sequence	Item	Procedure
1	IRL interrupt request and IRQ interrupt request masking	Write 1 to all bits in INTMSK0 and INTMSK1
2	Setting IRL/IRQ[7:4] pins to IRL7 to IRL4	Write B'01 to P2MSEL1, P2MSEL0 bits in P2MSELR, Write B'00 to PH3MD1, PH3MD0 bits in PHCR, Write B'00 to PH2MD1 PH2MD0 bits in PHCR, Write B'00 to PH1MD1, PH1MD0 bits in PHCR, Write B'00 to PH0MD1, PH0MD0 bits in PHCR.
3	Setting IRQ/IRL[7:0] pins to IRL or IRQ	Set bits IRLM1 to IRLM0 in ICR0
4	Start of IRL and IRQ interrupt detection	Write 1 to the corresponding bit in INTMSKCLR0 and INTMSKCLR1

10.7.2 Clearing IRQ and IRL Interrupt Requests

To clear the interrupt request retained in the INTC, follow the procedure below.

- Clearing IRQ interrupt requests at edge detection

To clear the interrupt requests IRQ7 to IRQ0 setting edge detection, read the IR7 to IR0 bits corresponding to INTREQ as 1 and write 0 to the bits. The IRQ interrupt request being detected cannot be cleared even if 1 is written to the corresponding bit in INTMSK0.

- The INTC does not retain the interrupt source even if IRQ interrupts are detected at level detection or IRL interrupt requests are detected.

Section 11 Local Bus State Controller (LBSC)

The local bus state controller (LBSC) divides the external memory space and outputs control signals according to the specification of each memory and bus interface. The LBSC function enables connection of the SRAM or ROM, etc. to this LSI. The LBSC also supports the PCMCIA interface protocol, which implements simple system design and high-speed data transfers in a compact system.

11.1 Features

The LBSC has the following features.

- Manages areas 0 to 6 of the external memory space divided into seven
  - Maximum 128 Mbytes for area 0 (specified by external pin)
  - Maximum 64 Mbytes for each of areas 1 to 6
  - Bus width of each area can be set by a register (Only the area-0 bus width is set by an external pin.)
  - Wait cycle insertion by the $\overline{\text{RDY}}$ pin
  - Wait cycle insertion can be controlled by a program
  - Type of memory to be connected is specifiable for each area
  - Control signals are output for memory connected to each area
  - Automatic wait cycle insertion to prevent data bus collision in consecutive memory accesses
  - The write strobe setup and hold time periods can be inserted in a write cycle to connect to low-speed memory
- SRAM interface
  - Wait cycle insertion can be controlled by a program

    Connectable area: 0 to 6

    Settable bus width: 32, 16 and 8 bits
- Burst ROM interface
  - Wait cycle insertion can be controlled by a program
  - Burst transfers for the number of times specified by the register

    Connectable area: 0 to 6

    Settable bus width: 32, 16 and 8 bits

- MPX interface
  - — Address/data multiplexing
    Connectable area: 0 to 6
    Settable bus width: 32 bits
- Byte control SRAM interface
  - — SRAM interface with byte control
    Connectable area: 1 and 4
    Settable bus width: 32 and 16 bits
- PCMCIA interface (Little endian only)
  - — Wait cycle insertion can be controlled by a program
  - — Bus sizing function for I/O bus width
  - — Supports the little endian
    Connectable area: 5 and 6
    Settable bus width: 16 and 8 bits
  - — Prepared only for ATA device accesses

Figure 11.1 shows a block diagram of the LBSC.

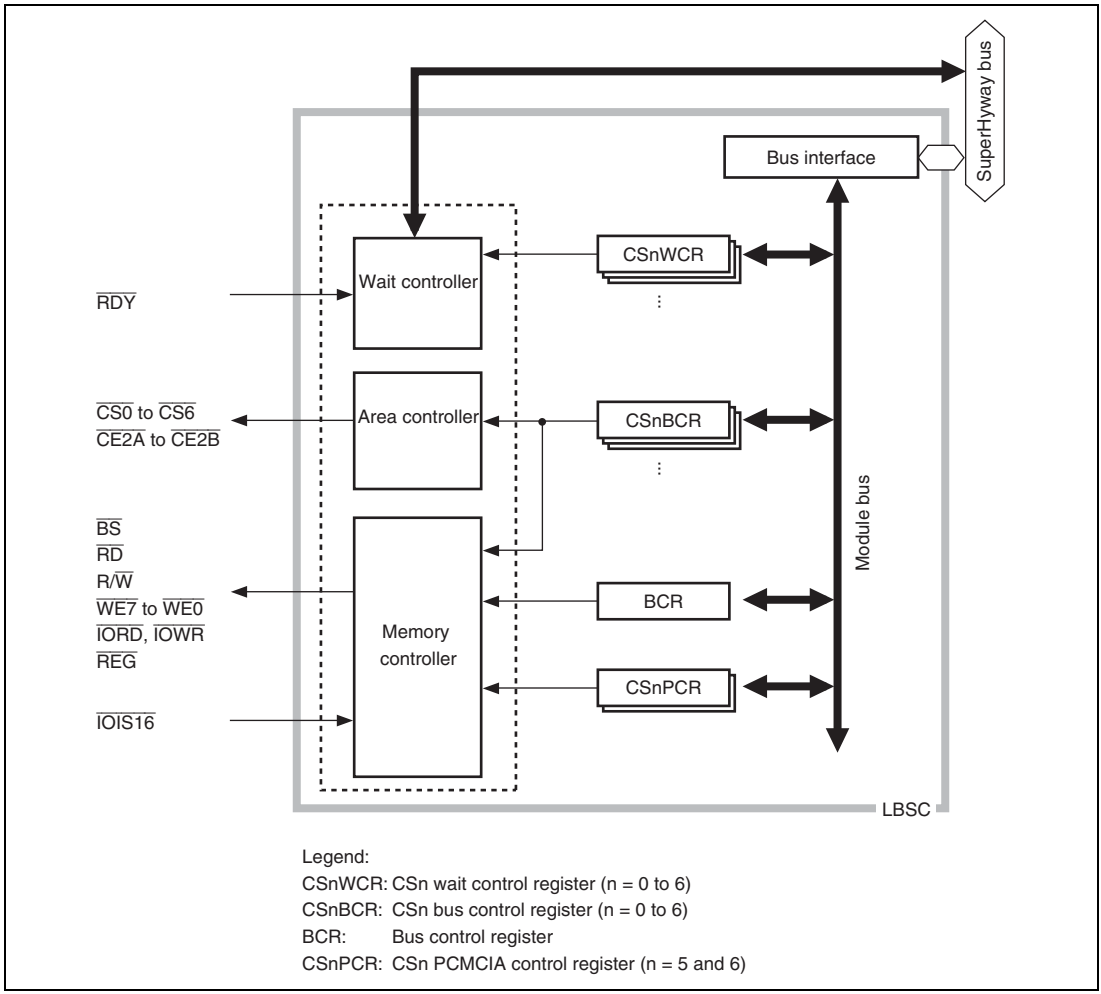


Figure 11.1 Block Diagram of LBSC

11.2 Input/Output Pins

Table 11.1 shows the LBSC pin configuration.

Table 11.1 Pin Configuration

Pin Name	Function	I/O	Description
A25 to A0	Address Bus	O	Address output
D31 to D8 D7/FD7 to D0/FD0	Data Bus	I/O	Data input/output D7/FD7 to D0/FD0 is used as a FLCTL data input/output, when FLCTL bus access right is granted.
$\overline{BS}$	Bus Cycle Start	O	Signal that indicates the start of a bus cycle Asserted once for a burst transfer when the MPX interface is set Asserted in every data cycle in other burst transfers
$\overline{CS6}$ to $\overline{CS2}$ $\overline{CS1}$/$\overline{A26}$ $\overline{CS0}$/$\overline{FCE0}$	Chip Select 6 to 0	O	Chip select signals that indicates the area being accessed. $\overline{CS5}$ and $\overline{CS6}$ can also be used as $\overline{CE1A}$ and $\overline{CE1B}$ of the PCMCIA respectively. $\overline{CS1}$/$\overline{A26}$ is used as A26 output, when CS0 is used with 128Mbytes size. $\overline{CS0}$/$\overline{FCE0}$ is used as CE signal, when FLCTL bus access right is granted.
R/$\overline{W}$	Read/Write	O	Data bus input/output direction designation signal. Also used as the PCMCIA interface write designation signal
$\overline{RD}$/$\overline{FRAME}$/$\overline{FRE}$	Read/Cycle Frame	O	Strobe signal indicating a read cycle. Used for $\overline{FRAME}$ signal when the MPX bus is used $\overline{RD}$/$\overline{FRAME}$/$\overline{FRE}$ is used as RE signal, when FLCTL bus access right is granted.
$\overline{WE0}$/$\overline{REG}$	Data Enable 0	O	Write strobe signal for D7 to D0 in SRAM interface setting $\overline{REG}$ signal in PCMCIA interface setting
$\overline{WE1}$	Data Enable 1	O	Write strobe signal for D15 to D8 in SRAM interface setting Write strobe signal in PCMCIA interface setting

Pin Name	Function	I/O	Description
WE2/IORD	Data Enable 2	O	Write strobe signal for D23 to D16 in SRAM interface setting IORD signal in PCMCIA interface setting
WE3/IOWR	Data Enable 3	O	Write strobe signal for D31 to D24 in SRAM interface setting IOWR signal in PCMCIA interface setting
RDY	Ready	I	Wait cycle request signal
IOIS16	16-Bit I/O	I	16-bit I/O designation signal in PCMCIA interface setting Valid only in little endian mode Multiplexed with MODE13, and SCIF5 RXD (SCIF5 input).
BREQ	Bus Release Request	I	Bus release request signal Multiplexed with DREQ1 (DMAC0 input), and USB OVC1 (USB input).
BACK	Bus Request Acknowledge	O	Bus request acknowledge signal Multiplexed with DACK1 (DMAC0 output), and FALE (FLCTL output).
CE2A*[1], CE2B*[2]	PCMCIA Card Select	O	CE2A, CE2B in PCMCIA interface setting Only valid in little endian mode CE2A: Multiplexed with MODE11, DRAK3 (DMAC0 output). CE2B: Multiplexed with MODE12, SCIF5 TXD (SCIF5 output).
MODE4, MODE5, MODE6	Bus Width and Memory Type for Area 0	I	Signals for setting area 0 bus width and MPX interface (MODE4, MODE5, MODE6) at a power-on reset by the PRESET pin MODE4: Multiplexed with SCIF0_CTS (SCIF0 output), DREQ2 (DMAC0 input) and SDIF0CLK (SD0 output). MODE5: Multiplexed with DREQ3 (DMAC0 input), SDIF0WP (SD0 input). MODE6: Multiplexed with DACK2 (DMAC0 output), SCIF0CD (SD0 input)
MODE7	Extension of Area 0	I	Signal for setting extension of Area 0 at a power-on reset by the PRESET pin Multiplexed with DACK3 (DMAC0 output), SDIF0CMD (SD0 output).

Pin Name	Function	I/O	Description
MODE8	Endian Switching	I	Signals for endian switching at a power-on reset by the $\overline{\text{PRESET}}$ pin Multiplexed with SCIF4_TXD (SCIF4 output), DRAK0 (DMAC0 output), SSI3_SCK (SSI3 output), and FSE (FLCTL output).
$\overline{\text{DACK0}}$*[3]	DMAC0 Acknowledge Signal	O	Data acknowledge in DMAC0 channel 0 Multiplexed with FCLE (FLCTL output)
$\overline{\text{DACK1}}$*[3]	DMAC1 Acknowledge Signal	O	Data acknowledge in DMAC0 channel 1 Multiplexed with BACK (LBSC output), and SDIF0_CD (SD0 input).
$\overline{\text{DACK2}}$*[3]	DMAC2 Acknowledge Signal	O	Data acknowledge in DMAC0 channel 2 Multiplexed with MODE6 (mode input), SDIF0_CD (SD0 input)
$\overline{\text{DACK3}}$*[3]	DMAC3 Acknowledge Signal	O	Data acknowledge in DMAC0 channel 3 Multiplexed with MODE7 (mode input), SDIF0CMD (SD0 output)

Notes: 1. $\overline{\text{CE2A}}$ is an output pin when the TYPE bits in CS5BCR are set to B'100.
2. $\overline{\text{CE2B}}$ is an output pin when the TYPE bits in CS6BCR are set to B'100.
3. The polarity of $\overline{\text{DACK0}}$ to $\overline{\text{DACK3}}$ can be selected by the AL bit in CHCR0 to CHCR3 of DMAC0 (the initial value selects active-low). For details, see section 15, Direct Memory Access Controller0 (DMAC0).

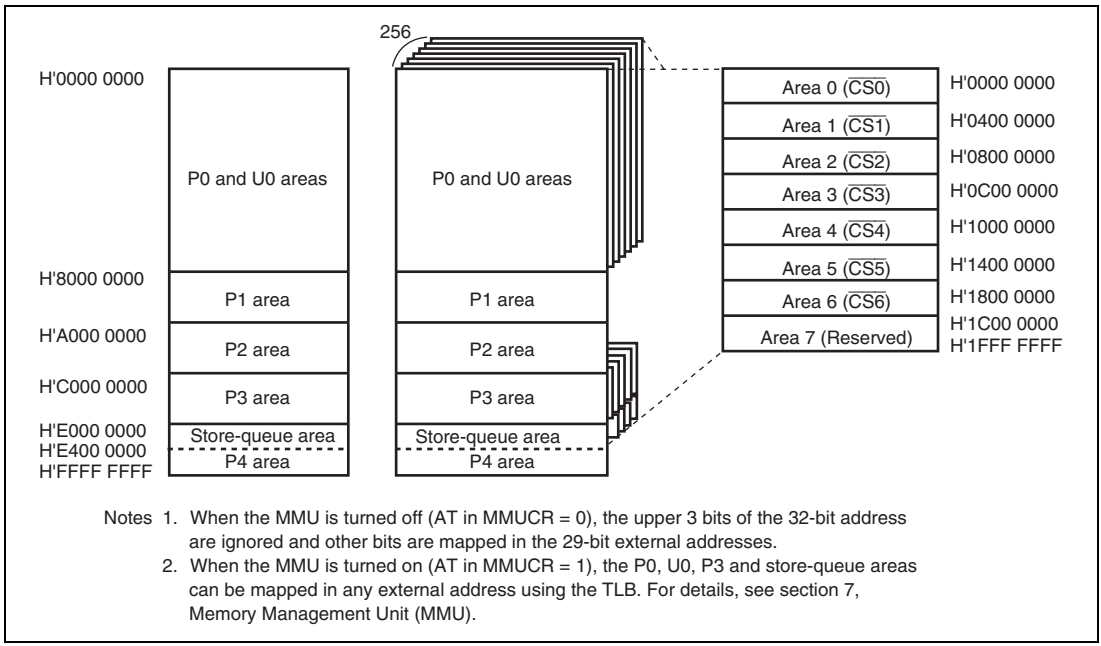
11.3 Overview of Areas

11.3.1 Space Divisions

The LSI has a 32-bit virtual address space as the architecture. The virtual address space is divided into five areas according to the upper address value. Also, the memory space of the local bus has a 29-bit address space, and it is divided into eight areas.

A virtual address can be allocated to any external address by the address changing unit (MMU). For details, see section 7, Memory Management Unit (MMU). This section describes the local bus address area division.

Various types of memory or PC cards can be connected to the external address seven areas as shown in table 11.2 and the chip select signals ($\overline{CS0}$ to $\overline{CS6}$, $\overline{CE2A}$, and $\overline{CE2B}$) are output for each area. $\overline{CS0}$ is asserted when area 0 is accessed, and $\overline{CS6}$ is asserted when areas 6 is accessed. When the PCMCIA interface is selected for area 5 or 6, $\overline{CE2A}$ or $\overline{CE2B}$ is asserted along with $\overline{CS5}$ or $\overline{CS6}$, according to the accessed bytes.



Notes 1. When the MMU is turned off (AT in MMUCR = 0), the upper 3 bits of the 32-bit address are ignored and other bits are mapped in the 29-bit external addresses.
2. When the MMU is turned on (AT in MMUCR = 1), the P0, U0, P3 and store-queue areas can be mapped in any external address using the TLB. For details, see section 7, Memory Management Unit (MMU).

Figure 11.2 Correspondence between Virtual Address Space and Local Bus Memory Space

Table 11.2 LBSC External Memory Space Map

Area	External addresses	Size	Connectable Memory	Specifiable Bus Width (bits)	Access Size*[7]
0	H'0000 0000 to H'03FF FFFF	64 Mbytes	SRAM	8, 16, 32*[1]	8/16/32 bits, 32 bytes
			Burst ROM	8, 16, 32*[1]	
			MPX	32*[1]	
1	H'0400 0000 to H'07FF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits, 32 bytes
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			Byte control SRAM	16, 32*[2]	
2	H'0800 0000 to H'0BFF FFFF	64 Mbytes	SRAM	8,16, 32*[2]	8/16/32 bits, 32 bytes
			Burst ROM	8,16, 32*[2]	
			MPX	32*[2]	
			(DDR3-SDRAM)*[3]	16, 32*[2]	
3	H'0C00 0000 to H'0FFF FFFF	64 Mbytes	SRAM	8,16, 32*[2]	8/16/32 bits, 32 bytes
			Burst ROM	8,16, 32*[2]	
			MPX	32*[2]	
			(DDR3-SDRAM)*[3]	16, 32*[3]	
4	H'1000 0000 to H'13FF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits, 32 bytes
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			Byte control SRAM	16, 32*[2]	
			(DDR3-SDRAM)*[3]	16, 32*[3]	8/16/32 bits, 32 bytes
			(PCIEC)*[4]	32	8/16/32 bits, 32 bytes
5	H'1400 0000 to H'17FF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits, 32 bytes
			MPX	32*[2]	
			Burst ROM	8, 16, 32*[2]	
			PCMCIA	8, 16*[2, 5]	
			(DDR3-SDRAM)*[2]	16, 32	8/16/32 bits, 32 bytes
			(LRAM)*[8]	—	8/16/32 bits, 16/32 bytes

Area	External addresses	Size	Connectable Memory	Specifiable Bus Width (bits)	Access Size*[7]
6	H'1800 0000 to H'1BFF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits, 32 bytes
			MPX	32*[2]	
			Burst ROM	8, 16, 32*[2]	
			PCMCIA	8, 16*[2, 5]	
7*[6]	H'1C00 0000 to H'1FFF FFFF	64 Mbytes	—	—	—

Notes:
1. The memory bus width is specified by the external pins.
2. The memory bus width is specified by the register.
3. These areas can be allocated to DDR3-SDRAM by setting MMSELR. For details, see section 12, DDR3-Contoller Interface.
4. This area can be allocated to PCI memory by setting MMSELR. For details, see section 13, PCI Express Controller.
5. When the PCMCIA interface is used, the bus width should be 8 or 16 bits.
6. Do not access the reserved area. If the reserved area is accessed, correct operation is not be guaranteed.
7. If the LBSC is requested to perform 8- or 16-byte access by the bus master, the LBSC performs accesses two or four times respectively with 32-bit access size.
8. This area can be allocated to LRAM area by setting MMSELR.

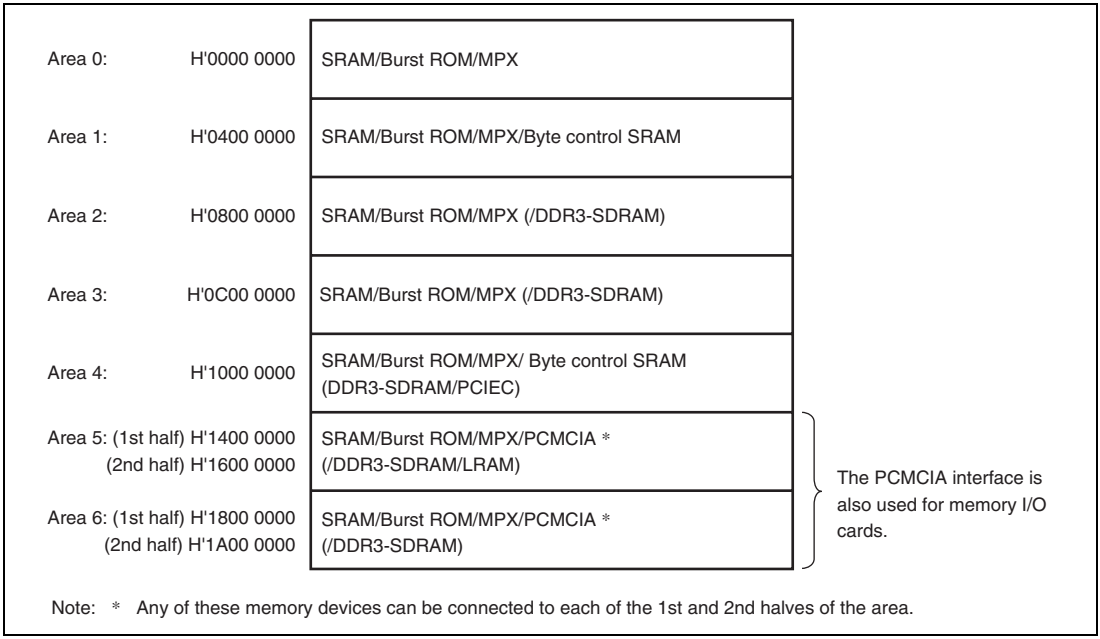


Figure 11.3 Local Bus Memory Space Allocation

11.3.2 Memory Bus Width

The memory bus width of the LBSC can be set independently for each area. In area 0, a bus width of 8, 16, or 32 bits is selected according to the external pin settings at a power-on reset by the $\overline{\text{PRESET}}$ pin. The relation between the external pins (MODE 6, MODE 5 and MODE 4) and the bus width at a power-on reset is shown below.

MODE6	MODE5	MODE4	Bus Width
0	0	0	32 bits (MPX)
0	0	1	8 bits
0	1	0	16 bits
0	1	1	32 bits

When the SRAM or ROM interface is used in areas 0 to 6, a bus width of 8, 16, or 32 bits can be selected by the CSn bus control register (CSnBCR). When the burst ROM interface is used, a bus width of 8, 16, or 32 bits can be selected. When the byte control SRAM interface is used, a bus width of 16, or 32 bits can be selected. When the MPX interface is used, the bus width should be set to 32 bits.

When the PCMCIA interface is used, the bus width should be set to 8 or 16 bits. For details, see section 11.5.5, PCMCIA Interface.

For details of memory bus width, see section 11.4.3, CSn Bus Control Register (CSnBCR).

The address range of area 7, from H'1C00 0000 to H'1FFF FFFF, is reserved and must not be used.

11.3.3 PCMCIA Support

This LSI supports the PCMCIA interface specifications for areas 5 and 6 in the external memory space.

The IC memory card interface and I/O card interface specified in JEIDA specifications version 4.2 (PCMCIA2.1) are supported.

Both the IC memory card interface and the I/O card interface are supported in areas 5 and 6 in the external memory space.

The PCMCIA interface is supported only in little endian mode.

Table 11.3 PCMCIA Interface Features

Item	Features
Access	Random access
Data bus	8/16 bits
Memory type	Mask ROM, OTPROM, EPROM, flash memory, SRAM, ATA device
Common memory capacity	Max. 64 Mbytes
Attribute memory capacity	Max. 64 Mbytes
Others	Dynamic bus sizing for I/O bus width, access to the ATA device control register

Table 11.4 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	Corresponding Pin of SH7786
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	$\overline{CS5}$ or $\overline{CS6}$
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	$\overline{WE}$	I	Write enable	$\overline{WE}$	I	Write enable	$\overline{WE1}$
16	$\overline{READY}$	O	Ready	$\overline{IREQ}$	O	Interrupt request	Sensed on port
17	VCC		Operating power supply	VCC		Operating power supply	—
18	VPP1 (VPP)		Programming power supply	VPP1 (VPP)		Programming/ 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
26	A3	I	Address	A3	I	Address	A3
27	A2	I	Address	A2	I	Address	A2

	IC Memory Card Interface			I/O Card Interface			
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	Corresponding Pin of SH7786
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	$\overline{WP}$*1	O	Write protect	$\overline{IOIS16}$	O	16-bit I/O port	$\overline{IOIS16}$
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	$\overline{CD1}$	O	Card detection	$\overline{CD1}$	O	Card detection	Sensed on port
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	$\overline{CE2}$	I	Card enable	$\overline{CE2}$	I	Card enable	$\overline{CE2A}$ or $\overline{CE2B}$
43	RFSH ($\overline{VS1}$)	I	Refresh request	RFSH ($\overline{VS1}$)	I	Refresh request	Output from port
44	RSRVD		Reserved	$\overline{IORD}$	I	I/O read	$\overline{IORD}$
45	RSRVD		Reserved	$\overline{IOWR}$	I	I/O write	$\overline{IOWR}$
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	—
52	VPP2 (VPP)		Programming power supply	VPP2 (VPP)		Programming/ 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

	IC Memory Card Interface			I/O Card Interface			
Pin	**Signal Name**	**I/O**	**Function**	**Signal Name**	**I/O**	**Function**	**Corresponding Pin of SH7786**
57	RSRVD		Reserved	RSRVD		Reserved	—
58	RESET	I	Reset	RESET	I	Reset	Output from port
59	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{RDY}}$*[2]
60	RSRVD		Reserved	$\overline{\text{INPACK}}$	O	Input acknowledge	—
61	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{REG}}$
62	BVD2	O	Battery voltage detection	$\overline{\text{SPKR}}$	O	Digital voice signal	Sensed on port
63	BVD1	O	Battery voltage detection	$\overline{\text{STSCHG}}$	O	Card status change	Sensed on port
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	Sensed on port
68	GND		Ground	GND		Ground	—

Notes: 1. $\overline{\text{WP}}$ is not supported.

2. Be careful of the polarity.

I/O means input/output in PCMCIA card.

The polarity of the PCMCIA card interface indicates that on the card side, and the polarity of the corresponding pin of the SH7786 indicates that on this LSI side.

11.4 Register Descriptions

Tables 11.5 and 11.6 show registers for the LBSC. These registers control the interface with each memory, wait state, etc.

Table 11.5 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size*	Sync Clock
Memory Address Map Select Register	MMSELR	R/W	H'FC40 0020	H'1C40 0020	32	SHck
Bus Control Register	BCR	R/W	H'FF80 1000	H'1F80 1000	32	Bck
CS0 Bus Control Register	CS0BCR	R/W	H'FF80 2000	H'1F80 2000	32	Bck
CS1 Bus Control Register	CS1BCR	R/W	H'FF80 2010	H'1F80 2010	32	Bck
CS2 Bus Control Register	CS2BCR	R/W	H'FF80 2020	H'1F80 2020	32	Bck
CS3 Bus Control Register	CS3BCR	R/W	H'FF80 2030	H'1F80 2030	32	Bck
CS4 Bus Control Register	CS4BCR	R/W	H'FF80 2040	H'1F80 2040	32	Bck
CS5 Bus Control Register	CS5BCR	R/W	H'FF80 2050	H'1F80 2050	32	Bck
CS6 Bus Control Register	CS6BCR	R/W	H'FF80 2060	H'1F80 2060	32	Bck
CS0 Wait Control Register	CS0WCR	R/W	H'FF80 2008	H'1F80 2008	32	Bck
CS1 Wait Control Register	CS1WCR	R/W	H'FF80 2018	H'1F80 2018	32	Bck
CS2 Wait Control Register	CS2WCR	R/W	H'FF80 2028	H'1F80 2028	32	Bck
CS3 Wait Control Register	CS3WCR	R/W	H'FF80 2038	H'1F80 2038	32	Bck
CS4 Wait Control Register	CS4WCR	R/W	H'FF80 2048	H'1F80 2048	32	Bck
CS5 Wait Control Register	CS5WCR	R/W	H'FF80 2058	H'1F80 2058	32	Bck
CS6 Wait Control Register	CS6WCR	R/W	H'FF80 2068	H'1F80 2068	32	Bck
CS5 PCMCIA Control Register	CS5PCR	R/W	H'FF80 2070	H'1F80 2070	32	Bck
CS6 PCMCIA Control Register	CS6PCR	R/W	H'FF80 2080	H'1F80 2080	32	BCk

Note: Do not access with non-specified access size.

Table 11.6 Register Configuration (2)

Register Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$ Pin/ WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction
Memory Address Map Select Register	MMSELR	H'0000 0000	H'0000 0000	Retained
Bus Control Register	BCR	H'x000 0000	Retained	Retained
CS0 Bus Control Register	CS0BCR	H'7777 77F0	Retained	Retained
CS1 Bus Control Register	CS1BCR	H'7777 77F0	Retained	Retained
CS2 Bus Control Register	CS2BCR	H'7777 77F0	Retained	Retained
CS3 Bus Control Register	CS3BCR	H'7777 77F0	Retained	Retained
CS4 Bus Control Register	CS4BCR	H'7777 77F0	Retained	Retained
CS5 Bus Control Register	CS5BCR	H'7777 77F0	Retained	Retained
CS6 Bus Control Register	CS6BCR	H'7777 77F0	Retained	Retained
CS0 Wait Control Register	CS0WCR	H'7777 770F	Retained	Retained
CS1 Wait Control Register	CS1WCR	H'7777 770F	Retained	Retained
CS2 Wait Control Register	CS2WCR	H'7777 770F	Retained	Retained
CS3 Wait Control Register	CS3WCR	H'7777 770F	Retained	Retained
CS4 Wait Control Register	CS4WCR	H'7777 770F	Retained	Retained
CS5 Wait Control Register	CS5WCR	H'7777 770F	Retained	Retained
CS6 Wait Control Register	CS6WCR	H'7777 770F	Retained	Retained
CS5 PCMCIA Control Register	CS5PCR	H'7700 0000	Retained	Retained
CS6 PCMCIA Control Register	CS6PCR	H'7700 0000	Retained	Retained

11.4.1 Memory Address Map Select Register (MMSELR)

MMSELR is a 32-bit register that selects memory address maps for areas 2 to 5. This register should be accessed at the address H'FC40 0020 in longword. To prevent incorrect writing, writing is accepted only when the upper 16-bit data is H'A5A5. The upper 29 bits are always read as 0. This register is initialized to H'0000 0000 by a power-on reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Key Code (H'A5A5)															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	Key Code	All 0	R/W	Key Code for writing to MM_SEL (bits 2 to 0). Write H'A5A5 to these bits, when writing to MM_SEL (bits 2 to 0) in this register. If key code does not match, MM_SEL is not modified. These bits are always read as 0.
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	MM_SEL	000	R/W	DDR3-SDRAM/PCI Express Memory /LRAM Space Select 000: Sets areas 2 and 3 (H'0800 0000 to H'0FFF FFFF) as the DDR3-SDRAM space, and the other areas as the local bus space. Sets H'4000 0000 to H'7FFF FFFF as the DDR3-SDRAM space, and H'8000 0000 to H'BFFF FFFF as the PCI Express space. 001: Sets areas 2 and 3 (H'0800 0000 to H'0FFF FFFF) as the DDR3-SDRAM space, area 4 (H'1000 0000 to H'13FF FFFF) as the PCI Express space, and the other areas as the local bus space. Sets H'4000 0000 to H'7FFF FFFF as the DDR3-SDRAM space, and H'8000 0000 to H'BFFF FFFF as the PCI Express space. 010: Sets areas 2 and 3 (H'0800 0000 to H'0FFF FFFF) as the DDR3-SDRAM space, area 4 (H'1000 0000 to H'13FF FFFF) as the PCI Express space, area 5 (H'1400 0000 to H'147F FFFF) as CPU0 LRAM space, area 5 (H'1480 0000 to H'14FF FFFF) as CPU1 LRAM space, and the other areas as the local bus space. Sets H'4000 0000 to H'7FFF FFFF as the DDR3-SDRAM space, and H'8000 0000 to H'BFFF FFFF as the PCI Express space. 011: Sets areas 2 to 5 (H'0800 0000 to H'17FF FFFF) as the DDR3-SDRAM space. Sets H'4000 0000 to H'7FFF FFFF as the DDR3-SDRAM space, and H'8000 0000 to H'BFFF FFFF as the PCIe space. 100: Sets area 2 to 5 (H'0800 0000 to H'17FF FFFF) as the local bus space. Sets H'4000 0000 to H'7FFF FFFF as the DDR3-SDRAM space, and H'8000 0000 to H'BFFF FFFF as the PCI Express space. 101: Sets area 4 (H'1000 0000 to H'13FF FFFF) as the PCI Express space, and the other areas as the local bus space. Sets H'4000 0000 to H'BFFF FFFF as the DDR3-SDRAM space. 110: Sets area 4 (H'1000 0000 to H'13FF FFFF) as the PCI Express space, area 5 (H'1400 0000 to H'147F FFFF) as the CPU0 LRAM space, area 5 (H'1480 0000 to H'14FF FFFF) as the CPU1 LRAM space, and the other areas as the local bus space. Sets H'4000 0000 to H'BFFF FFFF as the DDR3-SDRAM space. 111: Sets area 2 to 5 (H'0800 0000 to H'17FF FFFF) as the local bus space. Sets H'4000 0000 to H'BFFF FFFF as the DDR3-SDRAM space.

This register should be set in the initialization phase.

This register can be written only by the CPU0. The write accesses from the other modules are ignored.

Before writing to this register, set that no access will be generated from the DMAC0, DMAC2, HPB-DMAC, DU, USB, Ether or PCIEC and execute the SYNCO instruction immediately before the MOV instruction to write this register, etc. to prevent remaining unprocessed access.

Also, execute following instructions immediately after the MOV instruction to write to this register.

1. MOV instruction to read this register
2. MOV instruction to read this register
3. SYNCO instruction

Example:

```
----------------------------------------------------------------------
MOV.L   #H'FC400020, R0        ;
MOV.L   #MMSELR_DATA, R1       ; MMSELR_DATA=Writing value of MMSELR
SYNCO                          ;                  (upper word=H'A5A5)
MOV.L   R1, @R0                ; Writing to MMSELR
MOV.L    @R0, R2
MOV.L    @R0, R2
SYNCO
----------------------------------------------------------------------
```

The instruction to write to this register should be allocated to the uncacheable area P2, and to the area that is not affected by writing to this register.

This register should be set before enabling the instruction cache, operand cache, and MMU address translation, and should not be rewritten until after power-on reset is executed.

11.4.2 Bus Control Register (BCR)

BCR is a 32-bit readable/writable register that specifies the function, bus cycle state, etc for each area. BCR is initialized to H'0000 0000 in big endian mode and to H'8000 0000 in little endian mode by a power-on reset, however, not initialized by a manual reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ENDIAN	MASTER	CS0_SIZE	—	—	DPUP	—	OPUP	DACKBST[3:0]				—	—	BREQEN	DMABST
Initial value:	x*	0	x*	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

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

Note: * The initial values of bits 31 and 29 depend on the states of the external pins MODE8 and MODE7, respectively.

Bit	Bit Name	Initial Value	R/W	Description
31	ENDIAN	x	R	Endian Flag The value on the external pin (MODE8) that sets the endian mode is sampled and reflected in this bit at a power-on reset by the $\overline{\text{PRESET}}$ pin. This bit determines the endian for all spaces. 0: Indicates that a low level is on the MODE8 pin at a power-on reset and the LSI has been configured for big endian. 1: Indicates that a high level is on the MODE8 pin at a power-on reset and the LSI has been configured for little endian.
30	MASTER	0	R	Master/Slave Flag This is fixed at the master setting on the SH7786.

Bit	Bit Name	Initial Value	R/W	Description
29	CS0SIZE	X	R	Area 0 extension flag. The value on the external pin (MODE7) that sets the Area 0 extension is sampled and reflected in this bit at a power-on reset by the PRESET pin. This bit determines the LBSC CS0 area size. 0: Indicates that a low level is on the MODE7 pin at a power-on reset, and the CS0 area size is set to 128Mbytes. (CS1 is used as A26). 1: Indicates that a high level is on the MODE7 pin at a power-on reset, and CS0 area size is set to 64Mbytes.
28, 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	DPUP	0	R/W	Data Pin Pull-Up Resistor Control Specifies the pull-up resistor state of the data pins (D31 to D0). This bit is initialized at a power-on reset. The pins are not pulled up when access is performed or when the bus is released, even if the pull-up resistor is on. 0: Some cycles of the pull-up resistors of the data pins (D31 to D0) are turned on before and after a memory access.* 1: Pull-up resistors of the data pins (D31 to D0) are off. Note: * When data pin pull-up is necessary, it is recommended to connect a pull-up resistor externally.
25	—	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
24	OPUP	0	R/W	Control Output Pin Pull-Up Resistor Control Specifies the pull-up resistor state (A25 to A0, $\overline{BS}$, $\overline{CS}$n, $\overline{RD}$, $\overline{WE}$, R/$\overline{W}$, $\overline{CE2A}$, and $\overline{CE2B}$) when the control output pins are high-impedance. This bit is initialized at a power-on reset. 0: Pull-up resistors for control output pins (A25 to A0, $\overline{BS}$, $\overline{CS}$n, $\overline{RD}$, $\overline{WE}$, R/$\overline{W}$, $\overline{CE2A}$, and $\overline{CE2B}$) are on 1: Pull-up resistors for control output pins (A25 to A0, $\overline{BS}$, $\overline{CS}$n, $\overline{RD}$, $\overline{WE}$, R/$\overline{W}$, $\overline{CE2A}$, and $\overline{CE2B}$) are off
23 to 20	DACKBST [3:0]	All 0	R/W	DACKBST3 to DACKBST0 0: $\overline{DACK}$n signals asserted in synchronization with the bus cycle (n = 0 to 3) 1: $\overline{DACK}$n signals remain asserted from the start to the end of burst transfer when DMA transfer is performed in burst mode These bits can be set to 1 only when the memory type of the DACK output area in the corresponding DMA transfer channel is set to PCMCIA interface. In other cases, these bits should be cleared to 0. The pins corresponding to each bits are as follows. DACKBST[3]: $\overline{DACK3}$ DACKBST[2]: $\overline{DACK2}$ DACKBST[1]: $\overline{DACK1}$ DACKBST[0]: $\overline{DACK0}$
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	BREQEN	0	R/W	$\overline{BREQ}$ Enable Specifies whether an external request can be accepted or not. In the initialized state at a power-on reset, an external request is not accepted. 0: An external request is not accepted 1: An external request is accepted

Bit	Bit Name	Initial Value	R/W	Description
16	DMABST	0	R/W	DMAC Burst Mode Transfer Priority Setting Specifies the priority of burst mode transfers by DMA channels 0 to 5. When this bit is cleared to 0, the bus is released and the SuperHyway is the bus master. When this bit is set to 1, the bus is not released until completion of the DMAC burst transfer. This bit is initialized at a power-on reset. 0: DMAC burst mode transfer priority setting is off 1: DMAC burst mode transfer priority setting is on
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	HIZCNT	0	R/W	High Impedance (Hi-Z) Control Specifies the state of signals $\overline{\text{WEn}}$ and $\overline{\text{RD}}$/$\overline{\text{FRAME}}$ in the bus-released state. 0: Signals $\overline{\text{WEn}}$ and $\overline{\text{RD}}$/$\overline{\text{FRAME}}$ are high-impedance in the bus-released state 1: Signals $\overline{\text{WEn}}$ and $\overline{\text{RD}}$/$\overline{\text{FRAME}}$ are driven in the bus-released state
13 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	ASYNC[6:0]	All 0	R/W	Asynchronous Input These bits enable asynchronous inputs of the corresponding pins. 0: CLKOUT-synchronous inputs to the corresponding pins 1: CLKOUT-asynchronous inputs to the corresponding pins ASYNC[6]: $\overline{\text{DREQ3}}$ ASYNC[5]: $\overline{\text{DREQ2}}$ ASYNC[4]: $\overline{\text{DREQ1}}$ ASYNC[3]: $\overline{\text{DREQ0}}$ ASYNC[2]: $\overline{\text{IOIS16}}$ ASYNC[1]: $\overline{\text{BREQ}}$ ASYNC[0]: $\overline{\text{RDY}}$

When asynchronous input is set (ASYNCn = 1), the sampling timing is one cycle before the synchronous input setting (see figure 11.4).

The timing shown in this section and section 34 Electrical Characteristics is that with synchronous input setting (ASYNCn = 0). Note that the setup/hold time should be satisfied when synchronous input is set.

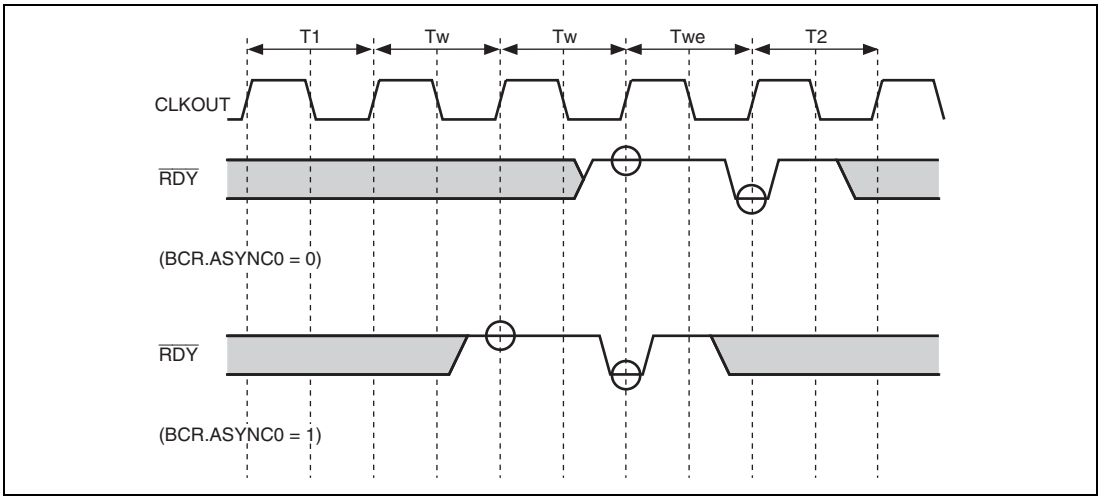


Figure 11.4 $\overline{\text{RDY}}$ Sampling Timings with ASYNCn Settings (Two Wait Cycles Inserted by CSnWCR.)

11.4.3 CSn Bus Control Register (CSnBCR)

CSnBCR are 32-bit readable/writable registers that specify the bus width for area n (n = 0 to 6), idle mode between cycles, burst ROM setting and memory types.

Some types of memory continue to drive the data bus immediately after the read signal is turned off. Therefore, data buses may collide with each other when different memory areas are accessed consecutively or memory writing is performed immediately after it is read. This LSI automatically inserts idle cycles as specified with CSnBCR when the data buses may collide. In the idle cycles, $\overline{CSn}$, $\overline{RD}$, $\overline{WEn}$, $\overline{CE2A}$, $\overline{CE2B}$, $\overline{BS}$ and R/$\overline{W}$ are set to high, and the data bus is not driven.

CSnBCR is initialized to H'7777 77F0 at a power-on reset, but is not initialized by a manual reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	IWW			—	IWRWD			—	IWRWS			—	IWRRD		
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	IWRRS			BST		SZ*		RDSPL	BW			MPX*	TYPE		
Initial value:	0	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W*	R/W*	R/W	R/W	R/W	R/W	R/W*	R/W	R/W	R/W

Note: * Bits SZ and MPX in CS0BCR are read-only.

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	IWW	111	R/W	Idle Cycles between Write-Read/Write-Write These bits specify the number of idle cycles to be inserted after the access of the memory connected to the space. The target cycles are write-read and write-write cycles. For details, see section 11.5.8, Wait Cycles between Access Cycles. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 24	IWRWD	111	R/W	Idle Cycles between Read-Write in Different Spaces These bits specify the number of idle cycles to be inserted after the access of the memory connected to the space. The target cycles are read-write cycles in which consecutive accesses are performed to different spaces. For details, see section 11.5.8, Wait Cycles between Access Cycles. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 20	IWRWS	111	R/W	Idle Cycles between Read-Write in Same Space These bits specify the number of idle cycles to be inserted after the access to the memory connected to the space. The target cycles are read-write cycles in which consecutive accesses are performed to the same space. For details, see section 11.5.8, Wait Cycles between Access Cycles. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	IWRRD	111	R/W	Idle Cycles between Read-Read in Different Spaces These bits specify the number of idle cycles to be inserted after the memory connected to the space is accessed. The target cycles are read-read cycles in which consecutive accesses are performed to different spaces. For details, see section 11.5.8, Wait Cycles between Access Cycles. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	IWRRS	111	R/W	Idle Cycles between Read-Read in Same Space These bits specify the number of idle cycles to be inserted after the memory connected to the space is accessed. The target cycles are read-read cycles in which consecutive accesses are performed to the same space. For details, see section 11.5.8, Wait Cycles between Access Cycles. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
11, 10	BST	01	R/W	Burst Number These bits specify the number of bursts when the burst ROM interface is used. The MPX interface is not affected. 00: 4 consecutive accesses (Can be used with 8-, 16-, or 32-bit bus width) 01: 8 consecutive accesses (Can be used with 8-, 16-, or 32-bit bus width) 10: 16 consecutive accesses (Can be used with 8-, or 16-bit bus width) 11: 32 consecutive accesses (Can be used with 8-bit bus width)
9, 8	SZ	11	R/W*	Bus Width In CS0BCR, the external pins (MODE4, MODE5, and MODE6) to specify the bus size are sampled at a power-on reset. When using the MPX interface, set these bits to 00 or 11 to specify 32 bits. When using the byte control SRAM interface, set these bits to 00, 10 or 11 to specify 16 or 32 bits. 00: 32 bits 01: 8 bits 10: 16 bits 11: 32 bits Note: * The SZ bits in CS0BCR are read-only. When area 0 is set to the MPX interface by the MODE4, MODE5, and MOD6 pin, the SZ bits in CS0BCR should be set to 00 or 11.
7	RDSPL	0	R/W	$\overline{\text{RD}}$ Hold Cycle Specifies the number of cycles to be inserted in the hold time for the read data sample timing of $\overline{\text{RD}}$. When setting this bit to 1, specify the number of $\overline{\text{RD}}$ negation-$\overline{\text{CS}}$n negation delay cycles set by the RDH bit in CSnWCR as 1 or more. Also the $\overline{\text{RD}}$ negation-$\overline{\text{CS}}$n negation delay cycle is reduced 1 cycle when this bit is set to 1 (valid only when the SRAM interface, burst ROM interface, or byte control SRAM interface is selected). 0: No hold cycle inserted 1: 1 hold cycle inserted

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	BW	111	R/W	Burst Pitch When the burst ROM interface is used, these bits specify the number of wait cycles to be inserted after the second data access in a burst transfer. 000: No idle cycle inserted, $\overline{\text{RDY}}$ pin disabled 001: 1 idle cycle inserted, $\overline{\text{RDY}}$ pin enabled 010: 2 idle cycles inserted, $\overline{\text{RDY}}$ pin enabled 011: 3 idle cycles inserted, $\overline{\text{RDY}}$ pin enabled 100: 4 idle cycles inserted, $\overline{\text{RDY}}$ pin enabled 101: 5 idle cycles inserted, $\overline{\text{RDY}}$ pin enabled 110: 6 idle cycles inserted, $\overline{\text{RDY}}$ pin enabled 111: 7 idle cycles inserted, $\overline{\text{RDY}}$ pin enabled
3	MPX	0	R/W*	MPX Interface Setting Selects the type of the MPX interface 0: Memory type set by bits TYPE2 to TYPE0 is selected 1: MPX interface is selected Note: * The MPX bit in CS0BCR is read-only.
2 to 0	TYPE	000	R/W	Memory Type Setting These bits specify the type of memory connected to the space. 000: SRAM (Initial value) 001: SRAM with byte-control*[1] 010: Burst ROM (burst at read/SRAM at write) 011: Reserved (Setting prohibited) 100: PCMCIA *[2] 101: Reserved (Setting prohibited) 110: Reserved (Setting prohibited) 111: Reserved (Setting prohibited) Notes: 1. Setting enabled only in CS1BCR and CS4BCR 2. Setting enabled only in CS5BCR and CS6BCR

11.4.4 CSn Wait Control Register (CSnWCR)

CSnWCR (n = 0 to 6) are 32-bit readable/writable registers that specify the number of wait cycles to be inserted for areas 0 to 6, the number of wait cycles to be inserted preceding the first data in burst memory access, the address setup time, which is the time from the point at which the output of address for access is started until assertion of the read/write strobe signal, and the number of cycles to be inserted as the data hold time from negation of the write strobe signal.

CSnWCR is initialized to H'7777 770F by a power-on reset, but it is not initialized by a manual reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ADS			—	ADH			—	RDS			—	RDH		
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	WTS			—	WTH			—	BSH			IW[3:0]			
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	1	1	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	ADS	111	R/W	Address Setup Cycle These bits specify the number of cycles to be inserted as the address setup time with respect to $\overline{\text{CSn}}$ assertion. (Only valid when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted
27	—	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
26 to 24	ADH	111	R/W	Address Hold Cycle These bits specify the number of cycles to be inserted as the address hold time with respect to $\overline{CSn}$ negation. (Only valid when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 20	RDS	111	R/W	$\overline{RD}$ Setup Cycle ($\overline{CSn}$ Assertion–$\overline{RD}$ Assertion Delay Cycle) These bits specify the number of cycles to be inserted as the time from $\overline{CSn}$ assertion to $\overline{RD}$ assertion. (Only valid only when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted (1 cycle delayed) 001: 1 cycle inserted (2 cycles delayed) 010: 2 cycles inserted (3 cycles delayed) 011: 3 cycles inserted (4 cycles delayed) 100: 4 cycles inserted (5 cycles delayed) 101: 5 cycles inserted (6 cycles delayed) 110: 6 cycles inserted (7 cycles delayed) 111: 7 cycles inserted (8 cycles delayed)
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	RDH	111	R/W	$\overline{RD}$ Hold Cycle ($\overline{RD}$ Negation–$\overline{CSn}$ Negation Delay Cycle) These bits specify the number of cycles to be inserted as the time from $\overline{RD}$ negation to $\overline{CSn}$ negation. (Only valid when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted (0 cycles delayed) 001: 1 cycle inserted (1 cycle delayed) 010: 2 cycles inserted (2 cycles delayed) 011: 3 cycles inserted (3 cycles delayed) 100: 4 cycles inserted (4 cycles delayed) 101: 5 cycles inserted (5 cycles delayed) 110: 6 cycles inserted (6 cycles delayed) 111: 7 cycles inserted (7 cycles delayed)
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	WTS	111	R/W	$\overline{WE}$ Setup Cycle ($\overline{CSn}$ Assertion–$\overline{WEn}$ Assertion Delay Cycle) These bits specify the number of cycles to be inserted as the time from $\overline{CSn}$ assertion to $\overline{WE}$ assertion. (Only valid when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted (0.5 cycle delayed) 001: 1 cycle inserted (1.5 cycles delayed) 010: 2 cycles inserted (2.5 cycles delayed) 011: 3 cycles inserted (3.5 cycles delayed) 100: 4 cycles inserted (4.5 cycles delayed) 101: 5 cycles inserted (5.5 cycles delayed) 110: 6 cycles inserted (6.5 cycles delayed) 111: 7 cycles inserted (7.5 cycles delayed)
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 to 8	WTH	111	R/W	$\overline{WE}$ Hold Cycle ($\overline{WEn}$ Negation–$\overline{CSn}$ Negation Delay Cycle) These bits specify the number of cycles to be inserted as the time from $\overline{WEn}$ negation to $\overline{CSn}$ negation. (Only valid when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted (0.5 cycle delayed) 001: 1 cycle inserted (1.5 cycles delayed) 010: 2 cycles inserted (2.5 cycles delayed) 011: 3 cycles inserted (3.5 cycles delayed) 100: 4 cycles inserted (4.5 cycles delayed) 101: 5 cycles inserted (5.5 cycles delayed) 110: 6 cycles inserted (6.5 cycles delayed) 111: 7 cycles inserted (7.5 cycles delayed)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	BSH	000	R/W	$\overline{BS}$ Hold Cycle These bits specify the number of cycles to extend $\overline{BS}$ assertion. The extension of the assertion is valid when the RDS bits in CSnWCR are not set to 000 in reading and when the WTS bits in CSnWCR are not set to 000 in writing. The total access cycle count is not changed by setting these bits. 000: $\overline{BS}$ assertion is 1 cycle 001: $\overline{BS}$ assertion is 2 cycles 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	IW[3:0]	1111	R/W	Insert Wait Cycle These bits specify the number of wait cycles to be inserted. The wait cycles are as follows when the SRAM interface, byte control SRAM interface, burst ROM interface (first data cycle only), or PCMCIA interface is selected. Insertion of external wait cycles by the $\overline{\text{RDY}}$ pin is not possible when "no cycle inserted" is selected. 0000: No cycle inserted 1000: 8 cycles inserted 0001: 1 cycle inserted 1001: 9 cycles inserted 0010: 2 cycles inserted 1010: 11 cycles inserted 0011: 3 cycles inserted 1011: 13 cycles inserted 0100: 4 cycles inserted 1100: 15 cycles inserted 0101: 5 cycles inserted 1101: 17 cycles inserted 0110: 6 cycles inserted 1110: 21 cycles inserted 0111: 7 cycles inserted 1111: 25 cycles inserted When MPX interface is selected, the wait cycles are inserted as follows according to the IW[2:0] setting. The IW[3] setting is invalid. The external wait cycles can be inserted by the $\overline{\text{RDY}}$ pin in any settings. IW[2] specifies wait cycle insertion for the second and subsequent data: 0: No cycle inserted 1: 1 cycle inserted IW[1:0] specifies wait cycle insertion for the first data: 00: 1 cycle inserted in reading and no cycle inserted in writing 01: 1 cycle inserted in reading and 1 cycle inserted in writing 10: 2 cycles inserted in reading and 2 cycles inserted in writing 11: 3 cycles inserted in reading and 3 cycles inserted in writing

11.4.5 CSn PCMCIA Control Register (CSnPCR)

CSnPCR is a 32-bit readable/writable register that specifies the timing for the PCMCIA interface connected to area n (CSnPCR, n = 5 or 6), the space property, and the assert/negate timing for the $\overline{OE}$ and $\overline{WE}$ signals. Also, areas 5 and 6 in CSnPCR can be set for the first half and second half individually. The first half of area 5 is allocated from H'1400 0000 to H'15FF FFFF, and the second half of area 5 is allocated from H'1600 0000 to H'17FF FFFF. The first half of area 6 is allocated from H'1800 0000 to H'19FF FFFF, and the second half of area 6 is allocated from H'1A00 0000 to H'1BFF FFFF (these addresses are the local bus address). The pulse widths of $\overline{OE}$ and $\overline{WE}$ assertion for the first half of area 5 and 6 are set by the IW bits in CSnWCR.

CSnPCR is initialized to H'7700 0000 by a power-on reset, but it is not initialized by a manual reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	SAA			—	SAB			PCWA		PCWB		PCIW			
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	SAA	111	R/W	Space Property A These bits specify the space property of PCMCIA connected to the first half of the area. 000: ATA complement mode 001: Dynamic I/O bus sizing 010: 8-bit I/O space 011: 16-bit I/O space 100: 8-bit common memory 101: 16-bit common memory 110: 8-bit attribute memory 111: 16-bit attribute memory
27	—	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
26 to 24	SAB	111	R/W	Space Property B These bits specify the space property of PCMCIA connected to the second half of the area. 000: ATA complement mode 001: Dynamic I/O bus sizing 010: 8-bit I/O space 011: 16-bit I/O space 100: 8-bit common memory 101: 16-bit common memory 110: 8-bit attribute memory 111: 16-bit attribute memory
23, 22	PCWA	00	R/W	PCMCIA Wait A These bits specify the number of wait cycles for low-speed PCMCIA, which is added to the number set by the IW bits in CSnWCR. The bit settings are selected when the access area of PCMCIA interface is the first half. 00: No wait cycle inserted 01: 15 wait cycles inserted 10: 30 wait cycles inserted 11: 50 wait cycles inserted
21, 20	PCWB	00	R/W	PCMCIA Wait B These bits specify the number of wait cycles for low-speed PCMCIA, which is added to the number set by the PCIW bits. The bit settings are selected when the access area of PCMCIA interface is the second half. 00: No wait cycle inserted 01: 15 wait cycles inserted 10: 30 wait cycles inserted 11: 50 wait cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
19 to 16	PCIW	0000	R/W	PCMCIA Insert Wait Cycle B These bits specify the number of wait cycles to be inserted. The bit settings are selected when the access area of PCMCIA interface is the second half. When the access area of PCMCIA interface is the first half, the number of wait cycles set by the IW bit in CSnWCR is selected. 0000: No cycle inserted 0001: 1 cycle inserted 0010: 2 cycles inserted 0011: 3 cycles inserted 0100: 4 cycles inserted 0101: 5 cycles inserted 0110: 6 cycles inserted 0111: 7 cycles inserted 1000: 8 cycles inserted 1001: 9 cycles inserted 1010: 11 cycles inserted 1011: 13 cycles inserted 1100: 15 cycles inserted 1101: 17 cycles inserted 1110: 21 cycles inserted 1111: 25 cycles inserted
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	TEDA	000	R/W	$\overline{OE}$/$\overline{WE}$ Assert Delay A These bits set the delay time from address output to $\overline{OE}$/$\overline{WE}$ assertion when the first half area is accessed with the connected PCMCIA interface. 000: No wait cycle inserted 001: 1 wait cycle inserted 010: 2 wait cycles inserted 011: 3 wait cycles inserted 100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	TEDB	000	R/W	$\overline{OE}$/$\overline{WE}$ Assert Delay B These bits set the delay time from address output to $\overline{OE}$/$\overline{WE}$ assertion when the second half area is accessed with the connected PCMCIA interface. 000: No wait cycle inserted 001: 1 wait cycle inserted 010: 2 wait cycles inserted 011: 3 wait cycles inserted 100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	TEHA	000	R/W	$\overline{OE}$/$\overline{WE}$ Negation-Address Delay A These bits set the delay time from $\overline{OE}$/$\overline{WE}$ negation to address hold when the first half area is accessed with the connected PCMCIA interface. 000: No wait cycle inserted 001: 1 wait cycle inserted 010: 2 wait cycles inserted 011: 3 wait cycles inserted 100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted
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 to 0	TEHB	000	R/W	$\overline{OE}$/$\overline{WE}$ Negation-Address Delay B These bits set the delay time from $\overline{OE}$/$\overline{WE}$ negation to address hold when the second half area is accessed with the connected PCMCIA interface. 000: No wait cycle inserted 001: 1 wait cycle inserted 010: 2 wait cycles inserted 011: 3 wait cycles inserted 100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted

11.5 Operation

11.5.1 Endian/Access Size and Data Alignment

This LSI supports both big and little endian modes. In big endian mode, the most significant byte (MSByte) in a string of byte data is stored at address 0, and in little endian mode, the least significant byte (LSByte) in a string of byte data is stored at address 0. The mode is specified by the external pin (MODE 8 pin) at a power-on reset by the $\overline{\text{PRESET}}$ pin. At a power-on reset by the $\overline{\text{PRESET}}$ pin, big endian mode is specified when the MODE8 pin is low, and little endian mode is specified when the MODE8 pin is high.

The data bus width can be selected from 8, 16 and 32 bits for the normal memory interface. For the PCMCIA interface, a data bus width of 8 or 16 bits can be selected. Data is aligned according to the data bus width and endian mode of each device. Therefore, when the data bus width is smaller than the access size, multiple bus cycles are automatically generated to reach the access size. In this case, access is performed by incrementing the addresses corresponding to the bus width. For example, when a longword access is performed in the area with an 8-bit width with the SRAM interface, each address is incremented by one, and accesses are performed four times. In the 32-byte transfer, a total of 32-byte data is continuously transferred according to the specified bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed on the subsequent data up to the nearest 32-byte boundary in a wraparound manner. The bus is not released during these transfers. This LSI automatically aligns data and changes the data length between interfaces.

In an 8- or 16-byte transfer, the LBSC executes the 4-byte accesses twice and four times respectively.

Tables 11.7 to 11.12 show the relationship between the device data width, endian mode and access size.

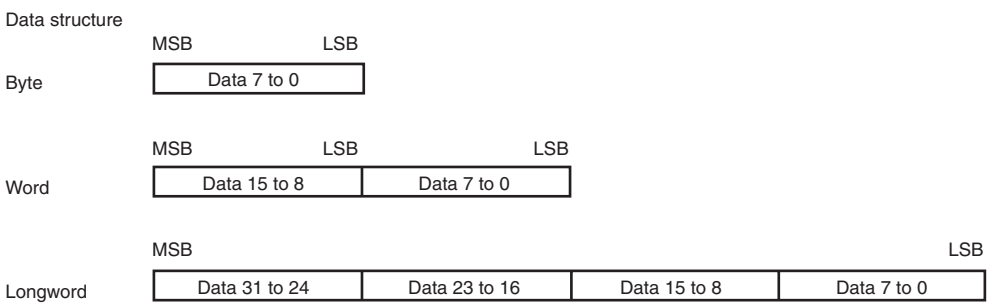


Table 11.7 32-Bit External Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	4n	1	Data 7 to 0	—	—	—	Asserted			
	4n + 1	1	—	Data 7 to 0	—	—		Asserted		
	4n + 2	1	—	—	Data 7 to 0	—			Asserted	
	4n + 3	1	—	—	—	Data 7 to 0				Asserted
Word	4n	1	Data 15 to 8	Data 7 to 0	—	—	Asserted	Asserted		
	4n + 2	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
Longword	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Asserted	Asserted	Asserted	Asserted
32 Bytes*	8n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Asserted	Asserted	Asserted	Asserted
	8n + 4	2	Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32	Asserted	Asserted	Asserted	Asserted
	8n + 8	3	Data 95 to 88	Data 87 to 80	Data 79 to 72	Data 71 to 64	Asserted	Asserted	Asserted	Asserted
	…	…	…	…	…	…	…	…	…	…
	8n + 28	8	Data 255 to 248	Data 247 to 240	Data 239 to 232	Data 231 to 224	Asserted	Asserted	Asserted	Asserted

Note: * This table shows an example when the access start address is on the 32-byte boundary. When the start address is not on the 32-byte boundary, accesses are performed up to immediately before the 32-byte boundary and the address is wrapped around to the previous 32-byte boundary.

Table 11.8 16-Bit External Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	2n	1	—	—	Data 7 to 0	—			Asserted	
	2n + 1	1	—	—	—	Data 7 to 0				Asserted
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
Longword	4n	1	—	—	Data 31 to 24	Data 23 to 16			Asserted	Asserted
	4n + 2	2	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
32 Bytes*	8n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
	8n + 2	2	—	—	Data 31 to 24	Data 23 to 16			Asserted	Asserted
	8n + 4	3	—	—	Data 47 to 40	Data 39 to 32			Asserted	Asserted
	…	…	…	…	…	…	…	…	…	…
	8n + 30	16	—	—	Data 255 to 248	Data 247 to 240			Asserted	Asserted

Note: * This table shows an example when the access start address is on the 32-byte boundary. When the start address is not on the 32-byte boundary, accesses are performed up to immediately before the 32-byte boundary and the address is wrapped around to the previous 32-byte boundary.

Table 11.9 8-Bit External Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	n	1	—	—	—	Data 7 to 0				Asserted
Word	2n	1	—	—	—	Data 15 to 8				Asserted
	2n + 1	2	—	—	—	Data 7 to 0				Asserted
Longword	4n	1	—	—		Data 31 to 24				Asserted
	4n + 1	2	—	—	—	Data 23 to 16				Asserted
	4n + 2	3	—	—	—	Data 15 to 8				Asserted
	4n + 3	4	—	—	—	Data 7 to 0				Asserted
32 Bytes*	8n	1	—	—	—	Data 7 to 0				Asserted
	8n + 1	2	—	—	—	Data 15 to 8				Asserted
	8n + 2	3	—	—	—	Data 23 to 16				Asserted
	…	…	…	…	…	…	…	…	…	…
	8n + 31	32	—	—	—	Data 255 to 248				Asserted

Note: * This table shows an example when the access start address is on the 32-byte boundary. When the start address is not on the 32-byte boundary, accesses are performed up to immediately before the 32-byte boundary and the address is wrapped around to the previous 32-byte boundary.

Table 11.10 32-Bit External Device/Little-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	4n	1	—	—	—	Data 7 to 0				Asserted
	4n + 1	1	—	—	Data 7 to 0	—			Asserted	
	4n + 2	1	—	Data 7 to 0	—	—		Asserted		
	4n + 3	1	Data 7 to 0	—	—	—	Asserted			
Word	4n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
	4n + 2	1	Data 15 to 8	Data 7 to 0	—	—	Asserted	Asserted		
Longword	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Asserted	Asserted	Asserted	Asserted
32 Bytes*	8n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Asserted	Asserted	Asserted	Asserted
	8n + 4	2	Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32	Asserted	Asserted	Asserted	Asserted
	8n + 8	3	Data 95 to 88	Data 87 to 80	Data 79 to 72	Data 71 to 64	Asserted	Asserted	Asserted	Asserted
	…	…	…	…	…	…	…	…	…	…
	8n + 28	8	Data 255 to 248	Data 247 to 240	Data 239 to 232	Data 231 to 224	Asserted	Asserted	Asserted	Asserted

Note: * This table shows an example when the access start address is on the 32-byte boundary. When the start address is not on the 32-byte boundary, accesses are performed up to immediately before the 32-byte boundary and the address is wrapped around to the previous 32-byte boundary.

Table 11.11 16-Bit External Device/Little-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	2n	1	—	—	—	Data 7 to 0				Asserted
	2n + 1	1	—	—	Data 7 to 0	—			Asserted	
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
Longword	4n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
	4n + 2	2	—	—	Data 31 to 24	Data 23 to 16			Asserted	Asserted
32 Bytes*	8n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
	8n + 2	2	—	—	Data 31 to 24	Data 23 to 16			Asserted	Asserted
	8n + 4	3	—	—	Data 47 to 40	Data 39 to 32			Asserted	Asserted
	…	…	…	…	…	…	…	…	…	…
	8n + 30	16	—	—	Data 255 to 248	Data 247 to 240			Asserted	Asserted

Note: * This table shows an example when the access start address is on the 32-byte boundary. When the start address is not on the 32-byte boundary, accesses are performed up to immediately before the 32-byte boundary and the address is wrapped around to the previous 32-byte boundary.

Table 11.12 8-Bit External Device/Little-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	n	1	—	—	—	Data 7 to 0				Asserted
Word	2n	1	—	—	—	Data 7 to 0				Asserted
	2n + 1	2	—	—	—	Data 15 to 8				Asserted
Longword	4n	1	—	—	—	Data 7 to 0				Asserted
	4n + 1	2	—	—	—	Data 15 to 8				Asserted
	4n + 2	3	—	—	—	Data 23 to 16				Asserted
	4n + 3	4	—	—	—	Data 31 to 24				Asserted
32 Bytes*	8n	1	—	—	—	Data 7 to 0				Asserted
	8n + 1	2	—	—	—	Data 15 to 8				Asserted
	8n + 2	3	—	—	—	Data 23 to 16				Asserted
	…	…	…	…	…	…	…	…	…	…
	8n + 31	32	—	—	—	Data 255 to 248				Asserted

Note: * This table shows an example when the access start address is on the 32-byte boundary. When the start address is not on the 32-byte boundary, accesses are performed up to immediately before the 32-byte boundary and the address is wrapped around to the previous 32-byte boundary.

11.5.2 Areas

(1) Area 0

Area 0 is an area where bits 28 to 26 in the local bus address are 000.

The interface that can be set for this area is the SRAM, burst ROM or MPX interface.

A bus width of 8, 16, or 32 bits is selectable by external pins MODE6, MODE5, and MODE4 at a power-on reset. For details, see section 11.3.2, Memory Bus Width.

The size of area 0 can be selected from 128Mbytes and 64Mbytes, by MODE7 pin at a power-on reset. For details, see section 11.4.2, Bus Control Register (BCR).

When 128Mbytes is selected for area 0 size, $\overline{CS1}$ pin is used as A26.

When area 0 is accessed, the $\overline{CS0}$ signal is asserted. In addition, the $\overline{RD}$ signal, which can be used as $\overline{OE}$, and write control signals $\overline{WE0}$ to $\overline{WE3}$ are asserted.

For the number of bus cycles, 0 to 25 wait cycles to be inserted can be selected with CS0WCR. When the burst ROM interface is used, the number of a burst pitch is selectable in the range from 0 to 7 with the BW bits in CS0BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{RDY}$). (when the number of inserted cycles is set to 0, the $\overline{RDY}$ signal is ignored.)

When the burst ROM interface is used, the number of transfer cycles for a burst cycle is selected in the range from 2 to 9 according to the number of wait cycles.

The setup/hold cycle of the address, the assert delay cycle of the read/write strobe signals for $\overline{CS0}$ assertion and the $\overline{CS0}$ negate delay cycle for the read/write strobe signals negation can be set in the range from 0 to 7 cycles by CS0WCR. The $\overline{BS}$ hold cycles can be set to 1 or 2 when the RDS bits in CS0WCR are not 000 in reading and the WTS bits in CS0WCR are not 000 in writing.

(2) Area 1

Area 1 is an area where bits 28 to 26 in the local bus address are 001.

The interface that can be set for this area is the SRAM, burst ROM, MPX and byte-control SRAM interface.

The bus width can be selected from 8, 16, and 32 bits by bits SZ in CS1BCR. When the MPX interface is used, the bus width should be set to 32 bits with bits SZ in CS1BCR. When the byte control SRAM interface is used, the bus width should be set to 16 or 32 bits.

When area 1 is accessed, the $\overline{CS1}$ signal is asserted. The $\overline{RD}$ signal, that can be used as $\overline{OE}$, and write control signals $\overline{WE0}$ to $\overline{WE3}$ are also asserted.

For the number of bus cycles, 0 to 25 wait cycles to be inserted can be selected by CS1WCR.

When the burst ROM interface is used, the number of a burst pitch is selectable in the range from 0 to 7 with the BW bits in CS1BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{RDY}$). (when no cycles are inserted, the $\overline{RDY}$ signal is ignored.)

The setup/hold time of the address, the assert delay cycle of the read/write strobe signals for $\overline{CS1}$ assertion and the $\overline{CS1}$ negate delay cycle for the read/write strobe signals negation can be set in the range from 0 to 7 cycles by CS1WCR. The $\overline{BS}$ hold cycles can be set to 1 or 2 when the RDS bits in CS1WCR are not 000 in reading and the WTS bits in CS1WCR are not 000 in writing.

(3) Area 2

Area 2 is an area where bits 28 to 26 in the local bus address are 010.

This area can be used as a interface for SRAM, MPX, or Burst ROM.

The bus width can be selected from 8, 16, and 32 bits, by bits SZ in CS2BCR. When the MPX interface is used, the bus width should be set to 32 bits.

When area 2 is accessed, the $\overline{CS2}$ signal is asserted. If area 2 is used as a SRAM interface, $\overline{RD}$ signal, which can be used as $\overline{OE}$, and write control signals $\overline{WE0}$ to $\overline{WE3}$ are also asserted.

For the bus cycle number, 0 to 25 wait cycles to be inserted can be selected by BW bits in CS2BCR. When Burst ROM interface is used, the number of a burst pitch is selectable in the range from 0 to 7, by the BW bits in CS2BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin (RDY). (when no cycles are inserted, the RDY signal is ignored.)

The setup/hold time of the address, the assert delay cycle of the read/write strobe signals for $\overline{CS2}$ assertion and the $\overline{CS2}$ negate delay cycle for the read/write strobe signals negation can be set in the range from 0 to 7 cycles by CS2WCR. The $\overline{BS}$ hold cycles can be set to 1 or 2 when the RDS bits in CS2WCR are not 000 in reading and the WTS bits in CS2WCR are not 000 in writing.

(4) Area 3

Area 3 is an area where bits 28 to 26 in the local bus address are 011.

This area can be used as a interface for SRAM, MPX, or Burst ROM.

The bus width can be selected from 8, 16, and 32 bits, by bits SZ in CS3BCR. When the MPX interface is used, the bus width should be set to 32 bits.

When area 3 is accessed, the $\overline{CS3}$ signal is asserted. If area 3 is used as a SRAM interface, $\overline{RD}$ signal, which can be used as $\overline{OE}$, and write control signals $\overline{WE0}$ to $\overline{WE3}$ are also asserted.

For the bus cycle number, 0 to 25 wait cycles to be inserted can be selected by BW bits in CS3BCR. When Burst ROM interface is used, the number of a burst pitch is selectable in the range from 0 to 7, by the BW bits in CS3BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin (RDY). (when no cycles are inserted, the RDY signal is ignored.)

The setup/hold time of the address, the assert delay cycle of the read/write strobe signals for $\overline{CS3}$ assertion and the $\overline{CS3}$ negate delay cycle for the read/write strobe signals negation can be set in the range from 0 to 7 cycles by CS3WCR. The $\overline{BS}$ hold cycles can be set to 1 or 2 when the RDS bits in CS3WCR are not 000 in reading and the WTS bits in CS3WCR are not 000 in writing.

(5) Area 4

Area 4 is an area where bits 28 to 26 in the local bus address are 100.

The interface that can be set for this area is the SRAM, MPX, byte control RAM, and burst ROM interface.

A bus width of 8, 16, or 32 bits is selectable by bits SZ in CS4BCR. When the MPX interface is used, a bus width of 32 bits should be selected through bits SZ in CS4BCR. When the byte control SRAM interface is used, select a bus width of 16 or 32 bits. For details, see section 11.3.2, Memory Bus Width.

When area 4 is accessed, the $\overline{CS4}$ signal is asserted.

When the SRAM interface is set, the $\overline{RD}$ signal, which can be used as $\overline{OE}$, and write control signals $\overline{WE0}$ to $\overline{WE3}$ are asserted. For details, see section 11.5.8, Wait Cycles between Access Cycles.

For the number of bus cycles, 0 to 25 wait cycles inserted can be selected by CS4WCR.

When the burst ROM interface is used, the number of a burst pitch is selectable in the range from 0 to 7 with the BW bits in CS4BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{\text{RDY}}$). (when no cycles are inserted, the $\overline{\text{RDY}}$ signal is ignored.)

The setup/hold time of the address, the assert delay cycle of the read/write strobe signals for $\overline{\text{CS4}}$ assertion and the $\overline{\text{CS4}}$ negate delay cycle for the read/write strobe signals negation can be set in the range from 0 to 7 cycles by CS4WCR. The $\overline{\text{BS}}$ hold cycles can be set to 1 or 2 when the RDS bits in CS4WCR are not 000 in reading and the WTS bits in CS4WCR are not 000 in writing.

(6) Area 5

Area 5 is an area where bits 28 to 26 in the local bus address are 101.

When the SRAM or burst ROM interface is used, a bus width of 8, 16, or 32 bits is selectable by bits SZ in CS5BCR. When the MPX interface is used, a bus width of 32 bits should be selected by bits SZ in CS5BCR. When the PCMCIA interface is used, select a bus width of 8 or 16 bits with SZ in CS5BCR. For details, see section 11.3.2, Memory Bus Width.

While the SRAM interface is used, the $\overline{\text{CS5}}$ signal is asserted when area 5 is accessed. The $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are also asserted. While the PCMCIA interface is used, the $\overline{\text{CE1A}}$ and $\overline{\text{CE2A}}$ signals, the $\overline{\text{RD}}$ signal, (which can be used as $\overline{\text{OE}}$), the $\overline{\text{WE0}}$, $\overline{\text{WE1}}$, $\overline{\text{WE2}}$, and $\overline{\text{WE3}}$ signals, (which can be used as, $\overline{\text{REG}}$, $\overline{\text{WE}}$, $\overline{\text{IORD}}$, and $\overline{\text{IOWR}}$, respectively) are asserted.

For the number of bus cycles, 0 to 25 wait cycles inserted by CS5WCR can be selected.

When the burst ROM interface is used, the number of a burst pitch is selectable the range from 0 to 7 with the BW bits in CS5BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{\text{RDY}}$). (when no cycles are inserted, the $\overline{\text{RDY}}$ signal is ignored.)

The setup/hold time of the address, the assert delay cycle of the read/write strobe signals for $\overline{\text{CS5}}$ assertion and the $\overline{\text{CS5}}$ negate delay cycle for the read/write strobe signals negation can be set in the range from 0 to 7 cycles by CS5WCR. The $\overline{\text{BS}}$ hold cycles can be set to 1 or 2 when the RDS bits in CS5WCR are not 000 in reading and the WTS bits in CS5WCR are not 000 in writing.

For the PCMCIA interface, the setup/hold time of the address, $\overline{\text{CE1A}}$ and $\overline{\text{CE2A}}$ to the read/write strobe signal can be specified in the range from 0 to 15 cycles by bits TEDA/B and TEHA/B in CS5PCR. In addition, the number of wait cycles can be specified in the range from 0 to 50 cycles

by the PCWA/B bit. The number of wait cycles specified by CS5PCR is added to the value specified by bits IW3 to IW0 in CS5WCR or bits PCIW3 to PCIW0 in CS5PCR.

(7) Area 6

Area 6 is an area where bits 28 to 26 in the local bus address are 110.

The interface is that can be set for this area is the SRAM, MPX, burst ROM, and PCMCIA interface.

When the SRAM interface is used, a bus width of 8, 16 or 32 bits is selectable by bits SZ in CS6BCR. When the MPX interface is used, a bus width of 32 bits should be selected by bits SZ in CS6BCR. When the PCMCIA interface is used, select a bus width of 8 or 16 bits with SZ in CS6BCR. For details, see section 11.3.2, Memory Bus Width.

When the SRAM interface is used, the $\overline{\text{CS6}}$ signal is asserted when area 6 is accessed. The $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are also asserted. When the PCMCIA interface is used, the $\overline{\text{CE1B}}$ and $\overline{\text{CE2B}}$ signals, the $\overline{\text{RD}}$ signal (which can be used as $\overline{\text{OE}}$), and the $\overline{\text{WE0}}$, $\overline{\text{WE1}}$, $\overline{\text{WE2}}$, and $\overline{\text{WE3}}$ signals which can be used as $\overline{\text{REG}}$, $\overline{\text{WE}}$, $\overline{\text{IORD}}$, and $\overline{\text{IOWR}}$, respectively are asserted.

For the number of bus cycles, 0 to 25 wait cycles inserted by CS6WCR can be selected.

When the burst ROM interface is used, the number of a burst pitch is selectable in the range from 0 to 7 with the BW bits in CS6BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{\text{RDY}}$). (when no cycles are inserted, the $\overline{\text{RDY}}$ signal is ignored.)

The setup/hold time of the address, the assert delay cycle of the read/write strobe signals for $\overline{\text{CS6}}$ assertion and the $\overline{\text{CS6}}$ negate delay cycle for the read/write strobe signals negation can be set in the range from 0 to 7 cycles by CS6WCR. The $\overline{\text{BS}}$ hold cycles can be set to 1 or 2 when the RDS bits in CS6WCR are not 000 in reading and the WTS bits in CS6WCR are not 000 in writing.

For the PCMCIA interface, the setup/hold time of the address, $\overline{\text{CE1B}}$ and $\overline{\text{CE2B}}$ to the read/write strobe signal can be specified within a range from 0 to 15 cycles by bits TEDA/B and TEHA/B in CS6PCR. In addition, the number of wait cycles can be specified in the range from 0 to 50 cycles by the PCWA/B bit. The number of wait cycles specified by CS6PCR is added to the value specified by bits IW3 to IW0 in CS6WCR or bits PCIW3 to PCIW0 in CS6PCR.

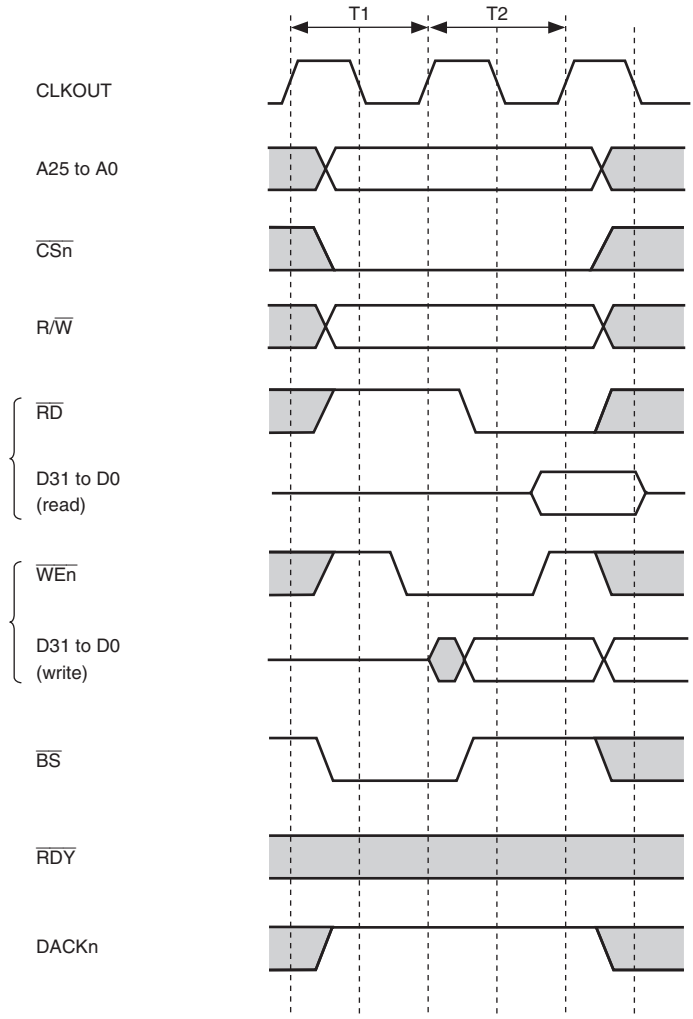
11.5.3 SRAM interface

(1) Basic Timing

The strobe signals for the SRAM interface in this LSI are output primarily based on the SRAM connection. Figure 11.5 shows the basic timing of the SRAM interface. Normal access without wait cycles is completed in two cycles. The $\overline{BS}$ signal is asserted for one or two cycles to indicate the start of a bus cycle. The $\overline{CSn}$ signal is asserted at the rising edge of the clock in the T1 state, and negated at the next rising edge of the clock in the T2 state. Therefore, there is no negation period in accesses at minimum pitch.

In reading, an access size is not specified. The output of an access address on the address pins (A25 to A0) is correct, however, since the access size is not specified, 32-bit data is always output when a 32-bit device is in use, and 16-bit data is output when a 16-bit device is in use. During writing, only the $\overline{WE}$ signal corresponding to the byte to be written is asserted. For details, see section 11.5.1, Endian/Access Size and Data Alignment.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the bus width set. The first access is performed on the data for which an access request is issued, and the remaining accesses are performed in wraparound method according to the set bus width. The bus is not released during this transfer.



In this example, DACKn is high-active.

Figure 11.5 Basic Timing of SRAM Interface

Figures 11.6 to 11.8 show examples of connections to SRAM with 32-, 16- and 8-bit data width, respectively.

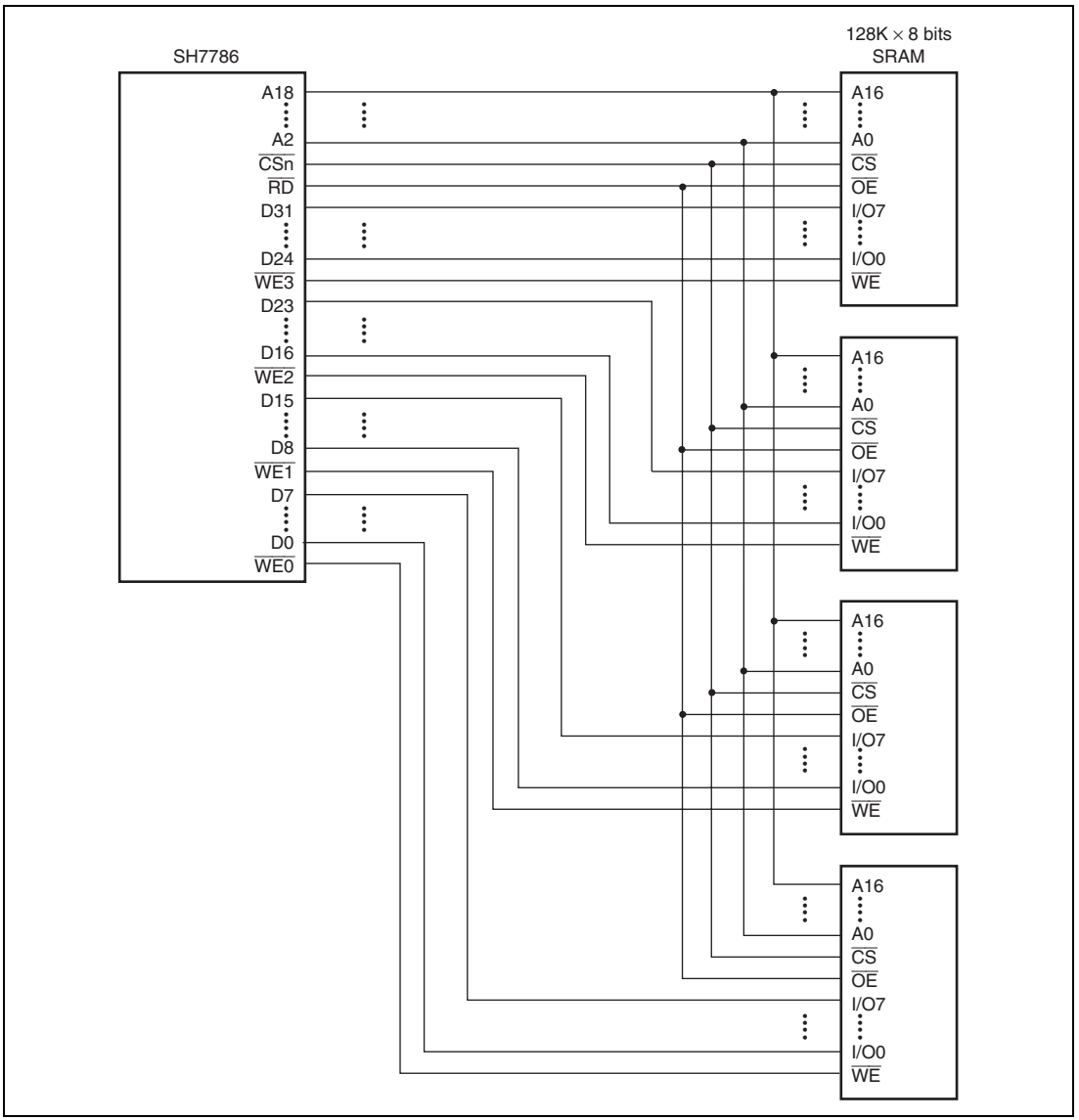


Figure 11.6 Example of 32-Bit Data Width SRAM Connection

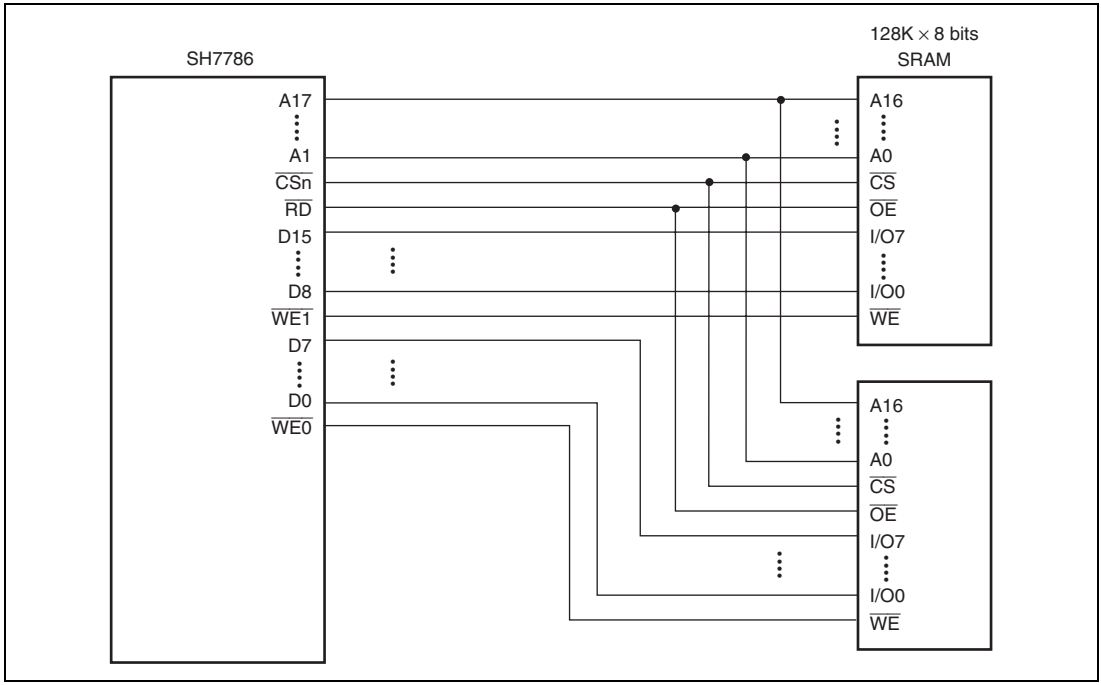


Figure 11.7 Example of 16-Bit Data Width SRAM Connection

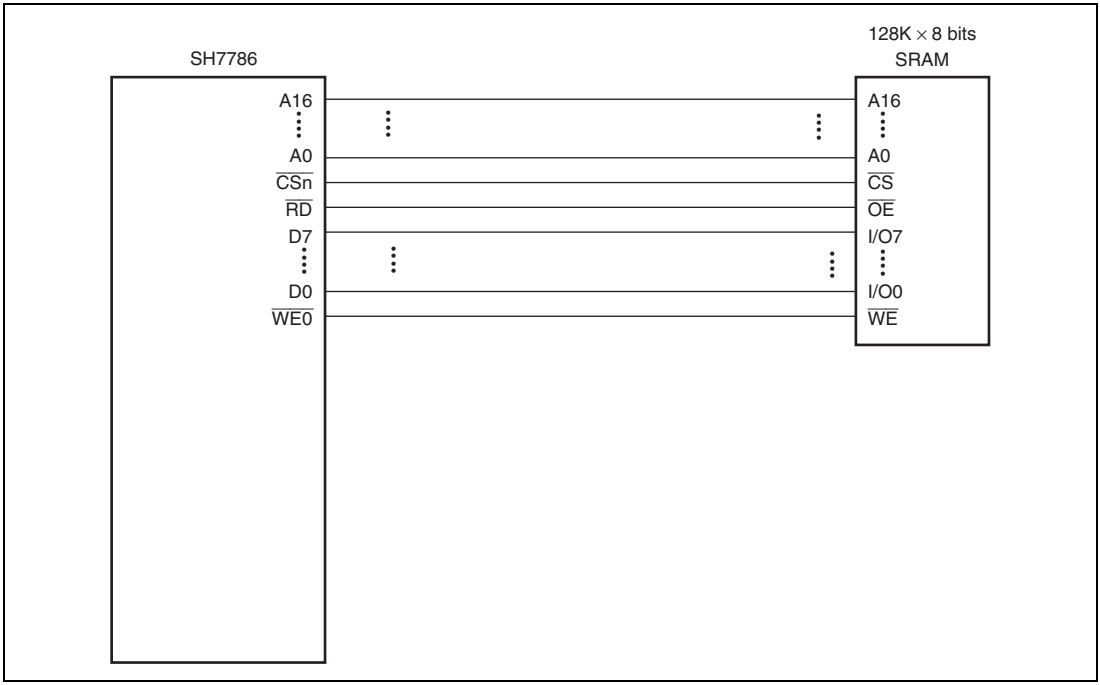


Figure 11.8 Example of 8-Bit Data Width SRAM Connection

(2) Wait Cycle Control

Wait cycle insertion for the SRAM interface can be controlled by CSnWCR. If the IW bits in CSnWCR are set to a value other than 0, a software wait is inserted in accordance with the wait control bits. For details, see section 11.4.4, CSn Wait Control Register (CSnWCR).

The specified number of Tw cycles is inserted as wait cycles in accordance with the CSnWCR setting. The wait cycle insertion timing is shown in figure 11.9.

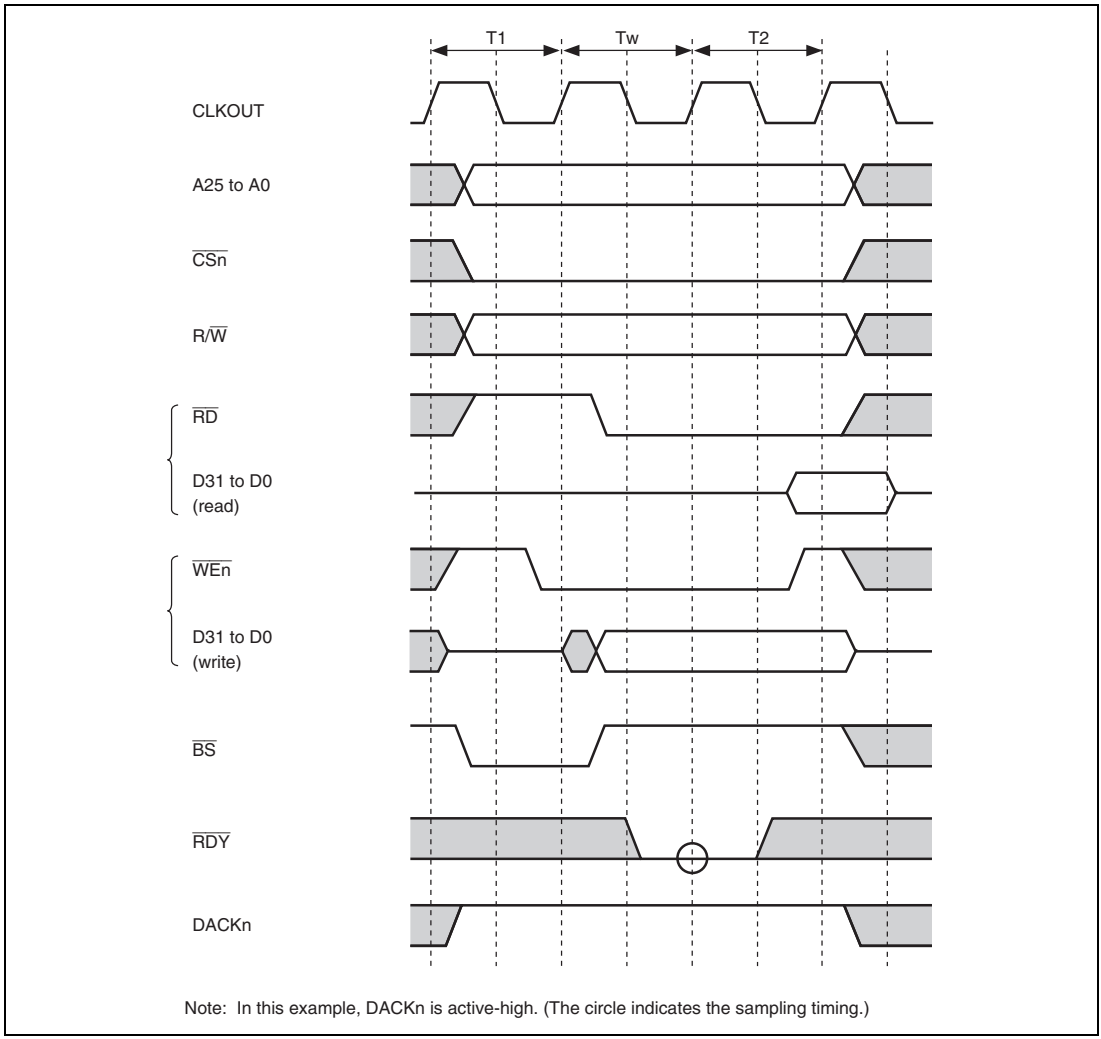
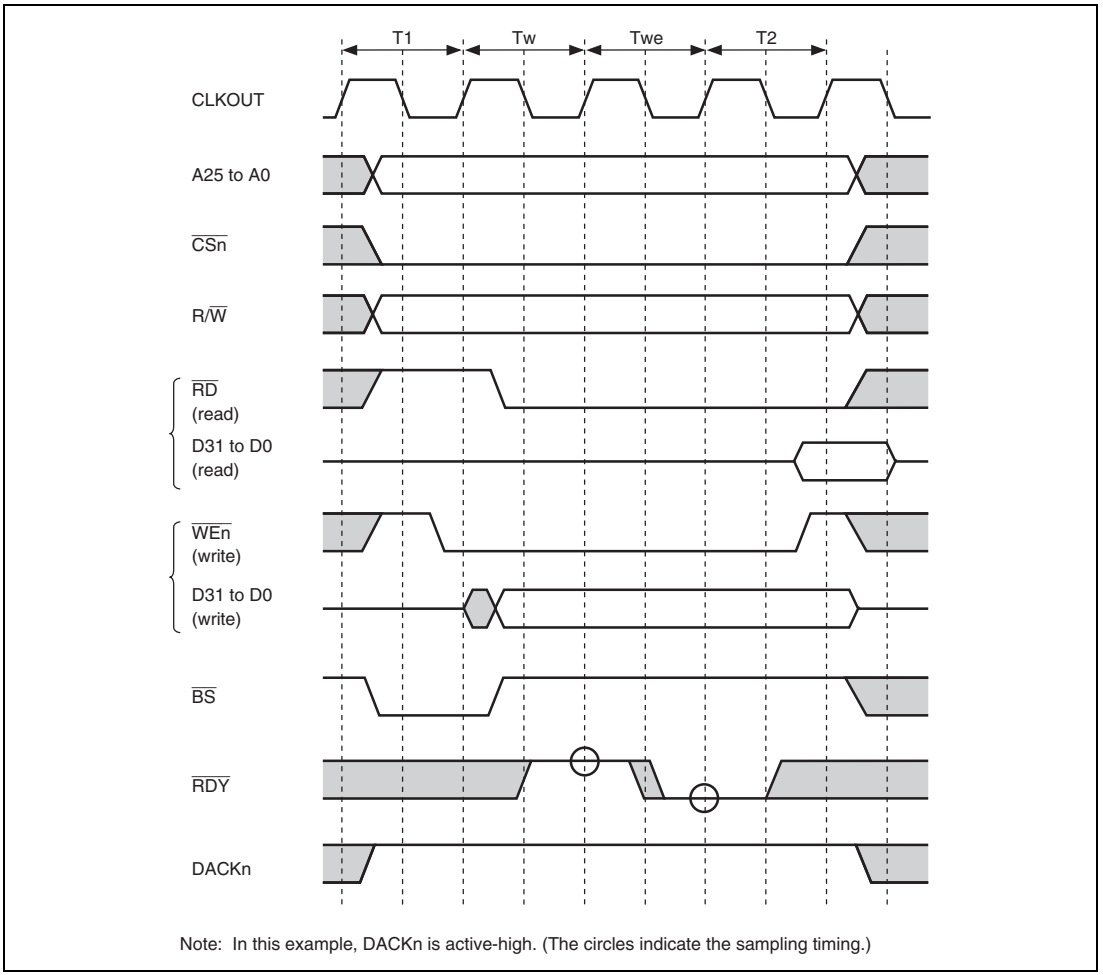


Figure 11.9 SRAM Interface Wait Timing (Software Wait Only)

When software wait insertion is specified by CSnWCR, the external wait input signal, $\overline{\text{RDY}}$, is also sampled. The $\overline{\text{RDY}}$ signal sampling timing is shown in figure 11.10, where a single wait cycle is specified as a software wait. The $\overline{\text{RDY}}$ signal is sampled at the transition from the Tw state to the T2 state. Therefore, the assertion of the $\overline{\text{RDY}}$ signal has no effect in the T1 cycle or in the first Tw cycle. The $\overline{\text{RDY}}$ signal is sampled on the rising edge of the clock.



Note: In this example, DACKn is active-high. (The circles indicate the sampling timing.)

Figure 11.10 SRAM Interface Wait Timing
(Wait Cycle Insertion by $\overline{\text{RDY}}$ Signal, $\overline{\text{RDY}}$ Signal is synchronous input)

(3) Read-Strobe Negate Timing

When the SRAM interface is used, the strobe signal negation timing in reading can be specified with the RDSPL bit in CSnBCR. For details of settings, see section 11.4.3, CSn Bus Control Register (CSnBCR). The RDSPL bit should be cleared to 0 when a byte control SRAM is specified.

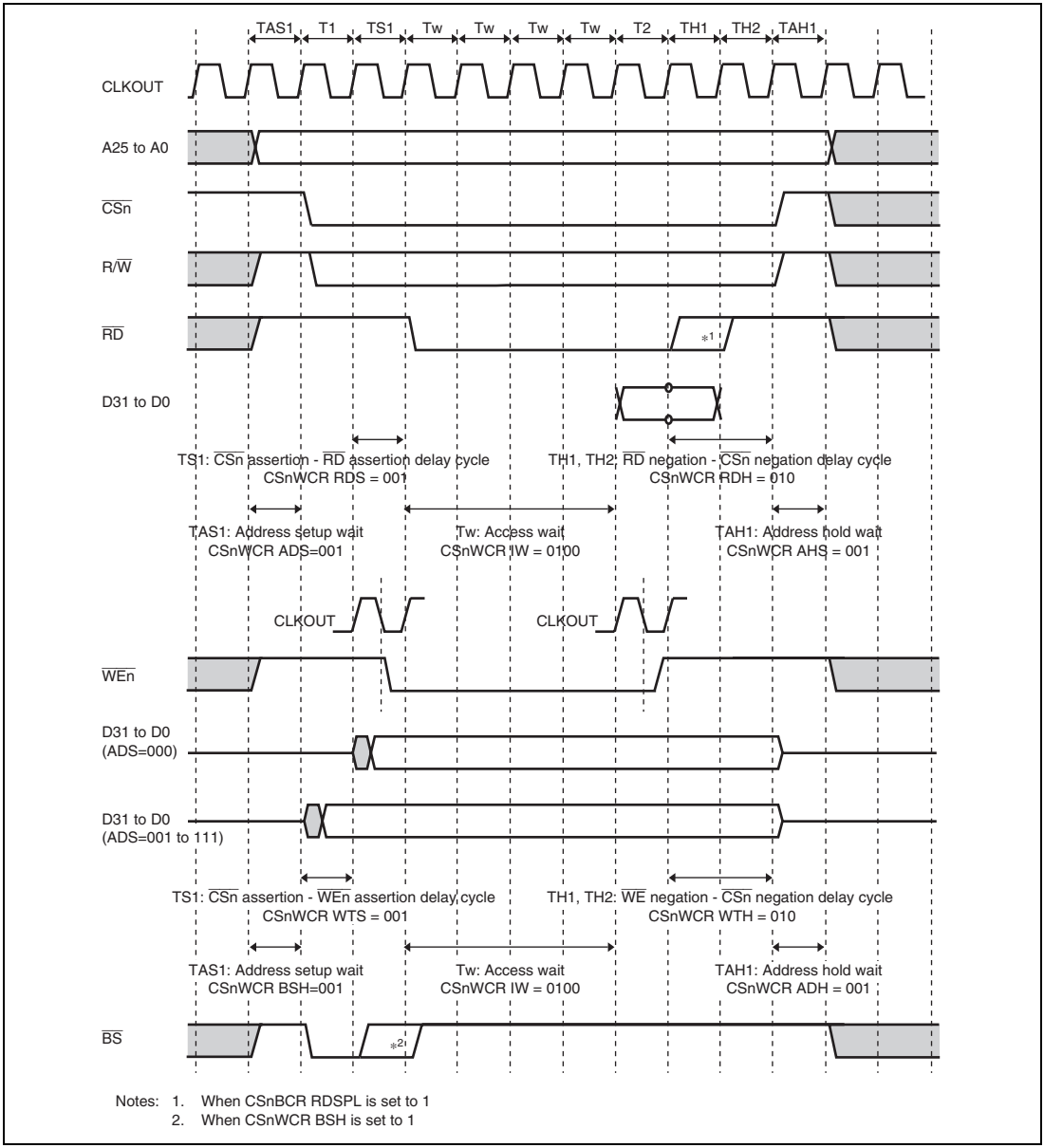


Figure 11.11 SRAM Interface Wait Timing (Read-Strobe Negate Timing Setting)

11.5.4 Burst ROM Interface

When the TYPE bit in CSnBCR is set to 010, a burst ROM can be connected to areas 0 to 6. The burst ROM interface provides high-speed access to ROM that has a burst access function. The burst access timing of burst ROM is shown in figure 11.12. The wait cycle is set to 0. Although the access is similar to that of the SRAM interface, only the address is changed when the first cycle ends and then the next access is started. When 8-bit ROM is used, the number of consecutive accesses can be set to 4, 8, 16 or 32 times through bits BST2 to BST0 in CSnBCR (n = 0 to 6). Similarly, when 16-bit ROM is used, 4, 8 or 16 times can be set; when 32-bit ROM is used, 4 or 8 times can be set.

The $\overline{\text{RDY}}$ signal is always sampled when the wait cycle is set to 1 or more. Even when no wait is specified in the burst ROM settings, the second and subsequent accesses are performed with two cycles as shown in figure 11.13.

Writing to this interface is performed in the same way as for the SRAM interface.

In a 32-byte transfer, a total of 32 bytes are transferred continuously according to the set bus width. The first access is performed on the data for which an access request is issued, and the remaining accesses are performed on wraparound method according to the set bus width. The bus is not released during this transfer.

Figure 11.14 shows the timing when the burst ROM is used and setup/hold is specified by CSnWCR.

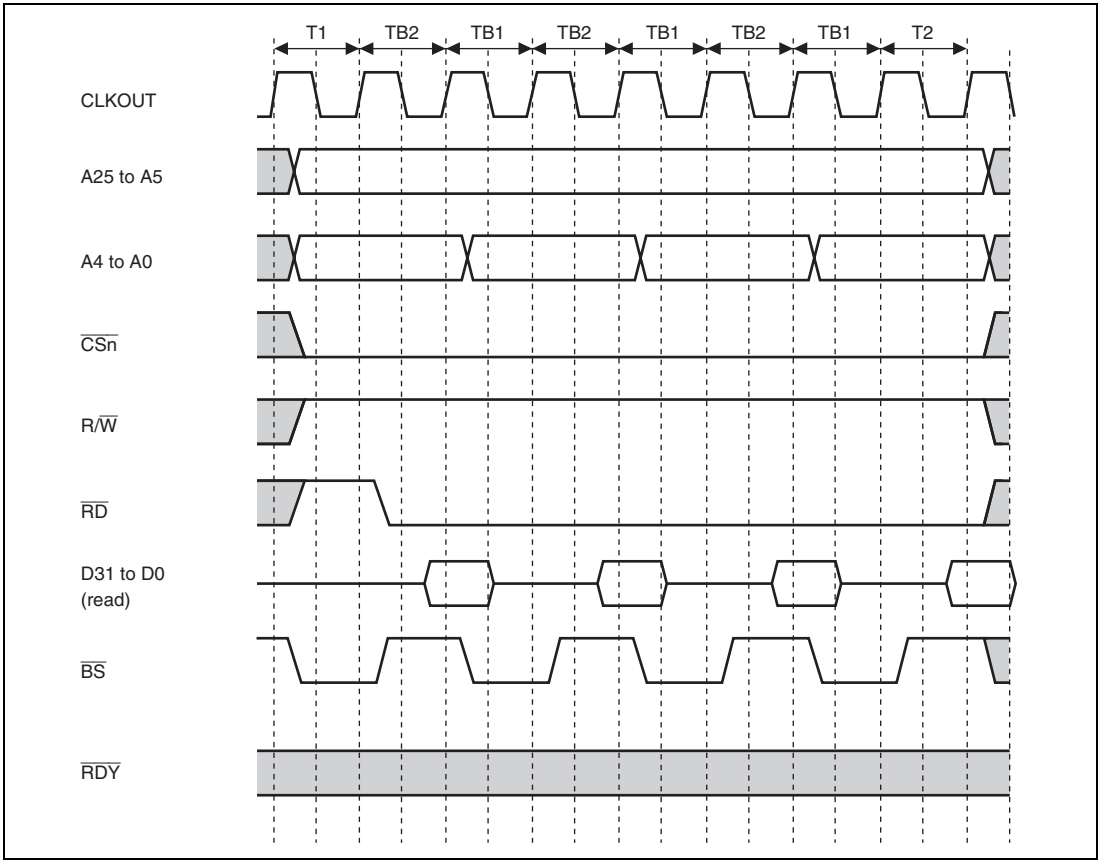


Figure 11.12 Burst ROM Basic Timing

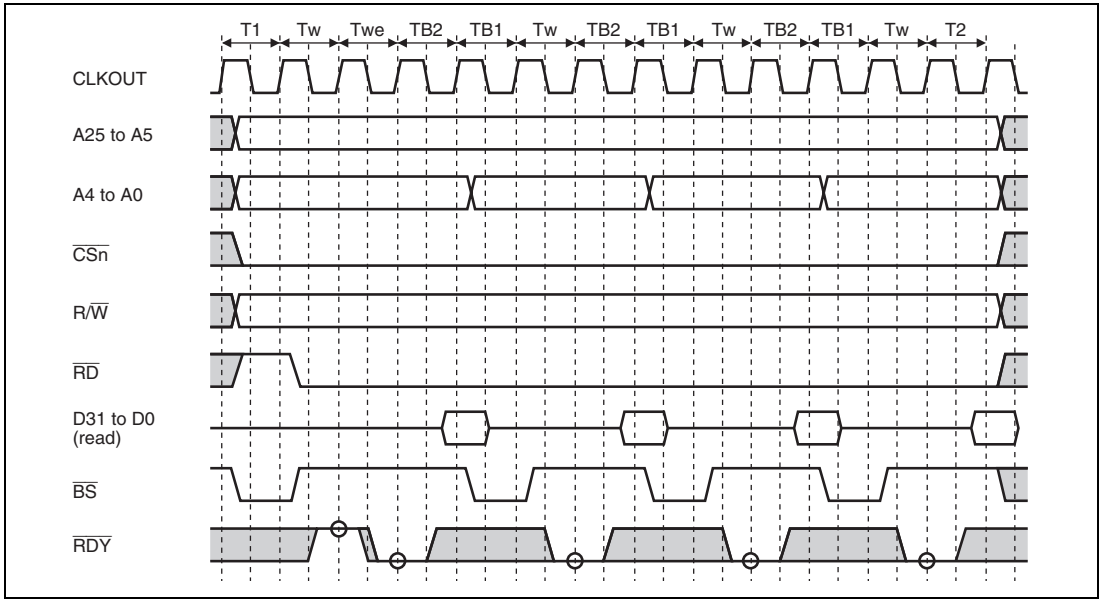
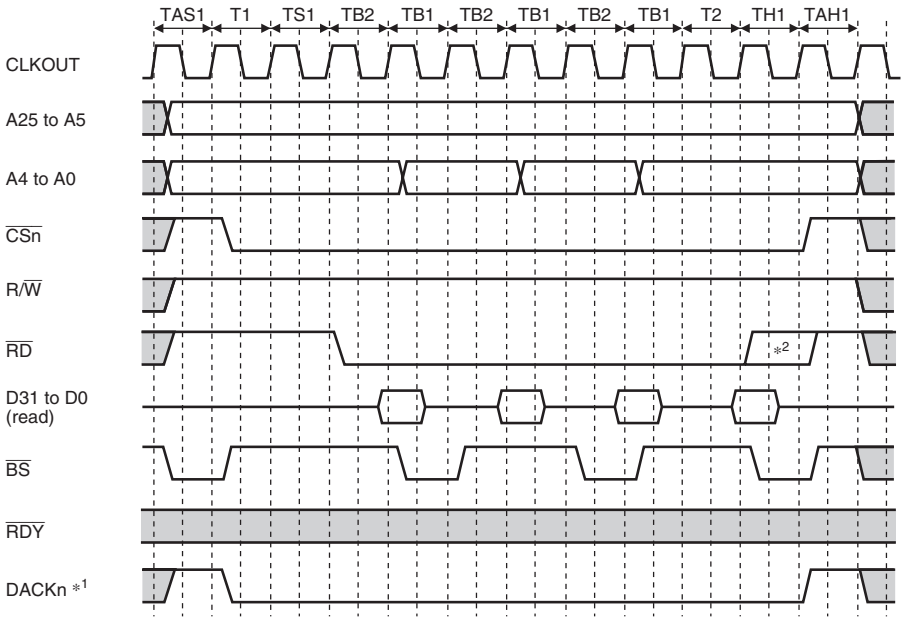


Figure 11.13 Burst ROM Wait Timing



Notes: 1. In this example, DACKn is active-high.
2. When RDSPL in CSnBCR is set to 1.

Figure 11.14 Burst ROM Wait Timing

11.5.5 PCMCIA Interface

By setting the TYPE bits in CS5BCR and CS6BCR, the bus interface for the external space areas 5 and 6 can be set to the IC memory card interface or I/O card interface, which is stipulated in JEIDA specification version 4.2 (PCMCIA 2.1).

Figure 11.15 shows the connection example of this LSI and PCMCIA card. PCMCIA card is required to connect a three state buffer between this LSI bus interface and PCMCIA card to perform hot swapping (a card is pulled out or plugged while the power supply of the system is turned on).

Since operation in big endian mode is not explicitly stipulated in the JEIDA/PCMCIA standard, this LSI only supports the little-endian PCMCIA interface with the little endian mode setting.

The PCMCIA interface space property can be selected from 8-bit common memory, 16-bit common memory, 8-bit attribute memory, 16-bit attribute memory, 8-bit I/O space, 16-bit I/O space, dynamic I/O bus sizing, and ATA complement mode by depending on the setting of the SAA and SAB bits in CSnPCR.

When the first half area is accessed, the IW bit in CSnWCR and the PCWA, TEDA, and TEHA bits in CSnPCR are selected. When the second half area is accessed, the IW bit in CSnWCR and the PCWB, TEDB, and TEHB bits in CSnPCR are selected.

The PCWA/B1 and PCWA/B bits can be used to set the number of wait cycles to be inserted in a low-speed bus cycle as 0, 15, 30, or 50. This value is added to the number of inserted wait cycles specified by the IW bit in CSnWCR or PCIW bit in CSnPCR. The setup time of the address of the $\overline{\text{RD}}$ and $\overline{\text{WE1}}$ signals, $\overline{\text{CSn}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$ and $\overline{\text{REG}}$ can be set with the TEDA/B bit (with a setting range from 0 to 15). The hold time of the address of the $\overline{\text{RD}}$ and $\overline{\text{WE1}}$ signals, $\overline{\text{CSn}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$ and $\overline{\text{REG}}$ can be set with the TEDA/B bit (with a setting range from 0 to 15).

The IW bits (IWW, IWWRD, IWWRS, IWRRD and IWRRS) in CS5BCR or CS6BCR are used to set the number of idle cycles between cycles. The selected number of wait cycles between cycles depends only on the area to be accessed (area 5 or 6). When area 5 is accessed, the IW bits in CS5WCR are selected, and when area 6 is accessed, the IW bits in CS6WCR are selected.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wraparound method according to the set bus width. The bus is not released during this transfer.

ATA complement mode is to access the ATA device register connected to this LSI. The Device Control Register, Alternate Status Register, Data Register, and Data Port can be accessed in ATA

complement mode. To access the Device Control Register and Alternate Status Register, use a CPU byte access (do not use a DMA transfer), and to access the Data Register, use the CPU word access (do not use a DMA transfer). To access the Data Port use a DMA transfer. When a CPU byte access is executed, $\overline{CE1x}$ is negated and $\overline{CE2x}$ is asserted (x = A, B). When a CPU word access is executed, $\overline{CE1x}$ is asserted and $\overline{CE2x}$ is negated. When a DMA access is executed, CE1x and CE2x are negated. The setting example of the DMAC0 (by DMA channel control register CHCR) is external request, burst mode, level detection, overrun 0, $\overline{DACK}$ output to the correspondent PCMCIA connected area. Set the DACKBST bit in BCR of the corresponding DMA transfer channel to 1, so that the corresponding $\overline{DACK}$ signal is asserted from the beginning to the end of the DMA transfer cycle. Even if the corresponding $\overline{DREQ}$ signal is negated during the transfer, the $\overline{DACK}$ signal is not negated. When DMA transfer that outputs $\overline{DACK}$ is made to access an area where ATA complement mode is set, neither $\overline{CE1x}$ nor $\overline{CE2x}$ is asserted.

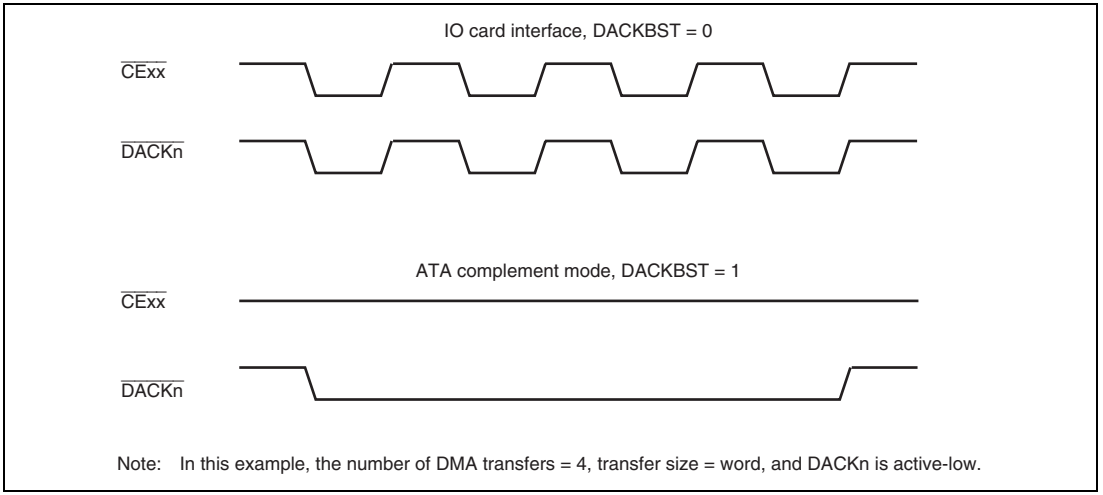


Figure 11.15 $\overline{CExx}$ and $\overline{DACKn}$ Output during DMA Transfer in Access to Space where ATA Complement Mode is Set

Table 11.13 Relationship between Address and CE When Using PCMCIA Interface

Bus (Bits)	Read/ Write	Access (bits)*[1]	Odd/ Even	IOIS16	Access	CE2	CE1	A0	D15 to D8	D7 to D0
8	Read	8	Even	×	—	H	L	L	Invalid	Read data
			Odd	×	—	H	L	H	Invalid	Read data
		16	Even	×	First	H	L	L	Invalid	Lower read data
			Even	×	Second	H	L	H	Invalid	Upper read data
			Odd	×	—	—	—	—	—	—
	Write	8	Even	×	—	H	L	L	Invalid	Write data
			Odd	×	—	H	L	H	Invalid	Write data
		16	Even	×	First	H	L	L	Invalid	Lower write data
			Even	×	Second	H	L	H	Invalid	Upper write data
			Odd	×	—	—	—	—	—	—
16	Read	8	Even	×	—	H	L	L	Invalid	Read data
			Odd	×	—	L	H	H	Read data	Invalid
		16	Even	×	—	L	L	L	Upper read data	Lower read data
			Odd	×	—	—	—	—	—	—
	Write	8	Even	×	—	H	L	L	Invalid	Write data
			Odd	×	—	L	H	H	Write data	Invalid
		16	Even	×	—	L	L	L	Upper write data	Lower write data
			Odd	×	—	—	—	—	—	—
Dynamic Bus Sizing*[2]	Read	8	Even	L	—	H	L	L	Invalid	Read data
			Odd	L	—	L	H	H	Read data	Invalid
		16	Even	L	—	L	L	L	Upper read data	Lower read data
			Odd	L	—	—	—	—	—	—
	Write	8	Even	L	—	H	L	L	Invalid	Write data
			Odd	L	—	L	H	H	Write data	Invalid
		16	Even	L	—	L	L	L	Upper write data	Lower write data
			Odd	L	—	—	—	—	—	—
	Read	8	Even	H	—	H	L	L	Invalid	Read data
			Odd	H	First	L	H	H	Invalid	Invalid
			Odd	H	Second	H	L	L	Invalid	Read data
		16	Even	H	First	L	L	L	Invalid	Lower read data
			Even	H	Second	H	L	H	Invalid	Upper read data
			Odd	H	—	—	—	—	—	—

Bus (Bits)	Read/ Write	Access (bits)*[1]	Odd/ Even	IOIS16	Access	CE2	CE1	A0	D15 to D8	D7 to D0
Dynamic Bus Sizing*[2]	Write	8	Even	H	—	H	L	L	Invalid	Write data
			Odd	H	First	L	H	H	Invalid	Write data
			Odd	H	Second	H	L	H	Invalid	Write data
		16	Even	H	First	L	L	L	Upper write data	Lower write data
			Even	H	Second	H	L	H	Invalid	Upper write data
			Odd	H	—	—	—	—	—	—
ATA comple-ment mode	Read (does not output DACK)	8	Even	×	—	L	H	L	Invalid	Read data
			Odd	×	—	—	—	—	—	—
		16	Even	×	—	H	L	L	Upper read data	Lower read data
			Odd	×	—	—	—	—	—	—
	Write (does not output DACK)	8	Even	×	—	L	H	L	Invalid	Write data
			Odd	×	—	—	—	—	—	—
		16	Even	×	—	H	L	L	Upper write data	Lower write data
			Odd	×	—	—	—	—	—	—
	Read (outputs DACK)	8	Even	×	—	H	H	L	Invalid	Read data
			Odd	×	—	H	H	L	Read data	Invalid
		16	Even	×	—	H	H	H	Upper read data	Lower read data
			Odd	×	—	—	—	—	—	—
	Write (outputs DACK)	8	Even	×	—	H	H	L	Invalid	Write data
			Odd	×	—	H	H	L	Write data	Invalid
		16	Even	×	—	H	H	H	Upper write data	Lower write data
			Odd	×	—	—	—	—	—	—

Legend:

×: Don't care

L: Low level

H: High level

Notes: 1. In 32-bit/64-bit/16-byte/32-byte transfer, the address is automatically incremented by the bus width, and the above accesses are repeated until the transfer data size is reached.

2. PCMCIA I/O card interface only.

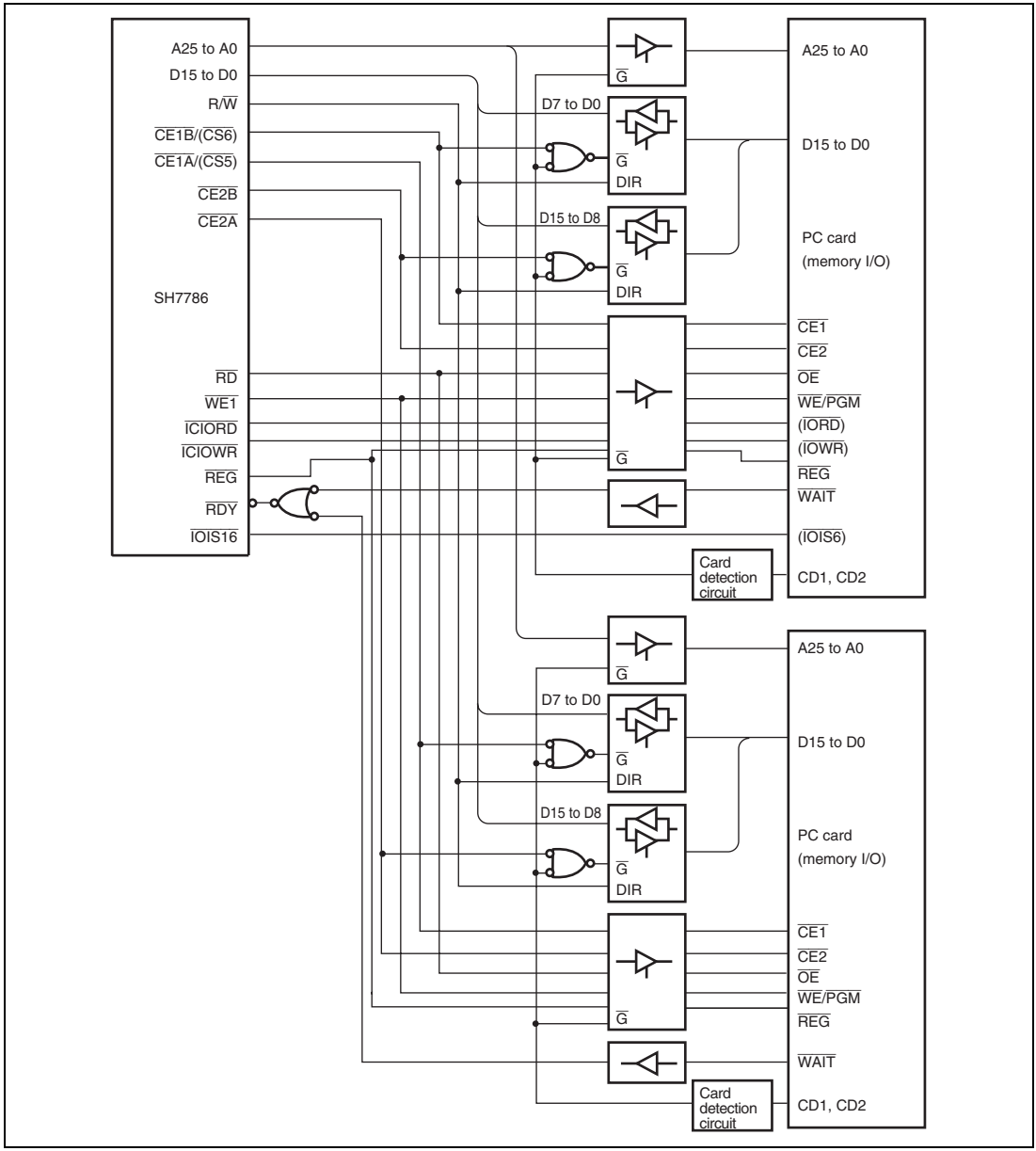


Figure 11.16 Example of PCMCIA Interface

(1) Memory Card Interface Basic Timing

Figure 11.17 shows the basic timing for the PCMCIA memory card interface, and figure 11.18 shows the wait timing for the PCMCIA memory card interface.

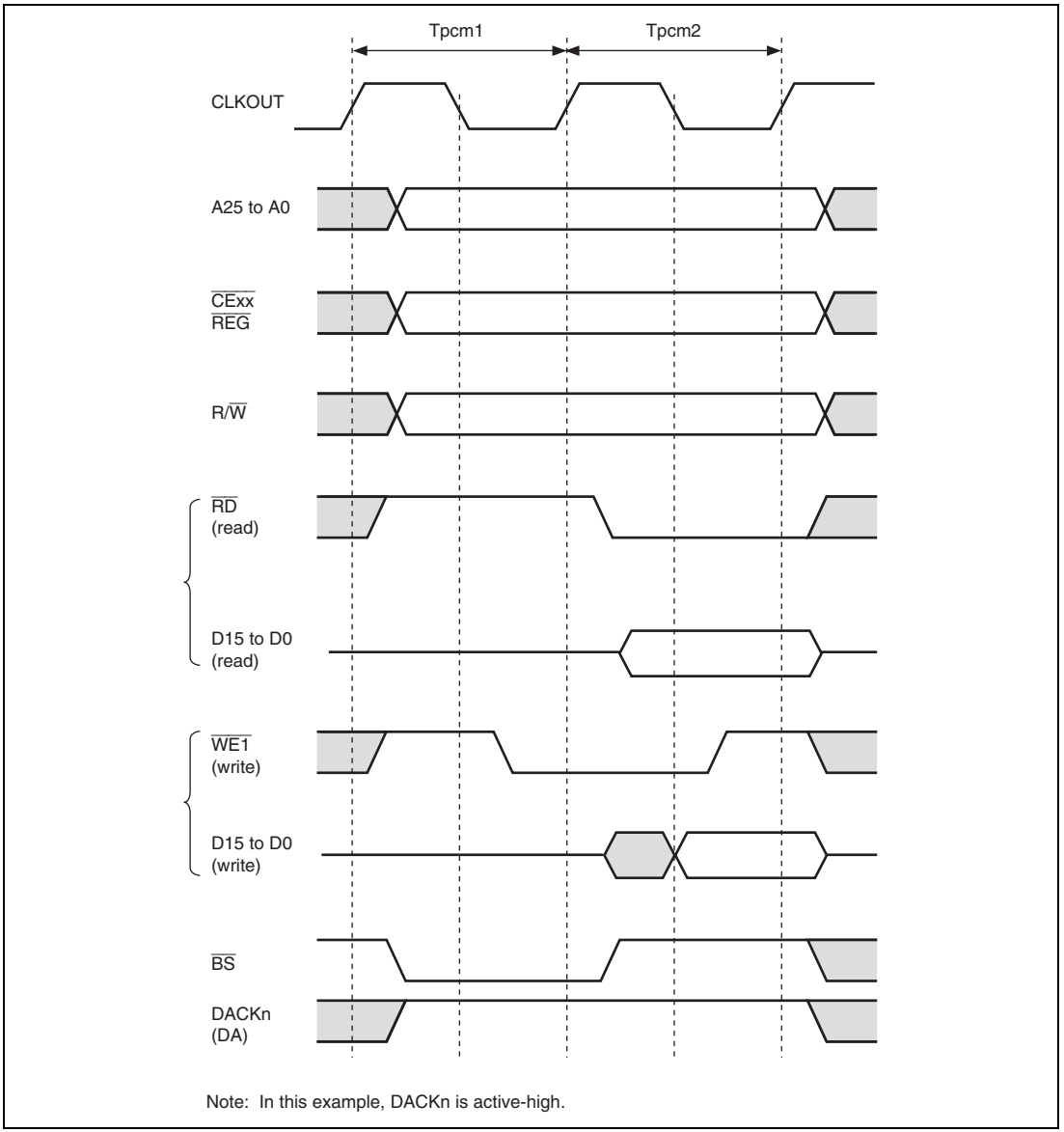
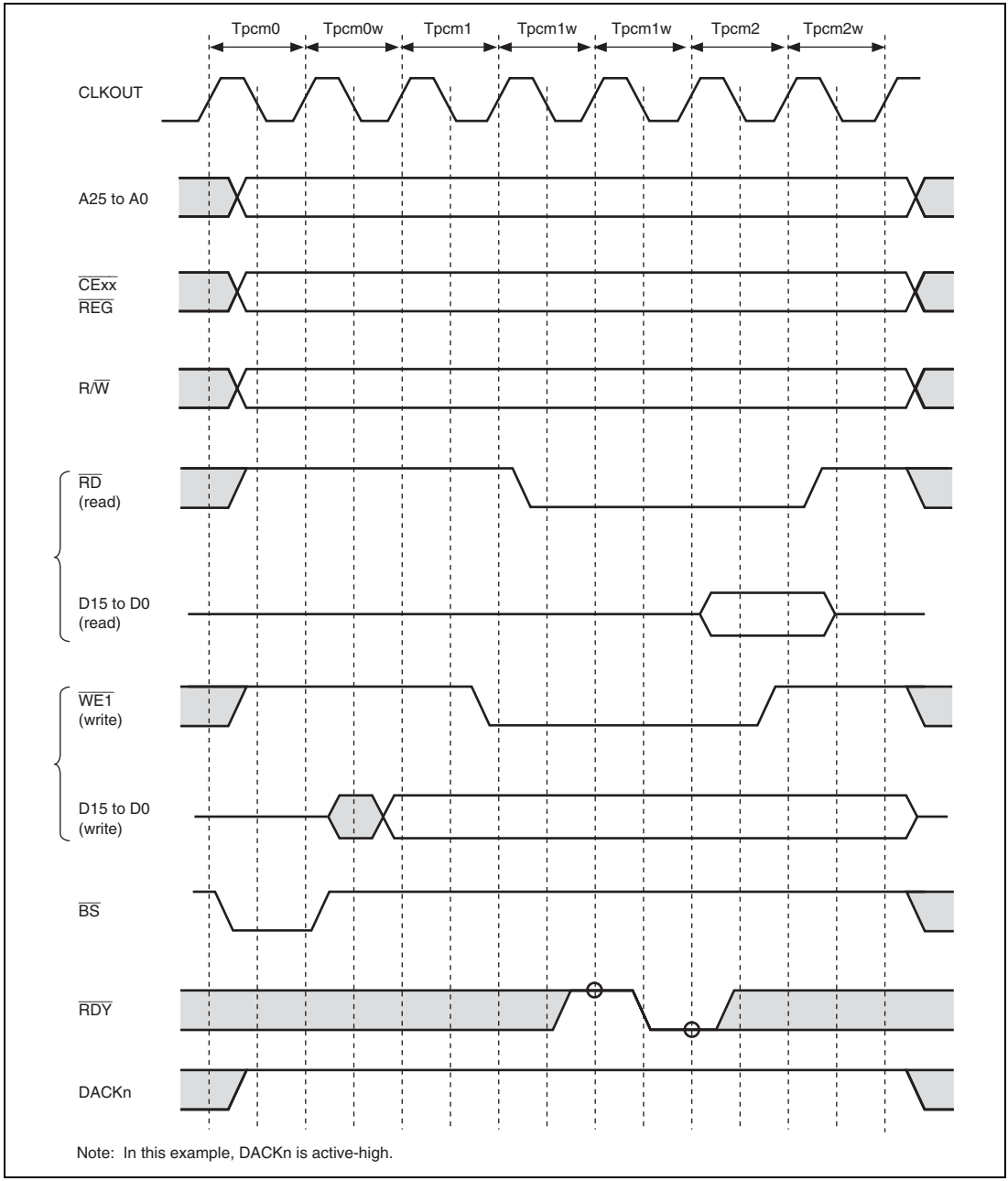


Figure 11.17 Basic Timing for PCMCIA Memory Card Interface



Note: In this example, DACKn is active-high.

Figure 11.18 Wait Timing for PCMCIA Memory Card Interface

(2) I/O Card Interface Timing

Figures 11.19 and 11.20 show the timing for the PCMCIA I/O card interface.

When a PCMCIA card is accessed as the I/O card interface, dynamic sizing with the I/O bus width can be performed using the $\overline{\text{IOIS16}}$ pin. With the 16-bit bus width selected, if the $\overline{\text{IOIS16}}$ signal is high during the word-size I/O bus cycle, the I/O port is recognized as eight bits in bus width. In this case, a data access for only eight bits is performed in the I/O bus cycle being executed, and this is automatically followed by a data access for the remaining eight bits. Dynamic bus sizing is also performed for byte-size access to address 2n + 1.

Figure 11.21 shows the basic timing for dynamic bus sizing.

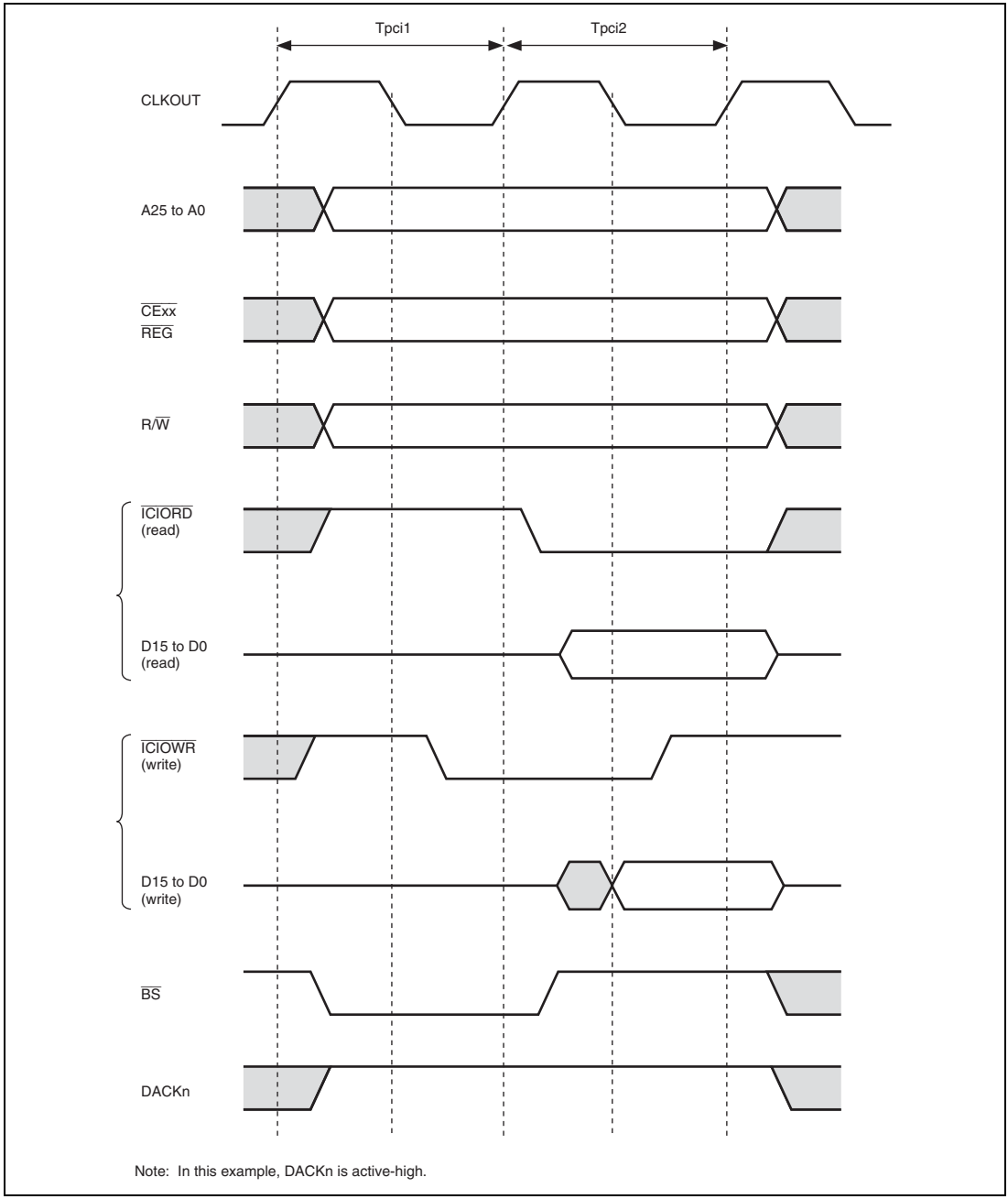


Figure 11.19 Basic Timing for PCMCIA I/O Card Interface

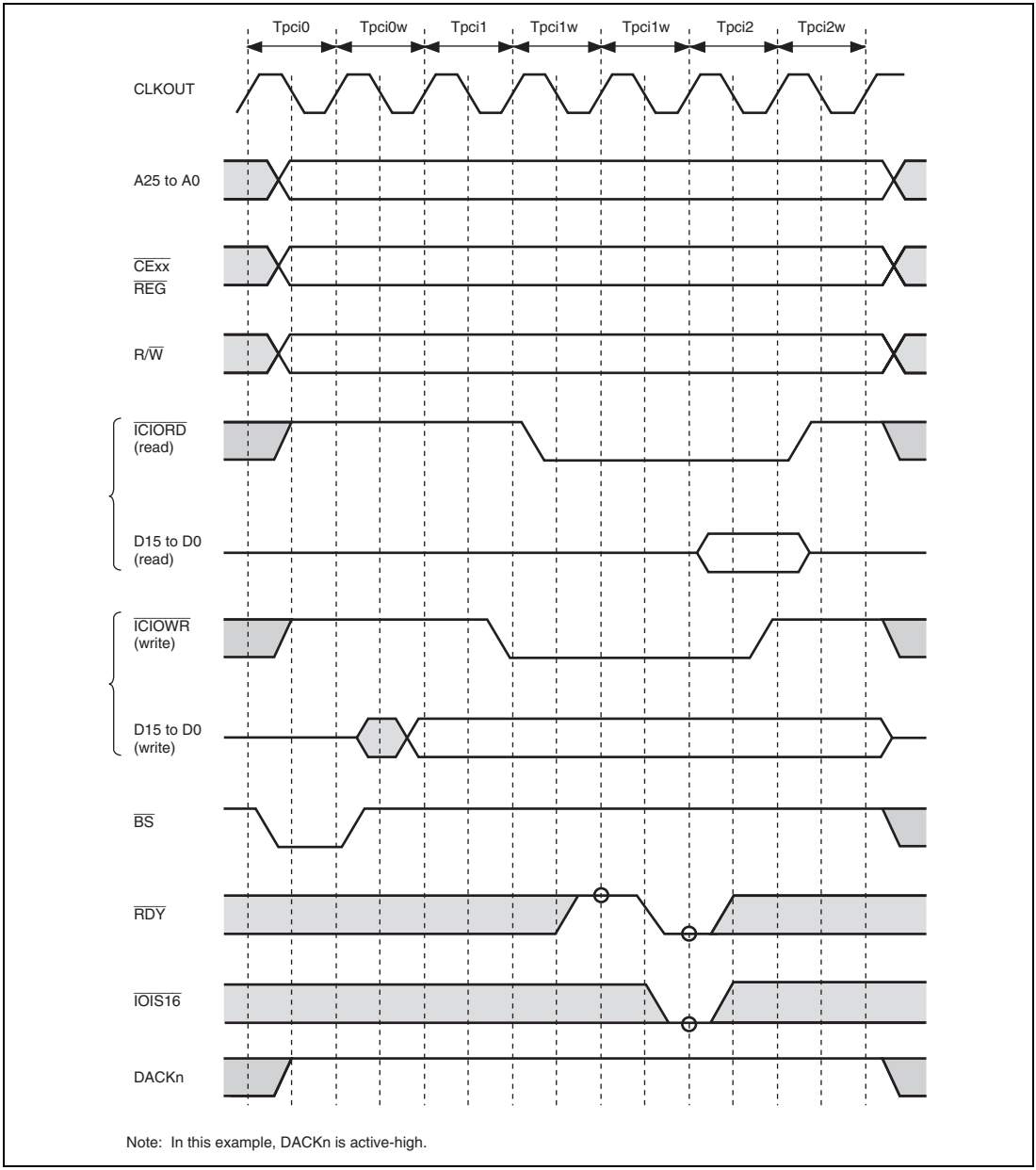
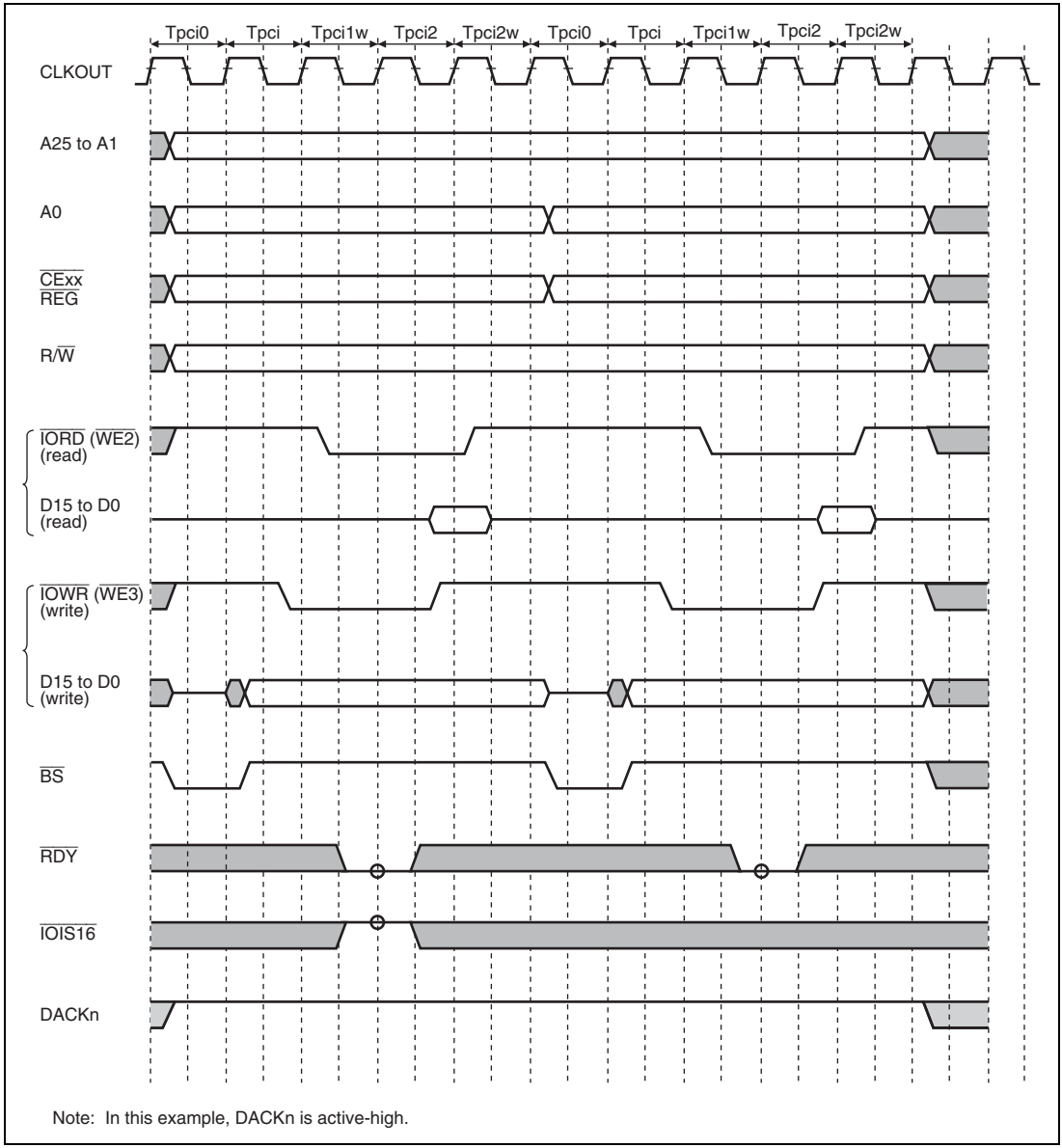


Figure 11.20 Wait Timing for PCMCIA I/O Card Interface



Note: In this example, DACKn is active-high.

Figure 11.21 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

11.5.6 MPX Interface

When both the MODE 6, MODE 5, MODE 4 pin is set to 0 at a power-on reset by the $\overline{\text{PRESET}}$ pin, the MPX interface is selected for area 0. The MPX interface is selected for areas 1 to 6 by the MPX bit in CS1BCR to CS6BCR. The MPX interface provides an address/data multiplex-type bus protocol and facilitates connection with external memory controller chips using an address/data multiplex-type 32-bit single bus. A bus cycle consists of an address phase and a data phase. In the address phase, address information is output on D25 to D0, and the access size is output on D31 to D29. The $\overline{\text{BS}}$ signal is asserted for one cycle to indicate the address phase. The $\overline{\text{CSn}}$ signal is asserted at the rising edge in Tm1 and is negated after the end of the last data transfer in the data phase. Therefore, a negation cycle is not generated in the case of minimum pitch access. The $\overline{\text{FRAME}}$ signal is asserted at the rising edge in Tm1 and negated at the start of the last data transfer cycle in the data phase. Therefore, an external device for the MPX interface must internally store the address information and access size output in the address phase and perform data input/output for the data phase. For details, see section 11.5.1, Endian/Access Size and Data Alignment.

Values output to address pins A25 to A20 are not guaranteed.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the set bus width. The first access is performed on the data for which an access request is issued, and the remaining accesses are performed according to the set bus width. If the access size is larger than the bus width, a burst access with continuing multiple data cycles occurs after one address output. The bus is not released during this transfer.

Table 11.14 Relationship between D31 to D29 and Access Size in Address Phase

D31	D30	D29	Access Size
0	0	0	Byte
		1	Word
	1	0	Longword
		1	Unused
1	X	X	32-byte burst

Legend:
X: Don't care

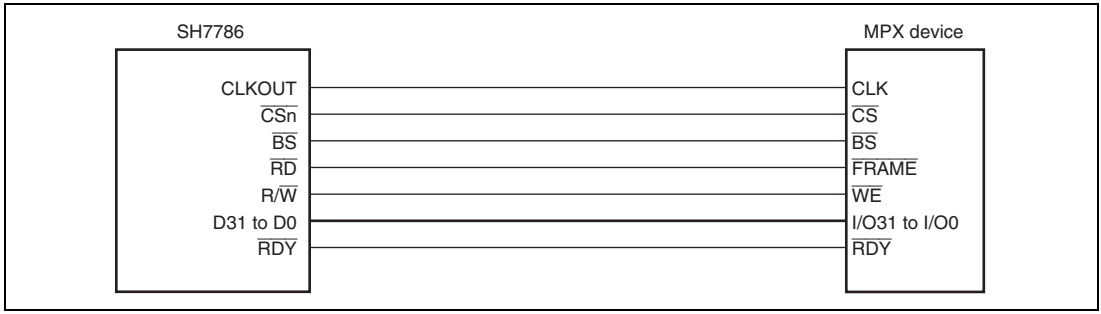


Figure 11.22 Example of 32-Bit Data Width MPX Connection

The MPX interface timing is shown below.

When the MPX interface is used for areas 1 to 6, the bus size should be set to 32 bits by CSnBCR.

Waits can be inserted by CSnWCR and the $\overline{\text{RDY}}$ pin.

In reading, one wait cycle is automatically inserted after address output even if CSnWCR is cleared to 0.

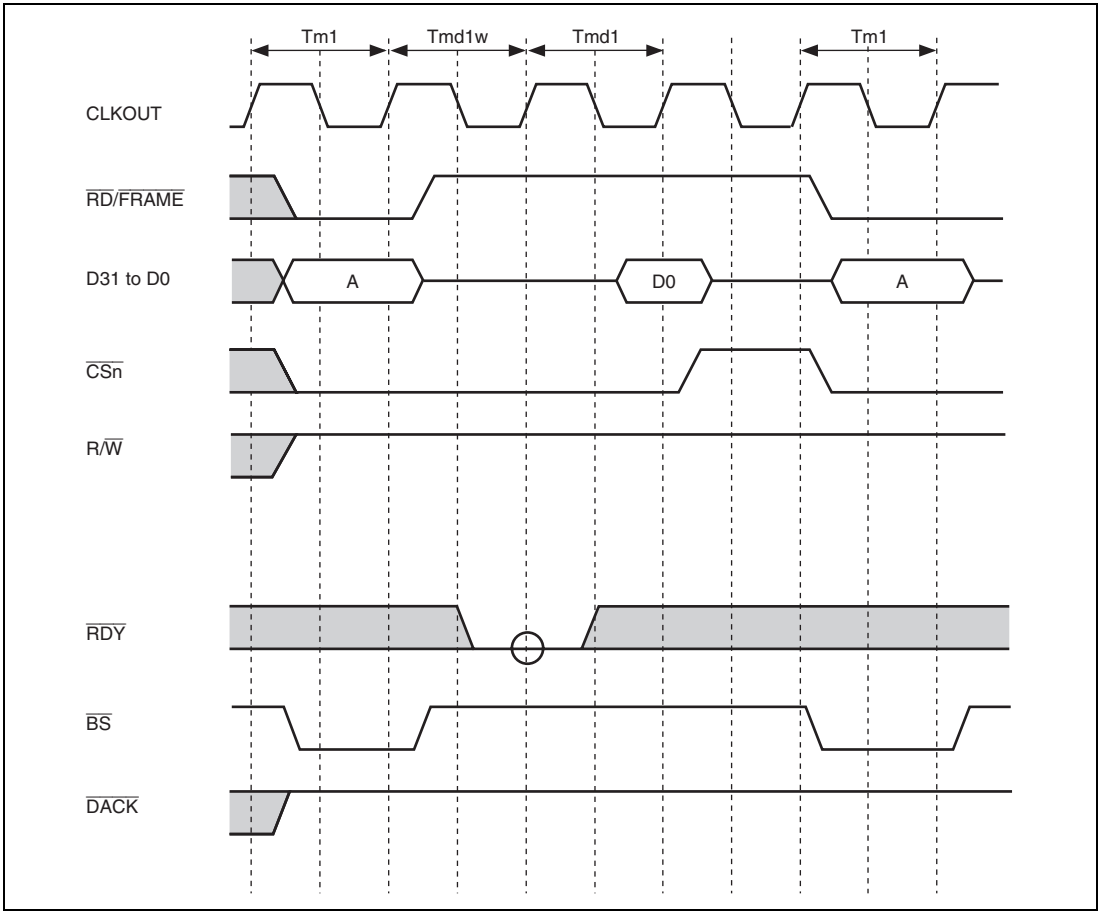
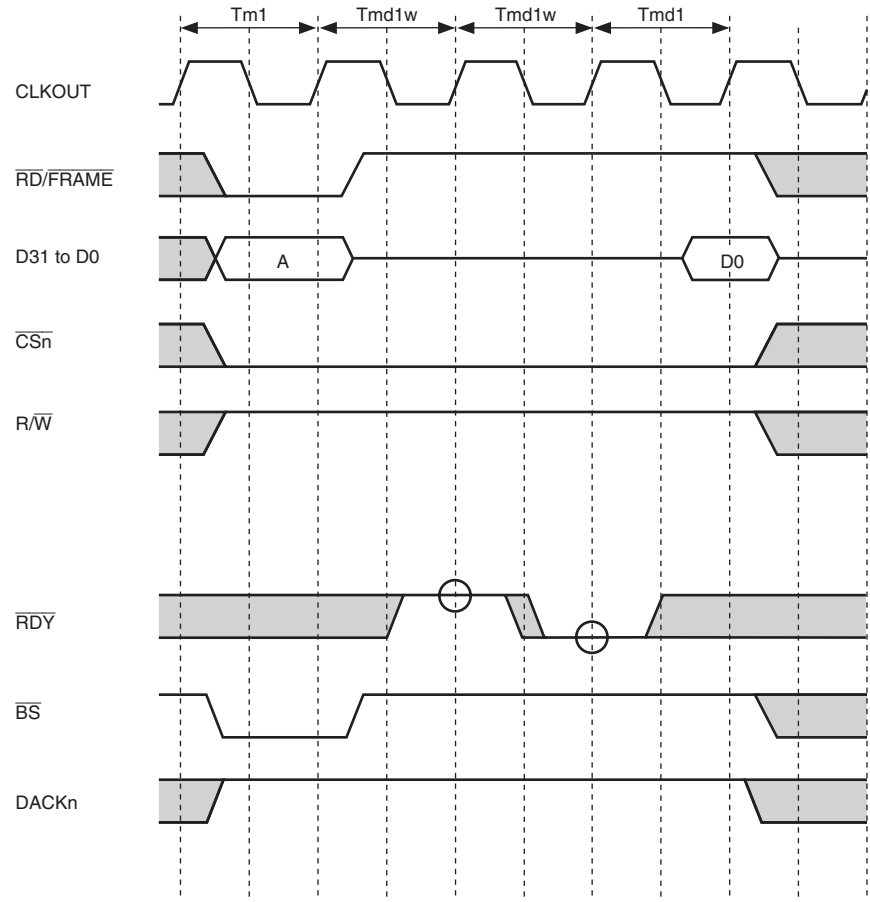
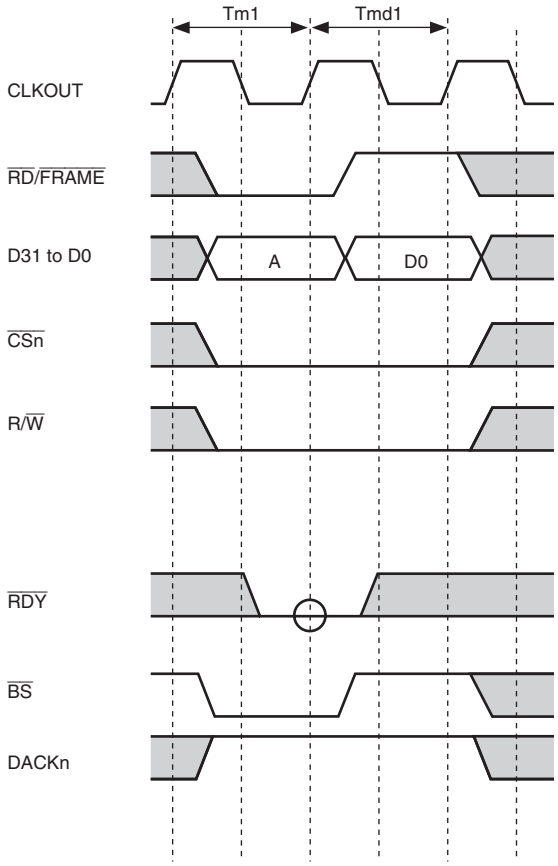


Figure 11.23 MPX Interface Timing 1
(Single Read Cycle, IW = 0000, No External Wait, 32-Bit Bus Width)



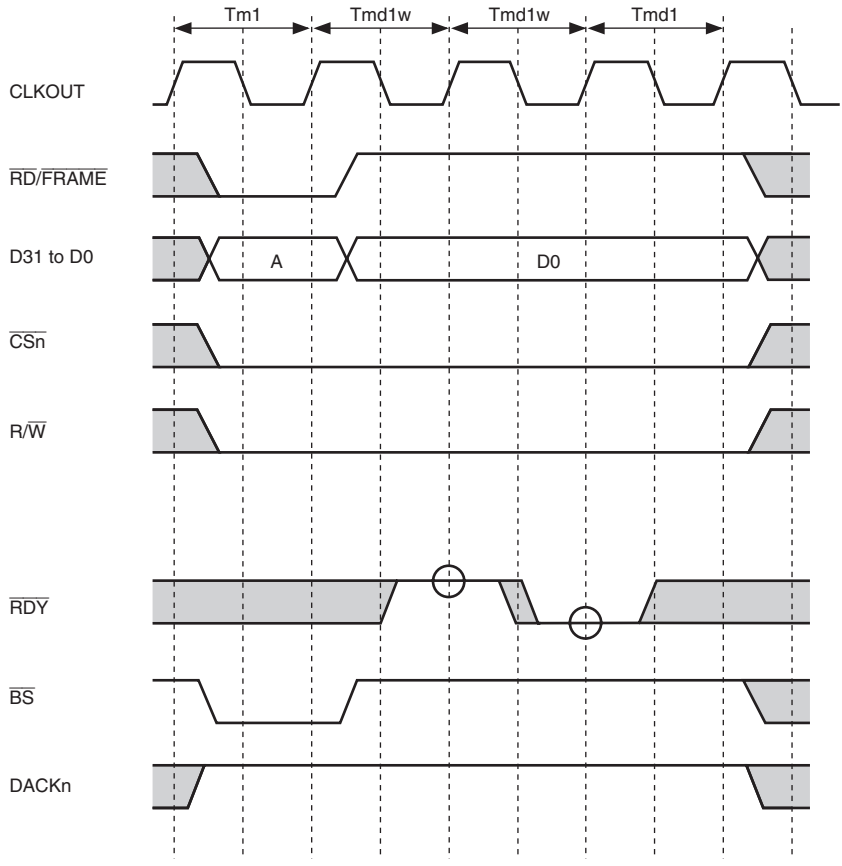
Note: In this example, DACKn is active-high. The circles indicate the sampling timing.

Figure 11.24 MPX Interface Timing 2
(Single Read, IW = 0000, One External Wait Inserted, 32-Bit Bus Width)



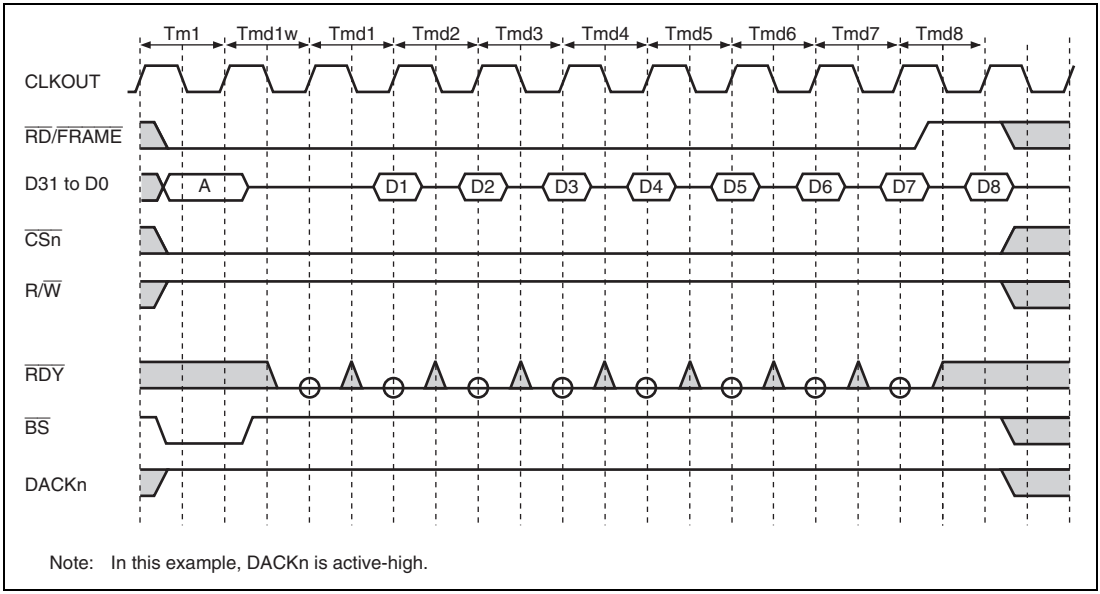
Note: In this example, DACKn is active-high. The circle indicates the sampling timing.

Figure 11.25 MPX Interface Timing 3
(Single Write Cycle, IW = 0000, No External Wait, 32-Bit Bus Width)



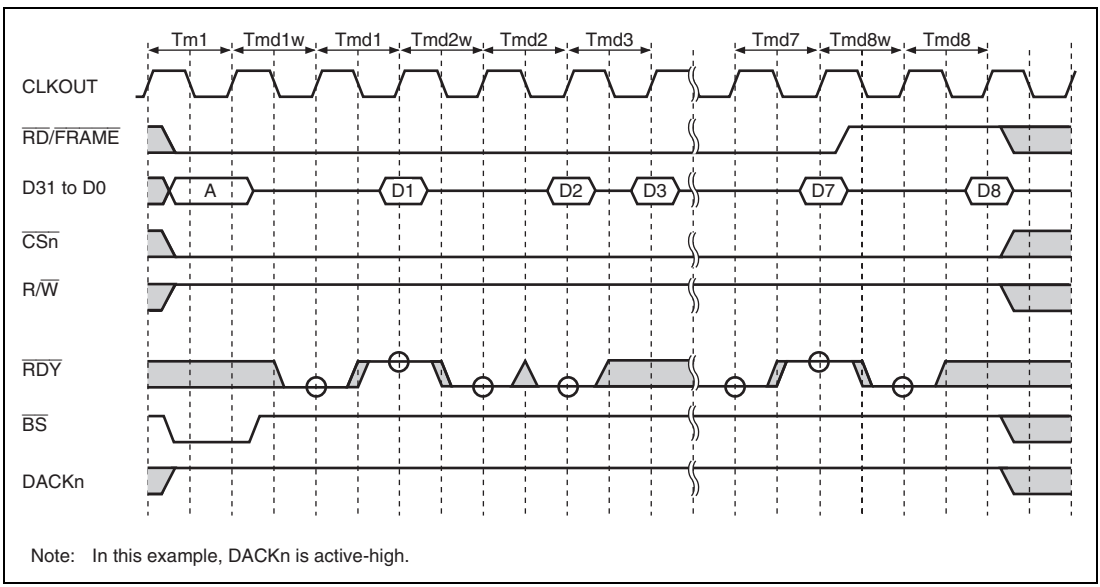
Note: In this example, DACKn is active-high.

Figure 11.26 MPX Interface Timing 4
(Single Write Cycle, IW = 0001, One External Wait Inserted, 32-Bit Bus Width)



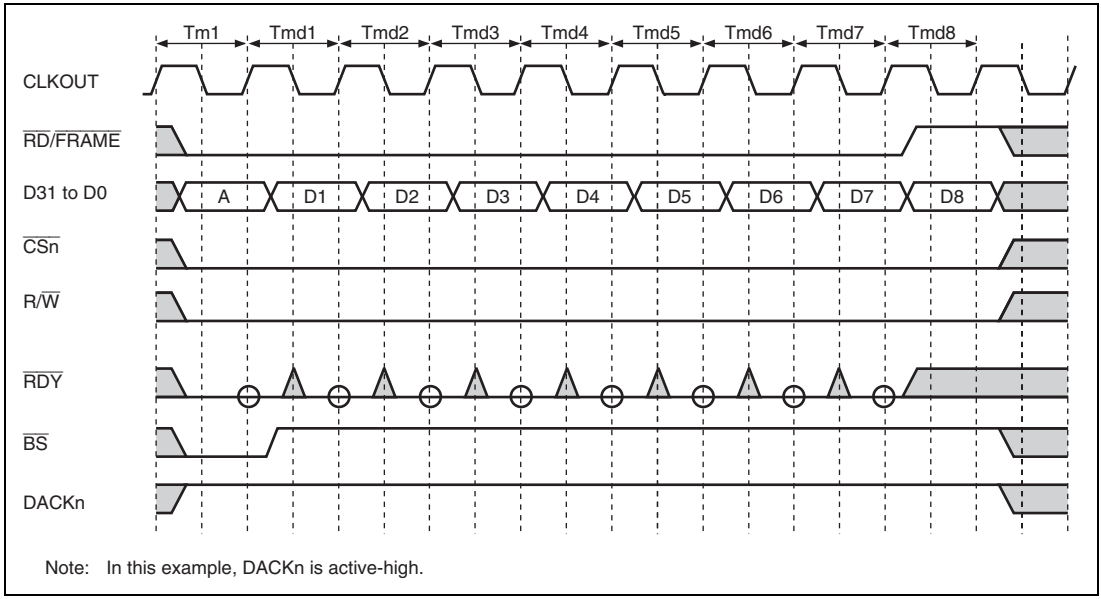
Note: In this example, DACKn is active-high.

Figure 11.27 MPX Interface Timing 5 (Burst Read Cycle, IW = 0000, No External Wait, 32-Bit Bus Width, 32-Byte Data Transfer)



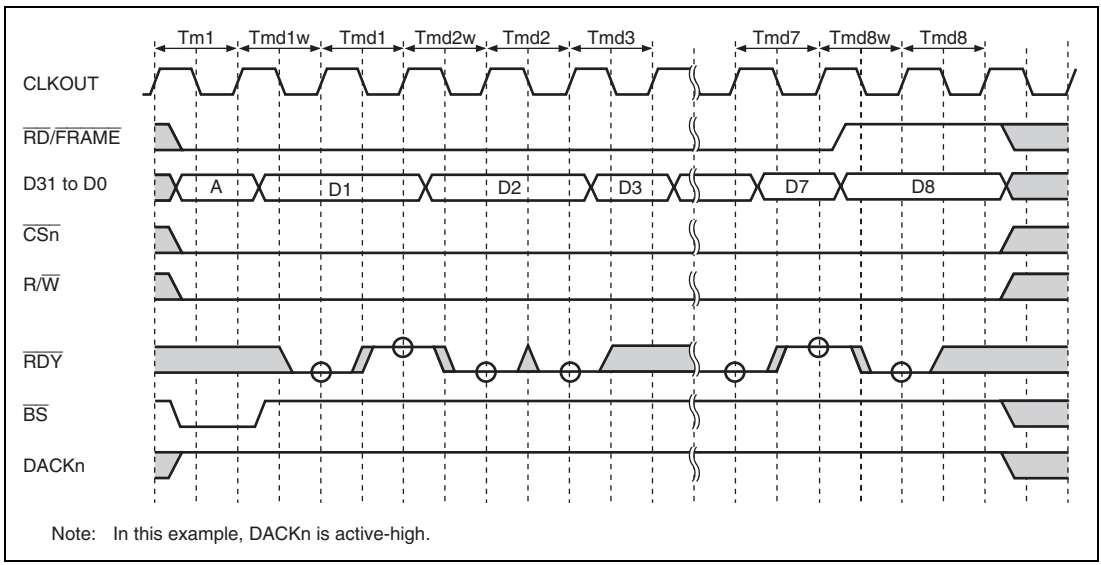
Note: In this example, DACKn is active-high.

Figure 11.28 MPX Interface Timing 6 (Burst Read Cycle, IW = 0000, External Wait Control, 32-Bit Bus Width, 32-Byte Data Transfer)



Note: In this example, DACKn is active-high.

Figure 11.29 MPX Interface Timing 7 (Burst Write Cycle, IW = 0000, No External Wait, 32-Bit Bus Width, 32-Byte Data Transfer)



Note: In this example, DACKn is active-high.

Figure 11.30 MPX Interface Timing 8 (Burst Write Cycle, IW = 0001, External Wait Control, 32-Bit Bus Width, 32-Byte Data Transfer)

11.5.7 Byte Control SRAM Interface

The byte control SRAM interface is a memory interface that outputs a byte-select strobe ($\overline{\text{WEn}}$) in both read and write bus cycles. This interface has 16-bit data pins and can be connected to SRAM having an upper byte select strobe and lower select strobe functions such as UB and LB.

Areas 1 and 4 can be specified as a byte control SRAM interface.

The write timing for the byte control SRAM interface is identical to that of a normal SRAM interface.

In reading operation, on the other hand, the $\overline{\text{WEn}}$ pin timing is different. In a read access, only the $\overline{\text{WEn}}$ signal for the byte being read is asserted. Assertion is synchronized with the falling edge of the CLKOUT clock in the same way as for the $\overline{\text{WEn}}$ signal, while negation is synchronized with the rising edge of the CLKOUT clock in the same way as for the $\overline{\text{RD}}$ signal.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wraparound method according to the set bus width. The bus is not released during this transfer.

Figures 11.31 shows examples of byte control SRAM connections, and figures 11.32 to 11.34 show examples of byte-control SRAM read cycles.

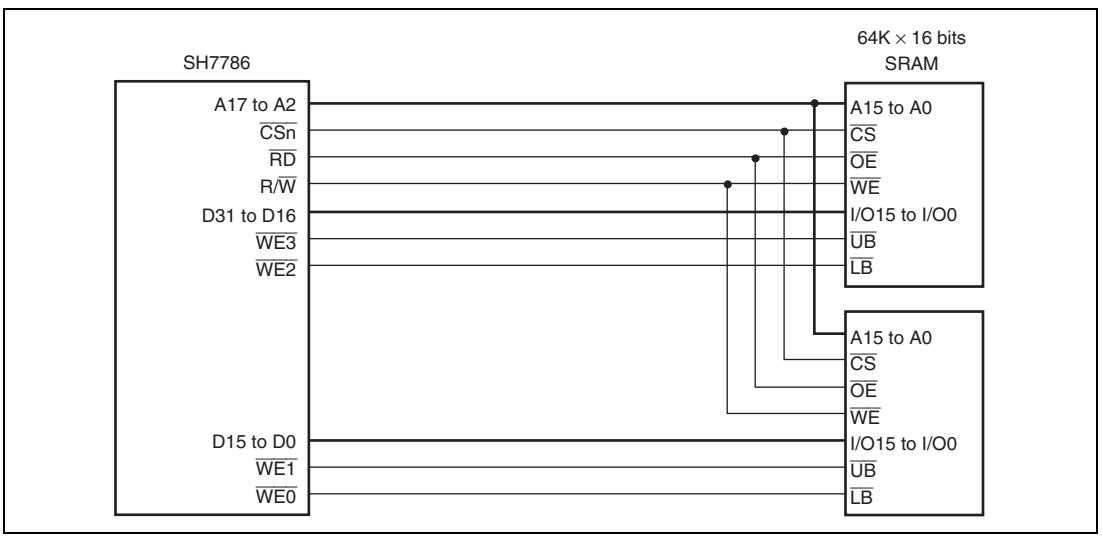
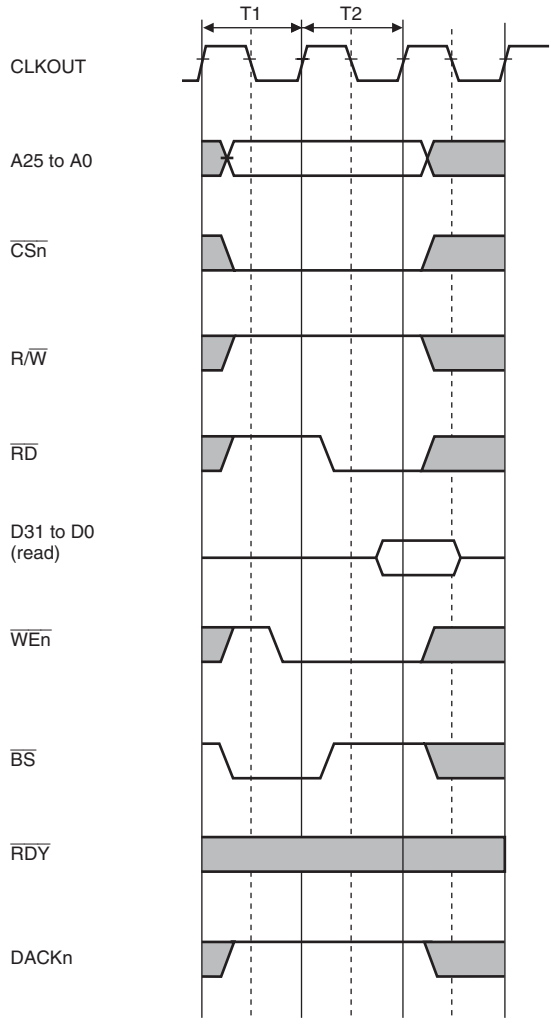
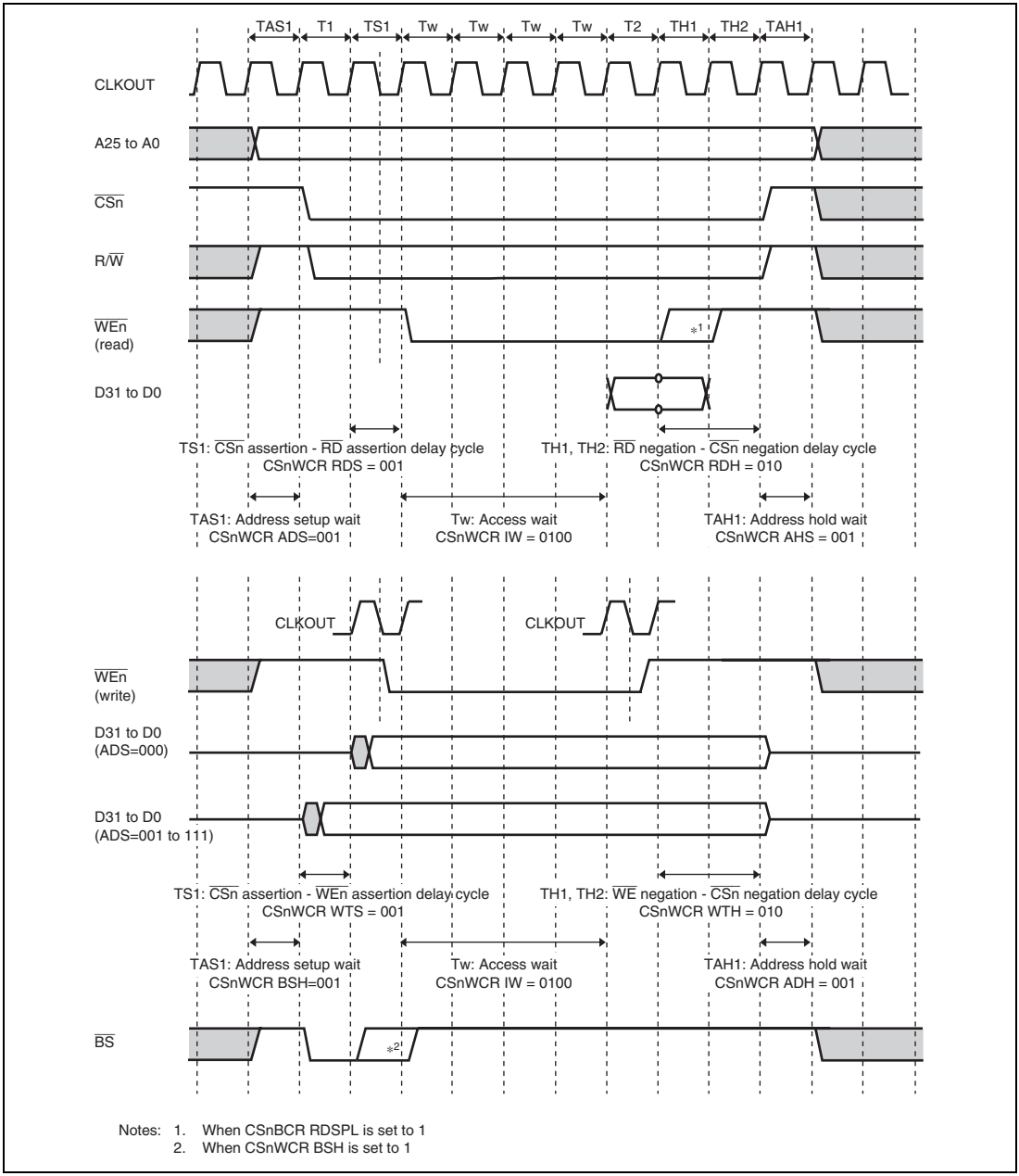


Figure 11.31 Example of Byte Control SRAM with 32-Bit Data Width



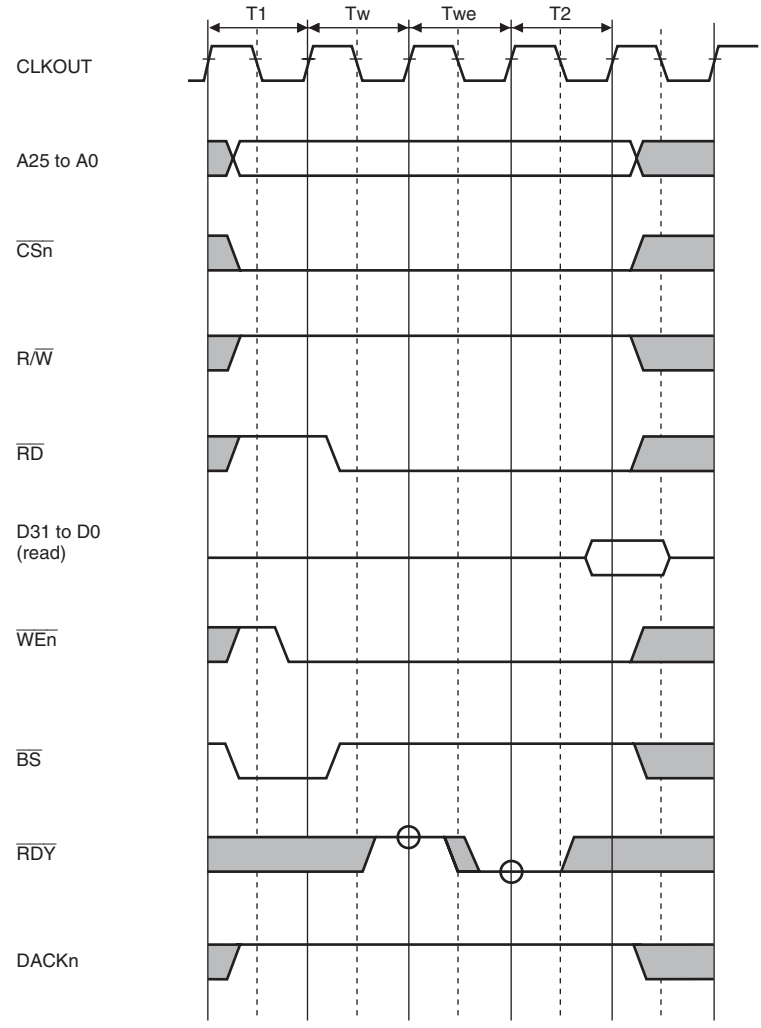
Note: In this example, DACKn is active-high.

Figure 11.32 Basic Read Cycle of Byte Control SRAM (No Wait)



Notes: 1. When CSnBCR RDSPL is set to 1
2. When CSnWCR BSH is set to 1

Figure 11.33 Wait State Timing of Byte Control SRAM



Note: In this example, DACKn is active-high.

Figure 11.34 Wait State Timing of Byte Control SRAM (One Internal Wait + One External Wait)

11.5.8 Wait Cycles between Access Cycles

When the external memory bus operating frequency is high, the turn-off of the data buffer performed on completion of reading from a low-speed device may not be made in time. This cause a collision with the next access data or a malfunction, which results in lower reliability. To prevent this problem, the data collision prevention function is provided. With this function, the preceding access area and the type of read/write are stored and a wait cycle is inserted before the access cycle if there is a possibility that a bus collision occurs when the next access is started. As an example of wait cycle insertion, idle cycles are inserted between the access cycles as shown in section 11.4.3, CSn Bus Control Register (CSnBCR). By using bits IWW, IWRWD, IWRWS, IWRRD and IWRRS in CSnBCR, at least the specified number of cycles can be inserted as idle cycles.

When bus arbitration is performed, the bus is released after wait cycles are inserted between the cycles.

When DMA transfer is performed in dual address mode, wait cycles are inserted as set in CSnBCR idle cycle bits.

When consecutive accesses to the MPX interface area are performed after a read access, 1 wait cycle is inserted even if the wait cycle is set to 0.

When the access size is 8-byte or 16-byte, wait cycles are inserted every 4-byte access.

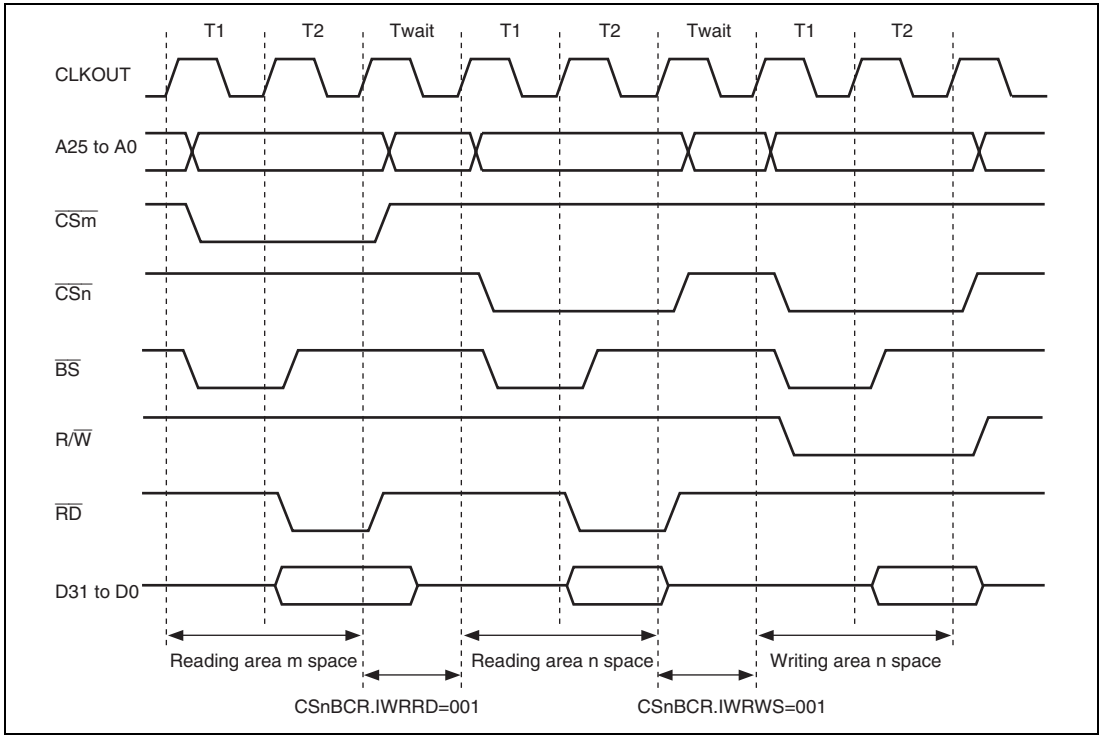


Figure 11.35 Wait Cycles between Access Cycles (Access Size is 4 Bytes)

11.5.9 Bus Arbitration

This LSI is provided with a bus arbitration function that gives the bus to an external device when a request is issued from the device.

While no bus access is requested from the external devices, the arbiter of this LSI keeps holding the access right to the bus, and by the request from the external devices, the arbiter releases and grants the access right to one of the devices. It is possible to connect an external device that issues bus requests. In the following description, an external device that issues bus requests is called a slave.

This LSI has nine internal bus masters, the CPU0, CPU1, DMAC0, DMAC2, HPB-DMAC, DU, USB, Ether, and PCIe. In addition to them, bus requests from external devices are issued. If requests occur simultaneously, higher priority is given to the external devices, and the round-robin scheme is adopted for the internal bus masters.

To prevent incorrect operation of connected devices when the bus is transferred between master and slave, all bus control signals are negated before the bus is released. In addition, when the bus mastership is received, bus control signals begin driving the bus from the negated state. Since the same signals are driven by the master and slave that exchange the bus, output buffer collisions can be avoided.

Bus transfer is executed between bus cycles.

When the bus release request signal ($\overline{\text{BREQ}}$) is asserted, this LSI releases the bus as soon as the currently executing bus cycle ends, and outputs the bus use permission signal ($\overline{\text{BACK}}$). However, bus release is not performed during multiple bus cycles generated because the data bus width is smaller than the access size (for example, when performing longword access to 8-bit bus width memory) or during a 32-byte transfer such as a cache fill or write-back. In addition, bus release is not performed between read and write cycles during execution of a TAS instruction When $\overline{\text{BREQ}}$ is negated, $\overline{\text{BACK}}$ is negated and use of the bus is resumed.

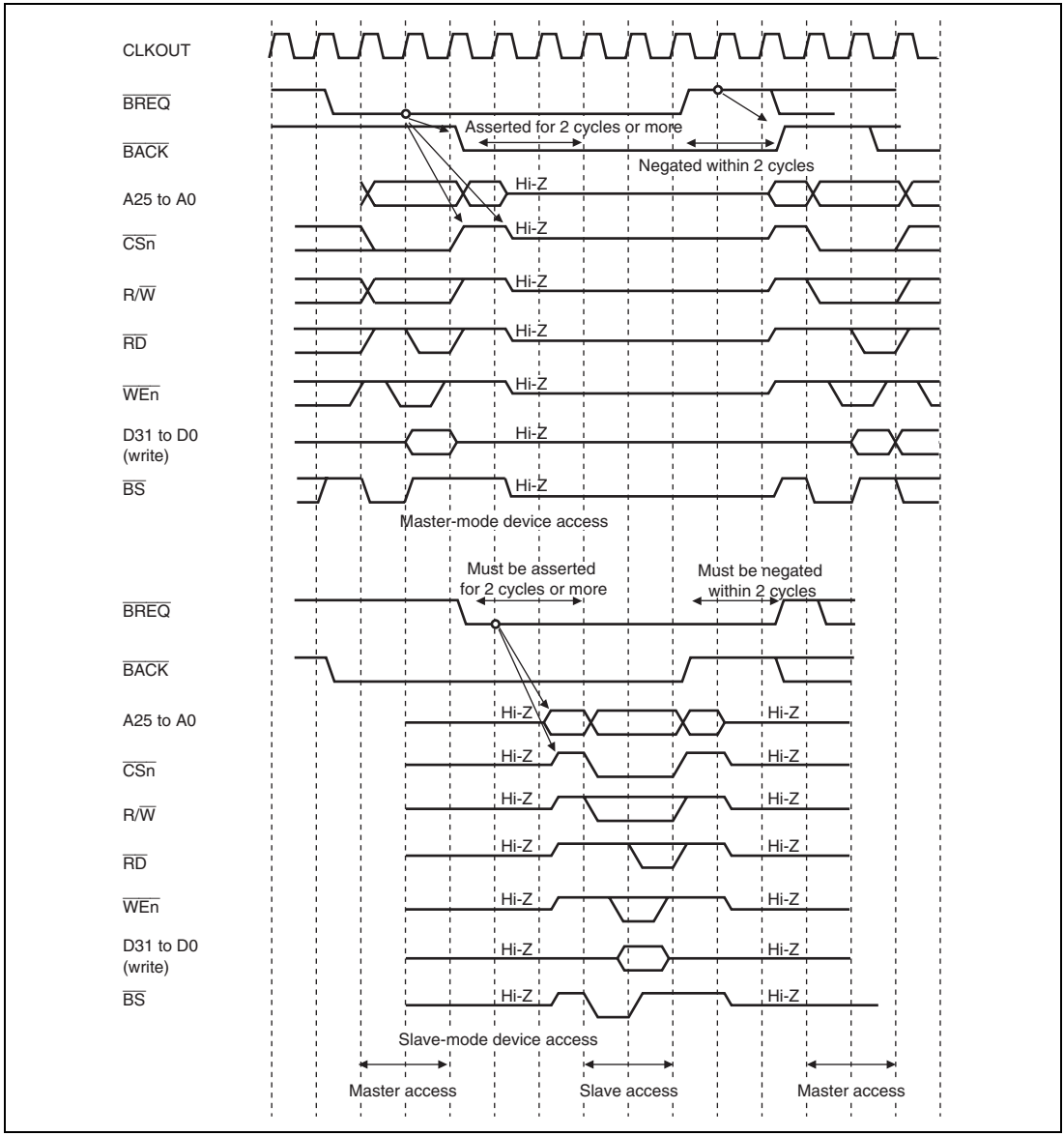


Figure 11.36 Arbitration Sequence

11.5.10 Master Mode

The processor in master mode holds the bus itself until it receives a bus request.

On receiving an assertion (low level) of the bus request signal ($\overline{\text{BREQ}}$) from the outside, the master mode processor releases the bus and asserts (drives low) the bus use permission signal ($\overline{\text{BACK}}$) as soon as the currently executing bus cycle ends. On receiving the $\overline{\text{BREQ}}$ negation (high level) indicating that the slave has released the bus, the processor negates (drives high) the $\overline{\text{BACK}}$ signal and resumes use of the bus.

When the bus is released, all bus control output signals and input/output signals related to bus interface enters a high-impedance state, except for $\overline{\text{BACK}}$ for bus arbitration and $\overline{\text{DACK0}}$ to $\overline{\text{DACK3}}$ for controlling DMA transfer.

The actual bus release sequence is as follows.

First, the bus use permission signal is asserted in synchronization with the rising edge of the clock. The address bus and data bus are put in a high-impedance state in synchronous with the rising edge of the clock next to the $\overline{\text{BACK}}$ assertion. At the same time, the bus control signals ($\overline{\text{BS}}$, $\overline{\text{CSn}}$, $\overline{\text{WEn}}$, $\overline{\text{RD}}$, R/$\overline{\text{W}}$, $\overline{\text{CE2A}}$, and $\overline{\text{CE2B}}$) enters a high-impedance state. These bus control signals are negated no later than one cycle before entering high-impedance. Bus request signal sampling is performed on the rising edge of the clock.

The sequence for re-acquiring the bus from the slave is as follows.

As soon as $\overline{\text{BREQ}}$ negation is detected on the rising edge of the clock, $\overline{\text{BACK}}$ is negated and bus control signal driving is started. Driving of the address bus and data bus starts at the next rising edge of an in-phase clock. The bus control signals are asserted and the bus cycle is actually started, at the earliest, at the clock rising edge at which the address and data signals are driven.

In order to reacquire the bus and start execution of bus access, the $\overline{\text{BREQ}}$ signal must be negated for at least two cycles.

Note: The SH7786 supports only master mode. Slave mode can not be used.

11.5.11 Cooperation between Master and Slave

To control system resources without contradiction by the master and slave, their respective roles must be clearly defined, as well as in the standby state implementing power-down mode.

The design of the SH7786 provides for all control, including initialization, and standby control, to be carried out by the master mode device.

If the SH7786 is specified as the master at a power-on reset, it will not accept bus requests from the slave until the $\overline{\text{BREQ}}$ enable bit (BREQEN in BCR) is set to 1.

To ensure that the slave processor does not access memory requiring initialization before completion of initialization, write 1 to the $\overline{\text{BREQ}}$ enable bit after the initialization is complete.

11.5.12 Pins Multiplexed with Other Modules Functions

Some pins used by the LBSC are multiplexed with general input output port (GPIO) and functions used in other peripheral modules. The pins to be used by the LBSC should start access after setting these pins to the LBSC functions with GPIO register. For example, when PCMCIA interface is used, the functions of CE2A and CE2B should be enabled with the GPIO bit in P2MSELR and the GPIO bit in PJCR before starting access.

11.5.13 Register Settings for Divided-Up $\overline{\text{DACKn}}$ Output

When the access size of DMAC1 transfer related to the local bus space is larger than the data bus width, multiple bus cycles are generated. When multiple bus cycles are generated and $\overline{\text{CS}}$ is negated between bus cycles, $\overline{\text{DACKn}}$ output is divided up, in the same way as $\overline{\text{CS}}$.

Tables 11.15 to 11.18 shows the register settings when $\overline{\text{DACKn}}$ output is not divided up in DMA1 transfer and when it is divided up.

Table 11.15 Register Settings for Divided-Up $\overline{\text{DACKn}}$ Output in DMA1 Transfer Using the SRAM/Burst ROM/Byte Control SRAM Interfaces

Bus Width [Bit]	Access Size in DMA Transfer	Bus Cycle Number	Not Divided		Divided
			IWRRD, IWRRS, or IWW in CSnBCR	ADS and ADH in CSnWCR.	ADS and ADH inCSnWCR
8	Byte	1	—	—	Undividable
	Word	2	—	B'000	B'111 to B'001
	Longword	4	—	B'000	B'111 to B'001
	16 bytes	16	B'000	B'000	B'111 to B'001
	32 bytes	32	—	B'000	B'111 to B'001
16	Byte	1	—	—	Undividable
	Word	1	—	—	Undividable
	Longword	2	—	B'000	B'111 to B'001
	16 bytes	8	B'000	B'000	B'111 to B'001
	32 bytes	16	—	B'000	B'111 to B'001
32	Byte	1	—	—	Undividable
	Word	1	—	—	Undividable
	Longword	1	—	—	Undividable
	16 bytes	4	B'000	B'000	B'111 to B'001
	32 bytes	8	—	B'000	B'111 to B'001

Note: "—" means an arbitrary setting value. When transfer is done in a single bus cycle, $\overline{\text{DACKn}}$ is not divided up because $\overline{\text{DACKn}}$ is output once in DMA1 transfer.

Table 11.16 Register Settings for Divided-Up $\overline{\text{DACKn}}$ Output in DMA1 Transfer Using the PCMCIA Interface

			Not Divided	Divided
Bus Width [Bit]	**Access Size**	**Bus Cycle Number**	**IWRRD, IWRRS, or IWW in CSnBCR**	**IWRRD, IWRRS, or IWW in CSnBCR**
8	Byte	1	—	Undividable
	Word	2	—	Undividable*[1]
	Longword	4	—	Undividable*[1]
	16 bytes	16	B'000	B'111 to B'001*[2]
	32 bytes	32	—	Undividable*[1]
16	Byte	1	—	Undividable
	Word	1	—	Undividable
	Longword	2	—	Undividable*[1]
	16 bytes	8	B'000	B'111 to B'001*[2]
	32 bytes	16	—	Undividable*[1]

Notes: "—" means an arbitrary setting value. When transfer is done in a single bus cycle, $\overline{\text{DACKn}}$ is not divided up because $\overline{\text{DACKn}}$ is output once in DMA1 transfer.

1. Multiple bus cycles are generated, however, $\overline{\text{DACKn}}$ cannot be divided.
2. Can be divided only in longword units.

Table 11.17 Register Settings for Divided-Up $\overline{\text{DACKn}}$ Output in DMA1 Transfer in Read Access Using the MPX Interface

			Not Divided	Divided
Bus Width [Bit]	**Access Size**	**Bus Cycle Number**	**IWRRD or IWRRS, in CSnBCR**	**IWRRD or IWRRS, in CSnBCR**
32	Byte	1	—	Undividable
	Word	1	—	Undividable
	Longword	1	—	Undividable
	16 bytes	4	Must be divided	—
	32 bytes	1	—	Undividable

Note: "—" means an arbitrary setting value. When transfer is done in a single bus cycle, $\overline{\text{DACKn}}$ is not divided up because $\overline{\text{DACKn}}$ is output once in DMA1 transfer.

Table 11.18 Register Settings for Divided-Up $\overline{\text{DACKn}}$ Output in DMA1 Transfer in Write Access Using the MPX Interface

			Not Divided		Divided
Bus Width [Bit]	**Access Size**	**Bus Cycle Number**	**IWW in CSnBCR**	**IW1 and IW0 in CSnWCR**	**IWW in CSnBCR**
32	Byte	1	—	—	Undividable
	Word	1	—	—	Undividable
	Longword	1	—	—	Undividable
	16 bytes	4	B'000	B'11 to B'01	B'111 to B'001
	32 bytes	1	—	—	Undividable

Note: "—" means an arbitrary setting value. When transfer is done in a single bus cycle, $\overline{\text{DACKn}}$ is not divided up because $\overline{\text{DACKn}}$ is output once in DMA1 transfer.

Section 12 DDR3-SDRAM Interface (DBSC3)

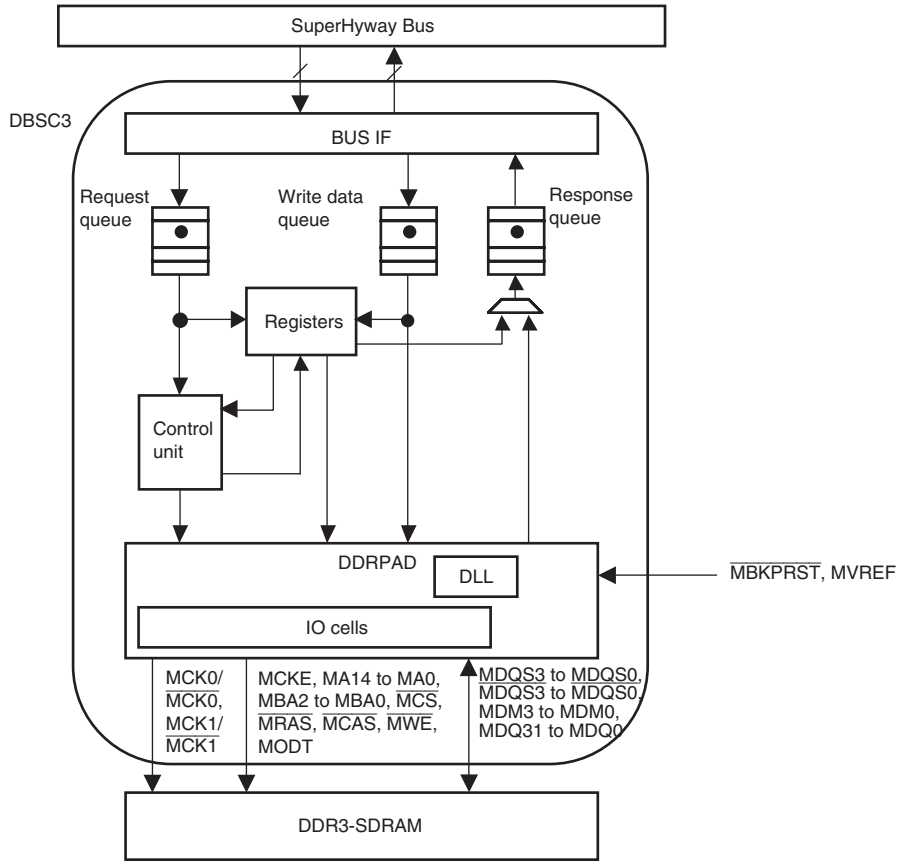
The DDR3-SDRAM interface (DBSC3) controls the DDR3-SDRAM.

12.1 Features

- Supports 32-bit external data bus widths
- Supports DDR3-1066 (controller operation at 533 MHz).
- Supports 1:1 clock ratio between SuperHyway clock and DDR clock.
- Queue provided for interface with SuperHyway.
- During power-on reset, pin MODE8 can be used to switch between big- and little-endian.
- Supports 8-bank DDR3-SDRAMs (multi-bank operation with up to 8 banks).
- Supports sequential mode with burst length 8.
- Supports Additive Latency (AL) of 0 only.
- Supports self-refresh.
- Supports power supply backup mode.
- Supported DDR3-SDRAM addresses × bit widths (total capacity) are as follows.
  For details, refer to tables 12.11 through 12.14.

  — DDR3-SDRAM data bus width: 32 bits
    - Two 512-Mbit (32 M × 16 bits) devices connected in parallel
      (total capacity = 128 Mbytes)
    - Four 512-Mbit (64 M × 8 bits) devices connected in parallel
      (total capacity = 256 Mbytes)
    - Two 1-Gbit (64 M × 16 bits) devices connected in parallel
      (total capacity = 256 Mbytes)
    - Four 1-Gbit (128 M × 8 bits) devices connected in parallel
      (total capacity = 512 Mbytes)
    - Two 2-Gbit (128 M × 16 bits) devices connected in parallel
      (total capacity = 512 Mbytes)
    - Four 2-Gbit (256 M × 8 bits) devices connected in parallel (total capacity = 1 Gbyte)
    - Two 4-Gbit (256 M × 16 bits) devices connected in parallel (total capacity = 1 Gbyte)
    - Four 4-Gbit (512 M × 8 bits) devices connected in parallel (total capacity = 2 Gbytes)

Figure 12.1 shows a block diagram of the DBSC3.



Request queue: Stores the access request of the SuperHyway bus.
Write data queue: Stores the write data sent from the SuperHyway bus.
Response queue: Stores the read data to be sent back to the SuperHyway bus.
Control unit: Controls each block depending on the request sent from the request queue.
Registers: Store timing parameters and SDRAM configuration information.
DDRPAD: Interfaces with the DDR3-SDRAM. This incorporates the DLL to perform phase adjustment of the DQS.

Figure 12.1 Block Diagram of the DBSC3

12.2 Input/Output Pins

Table 12.1 shows the pin configuration of the DBSC3.

Table 12.1 Pin Configuration of the DBSC3

Pin Name	Function	I/O	Description
MCK0	DDR3-SDRAM clock 0	Output	Clock output for the DDR3-SDRAM
$\overline{\text{MCK0}}$	DDR3-SDRAM clock 0	Output	Clock output for the DDR3-SDRAM or MCK0 inverted clock output
MCK1	DDR3-SDRAM clock 1	Output	Clock output for the DDR3-SDRAM
$\overline{\text{MCK1}}$	DDR3-SDRAM clock 1	Output	Clock output for the DDR3-SDRAM or MCK1 inverted clock output
MCKE0 to MCKE1	Clock enable	Output	Clock enable output signal
$\overline{\text{MCS0}}$ to $\overline{\text{MCS1}}$	Chip select	Output	Chip select output signal
$\overline{\text{MWE}}$	Write enable	Output	Write enable output signal
$\overline{\text{MRAS}}$	Row address strobe	Output	Row address strobe output signal
$\overline{\text{MCAS}}$	Column address strobe	Output	Column address strobe output signal
MA15 to MA0	Addresses	Output	Address output signals
MBA2 to MBA0	Bank active	Output	Bank address output signal
MDQ31 to MDQ0	Data	I/O	Data I/O signals
MDQS3 to MDQS0	I/O data strobe	I/O	Data strobe I/O signals
$\overline{\text{MDQS3}}$ to $\overline{\text{MDQS0}}$	I/O data strobe	I/O	Data strobe I/O signals or MDQS3 to MDQS0 inverted signals
MDM3 to MDM0	Data mask	Output	Data mask output signals
MODT0 to MODT1	ODT enable	Output	ODT enable output signal in the SDRAM
$\overline{\text{MRESET}}$	Reset	Output	Reset output for the DDR3-SDRAM
$\overline{\text{MBKPRST}}$	Power backup reset	Input	Used in power backup mode. When this pin is driven low level, CKE also goes low level.
SDBUP	Power backup mode	Input	This pin is used to fix $\overline{\text{MRESET}}$ low level at power-on. This pin must be maintained at high level during normal operation and in DDR backup mode.
MVREF	Reference voltage input	—	Input reference voltage
MZQ	Impedance matching	—	Impedance matching

The frequency of the SDRAM operating clock MCK0, $\overline{\text{MCK0}}$, MCK1, and $\overline{\text{MCK1}}$ is the same as the frequency of the DDR clock.

MDQ7 to MDQ0 correspond to MDQS0 and MDM0, MDQ15 to MDQ8 correspond to MDQS1 and MDM1, MDQ23 to MDQ16 correspond to MDQS2 and MDM2, and MDQ31 to MDQ24 correspond to MDQS3 and MDM3.

Table 12.2 shows an example of connections when a total of four 2-Gb DDR3-SDRAM units (256 M × 8 bits) are used, with the external data bus width set to 32 bits. Command-related signals (MCKE, $\overline{\text{MWE}}$, $\overline{\text{MCS}}$, $\overline{\text{MRAS}}$, $\overline{\text{MCAS}}$, MA14 to MA0, MBA2 to MBA0) are connected in common to four DDR3-SDRAM units. Data signals (MDQ31 to MDQ0, MDQS3 to MDQS0, $\overline{\text{MDQS3}}$ to $\overline{\text{MDQS0}}$, and MDM3 to MDM0) are connected to memory in 8-bit units. Clocks MCK1 and $\overline{\text{MCK1}}$ are connected to DDR3-SDRAM corresponding to the data signal upper sides (MDQ31 to MDQ16, MDQS3, MDQS2, $\overline{\text{MDQS3}}$, $\overline{\text{MDQS2}}$, MDM3, and MDM2), and MCK0 and $\overline{\text{MCK0}}$ are connected to DDR3-SDRAM corresponding to the lower sides (MDQ15 to MDQ0, MDQS1, MDQS0, $\overline{\text{MDQS1}}$, $\overline{\text{MDQS0}}$, MDM1, and MDM0). Address pins MA14 to MA0 should be connected to the DDR3-SDRAM address pins, without swapping the order.

Nothing should be connected to unused pins.

Table 12.2 An Example of DDR3-SDRAM Connection (When Four 2-Gbit DDR3-SDRAM Units (256 M × 8 Bits) Are Used)

Memory	MCK1, $\overline{\text{MCK1}}$, $\overline{\text{MCS1}}$, MCKE1, MODT1	MCK0, $\overline{\text{MCK0}}$, $\overline{\text{MCS0}}$, MCKE0, MODT0	MRAS, MCAS, MWE, MA14 to MA0, MBA2 to MBA0	MDQ31 to MDQ24, MDQS3, $\overline{\text{MDQS3}}$, MDM3	MDQ23 to MDQ16, MDQS2, $\overline{\text{MDQS2}}$, MDM2	MDQ15 to MDQ8, MDQS1, $\overline{\text{MDQS1}}$, MDM1	MDQ7 to MDQ0, MDQS0, $\overline{\text{MDQS0}}$, MDM0
Memory #1	Connected*[1]		Connected*[2]	Connected*[3]			
Memory #2	Connected*[1]		Connected*[2]		Connected*[4]		
Memory #3		Connected*[1]	Connected*[2]			Connected*[5]	
Memory #4		Connected*[1]	Connected*[2]				Connected*[6]
Memory #1	Connected*[1]		Connected*[2]	Connected*[3]			

Notes: 1. SDRAM pins should be connected as shown below.

Memory #1 and #2 Pins	SH7786 Pins	Memory #3 and #4 Pins	SH7786 Pins
CK	MCK1	CK	MCK0
$\overline{\text{CK}}$	$\overline{\text{MCK1}}$	$\overline{\text{CK}}$	$\overline{\text{MCK0}}$
$\overline{\text{CS}}$	$\overline{\text{MCS1}}$	$\overline{\text{CS}}$	$\overline{\text{MCS0}}$
ODT	MODT1	ODT	MODT0
CKE	MCKE1	CKE	MCK0

2. SDRAM pins should be connected as shown below.

Memory #1 to #4 Pins	SH7786 Pins	Memory #1 to #4 Pins	SH7786 Pins
$\overline{\text{RAS}}$	$\overline{\text{MRAS}}$	A7	MA7
$\overline{\text{CAS}}$	$\overline{\text{MCAS}}$	A6	MA6
$\overline{\text{WE}}$	$\overline{\text{MWE}}$	A5	MA5
A15	MA15	A4	MA4
A14	MA14	A3	MA3
A13	MA13	A2	MA2
A12	MA12	A1	MA1
A11	MA11	A0	MA0
A10	MA10	BA2	MBA2
A9	MA9	BA1	MBA1
A8	MA8	BA0	MBA0

3. SDRAM pins should be connected as shown below.

Memory #1 Pins	SH7786 Pins
DQS	MDQS3
$\overline{\text{DQS}}$	$\overline{\text{MDQS3}}$
DM	MDM3
DQ7	MDQ31
DQ6	MDQ30
DQ5	MDQ29
DQ4	MDQ28
DQ3	MDQ27
DQ2	MDQ26
DQ1	MDQ25
DQ0	MDQ24

4. SDRAM pins should be connected as shown below.

Memory #2 Pins	SH7786 Pins
DQS	MDQS2
$\overline{\text{DQS}}$	$\overline{\text{MDQS2}}$
DM	MDM2
DQ7	MDQ23
DQ6	MDQ22
DQ5	MDQ21
DQ4	MDQ20
DQ3	MDQ19
DQ2	MDQ18
DQ1	MDQ17
DQ0	MDQ16

5. SDRAM pins should be connected as shown below.

Memory #3 Pins	SH7786 Pins
DQS	MDQS1
$\overline{\text{DQS}}$	$\overline{\text{MDQS1}}$
DM	MDM1
DQ7	MDQ15
DQ6	MDQ14
DQ5	MDQ13
DQ4	MDQ12
DQ3	MDQ11
DQ2	MDQ10
DQ1	MDQ9
DQ0	MDQ8

6. SDRAM pins should be connected as shown below.

Memory #4 Pins	SH7786 Pins
DQS	MDQS0
$\overline{\text{DQS}}$	$\overline{\text{MDQS0}}$
DM	MDM0
DQ7	MDQ7
DQ6	MDQ6
DQ5	MDQ5
DQ4	MDQ4
DQ3	MDQ3
DQ2	MDQ2
DQ1	MDQ1
DQ0	MDQ0

12.3 Data Alignment

12.3.1 Data Alignment in DDR3-SDRAM

The DBSC3 accesses DDR3-SDRAM with a fixed burst length of 8 (figure 12.2). As shown in table 12.3, invalid read data is discarded during reading, and data mask signals are used to mask invalid data during writing, according to the access size. For example, in the case of little endian, the second access (falling edge of DQS) includes valid data if a byte access of address (8n + 0, 1, 2, 3) occurs.

Tables 12.4 and 12.5 show the correspondence with data on the external data bus for each access size. Endian switching is performed at power-on reset by switching using external pin MODE 8.

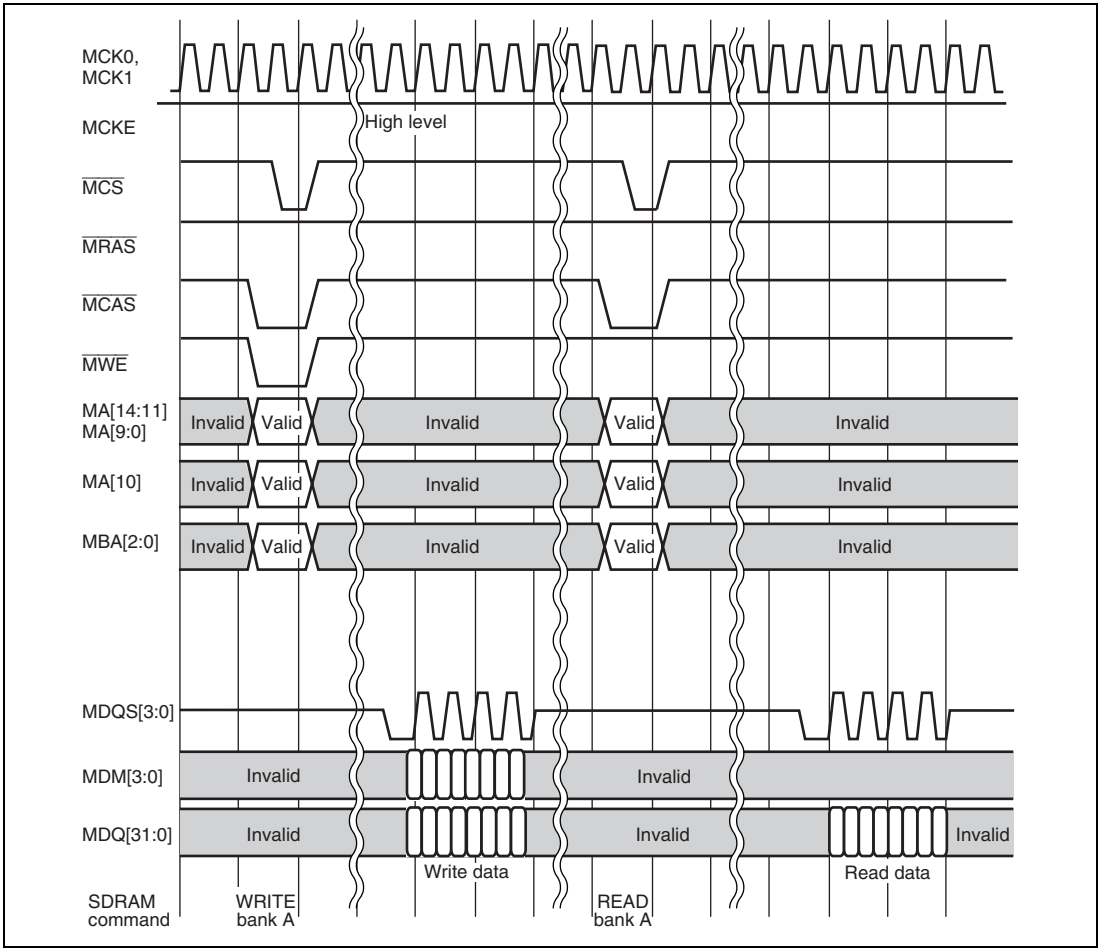


Figure 12.2 Burst Access Operation

Table 12.3 Positions of Valid Data for Access with Burst Length of 8

Little Endian	First Access	Second Access	Third Access	Fourth Access	Fifth Access	Sixth Access	Seventh Access	Eighth Access
Byte access (address 8n + 0,1,2,3)	Invalid	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Byte access (address 8n + 4,5,6,7)	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Word access (address 8n + 0,2)	Invalid	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Word access (address 8n + 4,6)	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Longword access (address 8n + 0)	Invalid	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Longword access (address 8n + 4)	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Quadword access (address 8n + 0)	Valid	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid

Big Endian	First Access	Second Access	Third Access	Fourth Access	Fifth Access	Sixth Access	Seventh Access	Eighth Access
Byte access (address 8n + 0,1,2,3)	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Byte access (address 8n + 4,5,6,7)	Invalid	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Word access (address 8n + 0,2)	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Word access (address 8n + 4,6)	Invalid	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Longword access (address 8n + 0)	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Longword access (address 8n + 4)	Invalid	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid
Quadword access (address 8n + 0)	Valid	Valid	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid

Table 12.4 Data Alignment for Access in Little Endian

Access Size	Address	MDQ31 to MDQ24	MDQ23 to MDQ16	MDQ15 to MDQ8	MDQ7 to MDQ0
Byte	Address 0				Data 7 to 0
	Address 1			Data 7 to 0	
	Address 2		Data 7 to 0		
	Address 3	Data 7 to 0			
	Address 4				Data 7 to 0
	Address 5			Data 7 to 0	
	Address 6		Data 7 to 0		
	Address 7	Data 7 to 0			
Word	Address 0			Data 15 to 8	Data 7 to 0
	Address 2	Data 15 to 8	Data 7 to 0		
	Address 4			Data 15 to 8	Data 7 to 0
	Address 6	Data 15 to 8	Data 7 to 0		
Longword	Address 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
	Address 4	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Quadword	Address 0 (First access: address 4)	Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32
	Address 0 (Second access: Address 0)	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0

Table 12.5 Data Alignment for Access in Big Endian

Access Size	Address	MDQ31 to MDQ24	MDQ23 to MDQ16	MDQ15 to MDQ8	MDQ7 to MDQ0
Byte	Address 0	Data 7 to 0			
	Address 1		Data 7 to 0		
	Address 2			Data 7 to 0	
	Address 3				Data 7 to 0
	Address 4	Data 7 to 0			
	Address 5		Data 7 to 0		
	Address 6			Data 7 to 0	
	Address 7				Data 7 to 0
Word	Address 0	Data 15 to 8	Data 7 to 0		
	Address 2			Data 15 to 8	Data 7 to 0
	Address 4	Data 15 to 8	Data 7 to 0		
	Address 6			Data 15 to 8	Data 7 to 0
Longword	Address 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
	Address 4	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Quadword	Address 0 (First access: Address 0)	Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32
	Address 0 (Second access: Address 4)	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0

12.4 Register Descriptions

Table 12.6 shows the DBSC3 register configuration, and table 12.7 shows register states in the different processing modes.

The register bit width is 32 bits, and the longword size (32 bits) should be used for register access. If registers are accessed with sizes other than the longword size, correct operation cannot be guaranteed.

The DBSC3 register area is, in P4 addresses, from H'FFA0 0000 to H'FFBF FFFF. If an address other than the register addresses indicated in table 12.6 is accessed, correct operation cannot be guaranteed.

Table 12.6 DBSC3 Register Configuration

Register Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size (Bits)	Synchroni-zation Clock
DBSC3 status register	DBSTATE	R	H'FFA0 000C	H'1FA0 000C	32	SHck
SDRAM access enable register	DBACEN	R/W	H'FFA0 0010	H'1FA0 0010	32	SHck
Auto-refresh enable register	DBRFEN	R/W	H'FFA0 0014	H'1FA0 0014	32	SHck
Manual command issuing register	DBCMD	R/W	H'FFA0 0018	H'1FA0 0018	32	SHck
Operation completion waiting register	DBWAIT	R/W	H'FFA0 001C	H'1FA0 001C	32	SHck
SDRAM kind setting register	DBKIND	R/W	H'FFA0 0020	H'1FA0 0020	32	SHck
SDRAM configuration setting register	DBCONF	R/W	H'FFA0 0024	H'1FA0 0024	32	SHck
SDRAM timing register 0	DBTR0	R/W	H'FFA0 0040	H'1FA0 0040	32	SHck
SDRAM timing register 1	DBTR1	R/W	H'FFA0 0044	H'1FA0 0044	32	SHck
SDRAM timing register 2	DBTR2	R/W	H'FFA0 0048	H'1FA0 0048	32	SHck

Register Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size (Bits)	Synchronization Clock
SDRAM timing register 3	DBTR3	R/W	H'FFA0 0050	H'1FA0 0050	32	SHck
SDRAM timing register 4	DBTR4	R/W	H'FFA0 0054	H'1FA0 0054	32	SHck
SDRAM timing register 5	DBTR5	R/W	H'FFA0 0058	H'1FA0 0058	32	SHck
SDRAM timing register 6	DBTR6	R/W	H'FFA0 005C	H'1FA0 005C	32	SHck
SDRAM timing register 7	DBTR7	R/W	H'FFA0 0060	H'1FA0 0060	32	SHck
SDRAM timing register 8	DBTR8	R/W	H'FFA0 0064	H'1FA0 0064	32	SHck
SDRAM timing register 9	DBTR9	R/W	H'FFA0 0068	H'1FA0 0068	32	SHck
SDRAM timing register 10	DBTR10	R/W	H'FFA0 006C	H'1FA0 006C	32	SHck
SDRAM timing register 11	DBTR11	R/W	H'FFA0 0070	H'1FA0 0070	32	SHck
SDRAM timing register 12	DBTR12	R/W	H'FFA0 0074	H'1FA0 0074	32	SHck
SDRAM timing register 13	DBTR13	R/W	H'FFA0 0078	H'1FA0 0078	32	SHck
SDRAM timing register 14	DBTR14	R/W	H'FFA0 007C	H'1FA0 007C	32	SHck
SDRAM timing register 15	DBTR15	R/W	H'FFA0 0080	H'1FA0 0080	32	SHck
SDRAM timing register 16	DBTR16	R/W	H'FFA0 0084	H'1FA0 0084	32	SHck
SDRAM timing register 17	DBTR17	R/W	H'FFA0 0088	H'1FA0 0088	32	SHck
Refresh configuration register 0	DBRFCNF0	R/W	H'FFA0 00E0	H'1FA0 00E0	32	SHck
Refresh configuration register 1	DBRFCNF1	R/W	H'FFA0 00E4	H'1FA0 00E4	32	SHck

Register Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size (Bits)	Synchronization Clock
Refresh configuration register 2	DBRFCNF2	R/W	H'FFA0 00E8	H'1FA0 00E8	32	SHck
PHY-unit control register 0	DBPDCNT0	R/W	H'FFA0 0200	H'1FA0 0200	32	SHck
PHY-unit control register 1	DBPDCNT1	R/W	H'FFA0 0204	H'1FA0 0204	32	SHck
PHY-unit control register 2	DBPDCNT2	R/W	H'FFA0 0208	H'1FA0 0208	32	SHck
PHY-unit control register 3	DBPDCNT3	R/W	H'FFA0 020C	H'1FA0 020C	32	SHck
PHY-unit lock register	DBPDLCK	R/W	H'FFA0 0280	H'1FA0 0280	32	SHck
PHY-unit internal register address register	DBPDRGA	R/W	H'FFA0 0290	H'1FA0 0290	32	SHck
PHY-unit internal register data register	DBPDRG	R/W	H'FFA0 02A0	H'1FA0 02A0	32	SHck
Bus control unit 0 control register 0	DBBS0CNT0	R/W	H'FFA0 0300	H'1FA0 0300	32	SHck
Bus control unit 0 control register 1	DBBS0CNT1	R/W	H'FFA0 0304	H'1FA0 0304	32	SHck

Table 12.7 Register Status in each Processing Mode

		Power-On Reset	Manual Reset	Sleep/Light Sleep
Register Name	**Abbreviation**	**By $\overline{\text{PRESET}}$ pin/ WDT/H-UDI**	**By WDT/Multiple Exception**	**By SLEEP Instruction**
DBSC3 status register	DBSTATE	H'0000 0x0x*	Retained	Retained
SDRAM access enable register	DBACEN	H'0000 0000	Retained	Retained
Auto-refresh enable register	DBRFEN	H'0000 0000	Retained	Retained
Manual command issuing register	DBCMD	H'0000 0000	Retained	Retained
Operation completion waiting register	DBWAIT	H'0000 0000	Retained	Retained
SDRAM kind setting register	DBKIND	H'0000 0000	Retained	Retained
SDRAM configuration setting register	DBCONF	H'0000 0000	Retained	Retained
SDRAM timing register 0	DBTR0	H'0000 0000	Retained	Retained
SDRAM timing register 1	DBTR1	H'0000 0000	Retained	Retained
SDRAM timing register 2	DBTR2	H'0000 0000	Retained	Retained
SDRAM timing register 3	DBTR3	H'0000 0000	Retained	Retained
SDRAM timing register 4	DBTR4	H'0000 0000	Retained	Retained
SDRAM timing register 5	DBTR5	H'0000 0000	Retained	Retained
SDRAM timing register 6	DBTR6	H'0000 0000	Retained	Retained
SDRAM timing register 7	DBTR7	H'0000 0000	Retained	Retained
SDRAM timing register 8	DBTR8	H'0000 0000	Retained	Retained

Register Name	Abbreviation	Power-On Reset By $\overline{\text{PRESET}}$ pin/ WDT/H-UDI	Manual Reset By WDT/Multiple Exception	Sleep/Light Sleep By SLEEP Instruction
SDRAM timing register 9	DBTR9	H'0000 0000	Retained	Retained
SDRAM timing register 10	DBTR10	H'0000 0000	Retained	Retained
SDRAM timing register 11	DBTR11	H'0000 0000	Retained	Retained
SDRAM timing register 12	DBTR12	H'0000 0000	Retained	Retained
SDRAM timing register 13	DBTR13	H'0000 0000	Retained	Retained
SDRAM timing register 14	DBTR14	H'0000 0000	Retained	Retained
SDRAM timing register 15	DBTR15	H'0000 0000	Retained	Retained
SDRAM timing register 16	DBTR16	H'0000 0000	Retained	Retained
SDRAM timing register 17	DBTR17	H'0000 0000	Retained	Retained
Refresh configuration register 0	DBRFCNF0	H'0000 0000	Retained	Retained
Refresh configuration register 1	DBRFCNF1	H'0000 0000	Retained	Retained
Refresh configuration register 2	DBRFCNF2	H'0000 0000	Retained	Retained
PHY-unit control register 0	DBPDCNT0	H'0000 0000	Retained	Retained
PHY-unit control register 1	DBPDCNT1	H'0000 0000	Retained	Retained
PHY-unit control register 2	DBPDCNT2	H'0000 0000	Retained	Retained
PHY-unit control register 3	DBPDCNT3	H'0000 0000	Retained	Retained

Register Name	Abbreviation	Power-On Reset By $\overline{\text{PRESET}}$ pin/ WDT/H-UDI	Manual Reset By WDT/Multiple Exception	Sleep/Light Sleep By SLEEP Instruction
PHY-unit lock register	DBPDLCK	H'0000 0000	Retained	Retained
PHY-unit internal register address register	DBPDRGA	H'0000 0000	Retained	Retained
PHY-unit internal register data register	DBPDRG	H'0000 0000	Retained	Retained
Bus control unit 0 control register 0	DBBS0CNT0	H'0000 0000	Retained	Retained
Bus control unit 0 control register 1	DBBS0CNT1	H'0000 0000	Retained	Retained

Note: * Initial value is specified by the external MODE8 and SDBUP pins.

12.4.1 DBSC3 Status Register (DBSTATE)

The DBSC3 status register (DBSTATE) is a read-only register. Writing is invalid. It is only initialized at the time of a power-on reset.

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

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

Note: * Initial value is specified by external pin MODE8.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0.
9	—	Undefined	R	Reserved Undefined value is read from this bit.
8	ENDN	X*	R	Endian Display Bit Displays the endian set by external pin MODE8. 0: Big endian 1: Little endian
7 to 2	—	All 0	R	Reserved These bits are always read as 0.
1	BKUP	X*	R	Power-back-up monitor Bit Displays the power-backup state set by the external SDBUP pin. 0: Non-power-back-up state (SDBUP = "L") 1: Power-back-up state (SDBUPH = "H")
0	—	Undefined	R	Reserved Undefined value is read from this bit.

Note: * Initial value is specified by the state of the external MODE8 and SDBUP pins.

12.4.2 SDRAM Access Enable Register (DBACEN)

The SDRAM access enable register (DBACEN) is a readable/writable register. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
0	ACCEN	0	R/W	SDRAM Access Enable Bit By setting this bit, data accessing of SDRAM is enabled. When set to 0, access is disabled; when set to 1, access is enabled. When access is disabled, attempts to access SDRAM are ignored. This bit is used for the initialization sequence or self-refresh operation. 0: Disables access 1: Enables access

Note: When making the setting to disable access (writing 0 to this register), a Precharge All or Precharge command for the SDRAM may be issued automatically.

12.4.3 Auto-Refresh Enable Register (DBRFEN)

The auto-refresh enable register (DBRFEN) is a readable/writable register. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
0	ARFEN	0	R/W	Auto-Refresh Enable Bit This bit is used to enable or disable the automatic issuing of auto-refresh commands. Auto-refresh commands are issued at regular intervals based on the internal refresh counter while the value of this bit is 1. For details on the timing with which auto-refresh commands are issued, refer to section 12.5.5, Auto-Refresh Operation. 0: Disables automatic issuing of auto-refresh commands. 1: Enables automatic issuing of auto-refresh commands.

12.4.4 Manual Command-Issuing Register (DMCMD)

The manual command-issuing register (DMCMD) is used to issue the required commands for the sequence of initializing the SDRAM and of transitions to and from the self-refresh mode. The command corresponding to the OPC bits is issued once as a result of writing to this register. For instance, issuing the refresh command twice requires that "001100" be written to the OPC bits twice. Writing to this register must not proceed while access to SDRAM is enabled (ACCEN = 1 in the DBACEN register). The timing with which an operation is complete (i.e. the timing with which the specified SDRAM command is output to the PHY-unit from DBSC3) may be later than the response of DBSC3 to writing to this register. Reading from the DBWAIT register will result in the actual output of the specified SDRAM command to SDRAM.

When this register is used to issue command, the issuing of a subsequent SDRAM command is delayed by a certain period of time from the moment at which each operation is completed (i.e. from the time when the specified SDRAM command is output to SDRAM). This facilitates securing of the required amounts of time between commands in the issuing of multiple consecutive commands.

These periods are given in the "Interval" column of table 12.8. They can also be customized by using the ARG bits (except in cases where the OPC bits indicate MRS0 to MRS3, READ, or WRITE).

The DMCMD register is readable and writable. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
29 to 24	OPC	All 0	R/W	Operation Code Specify the type of command to be issued. Refer to the table below.
23 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
15 to 0	ARG	All 0	R/W	Parameter The meaning can differ according to the operation code indicated by the OPC bits. When OPC indicates ModeRegisterSet (MRS0 to MRS3), the ARG bits specify the value to be issued on the address pins (MA) of SDRAM. When OPC indicates READ/WRITE, the ARG bits are not used. When OPC is any other value, the ARG bits specify the minimum interval to issuing of the next command in SDRAM cycles. When ARG = 0, however, the values default to those given in "Interval" column of table 12.8.

Notes: 1. TRPA, TRFC and TMOD in the "Interval" column of table 12.8 indicate timing registers. The intervals in these cases are determined by the corresponding register settings.
2. Only write to this register after having disabled SDRAM access (ACCEN in DBACEN = 0).
3. Only write to this register after having stopped the auto-refresh function (ARFEN in DBRFEN register = 0). However, sequences described in section 12.5, DBSC3 Operation, are not limited to these. If OPC = Wait is written to this register during automatic refresh operations, the auto-refresh function may issue a refresh command during the time secured for the Wait command.

Table 12.8 Manual Command Issue Functions

OPC	Code	Operation	Interval (Number of SDRAM Cycles)	ARG Function
00 0000	Wait	Issue "Device Deselected" (and insert waiting time)	4	Customizing the interval (for ARG = 0, the value is that given at left)
00 0010	ZQCS	Issue "ZQ Calibration Short"	4	
00 0011	ZQCL	Issue "ZQ Calibration Long"	4	
00 1011	PreA	Issue "Precharge All"	TRPA	
00 1100	Ref	Issue "Refresh"	TRFC	
01 0000	PDEn	Power Down Entry	4	
01 0001	PDXt	Power Down Exit	4	
01 1000	SREn	Self-Refresh Entry	4	
01 1001	SRXt	Self-Refresh Exit	TRFC	
10 0000	RstL	Set Reset Pins to Low	4	
10 0001	RstH	Set Reset Pins to High	4	
10 1000	MRS0	Issue "ModeRegisterSet" (for MRS/MR0)”	TMOD	Specifying the setting for a mode register in SDRAM
10 1001	MRS1	Issue "ModeRegisterSet" (for EMRS1/MR1)	TMOD	
10 1010	MRS2	Issue "ModeRegisterSet" (for EMRS2/MR2)	TMOD	
10 1011	MRS3	Issue "ModeRegisterSet" (for EMRS3/MR3)	TMOD	
10 0100	Rd	Issue Read command	4	(reserved) Available only for ARG = 0
10 0110	Wr	Issue Write command	4	

12.4.5 Operation Completion Waiting Register (DBWAIT)

The operation completion waiting register (DBWAIT) is read-only. Writing is invalid. It is only initialized at the time of a power-on reset. When this register is read, a response is returned after all command issuing that has been specified by using the DBCMD register by that time is completed. This register can be used to guarantee correctness for the relationship between the timing with which SDRAM commands are issued by DBSC3 and timing that is not managed by DBSC3 (e.g. clock control).

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
0	WAIT	0	R	Operation Completion Waiting Bits These bits are always read as 0.

12.4.6 SDRAM Kind Setting Register (DBKIND)

The SDRAM kind setting register (DBKIND) is readable/writable register. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
2 to 0	DDCG	000	R/W	SDRAM Kind Bits These bits can set the kind of SDRAM. Set the value that has been determined for the product in use. 111: DDR3-SDRAM Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. This register must only be written from within the initialization sequence (see section 12.5.3).
2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)
3. Set the defined value in each group.

12.4.7 SDRAM Configuration Setting Register (DBCONF)

The SDRAM configuration setting register (DBCONF) is readable/writable. This register sets the memory configuration to be used. Refer to "Method of setting for the SDRAM configuration setting register" for details on the memory configurations supported by DBSC3. For details on the relationship between the external pins of the SDRAM and the logical addresses of this LSI, refer to section 12.2 "Relationship between the external pins of SDRAM and logical addresses of this LSI". It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
28 to 24	AWRW	00000	R/W	Row Setting Bits Specify the width, in bits, of row addresses. 01011: 11 bits 01100: 12 bits 01101: 13 bits 01110: 14 bits 01111: 15 bits 10000: 16 bits Other settings are prohibited (if specified, correct operation cannot be guaranteed)
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
20	AWRK	0	R/W	Number of Ranks Setting Bits Specify the number of ranks. 0: 1 rank Other settings are prohibited (if specified, correct operation cannot be guaranteed)
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
17, 16	AWBK	00	R/W	Number of Banks Setting Bits Specify the number of banks. 11: 8 banks Other settings are prohibited (if specified, correct operation cannot be guaranteed)
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
11 to 8	AWCL	0000	R/W	Column Bit Width Setting Bits Specify the width, in bits, of column addresses. 1000: 8 bits 1001: 9 bits 1010: 10 bits 1011: 11 bits 1100: 12 bits Other settings are prohibited (if specified, correct operation cannot be guaranteed)
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	DW	00	R/W	External Data Bus Width Setting Bits Specify the width of the external bus. 10: 32 bits Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. This register must only be written from within the initialization sequence (see section 12.5.3).
2. Writing to this register should only be performed when the following conditions are met:
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)
3. Set the defined value in each group.
4. The supported memory configuration is two 16-bit-width SDRAM devices or four 8-bit width SDRAM devices connected to a 32-bit bus.
5. Although DBSC3 supports the connection of multiple SDRAM devices, the electrical characteristics of some chips may impose a limitation on the number of connectable SDRAM devices.

12.4.8 SDRAM Timing Register 0 (DBTR0)

SDRAM timing register 0 (DBTR0) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR0 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
3 to 0	CL	0000	R/W	CAS Latency Setting Bits These bits are for setting the CAS latency of the SDRAM. 0101: 5 cycles 0110: 6 cycles 0111: 7 cycles 1000: 8 cycles 1001: 9 cycles 1010: 10 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.9 SDRAM Timing Register 1 (DBTR1)

The SDRAM timing register 1 (DBTR1) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR1 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
3 to 0	CWL	0000	R/W	CAS Write Latency Setting Bits These bits are for setting the CAS-write latency of the SDRAM. 0101: 5 cycles 0110: 6 cycles 0111: 7 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.10 SDRAM Timing Register 2 (DBTR2)

The SDRAM timing register 2 (DBTR2) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR2 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
3 to 0	AL	0000	R/W	Additive Latency Setting Bits These bits are for setting the additive latency of the SDRAM. 0000: 0 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.11 SDRAM Timing Register 3 (DBTR3)

The SDRAM timing register 3 (DBTR3) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR3 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
3 to 0	TRCD	0000	R/W	ACT-READ/WRITE Interval Setting Bits These bits indicate the minimum interval from an ACT command to a READ/WRITE command. 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 7 cycles 1000: 8 cycles 1001: 9 cycles 1010: 10 cycles 1011: 11 cycles 1100: 12 cycles 1101: 13 cycles 1110: 14 cycles 1111: 15 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.12 SDRAM Timing Register 4 (DBTR4)

The SDRAM timing register 4 (DBTR4) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR4 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
19 to 16	TRPA	0000	R/W	PREA Time Setting Bits These bits indicate the minimum interval from a PRE ALL (precharge all banks) command to an ACT/REF command. The value set in these bits must be greater than or equal to that in the TRP bits. 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 7 cycles 1000: 8 cycles 1001: 9 cycles 1010: 10 cycles 1011: 11 cycles 1100: 12 cycles 1101: 13 cycles 1110: 14 cycles 1111: 15 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
3 to 0	TRP	0000	R/W	PRE Time Setting Bits These bits indicate the minimum interval from a PRE (precharge) command to an ACT/REF command. 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 7 cycles 1000: 8 cycles 1001: 9 cycles 1010: 10 cycles 1011: 11 cycles 1100: 12 cycles 1101: 13 cycles 1110: 14 cycles 1111: 15 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).
2. The following condition must be satisfied: TRPA ≥ TRP.
3. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.13 SDRAM Timing Register 5 (DBTR5)

The SDRAM timing register 5 (DBTR5) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR5 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
5 to 0	TRC	000000	R/W	ACT-ACT/REF Interval Setting Bits These bits indicate the minimum interval from one ACT command to another ACT command (for the same bank) or to a REF command. 000101: 5 cycles 000110: 6 cycles : : 111011: 59 cycles 111100: 60 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.14 SDRAM Timing Register 6 (DBTR6)

The SDRAM timing register 6 (DBTR6) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR6 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
5 to 0	TRAS	000000	R/W	ACT-PRE Interval Setting Bits These bits indicate the minimum interval from an ACT command to a PRE command. 000100: 4 cycles 000101: 5 cycles : : 100111: 39 cycles 101000: 40 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.15 SDRAM Timing Register 7 (DBTR7)

The SDRAM timing register 7 (DBTR7) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR7 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
3 to 0	TRRD	0000	R/W	ACT(A)-ACT(B) Interval Setting Bits These bits indicate the minimum interval between ACT commands issued for different banks. 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 7 cycles 1000: 8 cycles 1001: 9 cycles 1010: 10 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.16 SDRAM Timing Register 8 (DBTR8)

The SDRAM timing register 8 (DBTR8) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR8 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
7 to 0	TFAW	0000 0000	R/W	4 Activate Window Length Setting Bits These bits indicate the length of the 4 Activate window. The value of these bits must be at least four times as large as that of the TRRD bits in DBTR7. 000100: 4 cycles 000101: 5 cycles : : 100111: 39 cycles 101000: 40 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).
2. The following condition must be satisfied: TFRW $\geq 4 \times$ TRRD.
3. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.17 SDRAM Timing Register 9 (DBTR9)

The SDRAM timing register 9 (DBTR9) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR9 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
3 to 0	TRDPR	0000	R/W	READ-PRE Interval Setting Bits These bits indicate the minimum interval from a READ command to a PRE command. 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 7 cycles 1000: 8 cycles 1001: 9 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).
2. The following condition must be satisfied: TRDPR ≥ 4.
3. If the SDRAM in use conforms to the JEDEC standard, the value of the bits can be calculated from the following formula.
In the case of DDR3: TRDPR = 4 + max {2, ceil(tRTP / tCK)} – 2
4. Writing to this register should only be performed when the following conditions are met.
- Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
- Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.18 SDRAM Timing Register 10 (DBTR10)

The SDRAM timing register 10 (DBTR10) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR10 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	TWR	0000	R/W	Write-Recovery Period Setting Bits These bits indicate the write-recovery period. 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 7 cycles 1000: 8 cycles 1001: 9 cycles 1010: 10 cycles 1011: 11 cycles 1100: 12 cycles 1101: 13 cycles 1110: 14 cycles 1111: 15 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.19 SDRAM Timing Register 11 (DBTR11)

The SDRAM timing register 11 (DBTR11) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR11 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
5 to 0	TRDWR	000000	R/W	READ-WRITE Interval Setting Bits These bits indicate the minimum interval from a READ command to a WRITE command. 000100: 4 cycles 000101: 5 cycles 000110: 6 cycles 000111: 7 cycles 001000: 8 cycles 001001: 9 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).
2. The following condition must be satisfied: TRDWR ≥ CL + 4 + 2 – CWL.
3. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.20 SDRAM Timing Register 12 (DBTR12)

The SDRAM timing register 12 (DBTR12) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR12 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
5 to 0	TWRRD	000000	R/W	WRITE-READ Interval Setting Bits These bits indicate the minimum interval from a WRITE command to a READ command. 000100: 4 cycles 000101: 5 cycles : : 010101: 21 cycles 010110: 22 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).
2. The following condition must be satisfied: TWDRD ≥ CWL + 4.
3. If the SDRAM in use conforms to the JEDEC standard, the value of the bits can be calculated from the following formula.
TWDRD = WL + BL/2 + ceil(tWTR / tCK)
4. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.21 SDRAM Timing Register 13 (DBTR13)

The SDRAM timing register 13 (DBTR13) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR13 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
9 to 0	TRFC	00 0000 0000	R/W	REF-ACT/REF Interval Setting Bits These bits indicate the minimum interval from a REF (refresh) command to an ACT/REF command. 0000000110: 6 cycles 0000000111: 7 cycles : : 0011111110: 254 cycles 0011111111: 255 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.22 SDRAM Timing Register 14 (DBTR14)

The SDRAM timing register 14 (DBTR14) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR14 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	TCKEH	0000 0000	R/W	CKEH Period Setting Bits These bits indicate the minimum interval from the time the CKE signal goes high until the issuing of a further valid command. The minimum interval from the time the CKE signal goes high until issuing of the first READ command will be TCKEH + TRCD. 00000001: 1 cycle 00000010: 2 cycles 00000011: 3 cycles 00000100: 4 cycles 00000101: 5 cycles 00000110: 6 cycles 00000111: 7 cycles 00001000: 8 cycles 00001001: 9 cycles 00001010: 10 cycles 00001011: 11 cycles 00001100: 12 cycles 00001101: 13 cycles 00001110: 14 cycles 00001111: 15 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. When the SH7786 is to be used in a power-down mode, this register must be set accordingly.
   If the SDRAM in use conforms to the JEDEC standard, the value of the bits can be calculated from the following formula.
   In the case of DDR3-SDRAM used in Fast Exit Mode: ceil(tXP / tCK)
   In the case of DDR3-SDRAM used in Slow Exit Mode:
   max {ceil(tXP / tCK), ceil(tXPDLL / tCK) – ceil(tRCD / tCK)}

3. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.23 SDRAM Timing Register 15 (DBTR15)

The SDRAM timing register 15 (DBTR15) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR15 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
3 to 0	TCKEL	0000	R/W	CKEL Period Setting Bits These bits indicate the minimum time from the CKE signal going low until it goes high. 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 7 cycles 1000: 8 cycles 1001: 9 cycles 1010: 10 cycles 1011: 11 cycles 1100: 12 cycles 1101: 13 cycles 1110: 14 cycles 1111: 15 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).
2. Writing to this register should only be performed when the following conditions are met.
- Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
- Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.24 SDRAM Timing Register 16 (DBTR16)

The SDRAM timing register 16 (DBTR16) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR16 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
21 to 16	DQL	000000	R/W	dqltncy Setting Bits These bits indicate the input edge dqltncy for DDR3PHY. 010110: 22 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	WDQL	0000	R/W	wdqltncy Setting Bits These bits indicate the input edge wdqltncy for DDR3PHY. 0010: 2 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).
2. Writing to this register should only be performed when the following conditions are met.
- Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
- Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.25 SDRAM Timing Register 17 (DBTR17)

The SDRAM timing register 17 (DBTR17) is a readable/writable register which is used to set a timing parameter for the SDRAM. DBTR17 is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
21 to 16	TMOD	000000	R/W	MRS Time Setting Bits These bits indicate the minimum interval from an MRS (mode register set) command to a subsequent command. 000100: 4 cycles 000101: 5 cycles 000110: 6 cycles 000111: 7 cycles 001000: 8 cycles 001001: 9 cycles 001010: 10 cycles 001011: 11 cycles 001100: 12 cycles 001101: 13 cycles 001110: 14 cycles 001111: 15 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	TRDMR	000000	R/W	READ-MRS Interval Setting Bits These bits indicate the minimum interval from a READ command to an MRS command in MPR mode. 000100: 4 cycles 000101: 5 cycles 000110: 6 cycles 000111: 7 cycles 001000: 8 cycles 001001: 9 cycles 001010: 10 cycles 001011: 11 cycles 001100: 12 cycles 001101: 13 cycles 001110: 14 cycles 001111: 15 cycles 010000: 16 cycles 010001: 17 cycles 010010: 18 cycles 010011: 19 cycles 010100: 20 cycles 010101: 21 cycles 010110: 22 cycles 010111: 23 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. Writing to this register should only be performed when the following conditions are met.
   - Access to SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0)
   - Auto-refresh is disabled (i.e. the ARFEN bit in the DBRFEN register is 0)

12.4.26 Refresh Configuration Register 0 (DBRFCNF0)

Refresh configuration register 0 (DBRFCNF0) is a readable/writable register which is used to set the timing for refreshing of the SDRAM. DBRFCNF0 is only initialized at the time of a power-on reset.

If you change the value of this register while auto-refresh is active (i.e. the ARFEN bit in the DBRFEN register is 1), the auto-refresh facility must be re-activated to reflect the new value in refreshing. To re-activate the auto-refresh facility, write 1 to the ARFEN bit in the DBRFEN register.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
11 to 0	REFTHF	0000 0000 0000	R/W	Forcible Auto-Refresh Threshold Setting Bits These bits indicate the timing for forcible refreshing regardless of bus requests. The value represented by these bits affects the amount of jitter in the refresh interval and performance in access to memory. A smaller value means less jitter in the refresh interval but may reduce performance in access. For details on the amount of jitter in the refresh interval, see section 12.4.28, Refresh Configuration Register 1 (DBRFCNF1). 000000100000: 32 cycles 000000100001: 33 cycles : : 000111111110: 510 cycles 000111111111: 511 cycles Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

2. If auto-refresh is activated (i.e. the ARFEN bit in the DBRFEN register is set to 1) after a value smaller than the minimum value defined in the above table has been set in this register, correct operation cannot be guaranteed.

3. The minimum value can be calculated by the following formula (however, it is recommended that the user-specified number of cycles be the minimum value + 50 or more).
32 + max { TCKEL + TCKEH, TRPA + max {CWL + 4 + TWR, TRDPR, TRAS, TRC – TRP }}

12.4.27 Refresh Configuration Register 1 (DBRFCNF1)

Refresh configuration register 1 (DBRFCNF1) is a readable/writable register which is used to set the timing for refreshing of the SDRAM. DBRFCNF1 is only initialized at the time of a power-on reset.

If you change the value of this register while auto-refresh is active (i.e. the ARFEN bit in the DBRFEN register is 1), the auto-refresh facility must be re-activated to reflect the new value in refreshing. To re-activate the auto-refresh facility, write 1 to the ARFEN bit in the DBRFEN register.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
27 to 24	REFPMAX	0000	R/W	Maximum Post Number of Refresh Commands These bits indicate the maximum number of refresh commands (post number) accumulated by auto-refresh. As long as the number of refresh commands that has been accumulated is smaller than REFPMAX, refresh commands are issued while there are no bus requests. 0000: 0 cycles (minimum amount of jitter in the refresh interval) 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 7 cycles 1000: 8 cycles (maximum amount of jitter in the refresh interval) Other settings are prohibited (if specified, correct operation cannot be guaranteed)
23 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
15 to 0	REFINT	0000 0000 0000 0000	R/W	Average Refresh Interval These bits indicate the average interval for issuing of refresh commands. When the REFINTS bit of the DBRFCN2 register is 0, the average interval (in cycles) is REFINT. When the REFINTS bit of the DBRFCN2 register is 1, on the other hand, the average interval (in cycles) is floor(REFINT/2). This average interval is hereafter referred to as REFINT_E. Thus, REFINT_E = REFINT >> REFINTS (>>: logical right-shift operator). 0000000010000000: 128 cycles (H'0080) 0000000010000001: 129 cycles (H'0081) : : 0011111111111110: 16387 cycles (H'3FFE) 0011111111111111: 16388 cycles (H'3FFF) Other settings are prohibited (if specified, correct operation cannot be guaranteed)

Notes: 1. The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).
2. Other restrictions not mentioned in the above table may apply to some types of SDRAM.
3. If auto-refresh is activated (i.e. the ARFEN bit in the DBRFEN register is set to 1) after a value smaller than the minimum value defined in the above table has been set in this register, correct operation cannot be guaranteed.

12.4.28 Refresh Configuration Register 2 (DBRFCNF2)

Refresh configuration register 2 (DBRFCNF2) is a readable/writable register which is used to set the timing for refreshing of the SDRAM. DBRFCNF2 is only initialized at the time of a power-on reset.

If you change the value of this register while auto-refresh is active (i.e. the ARFEN bit in the DBRFEN register is 1), the auto-refresh facility must be re-activated to reflect the new value in refreshing. To re-activate the auto-refresh facility, write 1 to the ARFEN bit in the DBRFEN register.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
0	REFINTS	0	R/W	Average Refresh Interval Adjustment Bit When this bit is 0, the average interval (in cycles) is REFINT. When this bit is 1, on the other hand, the average interval (in cycles) is floor(REFINT/2). 0: Average interval is REFINT 1: Average interval is 1/2 REFINT

Note: The number of cycles is determined by the frequency of the SDRAM operating clock (MCK).

12.4.29 PHY-Unit Control Register 0 (DBPDCNT0)

PHY-unit control register 0 (DBPDCNT0) is a readable/writable register which controls the PHY unit. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
0	DLLRST	0	R/W	DLL Reset Bit Setting this bit to 0 places the DLL in the reset state. Setting this bit to 1 releases the DLL from the reset state. For details, refer to section 12.5.3, Initialization Sequence. 0: DLL is placed in the reset state (initial value) 1: DLL is released from the reset state

12.4.30 PHY-Unit Control Register 1 (DBPDCNT1)

PHY-unit control register 1 (DBPDCNT1) is a readable/writable register which controls the PHY unit. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
0	CALEN	0	R/W	Automatic Calibration Setting Bit This bit enables automatic calibration. It is only valid when CMODE in DBPDCNT3 is high. For details, refer to section 12.5.3, Initialization Sequence. 0: Disables automatic calibration (initial value) 1: Enables automatic calibration

12.4.31 PHY-Unit Control Register 2 (DBPDCNT2)

PHY-unit control register 2 (DBPDCNT2) is a readable/writable register which controls the PHY unit. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
5	DETFIXH	0	R/W	90-Degree Read Shifter Setting Bit This bit holds the setting for the Hi-Z detection circuit of the 90-degree shifter employed in reading. For details, refer to section 12.5.3, Initialization Sequence. 0: Hi-Z detection circuit is on (initial value) 1: Hi-Z detection circuit is off
4	CMDMOD	0	R/W	Command Hard-Macro Output Delay Mode Setting Bit This bit holds the setting for the command hard-macro output delay mode. For details, refer to section 12.5.3, Initialization Sequence. 0: The output delay mode for command address signals is off (initial value). 1: The output delay mode of the command address signals is on.

Bit	Bit Name	Initial Value	R/W	Description
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
1, 0	RLADD	00	R/W	Additional Read Latency Setting Bits These bits indicate the additional read latency for DDR3PHY when the round-trip time is large. For details, refer to section 12.5.3, Initialization Sequence. 00: No additional read latency (initial value) 01: One cycle of additional read latency 10: Two cycles of additional read latency 11: Three cycles of additional read latency

12.4.32 PHY-Unit Control Register 3 (DBPDCNT3)

PHY-unit control register 3 (DBPDCNT3) is a readable/writable register which controls the PHY unit. It is only initialized at the time of a power-on reset.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CMOD	DIC_AD		DIC_CK		DIC_DQ		—	ODT			—	FREQ		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
14	CMOD	0	R/W	Calibration Mode Setting Bit This bit holds the setting for calibration mode. For details, refer to section 12.5.3, Initialization Sequence. 0: External control mode (initial value) 1: Automatic calibration mode
13, 12	DIC_AD	00	R/W	Control-Signal Buffer-Drive Strength Setting Bits These bits indicate the strength of buffer driving by control signals. For details, refer to section 12.5.3, Initialization Sequence. 00: 40 Ω (initial value) Other settings are prohibited (if specified, correct operation cannot be guaranteed).
11, 10	DIC_CK	00	R/W	Clock-Signal Buffer-Drive Strength Setting Bits These bits indicate the strength of buffer driving by clock signals. For details, refer to section 12.5.3, Initialization Sequence. 00: 40 Ω (initial value) Other settings are prohibited (if specified, correct operation cannot be guaranteed).

Bit	Bit Name	Initial Value	R/W	Description
9, 8	DIC_DQ	00	R/W	Data-Signal Buffer-Drive Strength Setting Bits These bits indicate the strength of buffer driving by data signals. For details, refer to section 12.5.3, Initialization Sequence. 00: 40 Ω (initial value) Other settings are prohibited (if specified, correct operation cannot be guaranteed).
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
6 to 4	ODT	000	R/W	Internal ODT Impedance Setting Bits These bits indicate the internal ODT impedance. For details, refer to section 12.5.3, Initialization Sequence. 000: External control mode (initial value) 001: 60 Ω Other settings are prohibited (if specified, correct operation cannot be guaranteed).
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
2 to 0	FREQ	000	R/W	Lock-Waiting Skip Mode Setting Bits These bits indicate the lock-waiting skip mode. For details, refer to section 12.5.3, Initialization Sequence. 000: Normal mode (initial value) 001: Lock-waiting skip mode Other settings are prohibited (if specified, correct operation cannot be guaranteed).

12.4.33 PHY-Unit Lock Register (DBPDLCK)

The PHY-unit lock register (DBPDLCK) is a readable/writable register which controls the PHY unit. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
15 to 0	PLOCK	0000 0000 0000 0000	R/W	PHY-Unit Access-Lock Setting Bits Setting these bits to H'A55A enables access to the internal registers of the PHY unit. H'A55A: Enables access to the internal registers of the PHY unit Other: Disables access to the internal registers of the PHY unit

12.4.34 PHY-Unit Internal Register Address Register (DBPDRGA)

The PHY-unit internal register address register (DBPDRGA) is a readable/writable register which controls the PHY unit. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
7 to 0	PRA	0000 0000	R/W	PHY-Unit Internal Register Address These bits are used to set up addresses for access (setting or referring to values) to the internal registers of the PHY unit (registers incorporated in the PHY unit).

Table 12.9 Internal Register Configuration of DBSC3PHY

Register Name	Abbreviation	R/W	PRA Setting	Initial Value	Access Size (Bits)
Reserved	—	—	H'00 to H'27	—	—
Preamble detection control register	PACNT	R/W	H'28	H'00017100	32
Reserved	—	—	H'29 to H'FF	—	—

12.4.35 PHY-Unit Internal Register Data Register (DBPDRGD)

The PHY-unit internal register data register (DBPDRGD) is a readable/writable register which controls the PHY unit. It is only initialized at the time of a power-on reset.

For access to internal registers of the PHY unit, set the PHY-unit lock register (DBPDLCK) and the PHY-unit internal register address register (DBPDRGA) before writing to or reading from these bits.

Writing to these bits will be ignored if access is disabled by the PHY-unit lock register (DBPDLCK) at the time these bits are written to (neither writing nor an error will be generated).

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	PRD	Undefined	R/W	PHY-Unit Internal Registers Writing to and reading from these bits can give access (setting or referring to values) to the internal registers of the PHY unit is possible. The register to be accessed is selected by the value of the PRA bits of the PHY-unit internal register address register (DBPDRGA).

(1) Preamble Detection Control Register (PACNT) : DBPDRGA.PRA=H'28

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

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

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. If a value other than 0 is written, correct operation cannot be guaranteed.
16	en	1	R/W	FIFO Input Timing Control Enable Bit The write value should always be 1. When writing 0 to this bit, correct operation cannot be guaranteed.
15 to 12	Rprst_str	0111	R/W	FIFO Pointer Reset Period Start Bit Specifies additional cycle before the setting of RL+one MCK cycle. The default value is 3.5-MCK cycle. 0000: 0-MCK cycle 0001: 0.5-MCK cycle 0010: 1-MCK cycle 0011: 1.5-MCK cycle 0100: 2-MCK cycle 0101: 2.5-MCK cycle 0110: 3-MCK cycle 0111: 3.5-MCK cycle 1000: 4-MCK cycle Other settings are prohibited (if specified, correct operation cannot be guaranteed).
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
9, 8	Rprst_end	01	R/W	FIFO Pointer Reset Period End Bit Specifies additional cycle after the setting of RL+one MCK cycle. The default value is 0.5-MCK cycle. 00: 0-MCK cycle 01: 0.5-MCK cycle 10: 1-MCK cycle (recommendable) 11: setting prohibited (if specified, correct operation cannot be guaranteed).
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

12.4.36 Bus Control Unit 0 Control Register 0 (DBBS0CNT0)

Bus control unit 0 control register 0 (DBBS0CNT0) is a readable/writable register which controls the bank cache. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
0	BKCEN	0	R/W	Bank Cache Enable Bit This bit is used to specify whether or not the bank cache in the bus control unit is used. When the value is changed from 0 to 1, the contents of the bank cache are initialized. 0: Bank cache is not used. 1: Bank cache is used.

Notes: 1. This register must only be written from within the initialization sequence (see section 12.5.3).
2. Writing to this register must only proceed while access to the SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0).

12.4.37 Bus Control Unit 0 Control Register 1 (DBBS0CNT1)

Bus control unit 0 control register 1 (DBBS0CNT1) is a readable/writable register which controls the bank cache. It is only initialized at the time of a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
1, 0	BKADM	00	R/W	Bank Assignment Setting Bit This bit is used to set the method for assigning the SDRAM bank to the logical address space, i.e. whether and how logical addresses are divided into multiple blocks to which different SDRAM banks are assigned. 00: The whole logical address space is regarded as one block. 01: The logical address space is divided into two blocks, one for banks 0 to 3 and one for banks 4 to 7. 10: The logical address space is divided into three blocks, one for banks 0 to 1, one for banks 2 to 3, and one for banks 4 to 7. 11: The logical address space is divided into four blocks, one for bank 0, one for bank 1, one for banks 2 to 3, and one for banks 4 to 7.

Notes: 1. This register must only be written from within the initialization sequence (see section 12.5.3).
2. Writing to this register must only proceed while access to the SDRAM is disabled (i.e. the ACCEN bit in the DBACEN register is 0).

12.5 DBSC3 Operation

12.5.1 Supported SDRAM Commands

Table 12.10 lists the SDRAM commands issued by the DBSC3. These commands are issued to the DDR3-SDRAM in synchronously with MCK0, $\overline{\text{MCK0}}$, MCK1, and $\overline{\text{MCK1}}$. In the table, n-1 indicates the state of the signal applied to DDR3-SDRAM one cycle before SDRAM command issue; n indicates the state of the signal at the time of command issue.

Table 12.10 SDRAM Commands Issued by the DBSC3

Function	Symbol	MCKE n-1	MCKE n	$\overline{MCS0}$ $\overline{MCS1}$	$\overline{MRAS}$	$\overline{MCAS}$	$\overline{MWE}$	MBA [2:0]	MA [15:3]	MA12/ $\overline{BC}$	MA10/ AP	MA11, MA [9:0]
Device deselected	DES	H	H	H	X	X	X	X	X	X	X	X
Read	READ	H	H	L	H	L	H	V	V	V	L	V
Write	WRITE	H	H	L	H	L	L	V	V	V	H	V
Bank activate	ACT	H	H	L	L	H	H	V	V	V	V	V
Precharge select bank	PRE	H	H	L	L	H	L	V	V	V	L	V
Precharge all banks	PREA	H	H	L	L	H	L	V	V	V	H	V
Auto-refresh	REF	H	H	L	L	L	H	V	V	V	V	V
Self-refresh entry	SRE	H	L	L	L	L	H	V	V	V	V	V
Self-refresh exit	SRX	L	H	H	V	V	V	V	V	V	V	V
				L	H	H	H					
Power down entry	PDE	H	L	L	H	H	H	V	V	V	V	V
				H	V	V	V					
Power down exit	PDX	L	H	L	H	H	H	V	V	V	V	V
				H	V	V	V					
Mode register set	MRS	H	H	L	L	L	L	V	V	V	V	V
ZQ calibration short issue	ZQCS	H	H	L	H	H	L	X	X	X	H	X
ZQ calibration long issue	ZQCL	H	H	L	H	H	L	X	X	X	L	X

Legend:

H: High level

L: Low level

X: High or low level (don't care)

V: Valid data

The above DES command is issued when DDR3-SDRAM is not accessed, and so need not be explicitly issued by the user.

12.5.2 SDRAM Command Issue

(1) Basic Access

The DBSC3 stores in a queue the requests received via the SuperHyway bus. Request processing is begun around the time of preceding precharge/activate processing, but processing completion is in the order received in the queue.

When SDRAM initialization is completed, upon receiving a read/write request, a page miss occurs with all banks in the closed state. Hence the DBSC3 first issues an activate (ACT) command, to open the corresponding bank. After opening the bank, the read/write command of the SDRAM corresponding to the read/write request is issued. At this time, the number of issued read/write commands differs depending on the bus width and the request size (1/2/4/8/16/32 bytes), as indicated in figure 12.3. For example, when performing 32-byte reading from the SuperHyway bus with an external data bus width of 32 bytes, two read commands are executed. When issuing the read command in the first cycle, data is read with a burst length of 4 (two DDR clock cycles), so that it is necessary to wait until the third cycle to issue the second read command.

When access ends, the DBSC3 leaves the bank open, without using a precharge (PRE) command. The bank is closed when (1) the following request is for the same bank with a different row address; (2) there is an auto-refresh request; or (3) the user issues a precharge-all (PREA) command using the SDRAM command control register, for self-refresh processing.

Thus in normal access other than self-refresh, the DBSC3 uses hardware for bank management, so that except for the register settings upon initialization, the user need not execute control.

Further, the DBSC3 performs multi-bank operation of eight banks. Hence the maximum number of banks which can be opened simultaneously is eight. Refer to section 12.5.6, Regarding Address Multiplexing, for the correspondence between access addresses from the SuperHyway bus and SDRAM bank/row addresses.

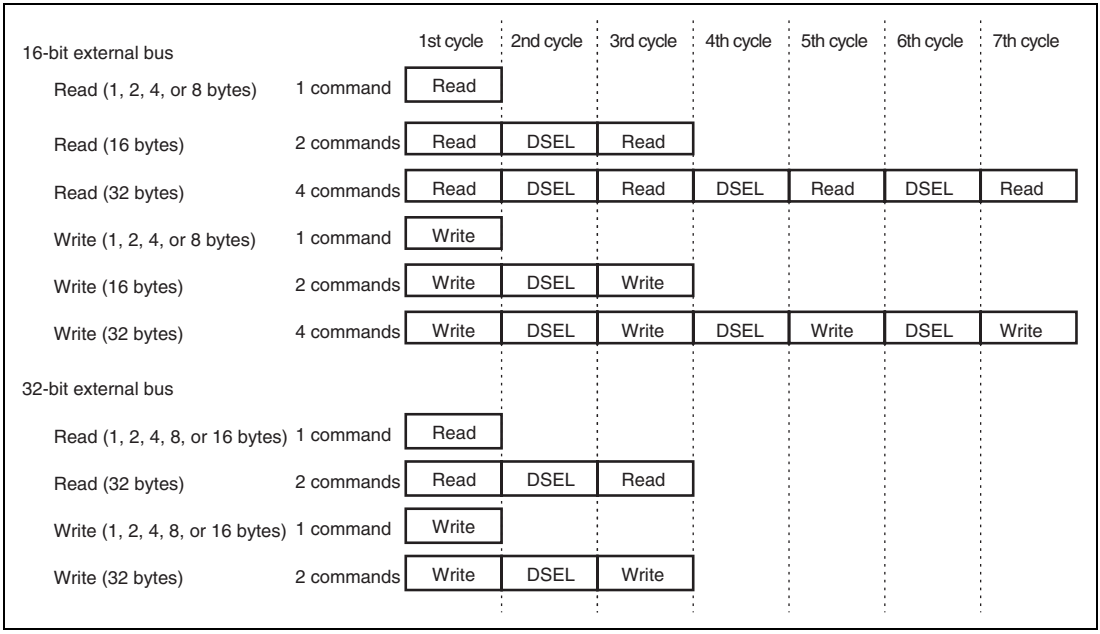


Figure 12.3 Read/Write Command Issued to the SDRAM in Response to the Request from the SuperHyway Bus

(2) Preceding Precharge/Activate Processing

In order to utilize DDR3-SDRAM multibank functions to reduce SDRAM command vacant cycles insofar as possible and improve the efficiency of bus use, the DBSC3 issues in advance a PRE/ACT command corresponding to the following request queue page miss processing. Only the PRE/ACT command is issued in advance, so there is no change in the read/write order. A PRE/ACT command is issued in advance only when the following request (1) results in a page miss, and moreover (2) entails access of a bank different from that of the request currently being processed. Figure 12.4 shows an example of execution of preceding precharge/activate processing. This is an example of a command issued to the SDRAM when the external data bus width is 32 bits, the PRE/ACT minimum time constraint is 3 cycles, the ACT-READ/WRITE minimum time constraint is 3 cycles, and the ACT (A)-ACT (B) minimum time constraint is 2 cycles. In this example, the first through fourth requests are accumulated, and the first request is the request initially provided to the queue.

First, at time 1 the DBSC3 issues to the SDRAM a PRE command for the first read (16-byte) request processing. Then, when determining the command to be issued at time 2, due to timing constraints it is not possible to issue at time 2 the ACT command necessary as request processing for the first read (16-byte) request, which has higher priority. Hence the DBSC3 searches for a

command to be issued at time 2 from the following request queue. From the search results it is seen that advance precharge processing can be executed for the third read (8-byte) request and the fourth read (16-byte) request. Because the DBSC3 gives priority to preceding requests, it decides to perform advance precharge processing for the third read (8-byte) request, and issues a PRE command to the SDRAM.

When the time advances to time 3, the ACT command cannot be issued for the first read (16-byte) request at time 3 either, and so a search of the following queue is performed for a command which can be issued. Due to timing constraints, the ACT command cannot be issued for the third read (8-byte) request, and as a result, issuance of the PRE command corresponding to the fourth read (16-byte) request is selected.

At time 4, it is possible to execute request processing for the first read (16-byte) request, and an ACT command is issued to the DDR3-SDRAM.

Thereafter, the processing described above is repeated.

Request No.	Request	Bank to be accessed	Page state during request	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6	Time 7	Time 8	Time 9	Time 10	Time 11	Time 12	Time 13	Time 14	Time 15
1	Read (16 bytes)	Bank 0	Miss	PRE			ACT			READ								
2	Read (32 bytes)	Bank 1	Hit									READ		READ				
3	Read (8 bytes)	Bank 2	Miss		PRE				ACT							READ		
4	Read (16 bytes)	Bank 3	Miss			PRE					ACT							READ
SDRAM command				PRE	PRE	PRE	ACT		ACT	READ	ACT	READ		READ		READ		READ

As the burst length is 4 in the DDR3-SDRAM, the interval between READ commands is always two cycles.

Figure 12.4 Example of Preceding Precharge/Activate Processing

12.5.3 Initialization Sequence

The following shows an example of the initialization sequence. For detailed information such as the power supply and timing parameters, please refer to the datasheet for the DDR3-SDRAM being used.

1. Supply the power and the reference voltage according to the guidance provided by the vender of the SDRAM being used. The procedure differs for system startup and for recovering from the power supply backup state.
2. After the LSI circuit have been released from the power-on reset state and the CPU has begun operating, the system judges whether the LSI was in power supply backup mode or the initialization sequence is normal (for information on entering settings in power backup mode, please refer to (2) Recovery from SDRAM Power Supply Backup Mode in section 12.5.8, DDR3-SDRAM Power Supply Backup Function). If it was an initialization sequence, the software is used to initiate a wait of at least 100 μs.
3. Use the manual command-issuing register (DBCMD) to set the RESET pin of the SDRAM to the low level. The value written to this register should be opc = RstL, arg = 0.
4. Use the manual command-issuing register (DBCMD) to set the CKE pin of SDRAM to the low level. The value written to this register should be opc = PDEn, arg = 100 μs.
5. Use the SDRAM-kind setting register (DBKIND) to set the type of memory.
6. Enter the settings for the SDRAM configuration setting register (DBCONF) and SDRAM timing registers 0 to 17 (DBTR0 to DBTR7).
7. Enter the settings of PHY-unit internal register address register (DBPDRGA) and PHY-unit internal register data register (DBPDRGD) for preamble detection control register (PACNT).
8. Use PHY-unit control register 3 (DBPDCNT3) to select auto-calibration.
9. Use PHY-unit control register 1 (DBPDCNT1) to start up auto-calibration.
10. Ensure that the software allows the required waiting time (at least 10 μs).
11. Use PHY-unit control register 0 (DBPDCNT1) to release the DLL from the reset state.
12. Ensure that the software allows the required waiting time (at least 50 μs).
13. After setting MBKPRSC = "H", set SDBUP = "H".
14. Use the manual command-issuing register (DBCMD) to insert a waiting time. The value written to this register should be opc = Wait, arg = 100 μs.
15. Use the manual command-issuing register (DBCMD) to set the RESET pin of SDRAM to high level. The value written value to this register should be opc = RstH, arg = 100 μs.
16. Use the manual command-issuing register (DBCMD) to insert a waiting time. Write opc = Wait, arg = 100 μs to this register four times.

17. Use the manual command-issuing register (DBCMD) to set the CKE pin of SDRAM to the high level. The value written to this register should be opc = PDXt, arg = tXPR (normally tRFC + 10ns).
18. Use the manual command-issuing register (DBCMD) to issue the MRS (MR2) command. Adjust the setting for CWL to the setting of the CWL bits in SDRAM timing register 1 (DBTR1).
19. Use the manual command-issuing register (DBCMD) to issue the MRS (MR3) command. Set the MPR for normal operation.
20. Use the manual command-issuing register (DBCMD) to issue the MRS (MR1) command. Set the Additive Latency to 0, DLL Enable to Enabled.
21. Use the manual command-issuing register (DBCMD) to issue the MRS (MR0) command. At this time, set the mode of operation to normal, reset the DLL, the burst length to eight, and the burst type to sequential. Also adjust the setting for CAS latency to the setting of the CL bits in the SDRAM timing register 0 (DBTR0). Adjust the setting of writing to the settings for TWR bits in the SDRAM timing register 10 (DBTR10).
22. Use the manual command-issuing register (DBCMD) to issue the ZQ Calibration Long command. The value written to this register should be opc = ZQCL, arg = max {tZQinit, tDLLK-tMOD} (normally 512).
23. If this is necessary, enter a setting for the bus control unit 0 setting register.
24. Enter settings for refresh configuration registers 0 to 2 (DBRFCNF0 to DBRFCNF2).
25. Set the ARFEN bit to 1 (access enabled) in the auto-refresh enable register (DBRFREN).
26. Set the ACCEN bit to 1 (access enabled) in the SDRAM access enable register (DBACEN).
27. Read the operation completion waiting register (DBWAIT) and wait for the response.

12.5.4 Self-Refresh Operation

Self-refreshing helps to reduce the amount of power consumed by the SDRAM.

Also, using the self-refresh operation in combination with power supply control makes it possible to operate in power supply backup mode, with all power supplies other than that of the SDRAM off. For details on power supply backup mode, refer to section 12.5.8, DDR3-SDRAM Power Supply Backup Function.

If it is not necessary to access the SDRAM, the SDRAM can be put in self-refresh mode to reduce power consumption while still retaining data contents.

Because access is disabled in self-refresh mode, any attempt to access data in the DDR3-SDRAM will be ignored.

The following procedure is used to make a transition to self-refresh mode.

1. Check to make sure the DBSC3 is not being accessed. The time required for transition to self-refresh must not exceed the auto-refresh interval requested by the SDRAM by interrupts or some other causes.
2. Set the ACCEN bit in the SDRAM operation enable register (DBACEN) to 0 (access disabled).
3. Use the manual command-issuing register (DBCMD) to issue the PREA (precharge all) command. The value written to the register should be opc = PreA, arg = 0.
4. Use the manual command-issuing register (DBCMD) to issue the Self-Refresh Entry command. The value written to the register should be opc = SREn, arg = 0.
5. Set the ARFEN bit in the auto-refresh enable register (DBRFEN) to 0.

Use the following procedure for release from the self-refresh mode.

1. Make sure that the processing will not be disrupted by interrupts, etc. to assure the auto-refresh interval.
2. Set the ARFEN bit in the auto-refresh enable register (DBRFEN) to 1.
3. Use the manual command-issuing register (DBCMD) to issue the Self-Refresh Exit command. The value written to the register should be opc = SRXt, arg = 0.
4. Use the manual command-issuing register (DBCMD) to insert the period of waiting until access to SDRAM is enabled. The value written to the register should be opc = Wait, arg = tXSDLL (normally 512).
5. Set the ACCEN bit in the SDRAM access enable register (DBACEN) to 1 (access enabled).

12.5.5 Auto-Refresh Operation

When the auto-refresh enable bit (ARFEN) in the auto-refresh enable register 0 (DBRFEN) is 1, auto-refresh commands are issued periodically. If accessing data in the SDRAM, always make sure this is set.

The average interval for refreshing is set in refresh configuration registers 1 and 2.

The following is an example of the settings in refresh configuration registers 1 and 2 and the timing of refresh generation.
In the explanations, "a ± b" indicates the range of values from a-b to a + b.

To minimize variation of the refresh interval
Set REFPMAX to 0; in this case, the time between one round of refresh generation and the next will be REFINT_E ± REFTHF cycles. More generally, taking n as a positive integer, the time from one round of refresh generation to refresh generation n rounds later is n × REFINT_E ± REFTHF cycles. However, this is on the assumption of no writing to the DBRFEN register during this period.

For a flexible refresh interval
Set REFPMAX to a value greater than 1.
REFINT holds the setting for the average refresh interval (tREFI in the normal range of operating temperatures), which is given in memory-vendor datasheets as an integer number of cycles. Set a value which has been rounded down from this integer. Set REFINTS to 0 or 1 according to the temperature at the time.
In this case, taking n as a positive integer, the time from one round of refresh generation to refresh generation n rounds later is n × REFINT_E ± REFTHF cycles. However, this is on the assumption of no writing to the DBRFEN register during this period.

To minimize the average refresh interval
The lower limit on configurable values of REFINT is obtained from the following expression.
The configurable lower limit value to REFINT is the following expression.
Here "<<" means left–shifted operator. (REFTHF + TRFC + 2 × TMOD + TRDMR + 2 × TFAW + max {TFAW, TRAS, TRC – TRP, TRCD} + 100) << REFINTS.

If there is a bank that is open before the auto-refresh is carried out, the DBSC3 automatically uses the PREA (precharge all banks) command to precharge all of the banks, and then issues the REF (auto-refresh) command. Consequently, after the refresh takes place, data access for all of the banks will be in the missed page state.

12.5.6 Regarding Address Multiplexing

Memory of various sizes can be connected through the settings of the SDRAM configuration register (DBCONF). The BWIDTH bits are used to set the external data bus width, and the SPLIT bit is used to set the size of the memory connected. The BASFT bit setting is used to move the position of the bank address toward the lower bits; depending on the application the possibility of page hits may be increased.

Table 12.11 Relation between Address Pins and Logical Addresses (BKADM = 00) (When Two 16-bit-width SDRAM Devices or Four 8-bit-width SDRAM Devices are Connected)

Memory Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
32M × 32b	ROW	A14	A13	A12	—	—	—	—	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A14	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
64M × 32b	ROW	A14	A13	A12	—	—	—	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A14	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
128M × 32b	ROW	A14	A13	A12	—	—	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A14	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
256M × 32b	ROW	A14	A13	A12	—	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A14	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
512M × 32b	ROW	A14	A13	A12	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A14	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0

Notes: 1. A31 to A0 are the logical address bits in byte units, with A31 and A0 indicating the MSB and LSB sides, respectively.
2. AP indicates the auto precharge option.
3. $\overline{BC}$: Burst chop is not supported.

Table 12.12 Relation between SDRAM Address Pins and Logical Addresses (BKADM = 01) (When Two 16-bit-width SDRAM Devices or Four 8-bit-width SDRAM Devices are Connected)

Memory Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
32M × 32b	ROW	A26	A13	A12	—	—	—	—	A14	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A26	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
64M × 32b	ROW	A27	A13	A12	—	—	—	A14	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A27	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
128M × 32b	ROW	A28	A13	A12	—	—	A14	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A28	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
256M × 32b	ROW	A29	A13	A12	—	A14	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A29	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
512M × 32b	ROW	A30	A13	A12	A14	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A30	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0

Notes: 1. A31 to A0 are the logical address bits in byte units, with A31 and A0 indicating the MSB and LSB sides, respectively.
2. AP indicates the auto precharge option.
3. $\overline{BC}$: Burst chop is not supported.

Table 12.13 Relation between SDRAM Address Pins and Logical Addresses (BKADM = 10) (When Two 16-bit-width SDRAM Devices or Four 8-bit-width SDRAM Devices are Connected)

Memory Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
32M × 32b	ROW	A26	A25	A12	—	—	—	—	A14	A13	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 0)	COL	A26	A25	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
32M × 32b	ROW	A26	A13	A12	—	—	—	—	A14	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 1)	COL	A26	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
64M × 32b	ROW	A27	A26	A12	—	—	—	A14	A13	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 0)	COL	A27	A26	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
64M × 32b	ROW	A27	A13	A12	—	—	—	A14	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 1)	COL	A27	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
128M × 32b	ROW	A28	A27	A12	—	—	A14	A13	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 0)	COL	A28	A27	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
128M × 32b	ROW	A28	A13	A12	—	—	A14	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 1)	COL	A28	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
256M × 32b	ROW	A29	A28	A12	—	A14	A13	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 0)	COL	A29	A28	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
256M × 32b	ROW	A29	A13	A12	—	A14	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 1)	COL	A29	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
512M × 32b	ROW	A30	A29	A12	A14	A13	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 0)	COL	A30	A29	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
512M × 32b	ROW	A30	A13	A12	A14	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 1)	COL	A30	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0

Notes: 1. A31 to A0 are the logical address bits in byte units, with A31 and A0 indicating the MSB and LSB sides, respectively.
2. AP indicates the auto precharge option.
3. $\overline{BC}$: Burst chop is not supported.

Table 12.14 Relation between SDRAM Address Pins and Logical Addresses (BKADM = 11) (When Two 16-bit-width SDRAM Devices or Four 8-bit-width SDRAM Devices are Connected)

Memory Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
32M × 32b	ROW	A26	A25	A24	—	—	—	—	A14	A13	A12	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A[26:25] = 00)	COL	A26	A25	A24	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
32M × 32b	ROW	A26	A25	A12	—	—	—	—	A14	A13	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A[26:25] = 01)	COL	A26	A25	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
32M × 32b	ROW	A26	A13	A12	—	—	—	—	A14	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 1)	COL	A26	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
64M × 32b	ROW	A27	A26	A25	—	—	—	A14	A13	A12	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A[26:25] = 00)	COL	A27	A26	A25	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
64M × 32b	ROW	A27	A26	A12	—	—	—	A14	A13	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A[26:25] = 01)	COL	A27	A26	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
64M × 32b	ROW	A27	A13	A12	—	—	—	A14	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 1)	COL	A27	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
128M × 32b	ROW	A28	A27	A26	—	—	A14	A13	A12	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A[26:25] = 00)	COL	A28	A27	A26	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
128M × 32b	ROW	A28	A27	A12	—	—	A14	A13	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A[26:25] = 01)	COL	A28	A27	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
128M × 32b	ROW	A28	A13	A12	—	—	A14	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 1)	COL	A28	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
256M × 32b	ROW	A29	A28	A27	—	A14	A13	A12	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A[26:25] = 00)	COL	A29	A28	27	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
256M × 32b	ROW	A29	A28	A12	—	A14	A13	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A[26:25] = 01)	COL	A29	A28	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
256M × 32b	ROW	A29	A13	A12	—	A14	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 1)	COL	A29	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
512M × 32b	ROW	A30	A29	A28	A14	A13	A12	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A[26:25] = 00)	COL	A30	A29	A28	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
512M × 32b	ROW	A30	A29	A12	A14	A13	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A[26:25] = 01)	COL	A30	A29	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
512M × 32b	ROW	A30	A13	A12	A14	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
(A26 = 1)	COL	A30	A13	A12	—	—	—	$\overline{BC}$	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0

Notes: 1. A31 to A0 are the logical address bits in byte units, with A31 and A0 indicating the MSB and LSB sides, respectively.
2. AP indicates the auto precharge option.
3. $\overline{BC}$: Burst chop is not supported.

12.5.7 Regarding SDRAM Access and Timing Constraints

In this section, waveforms at the various pins during basic DDR3-SDRAM access are explained first and then the relation between DDR3-SDRAM access and the CAS latency (CL), tRAS, tRFC, tRCD, tRP, tRRD, tWR, tRTP, tRC, READ-WRITE minimum interval, WRITE-READ minimum interval set using the SDRAM timing registers 0 to 17 (DBTR0 to DBTR17) is explained.

(1) Basic SDRAM Access

In this section, waveforms at the various pins during basic SDRAM access, including reading, writing, auto-refresh, and self-refresh operations, are explained.

Figure 12.5 shows waveforms for 1/2/4/8/16/32-byte reading. In this example, read access processing is executed for bank A after the ACT command is issued, but when there is a page hit, access begins with the issue of the READ command.

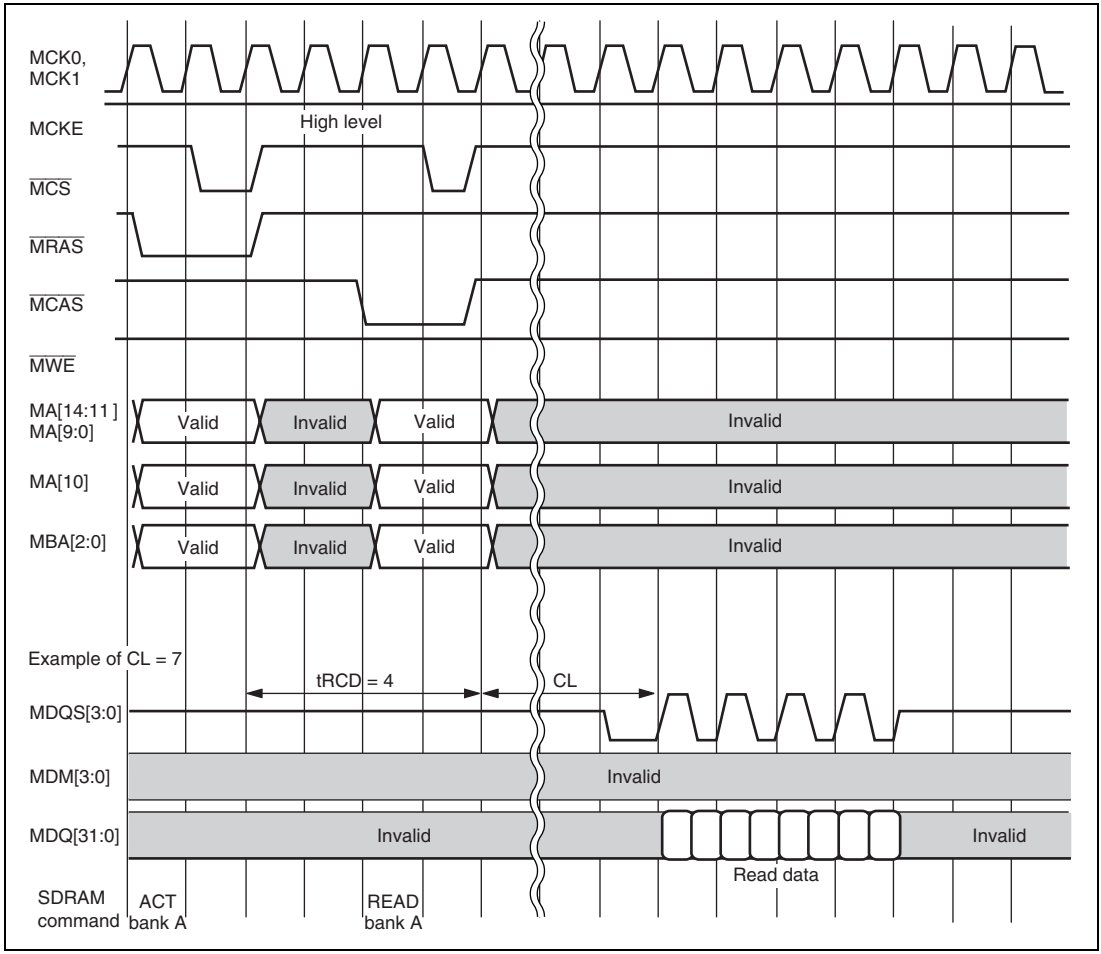


Figure 12.5 Waveforms for 1/2/4/8/16/32-Byte Reading

Figure 12.6 shows waveforms for 1/2/4/8/16/32-byte writing. In this example, write access processing is executed for bank A after the ACT command is issued, but when there is a page hit, access begins with the issue of the WRITE command.

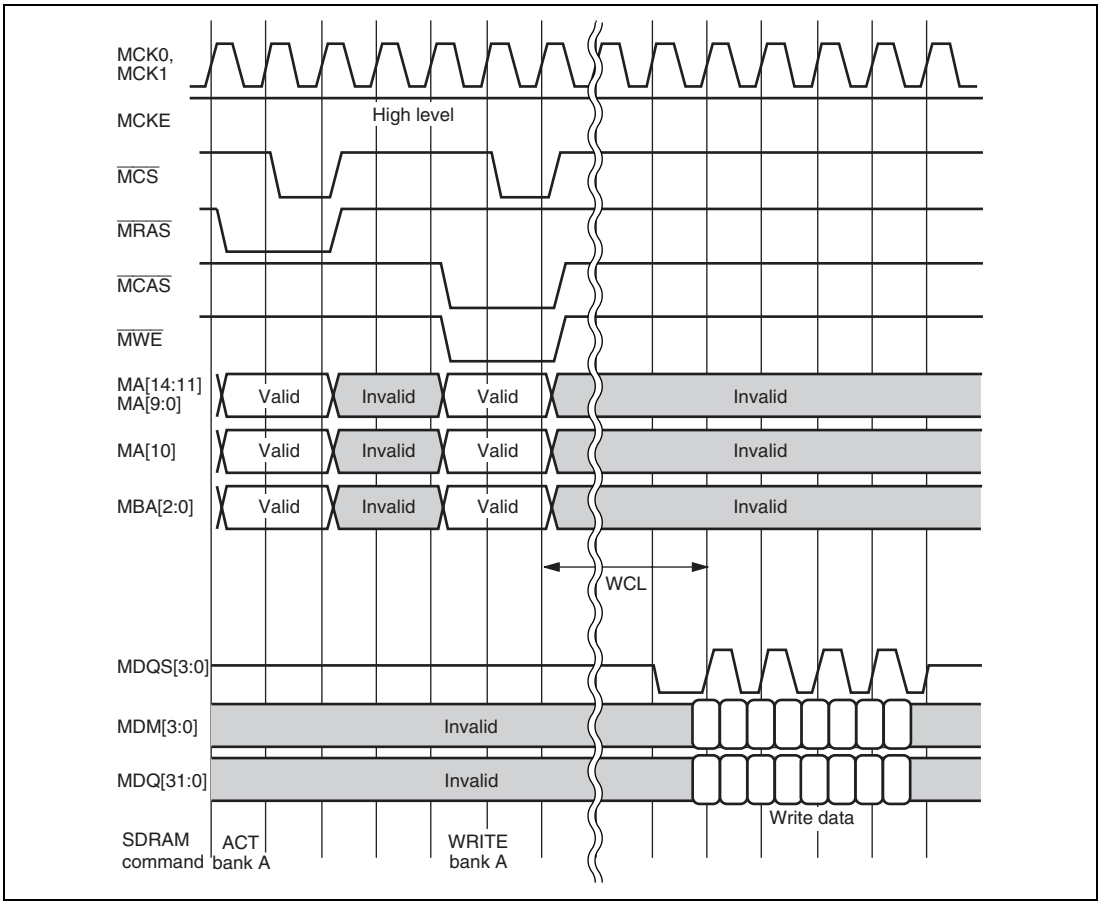


Figure 12.6 Waveforms for 1/2/4/8/16/32-Byte Writing

Figure 12.7 shows waveforms during auto-refresh operation resulting from settings of the SDRAM refresh control registers 0, 1, and 2. The DBSC3 issues a REF command automatically after the PALL command is issued when at least one DDR3-SDRAM bank is activated before the REF command. Consequently, there is no need to use software to manage precharging of all the banks for the auto-refresh operation.

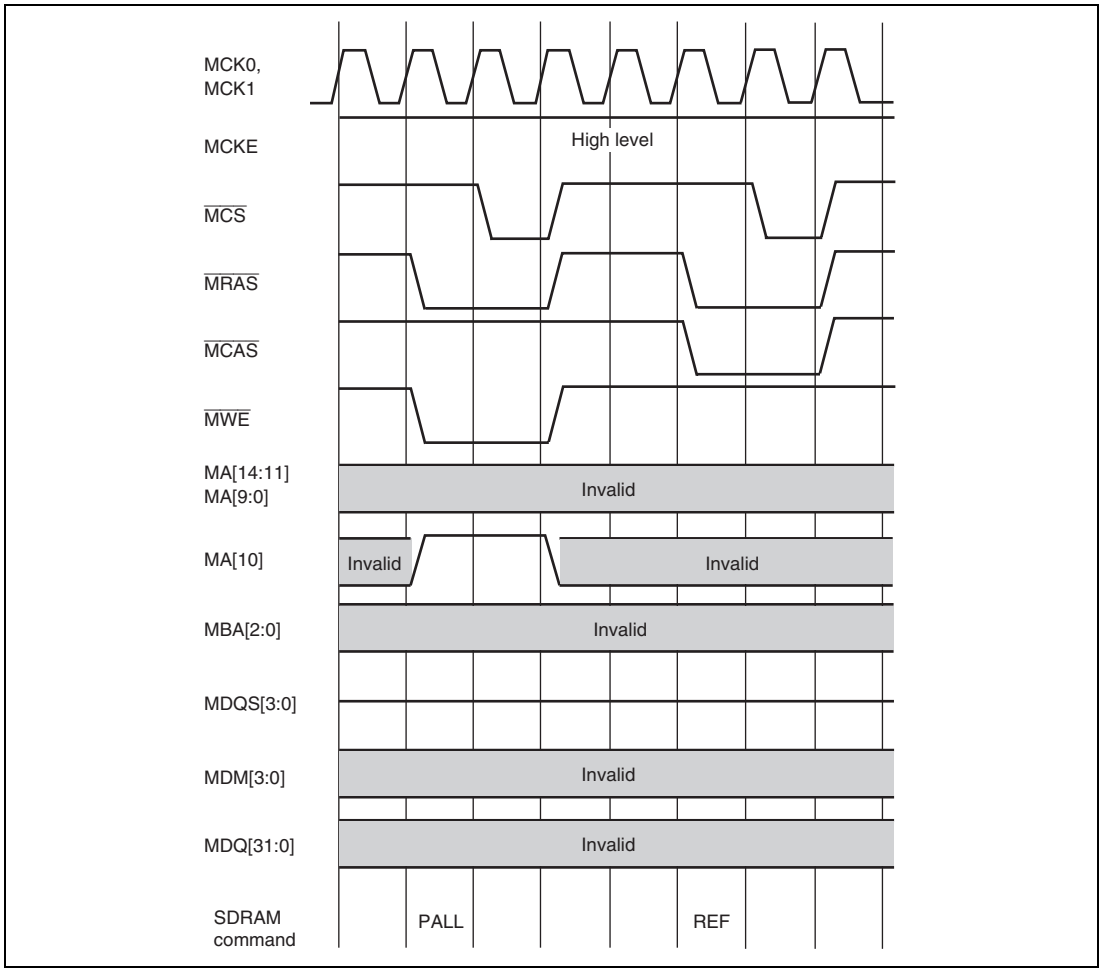


Figure 12.7 Auto-Refresh Operation

Figure 12.8 shows the self-refresh operation. In order to perform self-refresh operation, the sequence must be observed. For details, refer to section 12.5.4, Self-Refresh Operation.

When performing processing according to the sequence in section 12.5.4, Regarding Self-Refresh Operation, commands to be issued to the SDRAM are those shown in figure 12.10. Before the

transition to self-refresh, the PALL command is issued in software. Then, software is used to issue the REF command, and the SRX (self-refresh entry from IDLE) command is issued. The SDRAM continues in self-refresh mode until self-refresh is cancelled in software. After issuing the SLFRSHX (self-refresh exit) command in software, it is necessary to wait for the time (tXSNR) specified in the datasheet for the SDRAM being used until issuing a REF command.

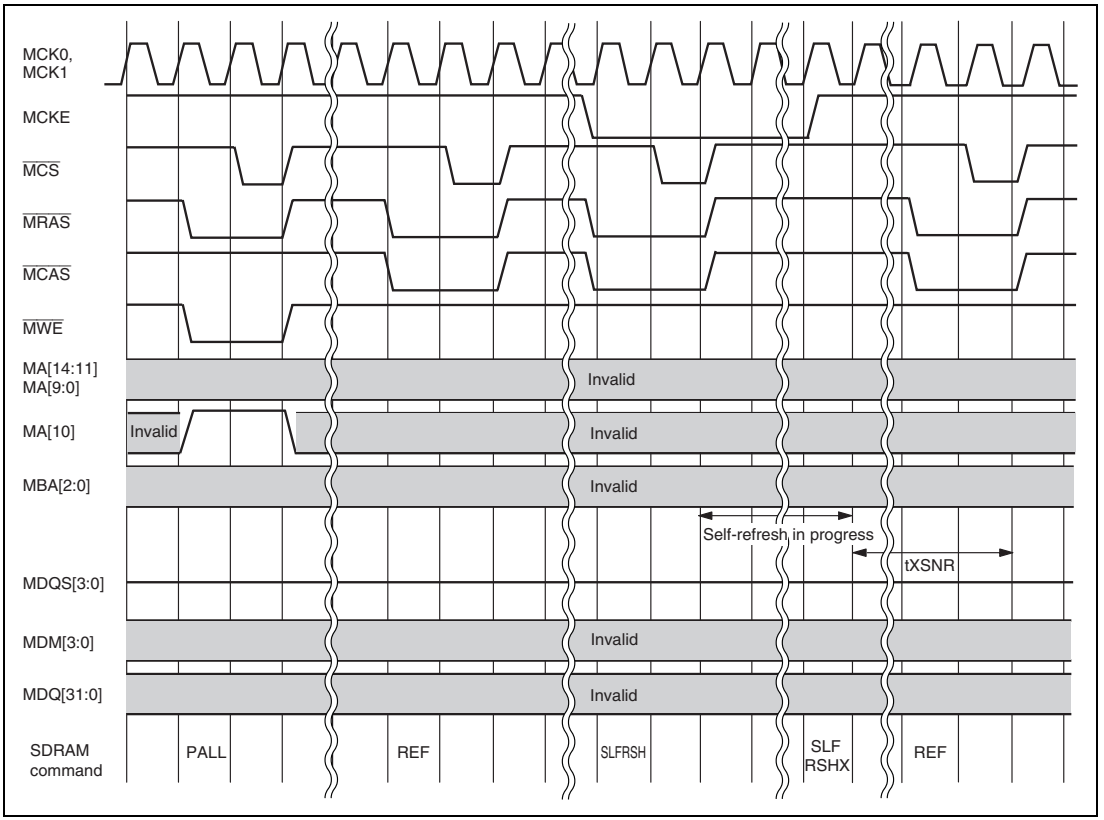


Figure 12.8 Self-Refresh Operation

(2) Regarding Timing Constraints

Figure 12.9 shows the relation between the settings of CL, tRAS, tRCD, and tRP, and the issuing of commands. Figure 12.10 shows the relation to tRRD and tRTP, figure 12.11 shows the relation to tWR, figure 12.12 shows the relation to tRC, figure 12.13 shows the relation to READ-WRITE, figure 12.14 shows the relation to WRITE-READ, and figure 12.15 shows the relation to tRFC.

Figure 12.9 corresponds to operation in a case in which, with bank A open, there is read access of bank A and a page miss occurs. The constraint tRP between the PRE command and ACT command, the constraint tRCD between the ACT command and READ command, and the constraint tRAS between the ACT command and the PRE command are involved. The DBSC3 waits to issue commands until each of the constraints is satisfied.

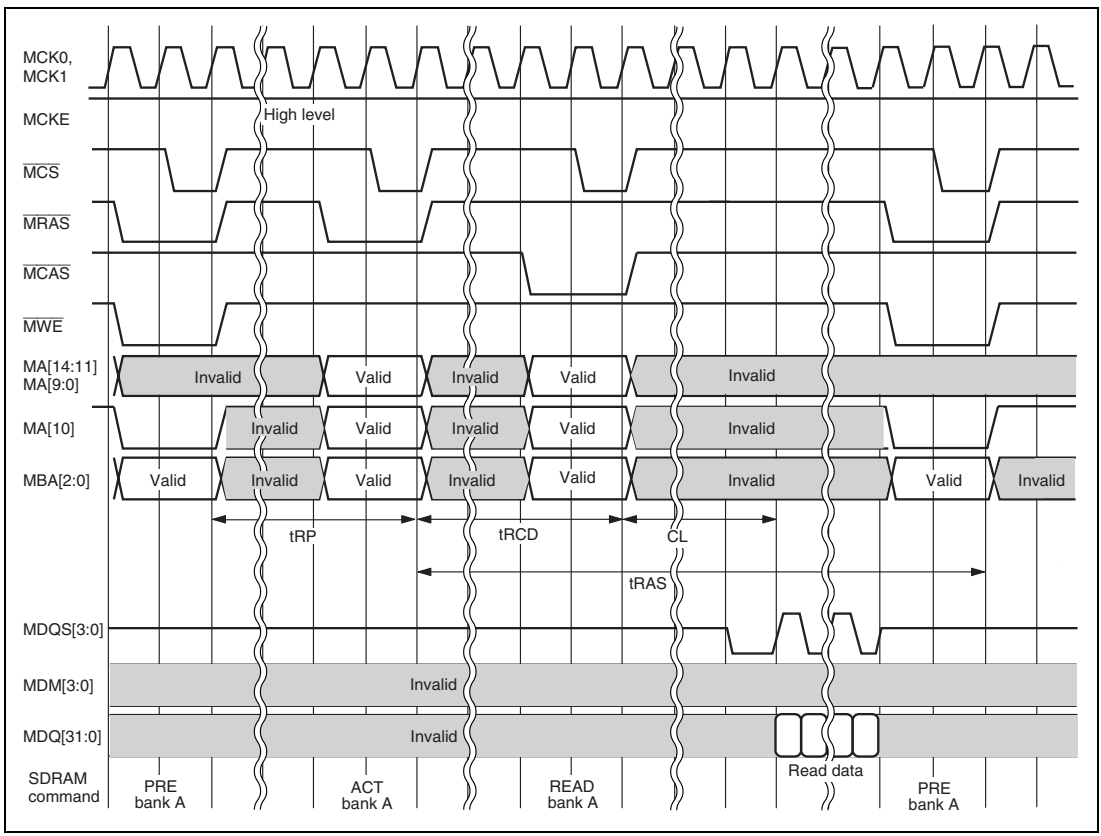


Figure 12.9 tRP, tRCD, CL, and tRAS

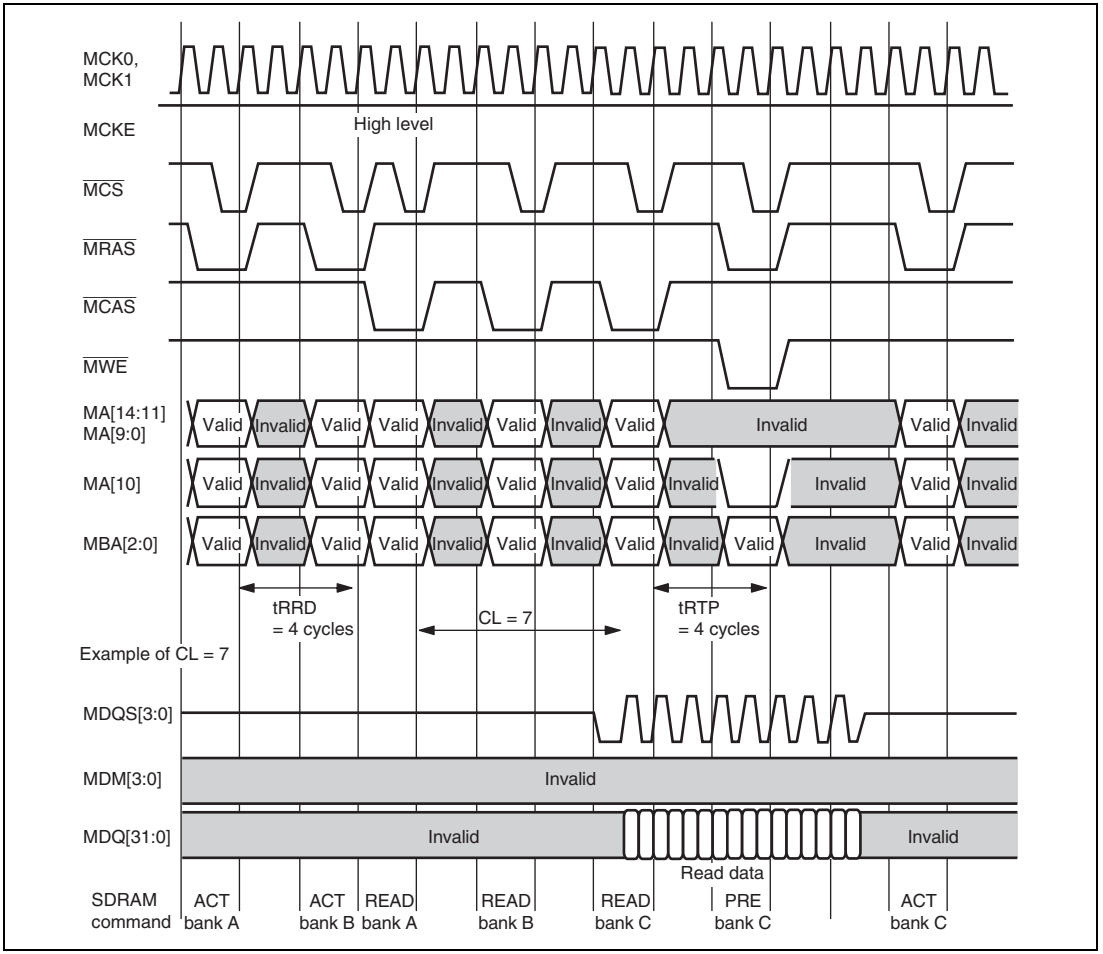


Figure 12.10 tRRD and tRTP

Figure 12.10 shows a case in which the pages for both of banks A and B are closed, the page for bank C is open, and a page hit has occurred. When the tRRD time constraint has been satisfied starting from issue of the ACT command for bank A, the ACT command for bank B is issued. Because time tRCD has elapsed from the issue of the ACT command for bank A, a READ command can be used. A further two cycles later, a READ command for bank C can be issued. However, the next request is access for which bank C must be closed, and so after the elapse of time tRTP a PRE command is issued.

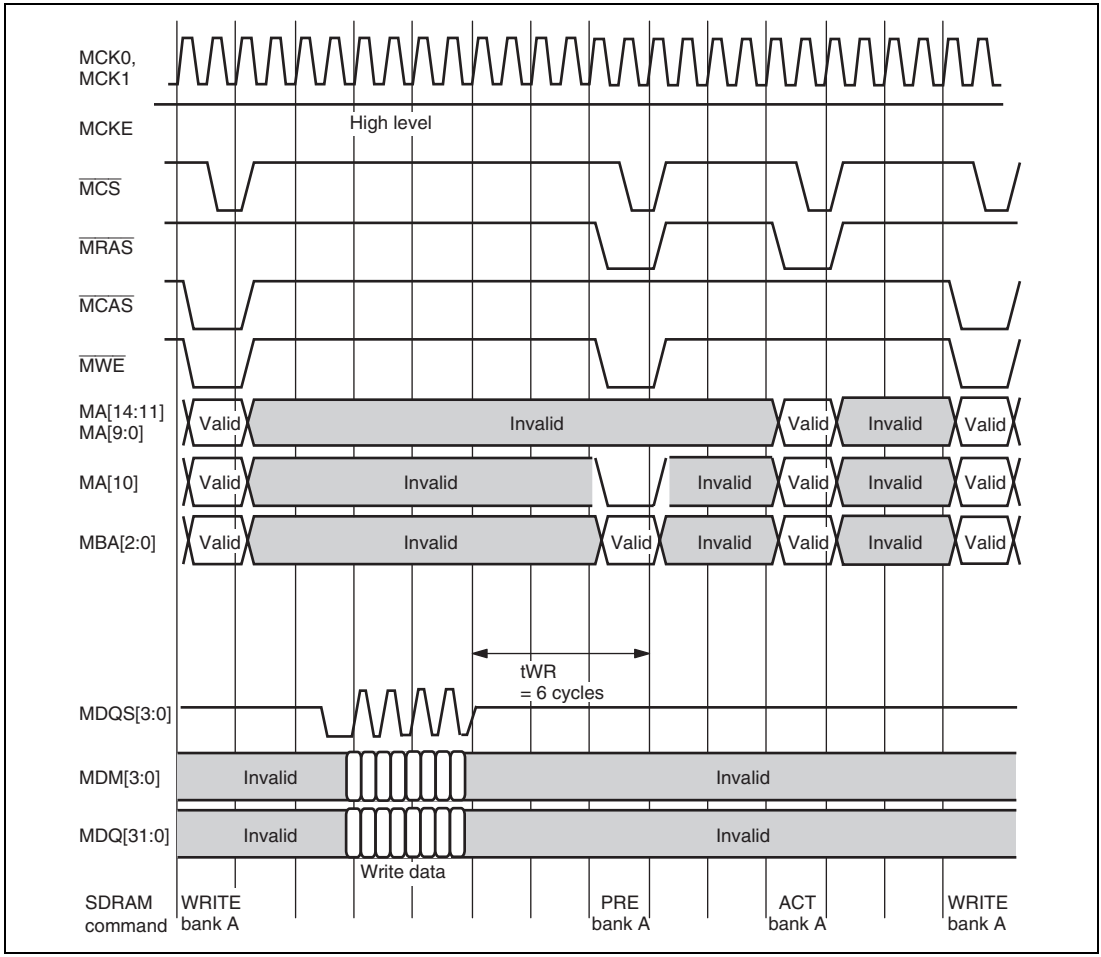


Figure 12.11 tWR

Figure 12.11 shows a case in which, after a write request, access occurs requiring that bank B be closed. After the issue of a WRITE command, it is necessary to wait for time tWR or longer after output of the write data before issuing a PRE command.

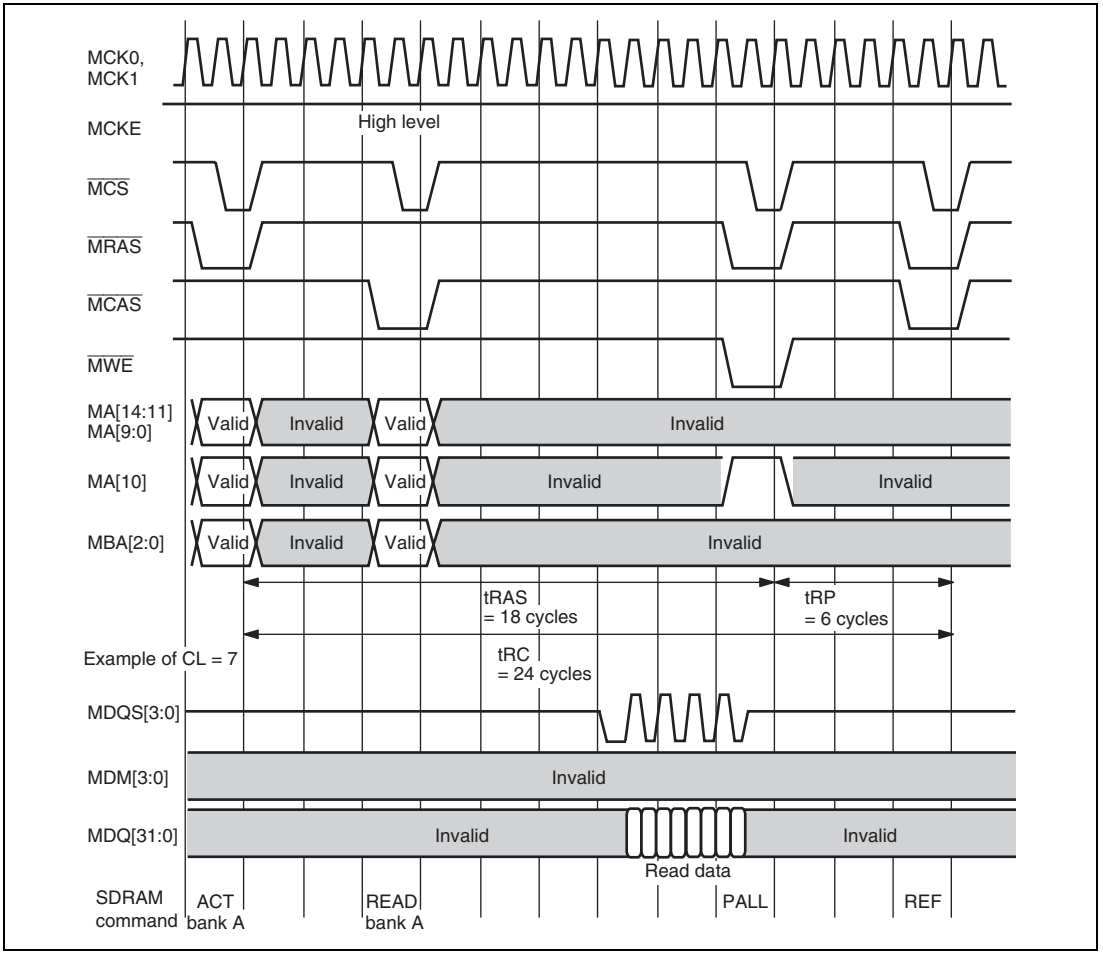


Figure 12.12 tRC

Figure 12.12 shows an example of performing auto-refresh after read access of bank A, the page for which had been closed. After issuing an ACT command and READ command for bank A and performing data reading, a PALL command must be used to close all banks in order to perform auto-refresh. In order to issue the PREA command, the tRAS time constraint must be satisfied, and issuing of the PREA command is delayed until this time. Then, when issuing the REF command, both of time constraints tRP and tRC must be satisfied simultaneously. When these constraints are both satisfied, the REF command is issued and auto-refresh is performed.

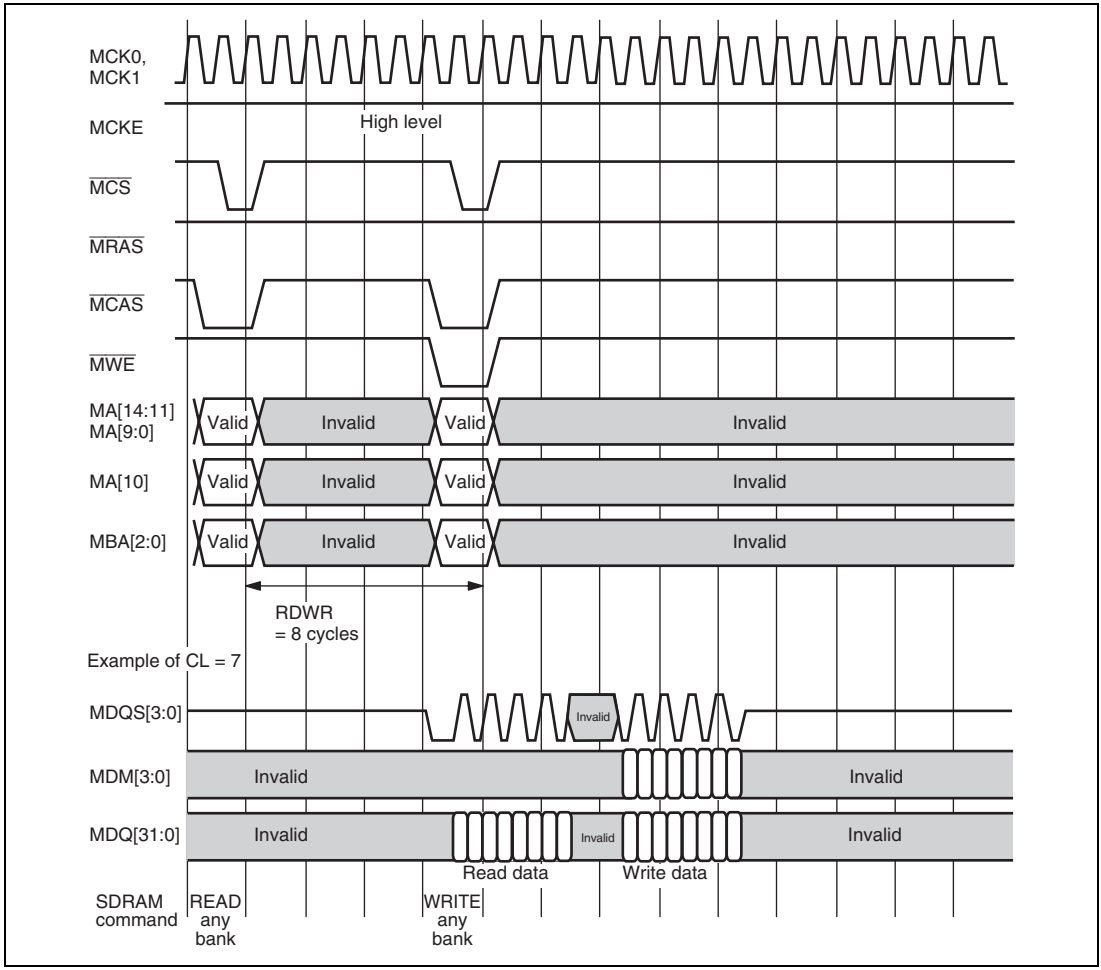


Figure 12.13 READ-WRITE Minimum Time

Figure 12.13 is an example of a case in which, after issuing a READ command, a WRITE command is issued. In order to issue the WRITE command after issuing the READ command, the DBSC3 waits for a minimum time stipulated by the RDWR bits.

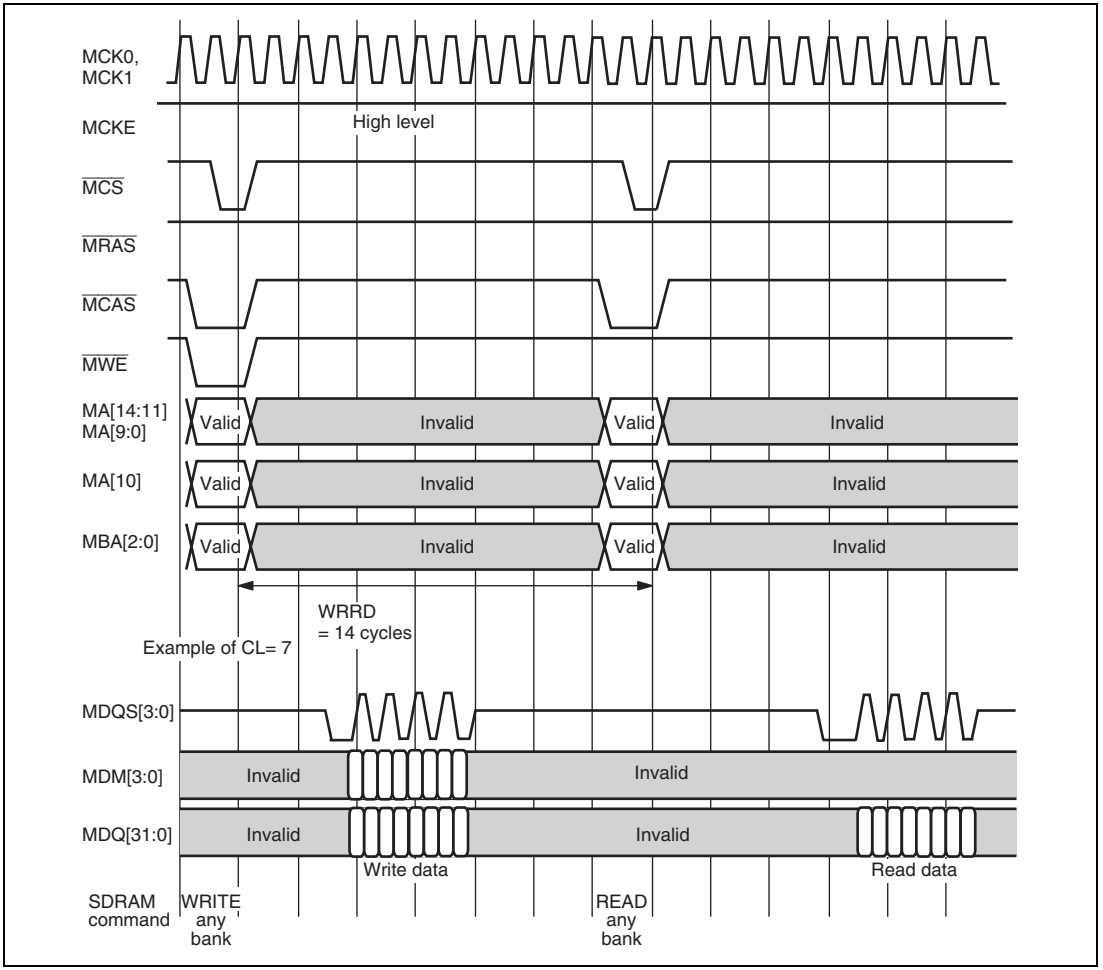


Figure 12.14 WRITE-READ Minimum Time

Figure 12.14 is an example of a case in which, after issuing a WRITE command, a READ command is issued. In order to issue the READ command after issuing the WRITE command, the DBSC3 waits for a minimum time stipulated by the WRRD bits.

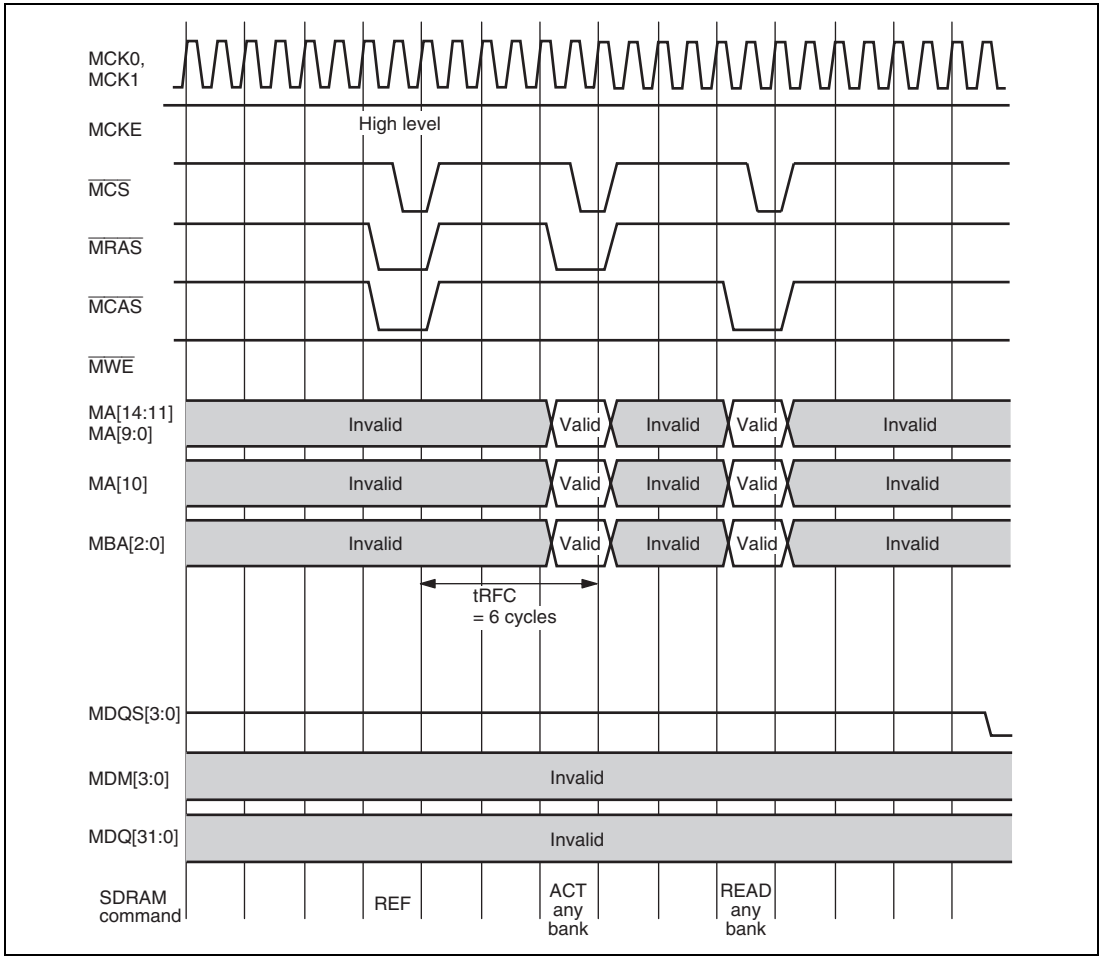


Figure 12.15 tRFC

Figure 12.15 is an example of a case in which, after issuing a REF command, a READ request is issued. In order to issue the ACT command after issuing the REF command, the DBSC3 waits for a time stipulated by tRFC.

12.5.8 DDR3-SDRAM Power Supply Backup Function

The SDRAM power supply backup function utilizes the SDRAM self-refresh state to turn off the power supply to the DBSC3 (1.2-V) except the IO cell block, while maintaining the data in the SDRAM. By using this function, not only is it possible to cut power consumption, but the time needed to transfer data once again to the SDRAM can be eliminated, since the valid data is maintained within the SDRAM (see figure 12.16). In order to realize this function, in addition to this LSI, a separate external control circuit (microcomputer or similar) is needed to monitor the states of this LSI and the memory.

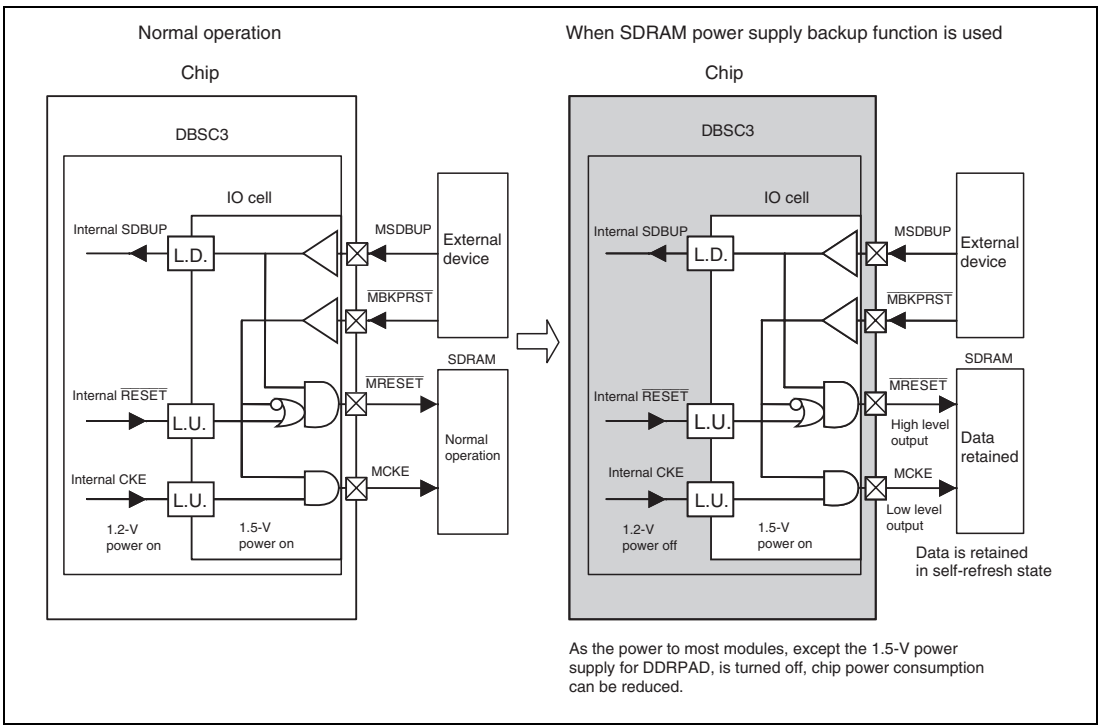


Figure 12.16 SDRAM Power Supply Backup Function

In order to implement the power supply backup function, a control signal $\overline{\text{MBKPRST}}$ is necessary to hold MCKE at low level even when power other than for the 1.5-V I/O is turned off. When this signal is at low level, pin MCKE can be held at low level even when the power supply within the chip is in the turned-off state. After using the DBSC3 to put the SDRAM into the self-refresh state, by using this $\overline{\text{MBKPRST}}$ signal to hold the MCKE signal at low level, the SDRAM self-refresh state can be maintained even when the power supply in the chip is turned off.

To cancel the power supply backup state, perform a power-on reset. As a result, the DBSC3 registers are initialized, and so the self-refresh control circuit is also initialized. In order to put the SDRAM into the self-refresh state before power-on reset, when the internal CKE signal is indefinite, and also during power-on reset, the $\overline{\text{MBKPRST}}$ signal must be held at low level. Power-on reset causes the DBSC3 to fix the internal CKE signal at low level, so that after power-on reset is released the $\overline{\text{MBKPRST}}$ signal is raised to high level. (If not in the power supply backup state, $\overline{\text{MBKPRST}}$ is always at high level and there is no problem).

Thus the power supply backup state is cancelled through power-on reset, and so the software must decide whether the normal SDRAM initialization sequence is necessary, or whether the LSI was in the power supply backup state. In order to perform this decision, a signal which indicates the back-up state to pin SDBUP is input to this LSI from an external control circuit. Pin SDUP is the 1.5-V I/O pin. Furthermore, the MRESET signal to the SRAM can be held at the high level while pin SDBUP is at the high level. The state of pin SDBUP can be checked by reading DBSC3 status register 1 (DBSTATE1).

After power-on reset, the software monitors the SDBUP-state signal, and judges whether the state should be the power supply backup state or whether SDRAM initialization is necessary. Before using register settings to send MCKE to high level, the state signal must be made to signify a state other than the power supply backup state. (After driving pin MCKE to high level, upon power-on reset the data within the SDRAM is destroyed. Hence if the state signal is not set in advance to a state other than power supply backup state, there is the danger that the destroyed data may be treated as the correct data).

In this way, procedures are used to make a transition to and cancel SDRAM power supply backup mode; if these procedures are not followed, the data in the SDRAM may be destroyed.

These procedures are explained below.

(1) Transition to SDRAM Power Supply Backup Mode

The following method is used.

1. Confirm that the controller is not being accessed. The time required for transition must not exceed the auto-refresh interval requested by the SDRAM by interrupts or some other causes.
2. Set the ACCEN bit in the SDRAM operation enable register (DBACEN) to 0 (access disabled).
3. Use the manual command-issuing register (DBCMD) to issue a PREA (precharge all banks) command. The value written to this register should be opc = PreA, arg = 0.
4. Use the manual command-issuing register (DBCMD) to issue a Self-Refresh Entry command. The value written to this register should be opc = SREn, arg = 0.

5. Set the ARFEN bit in the auto-refresh enable register (DBRFEN) to 0.
6. Use the manual command-issuing register (DBCMD) to insert a time to wait for the clock to stop. The value written to this register should be opc = Wait, arg = tCKSRE (normally, max {5,10ns}).
7. Read the operation completion waiting register (DBWAIT) and wait for the response.
8. Use a general-purpose port or other means to convey to the external device that the SDRAM has entered the self-refresh state. Upon receiving this notification, the external device should change the $\overline{\text{MBKPRST}}$ signal from high to low level and maintain the SDBUP at high level.
9. Turn off the unnecessary power, other than the DBSC3 1.5-V I/O.

(2) Recovery from SDRAM Power Supply Backup Mode

The following method is used.

1. Turn on the power supply to the LSI.
2. Input a power-on reset to the LSI.
3. After release of power-on reset, an external device should change the $\overline{\text{MBKPRST}}$ signal from low to high level.
4. The DBSC3 status register 1 (DBSTATE1) to determine whether the normal initialization sequence of the SDRAM, or to recover from power supply backup mode. For normal initialization sequence of the SDRAM, refer to section 12.5.3, Initialization Sequence.
5. Use the SDRAM kind setting register (DBKIND) to set the type of memory.
6. Set the SDRAM configuration setting register (DBCONF), SDRAM timing register 0 to 17 (DBTR0 to DBTR17).
7. Enter the settings of PHY-unit internal register address (DBPDRGA) and PHY-unit internal register access (DBPDRGD) for read enable setting registers 0 and 1.
8. Use the PHY-unit internal register 3 (DBPDCNT3) to select the auto-calibration.
9. Use the PHY-unit internal register 1 (DBPDCNT1) to start up the auto-calibration.
10. Ensure that the software allows the required waiting time (at least 10 µs).
11. Use PHY-unit control register 0 (DBPDCNT1) to release the DLL from the reset state.
12. Ensure that the software allows the required waiting time (at least 50 µs).
13. Use the manual command-issuing register (DBCMD) to insert the time to wait until release from self-refreshing (at the same time, make DBSC3 recognize that self-refreshing is proceeding). The value written to this register should be opc = SREn, arg = tCKSRX (normally, max {5,10ns}).
14. Set refresh configuration registers 0 to 2 (DBRFCNF0 to DBRFCNF2).
15. Set the ARFEN bit in the auto-refresh enable register (DBRFEN) to 1.

16. Use the manual command-issuing register (DBCMD) to issue the Self-Refresh Exit command. The value written to this register should be opc = SRXt, arg = 0.
17. Use manual command-issuing register (DBCMD) to insert time to wait until access to the SDRAM is enabled. The value written to this register should be opc = Wait, arg = tXSDLL (normally, 512).
18. Set the ACCEN bit in the SDRAM access enable register (DBACEN) to 1 (access enabled).
19. Wait for the response after reading the operation completion waiting register (DBWAIT). After that, normal access will be enabled.

Section 13 PCI Express Controller (PCIEC)

This module (PCIEC: PCI Express controller) performs PCI Express controls, and transfers data between this LSI internal bus (SuperHyway) and the PCI devices connected to the PCI Express. This module facilitates the design of systems using the PCI Express while at the same time permitting high-speed data transfers in a compact system.

This LSI incorporates a PCI Express physical module that supports a maximum of four lanes (×4) and another PCI Express physical module that supports one lane (×1). The module supporting four lanes (×4) can be separated into two lanes + one lane. Accordingly, the PCIEC can be used as two PCI Express modules (PCIEC0 with four lanes and PCIEC1 with one lane) or three PCI Express modules (PCIEC0 with two lanes, PCIEC1 with one lane, and PCIEC2 with one lane), which can be selected through mode pins.

The PCIEC has two modes: a Root Port and an Endpoint. The PCIEC can operate as a Root Port or Endpoint that is defined in the PCI Express specification.

13.1 Features

The PCI Express has the following features.

- Functions as a bridge that connects the PCI Express to the SuperHyway bus
  - Generates a PCI Express packet from the SuperHyway transaction to the PCI Express area
  - Generates an SuperHyway transaction from a PCI Express packet
- Supports high-speed data transfer between PCI Express devices and SuperHyway bus via the internal DMAC
- Supports the subset of PCI Express Base Specification Revision 1.1 (see Note for unsupported functions)
- Operates as a PCI Express Endpoint or a PCI Express Root Port
- Supports the requester and completer functions for the PCI Express transaction
- Incorporates configuration registers and the following capability structures
  - Power management capability structure
  - MSI capability structure
  - PCI Express capability structure
  - Virtual channel capability structure
- Supports interrupts by INTx and MSI
- Supports the error handling function by hardware

- Supports four lanes (×4), two lanes (×2), and one lane (×1) (PCIEC1 and PCIEC2 supports only one lane)
- Automatically performs link training and link configurations
- Supports the D+/D-line automatic swap function (lane polarity inversion)
- Guarantees the data integrity by automatic retry (retransfer), and LCRC generation and check functions using the Ack/Nak unit
- Supports the hardware flow control by the credit management
- Supports a maximum of 250 Mbytes/sec transfer rate per a lane
- Supports the link power control function (L0, L0s, L1, and L3 states)

Note: This PCIEC does not support the following PCI Express functions.

- Expansion ROM for initialization and system boot
- Card bus
- Extended 8-bit tag ID function
- Multiple function and phantom function
- Power-management event (PME) handling by hardware
- Hot Plug (including MRL, Power Controller, and Attention Button) support
- Low-power L2 state
- ECRC (CRC error check in the TLP layer)

13.2 Block Diagram

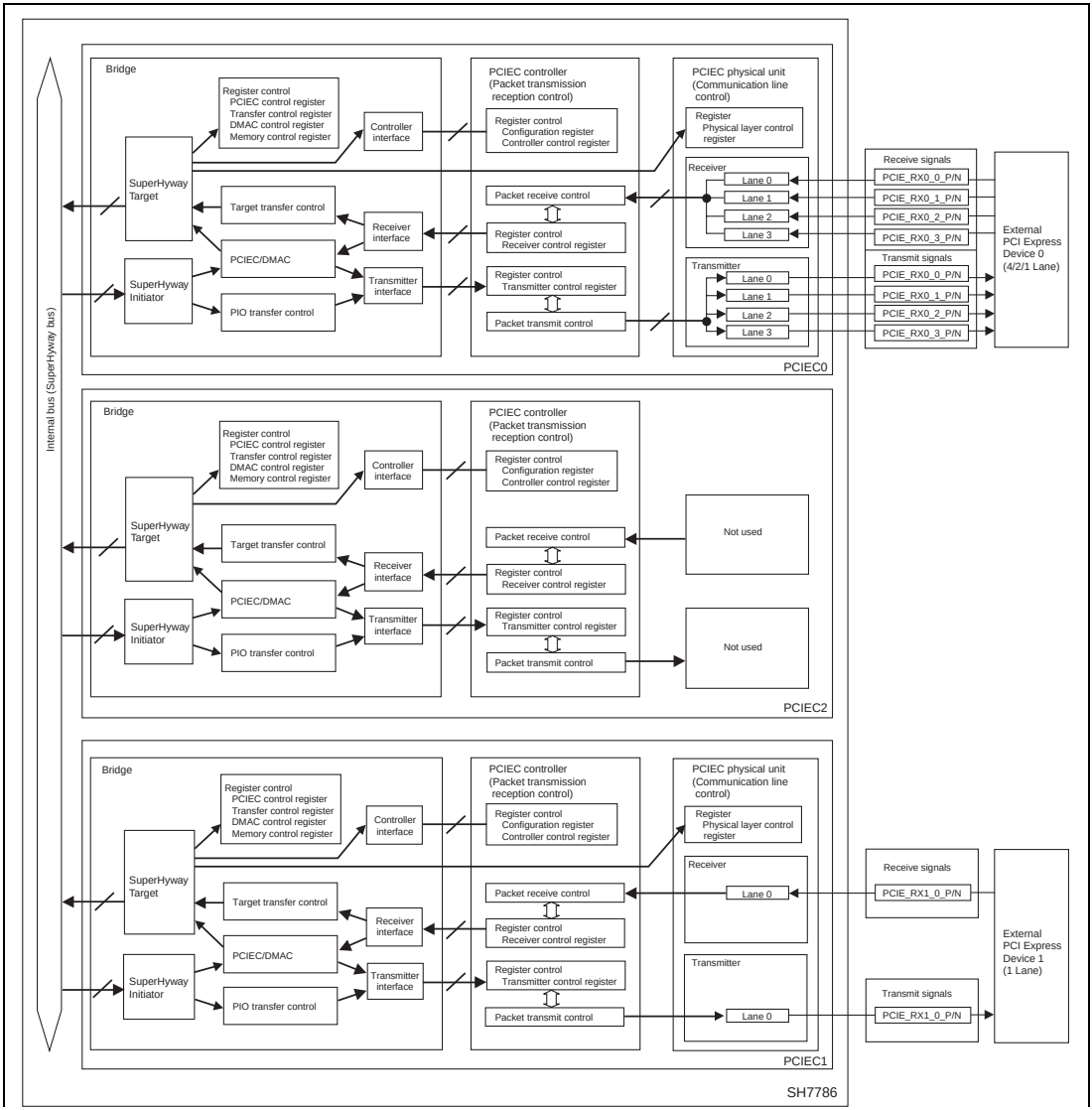
Figure 13.1 shows a block diagram of the PCIEC.

The PCIEC comprises a bridge, PCIEC controller and PCIEC physical unit.

The bridge provides the bridge function of connecting the SuperHyway bus to the PCI Express, and principally generates the PCI Express packets to be transmitted and processes received PCI Express packets. The PIO transfer controller generates PCI Express packets in response to a request by the SuperHyway bus; the target transfer controller generates SuperHyway requests from PCI Express packets, thus permitting data communications between the PCI Express and SuperHyway. The DMAC for high-speed data communications generates SuperHyway requests and PCI Express packets from a specified transfer setting, and transfers data in a manner that does not burden the CPU.

The PCIEC controller is a block that controls the PCI Express packet transmission and reception, and it implements the functions of TLP and data link layers that are defined by the PCI Express standards. The configuration register defined by the PCI Express standards is installed in this block.

The PCIEC physical unit is a block that controls PCI Express transmission paths, and implements the physical layer function that is defined by the PCI Express standards.


Internal bus (SuperHyway bus)
Bridge
Register control
PCIEC control register
Transfer control register
DMAC control register
Memory control register
Controller interface
SuperHyway Target
SuperHyway Initiator
Target transfer control
PCIEC/DMAC
PIO transfer control
Receiver interface
Transmitter interface
PCIEC controller (Packet transmission reception control)
Register control
Configuration register
Controller control register
Packet receive control
Register control
Receiver control register
Register control
Transmitter control register
Packet transmit control
PCIEC physical unit (Communication line control)
Register
Physical layer control register
Receiver
Lane 0
Lane 1
Lane 2
Lane 3
Transmitter
Receive signals
Transmit signals
PCIE_RX0_0_P/N
PCIE_RX0_1_P/N
PCIE_RX0_2_P/N
PCIE_RX0_3_P/N
External PCI Express Device 0 (4/2/1 Lane)
PCIEC0
Not used
PCIEC2
PCIE_RX1_0_P/N
External PCI Express Device 1 (1 Lane)
PCIEC1
SH7786

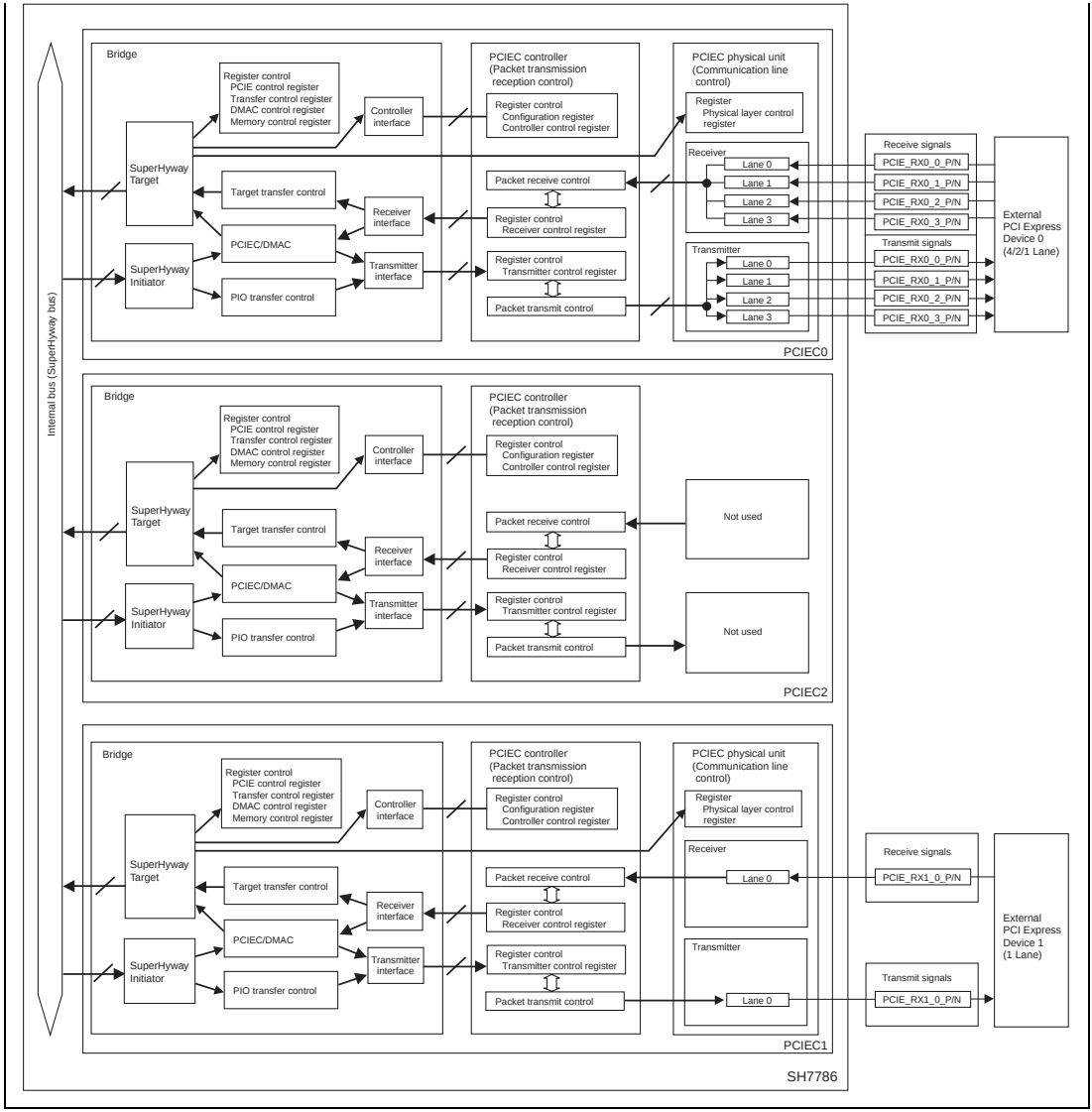


Figure 13.1 PCIEC Block Diagram (a) (×4)

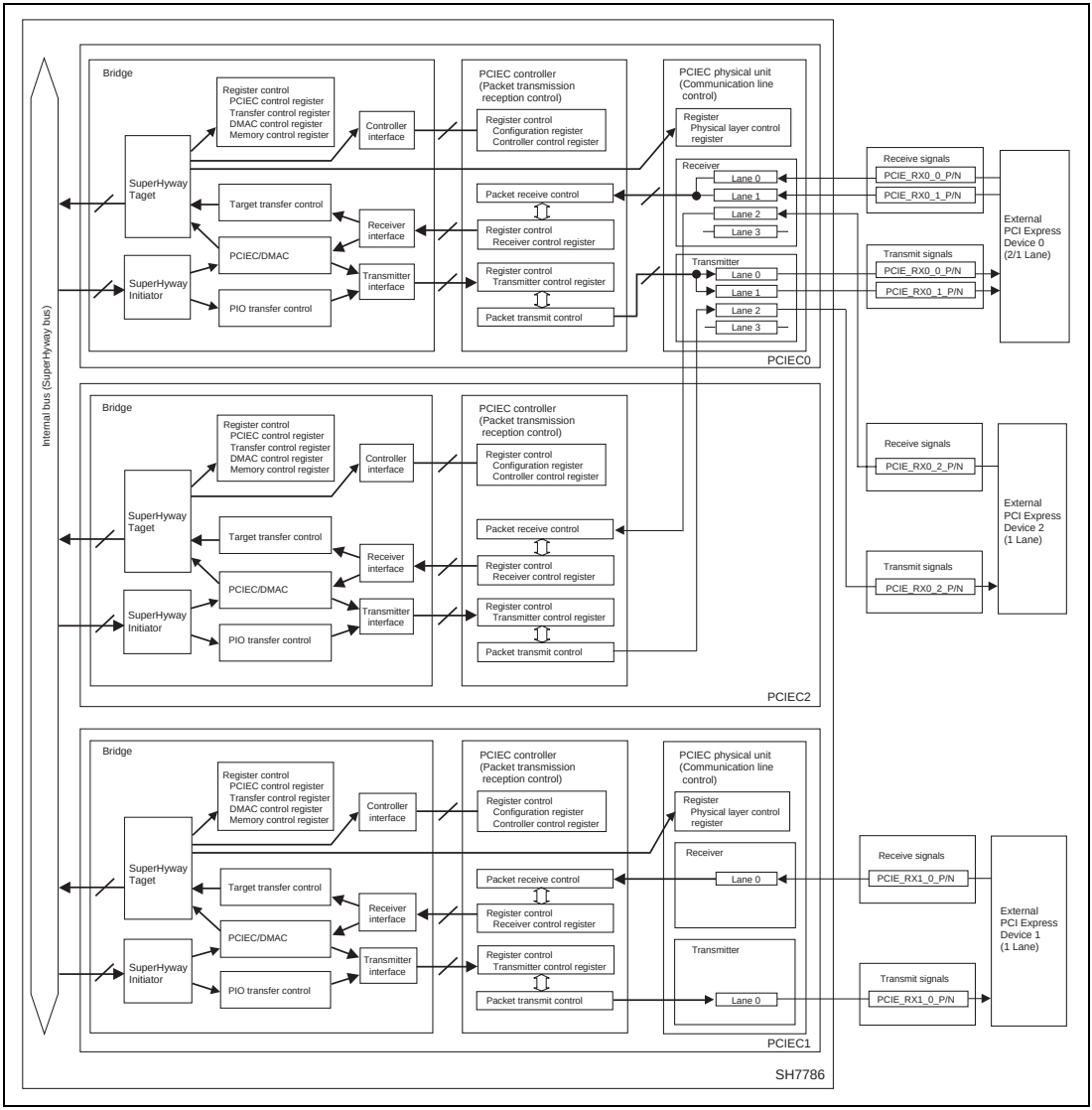


Figure 13.1 PCIEC Block Diagram (b) (×2/×1 Division)

13.3 Input/Output Pins

Table 13.1 shows the PCIE pin configuration.

Table 13.1 Pin Configuration

Signal Name	Signal Conforming to PCI Standard	Input/ Output	Description
GCLKP GCLKN	REFCLK+ REFCLK-	IN	Reference clock inputs to the PLL incorporated in the PCI Express module (differential inputs). 100-MHz clock should be applied.
PCIE_TX0_0_P PCIE_TX0_0_N PCIE_TX0_1_P PCIE_TX0_1_N PCIE_TX0_2_P PCIE_TX0_2_N PCIE_TX0_3_P PCIE_TX0_3_N	PETp0 PETn0 PETp1 PETn1 PETp2 PETn2 PETp3 PETn3	OUT	Transmit data pins used by a ×4 physical module, where 2.5-GHz signals are propagated. (Differential output) When the ×4 physical module is used as a four-lane module, all of these pins configure four lanes. When the module is separated into two lanes + one lane, PCIE_TX0_0_P, PCIE_TX0_0_N, PCIE_TX0_1_P, and PCIE_TX0_1_N configure two lanes, PCIE_TX0_2_P and PCIE_TX0_2_N configure one lane, and the other ports are not used.
PCIE_RX0_0_N PCIE_RX0_0_P PCIE_RX0_1_N PCIE_RX0_1_P PCIE_RX0_2_N PCIE_RX0_2_P PCIE_RX0_3_N PCIE_RX0_3_P	PERp0 PERn0 PERp1 PERn1 PERp2 PERn2 PERp3 PERn3	IN	Receive data pins used by a ×4 physical module, where 2.5-GHz signals are propagated. (Differential input) When the ×4 physical module is used as a four-lane module, all of these pins configure four lanes of PCIEC0. When the module is separated into two lanes + one lane, PCIE_RX0_0_P, PCIE_RX0_0_N, PCIE_RX0_1_P, and PCIE_RX0_1_N configure two lanes, PCIE_RX0_2_P and PCIE_RX0_2_N configure one lane, and the other ports are not used.
PCIE_TX1_0_P PCIE_TX1_0_N	PETp0 PETn0	OUT	Transmit data pins used by a ×1 physical module, where 2.5-GHz signals are propagated. (Differential output)
PCIE_RX1_0_P PCIE_RX1_0_N	PERp0 PERn0	IN	Receive data pins used by a ×1 physical module, where 2.5-GHz signals are propagated. (Differential input)

Signal Name	Signal Conforming to PCI Standard	Input/ Output	Description
MODE11	—	IN	PCI Express Operating Mode Selection 0: Root Port Specifies that the PCIEC operates as a PCI Express Root Port. 1: Endpoint Specifies that the PCIEC operates as a PCI Express Endpoint. The setting through this signal is valid for PCIEC0, PCIEC1, and PCIEC2.
MODE12	—	IN	PCI Express Lane Separation Selection 0: The module supporting four lanes (×4) is used as a four-lane module. PCIEC0 supports four lanes and PCIEC2 is not used. 1: The module supporting four lanes (×4) is separated into two lanes and one lane. PCIEC0 supports two lanes and PCIEC2 supports one lane. For either setting, the module supporting one lane (×1) works as PCIEC1 supporting one lane.

13.4 Register Descriptions

Table 13.3 shows the PCIEC register configuration. Table 13.5 shows the register states in each operating mode.

In table 13.3, the Address Offset column shows an offset for each register from the PCIEC0 to PCIEC2 base address. The actual access address is obtained by adding each offset to the base address. Table 13.2 shows the base addresses of PCIEC0 to PCIEC2. These registers should be accessed through the SuperHyway bus. The addresses and offsets of the PCI configuration registers are those in little endian mode. Addresses and offsets of the PCI configuration registers are those in little endian mode. The maximum access size of each register is shown as access size. 32-/16-/8-bit access to the registers in the PCI configuration register space is possible.

Table 13.4 shows the physical layer control register configuration and table 13.6 shows the physical layer register states in each operating mode. These registers should be accessed through the PCIEC registers. For details of access procedure, refer to section 13.5.14 Access to Physical Layer Control Register.

Access to the registers, which are specified as reserved, is prohibited.

Table 13.2 PCIEC Base Address

	PCIEC0	PCIEC1	PCIEC2
Base Address	H'FE00_0000	H'FE20_0000	H'FCC0_0000

Table 13.3 Register Configuration

Register	Abbreviation	SH* R/W	PCI* R/W	Address Offset	PCIEC0 Initial Value	PCIEC1 Initial Value	PCIEC2 Initial Value	Access Size
PCIEC control registers								
Enable register	PCIEENBLR	R/W	—	H'00_0008	H'0000 0001	H'0000 0001	H'0000 0001	32
Enable control register	PCIEECR	R/W	—	H'00_000C	H'0000 0000	H'0000 0000	H'0000 0000	32
PIO address register	PCIEPAR	R/W	—	H'00_0010	H'0000 0000	H'0000 0000	H'0000 0000	32
PIO control register	PCIEPCTLR	R/W	—	H'00_0018	H'0000 0000	H'0000 0000	H'0000 0000	32
PIO data register	PCIEPDR	R/W	—	H'00_0020	—	—	—	32
Message address lower register	PCIEMSGALR	R/W	—	H'00_0030	H'0000 0000	H'0000 0000	H'0000 0000	32
Message address upper register	PCIEMSGAHR	R/W	—	H'00_0034	H'0000 0000	H'0000 0000	H'0000 0000	32
Message control register	PCIEMSGCTLR	R/W	—	H'00_0038	H'0000 0000	H'0000 0000	H'0000 0000	32
Message data register	PCIEMSGD	W	—	H'00_0040	—	—	—	32
Unlock control register	PCIEUNLOCKCR	R/W	—	H'00_0048	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI-ID register	PCIEIDR	R	—	H'00_0060	H'0101 1101	H'0101 1101	H'0101 1101	32
Debug control register	PCIEDBGCTLR	R/W	—	H'00_0100	H'0000 0000	H'0000 0000	H'0000 0000	32
INTx register	PCIEINTXR	R/W	—	H'00_4000	H'0000 0000	H'0000 0000	H'0000 0000	32
Receive message register	PCIERMSGR	R/W1C	—	H'00_4010	H'0000 0000	H'0000 0000	H'0000 0000	32
Receive message interrupt enable register	PCIERMSGIER	R/W	—	H'00_4040	H'0000 0000	H'0000 0000	H'0000 0000	32
Reset control register 0	PCIERSTR0	R/W	—	H'00_8000	H'0000 0000	H'0000 0000	H'0000 0000	32
Reset control register 1	PCIERSTR1	R/W	—	H'00_8004	H'0000 0000	H'0000 0000	H'0000 0000	32
Reset control register 2	PCIERSTR2	R/W	—	H'00_8008	H'0000 0000	H'0000 0000	H'0000 0000	32
Reset control register 4	PCIERSTR3	R/W	—	H'00_800C	H'0000 0000	H'0000 0000	H'0000 0000	32
Software reset control register	PCIESRSTR	R/W	—	H'00_8040	H'0000 0000	H'0000 0000	H'0000 0000	32
Physical layer control registers								
Physical layer control register	PCIEPHYCTLR	R/W	—	H'01_0000	H'0000 0000	H'0000 0000	H'0000 0000	32
Physical layer address register	PCIEPHYADRR	R/W	—	H'01_0004	H'0000 0000	H'0000 0000	H'0000 0000	32
Physical layer data in register	PCIEPHYDINR	R	—	H'01_0008	H'0000 0000	H'0000 0000	H'0000 0000	32
Physical layer data out register	PCIEPHYDOUTR	R/W	—	H'01_000C	H'0000 0000	H'0000 0000	H'0000 0000	32
Physical layer status register	PCIEPHYSR	R	—	H'01_0010	H'0000 0000	H'0000 0000	H'0000 0000	32

Register	Abbreviation	SH* R/W	PCI* R/W	Address Offset	PCIEC0 Initial Value	PCIEC1 Initial Value	PCIEC2 Initial Value	Access Size
PCIEC transfer control registers								
Transfer control register	PCIETCTLR	R/W	—	H'02_0000	H'0000 0008	H'0000 0008	H'0000 0008	32
Transfer status register	PCIETSTR	R	—	H'02_0004	H'0000 0000	H'0000 0000	H'0000 0000	32
Interrupt register	PCIEINTR	R/W1C	—	H'02_0008	H'0000 0000	H'0000 0000	H'0000 0000	32
Interrupt enable register	PCIEINTER	R/W	—	H'02_000C	H'0000 0000	H'0000 0000	H'0000 0000	32
Error header 0 register	PCIEEH0R	R	—	H'02_0010	H'0000 0000	H'0000 0000	H'0000 0000	32
Error header 1 register	PCIEEH1R	R	—	H'02_0014	H'0000 0000	H'0000 0000	H'0000 0000	32
Error header 2 register	PCIEEH2R	R	—	H'02_0018	H'0000 0000	H'0000 0000	H'0000 0000	32
Error header 3 register	PCIEEH3R	R	R	H'02_001C	H'0000 0000	H'0000 0000	H'0000 0000	32
Error source register	PCIEERRFR	R/W	R/W	H'02_0020	H'0000 0000	H'0000 0000	H'0000 0000	32
Error interrupt register	PCIEERRFER	R/W	R/W	H'02_0024	H'0000 0000	H'0000 0000	H'0000 0000	32
Error source register 2	PCIEERRFR2	R/W	R/W	H'02_0028	H'0000 0000	H'0000 0000	H'0000 0000	32
MSI register	PCIEMSIR	R/W	—	H'02_0040	H'0000 0000	H'0000 0000	H'0000 0000	32
MSIF register	PCIEMSIFR	R/W	R/W	H'02_0044	H'0000 0000	H'0000 0000	H'0000 0000	32
Power-down control register	PCIEPWRCTLR	R/W	—	H'02_0100	H'0000 0000	H'0000 0000	H'0000 0000	32
Packet connection control register	PCIEPCCTLR	R/W	—	H'02_0180	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address register 0	PCIELAR0	R/W	R/W	H'02_0200	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address mask register 0	PCIELAMR0	R/W	R/W	H'02_0208	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address register 1	PCIELAR1	R/W	R/W	H'02_0220	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address mask register 1	PCIELAMR1	R/W	R/W	H'02_0228	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address register 2	PCIELAR2	R/W	R/W	H'02_0240	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address mask register 2	PCIELAMR2	R/W	R/W	H'02_0248	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address register 3	PCIELAR3	R/W	R/W	H'02_0260	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address mask register 3	PCIELAMR3	R/W	R/W	H'02_0268	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address register 4	PCIELAR4	R/W	R/W	H'02_0280	H'0000 0000	H'0000 0000	H'0000 0000	32

Register	Abbreviation	SH* R/W	PCI* R/W	Address Offset	PCIEC0 Initial Value	PCIEC1 Initial Value	PCIEC2 Initial Value	Access Size
Local (SuperHyway) address mask register 4	PCIELAMR4	R/W	R/W	H'02_0288	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address register 5	PCIELAR5	R/W	R/W	H'02_02A0	H'0000 0000	H'0000 0000	H'0000 0000	32
Local (SuperHyway) address mask register 5	PCIELAMR5	R/W	R/W	H'02_02A8	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address lower register 0	PCIEPALR0	R/W	—	H'02_0400	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address upper register 0	PCIEPAHR0	R/W	—	H'02_0404	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address mask register 0	PCIEPAMR0	R/W	—	H'02_0408	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI conversion control register 0	PCIEPTCTLR0	R/W	—	H'02_040C	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address lower register 1	PCIEPALR1	R/W	—	H'02_0420	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address upper register 1	PCIEPAHR1	R/W	—	H'02_0424	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address mask register 1	PCIEPAMR1	R/W	—	H'02_0428	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI conversion control register 1	PCIEPTCTLR1	R/W	—	H'02_042C	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address lower register 2	PCIEPALR2	R/W	—	H'02_0440	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address upper register 2	PCIEPAHR2	R/W	—	H'02_0444	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address mask register 2	PCIEPAMR2	R/W	—	H'02_0448	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI conversion control register 2	PCIEPTCTLR2	R/W	—	H'02_044C	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address lower register 3	PCIEPALR3	R/W	—	H'02_0460	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address upper register 3	PCIEPAHR3	R/W	—	H'02_0464	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI address mask register 3	PCIEPAMR3	R/W	—	H'02_0468	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI conversion control register 3	PCIEPTCTLR3	R/W	—	H'02_046C	H'0000 0000	H'0000 0000	H'0000 0000	32
PCIEC-DMAC control registers								
PCI DMAC DMA operation register	PCIEDMAOR	R/W	—	H'02_1000	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC PCI address lower register 0	PCIEDMPALR0	R/W	—	H'02_1100	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC PCI address upper register 0	PCIEDMPAHR0	R/W	—	H'02_1104	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC SuperHyway address lower register 0	PCIEDMSALR0	R/W	—	H'02_1108	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC byte count register 0	PCIEDMBCNTR0	R/W	—	H'02_1110	H'0000 0000	H'0000 0000	H'0000 0000	32

Register	Abbreviation	SH* R/W	PCI* R/W	Address Offset	PCIEC0 Initial Value	PCIEC1 Initial Value	PCIEC2 Initial Value	Access Size
PCI DMAC stride count register 0	PCIEDMSBCNTR0	R/W	—	H'02_1114	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC stride register 0	PCIEDMSTRR0	R/W	—	H'02_1118	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC command chain address register 0	PCIEDMCCAR0	R/W	—	H'02_1120	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC channel control register 0	PCIEDMCHCR0	R/W	—	H'02_1128	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC channel status register 0	PCIEDMCHSR0	R/W	—	H'02_112C	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC PCI address lower register 1	PCIEDMPALR1	R/W	—	H'02_1140	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC PCI address upper register 1	PCIEDMPAHR1	R/W	—	H'02_1144	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC SuperHyway address lower register 1	PCIEDMSALR1	R/W	—	H'02_1148	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC byte count register 1	PCIEDMBCNTR1	R/W	—	H'02_1150	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC stride count register 1	PCIEDMSBCNTR1	R/W	—	H'02_1154	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC stride register 1	PCIEDMSTRR1	R/W	—	H'02_1158	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC command chain address register 1	PCIEDMCCAR1	R/W	—	H'02_1160	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC channel control register 1	PCIEDMCHCR1	R/W	—	H'02_1168	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC channel status register 1	PCIEDMCHSR1	R/W	—	H'02_116C	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC PCI address lower register 2	PCIEDMPALR2	R/W	—	H'02_1180	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC PCI address upper register 2	PCIEDMPAHR2	R/W	—	H'02_1184	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC SuperHyway address lower register 2	PCIEDMSALR2	R/W	—	H'02_1188	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC byte count register 2	PCIEDMBCNTR2	R/W	—	H'02_1190	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC stride count register 2	PCIEDMSBCNTR2	R/W	—	H'02_1194	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC stride register 2	PCIEDMSTRR2	R/W	—	H'02_1198	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC command chain address register 2	PCIEDMCCAR2	R/W	—	H'02_11A0	H'0000 0000	H'0000 0000	H'0000 0000	32

Register	Abbreviation	SH* R/W	PCI* R/W	Address Offset	PCIEC0 Initial Value	PCIEC1 Initial Value	PCIEC2 Initial Value	Access Size
PCI DMAC channel control register 2	PCIEDMCHCR2	R/W	—	H'02_11A8	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC PCI address lower register 3	PCIEDMPALR3	R/W	—	H'02_11C0	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC PCI address upper register 3	PCIEDMPAHR3	R/W	—	H'02_11C4	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC SuperHyway address lower register 3	PCIEDMSALR3	R/W	—	H'02_11C8	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC byte count register 3	PCIEDMBCNTR3	R/W	—	H'02_11D0	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC stride count register 3	PCIEDMSBCNTR3	R/W	—	H'02_11D4	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC stride register 3	PCIEDMSTRR3	R/W	—	H'02_11D8	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC command chain address register 3	PCIEDMCCAR3	R/W	R/W	H'02_11E0	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC channel control register 3	PCIEDMCHCR3	R/W	R/W	H'02_11E8	H'0000 0000	H'0000 0000	H'0000 0000	32
PCI DMAC channel status register 3	PCIEDMCHSR3	R/W	R/W	H'02_11EC	H'0000 0000	H'0000 0000	H'0000 0000	32
Configuration registers								
PCI configuration register 0	PCIEPCICONF0	R	R	H'04_0000	H'0010_1912	H'0010_1912	H'0010_1912	8/16/32
PCI configuration register 1	PCIEPCICONF1	R/W	R/W	H'04_0004	H'0010 0000	H'0010 0000	H'0010 0000	8/16/32
PCI configuration register 2	PCIEPCICONF2	R/W	R/W	H'04_0008	H'FF00 0000	H'FF00 0000	H'FF00 0000	8/16/32
PCI configuration register 3	PCIEPCICONF3	R/W	R/W	H'04_000C	H'0001 0000	H'0001 0000	H'0001 0000	8/16/32
PCI configuration register 4	PCIEPCICONF4	R/W	R/W	H'04_0010	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCI configuration register 5	PCIEPCICONF5	R/W	R/W	H'04_0014	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCI configuration register 6	PCIEPCICONF6	R/W	R/W	H'04_0018	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCI configuration register 7	PCIEPCICONF7	R/W	R/W	H'04_001C	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCI configuration register 8	PCIEPCICONF8	R/W	R/W	H'04_0020	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCI configuration register 9	PCIEPCICONF9	R/W	R/W	H'04_0024	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCI configuration register 10	PCIEPCICONF10	R/W	R/W	H'04_0028	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCI configuration register 11	PCIEPCICONF11	R/W	R/W	H'04_002C	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCI configuration register 12	PCIEPCICONF12	R/W	R/W	H'04_0030	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCI configuration register 13	PCIEPCICONF13	R/W	R/W	H'04_0034	H'0000 0040	H'0000 0040	H'0000 0040	8/16/32
PCI configuration register 14	PCIEPCICONF14	R/W	R/W	H'04_0038	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32

Register	Abbreviation	SH* R/W	PCI* R/W	Address Offset	PCIEC0 Initial Value	PCIEC1 Initial Value	PCIEC2 Initial Value	Access Size
PCI configuration register 15	PCIEPCICONF15	R/W	R/W	H'04_003C	H'0000 00FF	H'0000 00FF	H'0000 00FF	8/16/32
PCI power management capability register 0	PCIEPMCAP0	R/W	R	H'04_0040	H'0003 5001	H'0003 5001	H'0003 5001	8/16/32
PCI power management capability register 1	PCIEPMCAP1	R/W	R/W	H'04_0044	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
MSI capability register 0	PCIEMSICAP0	R/W	R/W	H'04_0050	H'0180 7005	H'0180 7005	H'0180 7005	8/16/32
MSI capability register 1	PCIEMSICAP1	R/W	R/W	H'04_0054	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
MSI capability register 2	PCIEMSICAP2	R/W	R/W	H'04_0058	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
MSI capability register 3	PCIEMSICAP3	R/W	R/W	H'04_005C	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
MSI capability register 4	PCIEMSICAP4	R/W	R/W	H'04_0060	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
MSI capability register 5	PCIEMSICAP5	R/W	R/W	H'04_0064	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCIE capability register 0	PCIEEXPCAP0	R/W	R/W	H'04_0070	H'0041 0010	H'0041 0010	H'0041 0010	8/16/32
PCIE capability register 1	PCIEEXPCAP1	R/W	R	H'04_0074	H'0000_8000	H'0000_8000	H'0000_8000	8/16/32
PCIE capability register 2	PCIEEXPCAP2	R/W	R/W	H'04_0078	H'0000 0810	H'0000 0810	H'0000 0810	8/16/32
PCIE capability register 3	PCIEEXPCAP3	R/W	R	H'04_007C	H'0002_0441	H'0002_0411	H'0002_0411	8/16/32
PCIE capability register 4	PCIEEXPCAP4	R/W	R/W	H'04_0080	H'0011 0000	H'0011 0000	H'0011 0000	8/16/32
PCIE capability register 5	PCIEEXPCAP5	R/W	R/W	H'04_0084	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCIE capability register 6	PCIEEXPCAP6	R/W	R/W	H'04_0088	H'0000 03C0	H'0000 03C0	H'0000 03C0	8/16/32
PCIE capability register 7	PCIEEXPCAP7	R/W	R/W	H'04_008C	H'0001 0000	H'0001 0000	H'0001 0000	8/16/32
PCIE capability register 8	PCIEEXPCAP8	R/W	R/W	H'04_0090	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
VC capability register 0	PCIEVCCAP0	R/W	R	H'04_0100	H'0001 0002	H'0001 0002	H'0001 0002	8/16/32
VC capability register 1	PCIEVCCAP1	R	R	H'04_0104	H'0000 0001	H'0000 0001	H'0000 0001	8/16/32
VC capability register 2	PCIEVCCAP2	R	R	H'04_0108	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
VC capability register 3	PCIEVCCAP3	R	R/W	H'04_010C	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
VC capability register 4	PCIEVCCAP4	R/W	R/W	H'04_0110	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
VC capability register 5	PCIEVCCAP5	R/W	R/W	H'04_0114	H'8000 00FF	H'8000 00FF	H'8000 00FF	8/16/32
VC capability register 6	PCIEVCCAP6	R/W	R	H'04_0118	H'0002 0000	H'0002 0000	H'0002 0000	8/16/32
VC capability register 7	PCIEVCCAP7	R/W	R/W	H'04_011C	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
VC capability register 8	PCIEVCCAP8	R/W	R/W	H'04_0120	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
VC capability register 9	PCIEVCCAP9	R/W	R	H'04_0124	H'0002 0000	H'0002 0000	H'0002 0000	8/16/32

Register	Abbreviation	SH* R/W	PCI* R/W	Address Offset	PCIEC0 Initial Value	PCIEC1 Initial Value	PCIEC2 Initial Value	Access Size
Device serial number capability register 0	PCIENUMCAP0	R	R	H'04_01B0	H'0001 0003	H'0001 0003	H'0001 0003	8/16/32
Device serial number capability register 1	PCIENUMCAP1	R	R	H'04_01B4	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
Device serial number capability register 2	PCIENUMCAP2	R	R	H'04_01B8	H'0000 0000	H'0000 0000	H'0000 0000	8/16/32
PCI Express control system registers								
ID setting register 1	PCIEIDSETR1	R/W	—	H'04_1004	H'FF00 0000	H'FF00 0000	H'FF00 0000	16/32
ID setting register 2	PCIEIDSETR2	R/W	—	H'04_1024	H'0000 0000	H'0000 0000	H'0000 0000	16/32
Device serial number setting register 0	PCIEDSERSETR0	R/W	—	H'04_102C	H'0000 0000	H'0000 0000	H'0000 0000	16/32
Device serial number setting register 1	PCIEDSERSETR1	R/W	—	H'04_1030	H'0000 0000	H'0000 0000	H'0000 0000	16/32
TL status register	PCIETLSR	R/W1C	—	H'04_1044	H'0000 0000	H'0000 0000	H'0000 0000	16/32
TL control register	PCIETLCTLR	R/W	—	H'04_1048	H'0000 3200	H'0000 3200	H'0000 3200	16/32
DL status register	PCIEDLSR	R/W1C	—	H'04_104C	H'4003 0000	H'4003 0000	H'4003 0000	16/32
DL control register	PCIEDLCTLR	R	—	H'04_1050	H'0000 0000	H'0000 0000	H'0000 0000	16/32
MAC status register	PCIEMACSR	R/W1C	—	H'04_1054	H'0041 0000	H'0041 0000	H'0041 0000	16/32
MAC control register	PCIEMACCTLR	R/W	—	H'04_1058	H'80FF 0000	H'80FF 0000	H'80FF 0000	16/32
PM status register	PCIEPMSR	R/W1C	—	H'04_105C	H'0000 0000	H'0000 0000	H'0000 0000	16/32
PM control register	PCIEPMCTLR	R/W	—	H'04_1060	H'0000 0000	H'0000 0000	H'0000 0000	16/32
TL interrupt mask register	PCIETLINTENR	R/W	—	H'04_1064	H'0000 0000	H'0000 0000	H'0000 0000	16/32
DL interrupt mask register	PCIEDLINTENR	R/W	—	H'04_1068	H'0000 0000	H'0000 0000	H'0000 0000	16/32
MAC interrupt mask register	PCIEMACINTENR	R/W	—	H'04_106C	H'0000 0000	H'0000 0000	H'0000 0000	16/32
PM interrupt mask register	PCIEPMINTENR	R/W	—	H'04_1070	H'0000 0000	H'0000 0000	H'0000 0000	16/32
PCI Express transmission system register								
Transmit status register	PCIETXSR	R	—	H'04_4028	H'8000 0000 0000 0000	H'8000 0000 0000 0000	H'8000 0000 00000000	32/64
Transmit VCO status register	PCIETXVC0SR	R/W	—	H'04_4108	H'0088 8000 0000 0000	H'0088 8000 0000 0000	H'0088 8000 0000 0000	32/64

Note: * SH: SuperHyway bus (internal bus)

PCI: PCI local bus

A dash (–) in the R/W column indicates access prohibited (access not possible), while 1C indicates that the bit is cleared by writing 1 to it.

Addresses other than the above are reserved areas. Accessing the reserved areas is prohibited.

Table 13.4 Physical Layer Control Register Configuration

Register	Abbreviation	R/W	Address Offset *	PCIEC0 Initial Value	PCIEC1 Initial Value	PCIEC2 Initial Value	Access Size
Physical Layer Control Register							
Physical layer control registers	PCIEPLCTLR	R/W	H'67	H'0000 0480	H'0000 0480	—	32

Note: * Physical Layer Control Register is not mapped on the SuperHyway address space. For detail, refer to section 13.5.14 Access to Physical Layer Control Registers.

Table 13.5 Register States in Each Operating Mode

Register	Abbreviation	Power-On Reset			Manual Reset	Sleep	Light Sleep
		PCIEC0	PCIEC1	PCIEC2			
PCIEC control registers							
Enable register	PCIEENBLR	H'0000 0001	H'0000 0001	H'0000 0001	Retained	Retained	Retained
Enable control register	PCIEECR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PIO address register	PCIEPAR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PIO control register	PCIEPCTLR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PIO data register	PCIEPDR	—	—	—	Retained	Retained	Retained
Message address lower register	PCIEMSGALR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Message address upper register	PCIEMSGAHR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Message control register	PCIEMSGCTLR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Message data register	PCIEMSGD	—	—	—			
Unlock control register	PCIEUNLOCKCR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI-ID register	PCIEIDR	H'0101 1101	H'0101 1101	H'0101 1101	Retained	Retained	Retained
Debug control register	PCIEDBGCTLR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
INTx register	PCIEINTXR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Receive message register	PCIERMSGR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Receive message interrupt enable register	PCIERMSGIER	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Reset control register 0	PCIERSTR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Reset control register 1	PCIERSTR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Reset control register 2	PCIERSTR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Reset control register 4	PCIERSTR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained

Register	Abbreviation	Power-On Reset			Manual Reset	Sleep	Light Sleep
		PCIEC0	PCIEC1	PCIEC2			
Software reset control register	PCIESRSTR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Physical layer control registers							
Physical layer control register	PCIEPHYCTLR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Physical layer address register	PCIEPHYADRR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Physical layer data in register	PCIEPHYDINR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Physical layer data out register	PCIEPHYDOUTR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Physical layer status register	PCIEPHYSR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCIEC transfer control registers							
Transfer control register	PCIETCTLR	H'0000 0008	H'0000 0008	H'0000 0008	Retained	Retained	Retained
Transfer status register	PCIETSTR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Interrupt register	PCIEINTR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Interrupt enable register	PCIEINTER	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Error header 0 register	PCIEEH0R	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Error header 1 register	PCIEEH1R	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Error header 2 register	PCIEEH2R	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Error header 3 register	PCIEEH3R	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Error source register	PCIEERRFR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Error interrupt register	PCIEERRFER	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Error source register 2	PCIEERRFR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MSI register	PCIEMSIR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MSIF register	PCIEMSIFR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Power-down control register	PCIEPWRCTLR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Packet control connection register	PCIEPCCTLR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address register 0	PCIELAR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address mask register 0	PCIELAMR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address register 1	PCIELAR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained

Register	Abbreviation	Power-On Reset			Manual Reset	Sleep	Light Sleep
		PCIEC0	PCIEC1	PCIEC2			
Local (SuperHyway) address mask register 1	PCIELAMR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address register 2	PCIELAR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address mask register 2	PCIELAMR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address register 3	PCIELAR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address mask register 3	PCIELAMR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address register 4	PCIELAR4	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address mask register 4	PCIELAMR4	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address register 5	PCIELAR5	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Local (SuperHyway) address mask register 5	PCIELAMR5	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address lower register 0	PCIEPALR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address upper register 0	PCIEPAHR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address mask register 0	PCIEPAMR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI conversion control register 0	PCIEPTCTLR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address lower register 1	PCIEPALR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address upper register 1	PCIEPAHR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address mask register 1	PCIEPAMR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI conversion control register 1	PCIEPTCTLR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address lower register 2	PCIEPALR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address upper register 2	PCIEPAHR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address mask register 2	PCIEPAMR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI conversion control register 2	PCIEPTCTLR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address lower register 3	PCIEPALR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address upper register 3	PCIEPAHR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI address mask register 3	PCIEPAMR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained

Register	Abbreviation	Power-On Reset			Manual Reset	Sleep	Light Sleep
		PCIEC0	PCIEC1	PCIEC2			
PCI conversion control register 3	PCIEPTCTLR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCIEC-DMAC control registers							
PCI DMAC DMA operation register	PCIEDMAOR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC PCI address lower register 0	PCIEDMPALR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC PCI address upper register 0	PCIEDMPAHR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC SuperHyway address lower register 0	PCIEDMSALR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC byte count register 0	PCIEDMBCNTR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC stride count register 0	PCIEDMSBCNTR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC stride register 0	PCIEDMSTRR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC command chain address register 0	PCIEDMCCAR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC channel control register 0	PCIEDMCHCR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC channel status register 0	PCIEDMCHSR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Reserved							
PCI DMAC PCI address lower register 1	PCIEDMPALR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC PCI address upper register 1	PCIEDMPAHR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC SuperHyway address lower register 1	PCIEDMSALR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC byte count register 1	PCIEDMBCNTR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC stride count register 1	PCIEDMSBCNTR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC stride register 1	PCIEDMSTRR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC command chain address register 1	PCIEDMCCAR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC channel control register 1	PCIEDMCHCR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained

Register	Abbreviation	Power-On Reset			Manual Reset	Sleep	Light Sleep
		PCIEC0	PCIEC1	PCIEC2			
PCI DMAC channel status register 1	PCIEDMCHSR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC PCI address lower register 2	PCIEDMPALR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC PCI address upper register 2	PCIEDMPAHR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC SuperHyway address lower register 2	PCIEDMSALR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC byte count register 2	PCIEDMBCNTR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC stride count register 2	PCIEDMSBCNTR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC stride register 2	PCIEDMSTRR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC command chain address register 2	PCIEDMCCAR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC channel control register 2	PCIEDMCHCR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC channel status register 2	PCIEDMCHSR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC PCI address lower register 3	PCIEDMPALR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC PCI address upper register 3	PCIEDMPAHR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC SuperHyway address lower register 3	PCIEDMSALR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC byte count register 3	PCIEDMBCNTR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC stride count register 3	PCIEDMSBCNTR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC stride register 3	PCIEDMSTRR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC command chain address register 3	PCIEDMCCAR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC channel control register 3	PCIEDMCHCR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI DMAC channel status register 3	PCIEDMCHSR3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Configuration registers							
PCI configuration register 0	PCIEPCICONF0	H'0010_1912	H'0010_1912	H'0010_1912	Retained	Retained	Retained
PCI configuration register 1	PCIEPCICONF1	H'0010 0000	H'0010 0000	H'0010 0000	Retained	Retained	Retained

Register	Abbreviation	Power-On Reset			Manual Reset	Sleep	Light Sleep
		PCIEC0	PCIEC1	PCIEC2			
PCI configuration register 2	PCIEPCICONF2	H'FF00 0000	H'FF00 0000	H'FF00 0000	Retained	Retained	Retained
PCI configuration register 3	PCIEPCICONF3	H'0001 0000	H'0001 0000	H'0001 0000	Retained	Retained	Retained
PCI configuration register 4	PCIEPCICONF4	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI configuration register 5	PCIEPCICONF5	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI configuration register 6	PCIEPCICONF6	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI configuration register 7	PCIEPCICONF7	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI configuration register 8	PCIEPCICONF8	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI configuration register 9	PCIEPCICONF9	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI configuration register 10	PCIEPCICONF10	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI configuration register 11	PCIEPCICONF11	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI configuration register 12	PCIEPCICONF12	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI configuration register 13	PCIEPCICONF13	H'0000 0040	H'0000 0040	H'0000 0040	Retained	Retained	Retained
PCI configuration register 14	PCIEPCICONF14	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI configuration register 15	PCIEPCICONF15	H'0000 00FF	H'0000 00FF	H'0000 00FF	Retained	Retained	Retained
PCI power management capability register 0	PCIEPMCAP0	H'0003 5001	H'0003 5001	H'0003 5001	Retained	Retained	Retained
PCI power management capability register 1	PCIEPMCAP1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MSI capability register 0	PCIEMSICAP0	H'0180 7005	H'0180 7005	H'0180 7005	Retained	Retained	Retained
MSI capability register 1	PCIEMSICAP1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MSI capability register 2	PCIEMSICAP2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MSI capability register 3	PCIEMSICAP3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MSI capability register 4	PCIEMSICAP4	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MSI capability register 5	PCIEMSICAP5	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCIE capability register 0	PCIEEXPCAP0	H'0041 0010	H'0041 0010	H'0041 0010	Retained	Retained	Retained
PCIE capability register 1	PCIEEXPCAP1	H'0000_8000	H'0000_8000	H'0000_8000	Retained	Retained	Retained
PCIE capability register 2	PCIEEXPCAP2	H'0000 0810	H'0000 0810	H'0000 0810	Retained	Retained	Retained
PCIE capability register 3	PCIEEXPCAP3	H'0002_0411	H'0002_0411	H'0002_0411	Retained	Retained	Retained
PCIE capability register 4	PCIEEXPCAP4	H'0011 0000	H'0011 0000	H'0011 0000	Retained	Retained	Retained
PCIE capability register 5	PCIEEXPCAP5	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCIE capability register 6	PCIEEXPCAP6	H'0000 03C0	H'0000 03C0	H'0000 03C0	Retained	Retained	Retained

Register	Abbreviation	Power-On Reset PCIEC0	Power-On Reset PCIEC1	Power-On Reset PCIEC2	Manual Reset	Sleep	Light Sleep
PCIE capability register 7	PCIEEXPCAP7	H'0001 0000	H'0001 0000	H'0001 0000	Retained	Retained	Retained
PCIE capability register 8	PCIEEXPCAP8	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
VC capability register 0	PCIEVCCAP0	H'0001 0002	H'0001 0002	H'0001 0002	Retained	Retained	Retained
VC capability register 1	PCIEVCCAP1	H'0000 0001	H'0000 0001	H'0000 0001	Retained	Retained	Retained
VC capability register 2	PCIEVCCAP2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
VC capability register 3	PCIEVCCAP3	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
VC capability register 4	PCIEVCCAP4	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
VC capability register 5	PCIEVCCAP5	H'8000 00FF	H'8000 00FF	H'8000 00FF	Retained	Retained	Retained
VC capability register 6	PCIEVCCAP6	H'0002 0000	H'0002 0000	H'0002 0000	Retained	Retained	Retained
VC capability register 7	PCIEVCCAP7	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
VC capability register 8	PCIEVCCAP8	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
VC capability register 9	PCIEVCCAP9	H'0002 0000	H'0002 0000	H'0002 0000	Retained	Retained	Retained
Device serial number capability register 0	PCIENUMCAP0	H'0001 0003	H'0001 0003	H'0001 0003	Retained	Retained	Retained
Device serial number capability register 1	PCIENUMCAP1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Device serial number capability register 2	PCIENUMCAP2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI Express control system registers							
ID setting register 1	PCIEIDSETR1	H'FF00 0000	H'FF00 0000	H'FF00 0000	Retained	Retained	Retained
ID setting register 2	PCIEIDSETR2	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Device serial number setting register 0	PCIEDSERSETR0	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Device serial number setting register 1	PCIEDSERSETR1	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
TL status register	PCIETLSR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
TL control register	PCIETLCTLR	H'0000 3200	H'0000 3200	H'0000 3200	Retained	Retained	Retained
DL status register	PCIEDLSR	H'4003 0000	H'4003 0000	H'4003 0000	Retained	Retained	Retained
DL control register	PCIEDLCTLR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MAC status register	PCIEMACSR	H'0041 0000	H'0041 0000	H'0041 0000	Retained	Retained	Retained
MAC control register	PCIEMACCTLR	H'80FF 0000	H'80FF 0000	H'80FF 0000	Retained	Retained	Retained
PM status register	PCIEPMSR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained

Register	Abbreviation	Power-On Reset PCIEC0	Power-On Reset PCIEC1	Power-On Reset PCIEC2	Manual Reset	Sleep	Light Sleep
PM control register	PCIEPMCTLR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
TL interrupt mask register	PCIETLINTENR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
DL interrupt mask register	PCIEDLINTENR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MAC interrupt mask register	PCIEMACINTENR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PM interrupt mask register	PCIEPMINTENR	H'0000 0000	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PCI Express transmission system registers							
Transmit status register	PCIETXSR	H'8000 0000 0000 0000	H'8000 0000 0000 0000	H'8000 0000 0000 0000	Retained	Retained	Retained
Transmit VCO status register	PCIETXVC0SR	H'0088 8000 0000 0000	H'0088 8000 0000 0000	H'0088 8000 0000 0000	Retained	Retained	Retained

Table 13.6 Phy Register States in Each Operating Mode

Register	Abbreviation	Power-On Reset PCIE0	Power-On Reset PCIE1	Power-On Reset PCIE2	Manual Reset	Sleep	Light Sleep
Physical-layer Control register	PCIEPLCTLR	H'0000 0480	H'0000 0480	—	Retained	Retained	Retained

13.4.1 PCIEC Control Registers

PCIEC Control Registers control PCIEC module.

These registers are accessible via the SuperHyway bus. When a register is accessed from the PCI Express by target transfer, the PCIEC handles the access as a completer abort and executes error processing.

(1) Enable Register (PCIEENBLR)

PCIEENBLR specifies this module bridge operation and stop. To perform packet generation and packet reception using the bridge, set the corresponding bit in this register to 1 (initial value).

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ENBL	1	—	R/W	Enables the PCIEC module. To use the PCIEC module, set this bit to 1.

(2) Enable Control Register (PCIEECR)

PCIEECR clears the PCIEENBLR register to stop the bridge operation when a specific event occurs. PCIEECR is used to check the cause of the event after the event has occurred without executing the next packet processing.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	STPFERR	STPNFERR	STPCERR	—	—	—	—	—	—	—	STPSERR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 25	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
24	STPDMACE	0	—	R/W	Stops the module function due to a DMAC error. If a DMAC error occurs while this bit is set to 1, the PCIEENBLR.ENBL bit to 0 to stop packet transmission or reception.
23 to 11	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
10	STPFERR	0	—	R/W	Stops the module function due to a fatal error in the PCI. If a fatal error occurs in the PCI while this bit is set to 1, the PCIEENBLR.ENBL bit is cleared to 0 to stop packet transmission or reception.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
9	STPNFERR	0	—	R/W	Stops the module function due to a non-fatal error in the PCI. If a non-fatal error occurs in the PCI while this bit is set to 1, the PCIEENBLR.ENBL bit is cleared to 0 to stop packet transmission or reception.
8	STPCERR	0	—	R/W	Stops the module function due to a correctable error in the PCI. If a correctable error occurs in the PCI while this bit is set to 1, the PCIEENBLR.ENBL bit is cleared to 0 to stop packet transmission or reception.
7 to 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	STPSERR	0	—	R/W	Stops the module function due to a system error. If a system error (SERR) occurs while this bit is set to 1, the PCIEENBLR.ENBL bit is cleared to 0 to stop packet transmission or reception.

(3) PIO Address Register (PCIEPAR)

PCIEPAR is used to issue a configuration request from the PCIEC module. For details on the configuration request, refer to section 13.5.4 (1), Generating Configuration Access.

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	BN	All 0	—	R/W	These bits specify the bus number of access destination when a configuration cycle is generated.
23 to 19	DN	All 0	—	R/W	These bits specify the device number when a configuration cycle is generated.
18 to 16	FN	All 0	—	R/W	These bits specify the function number of access destination when a configuration cycle is generated.
15 to 12	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	EREGNO	All 0	—	R/W	These bits specify the expansion register number of access destination when a configuration cycle is generated.
7 to 2	REGNO	All 0	—	R/W	These bits specify the register number of access destination when a configuration cycle is generated.
1, 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(4) PIO Control Register (PCIEPCTLR)

PCIEPDR is used to issue a configuration request from the PCIEC module. For details on the configuration request, refer to section 13.5.4 (1), Generating Configuration Access.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCIE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CRS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W1C
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	CCIE	0	—	R/W	Enables the generation of a configuration request. To generate a configuration cycle by the PCI Express PIO address registers or PCI Express PIO data registers, set this bit to 1.
30 to 17	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
16	CRS	0	—	R/W1C	Indicates that a configuration retry status (CRS) is received. This bit is set to 1 when the CRS is received.
15 to 9	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TYPE	0	—	R/W	Specifies a type of configuration request which is issued during g PCIERDR register write access. 0: Generates a Type 0 configuration request 1: Generates a Type 1 configuration request
7 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(5) PIO Data Register (PCIEPDR)

PCIEPDR is used to issue a configuration request from the PCIEC module. For details on the configuration request, refer to section 13.5.4 (1), Generating Configuration Access.

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	PDR	(Undefined)	—	R/W	These bits issue a configuration request. Write accessing this field issues a configuration write request where the write contents in this field is handled as data. Read accessing this field issues a configuration read request to read data included in the response.

(6) Message Address Lower Register (PCIEMSGALR)

PCIEMSGALR is used to issue a message from the PCIEC module. For details on message issuance, refer to section 13.5.8, Message Transmission and Reception.

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	MSGADRL	All 0	—	R/W	These bits specify an address (lower 32 bits) when a message is issued.

(7) Message Address Upper Register (PCIEMSGAHR)

PCIEMSGAHR is used to issue a message from the PCIEC module. For details on message issuance, refer to section 13.5.8, Message Transmission and Reception.

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	MSGADRH	All 0	—	R/W	These bits specify an address (upper 32 bits) when a message is issued.

(8) Message Control Register (PCIEMSGCTLR)

PCIEMSGCTLR is used to issue a message from the PCIEC module. For details on message issuance, refer to section 13.5.8, Message Transmission and Reception.

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	MSGIE	0	—	R/W	Enables or disables issuance of a message request. 1: Enables issuance of a message request. 0: Disables issuance of a message request.
30 to 21	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
20	MTYPE	0	—	R/W	Specifies a message type when a message request is issued. 0: Msg 1: MsgD
19	—	0	—	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	MROUTE	All 0	—	R/W	These bits specify a routing method when a message request is issued. 000: Performs routing to root complex. 001: Performs routing by address. 010: Performs routing by ID. 011: Broadcasts from root complex. 100: Local (ends routing by receiver) 101: Collects and performs routing to root complex. Other than the above: Setting prohibited
15 to 8	MCODE	All 0	—	R/W	These bits specify message code when a message request is issued.
7 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(9) Message Data Register (PCIEMSGD)

PCIEMSGD is used to issue a message from the PCIEC module. For details on message issuance, refer to section 13.5.8, Message Transmission and Reception.

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	MDATA	All 0	—	R/W	These bits issue a message. Write accessing this field issues a message. Data to be added to the message should be written when the message is transmitted. These bits are read as undefined.

(10) Unlock Control Register (PCIEUNLOCKCR)

PCIEUNLOCKCR is used to issue an Unlock message from the PCIEC module to cancel the lock. For details of unlock transfer, refer to 13.5.5 (6) Lock Requests.

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ASTUNLOCK	0	—	R/W	Issues an Unlock message. Writing 1 to this bit issues an Unlock message (Message code = B'0000_0000, Routing = B'011). This bit is always read as 0. Writing to this bit to issue an Unlock message is allowed only in the Root Port. Sending an Unlock message from an Endpoint is not allowed in the PCI-Express standard. Do not write 1 to this bit until connection with the destination is established.

(11) PCI-ID Register (PCIEIDR)

PCIEIDR indicates the ID number of the subblocks composing the PCIEC module.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BRIDGE-ID								CTRL-ID							
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	BRIDGE-ID	H'01	—	R	These bits indicate the ID of the PCIEC bridge.
23 to 16	CTRL-ID	H'01	—	R	These bits indicate the ID of the PCIEC controller.
15 to 0	PHY-ID	H'1101	—	R	These bits indicate the ID of the PCIEC physical unit.

(12) Debug Control Register (PCIEDBGCTLR)

PCIEDBGCTLR is used to debug the module. Typically, the initial value of this register should not be changed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	USEX41F
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1/0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 17	—	All 0	—	R/W	Reserved These bits are always read as 0. The write value should always be 0.
16	USEX4IF	PCIEC0: 1 PCIEC1/2: 0	—	R/W	Reserved. The initial value should not be changed.
15 to 14	—	All 0	—	R/W	Reserved These bits are always read as 0. The write value should always be 0.
13 to 12	—	All 0	—	R/W	Reserved. The initial value should not be changed.
11 to 9	—	All 0	—	R/W	Reserved These bits are always read as 0. The write value should always be 0.
8	—	0	—	R/W	Reserved. The initial value should not be changed.
7 to 5	—	All 0	—	R/W	Reserved These bits are always read as 0. The write value should always be 0.
4	—	0	—	R/W	Reserved. The initial value should not be changed.
3 to 1	—	All 0	—	R/W	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
0	—	0	—	R/W	Reserved. The initial value should not be changed.

(13) INTx Register (PCIEINTXR)

PCIEINTX is used to check the INTx interrupt generation status in a Root Port and to generate an INTx interrupt in an Endpoint. For details, refer to section 13.5.9, INTx Interrupts.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	INTD	INTC	INTB	INTA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 17	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
16	ASTINTX	0	—	R/W	Generates an INTx interrupt. Writing 1 to this bit asserts an INTx interrupt specified by PCICONF15[15:8].INTPIN. Writing 0 to this bit deasserts an INTx interrupt. This bit is valid only in an Endpoint. In a Root Port, a write to this bit is invalid and this bit is always read as 0. Note that INTx interrupts cannot be used while the MSI is used. Accordingly, this bit setting is invalid while the MSICAP0[16].MSI Enable bit is set to 1. To generate an INTx interrupt by this bit, the PCICONF1[10].Interrupt Disable bit should be cleared to 0. In addition, if the PCICONF1[10].Interrupt Disable bit is set to 1 while this bit is set to 1, this bit is cleared to 0.
15 to 4	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
3	INTD	0	—	R	Indicates that an INTD interrupt on the PCIEC is asserted. This bit is asserted by an Assert INTD message; while this bit is deasserted by a Deassert INTD message. Asserting this bit generates an interrupt to the INTC while PCICONF15[7:0].INTLINE is other than FFh.
2	INTC	0	—	R	Indicates that an INTC interrupt on the PCIEC is asserted. This bit is asserted by an Assert INTC message; while this bit is deasserted by a Deassert INTC message. Asserting this bit generates an interrupt to the INTC while PCICONF15[7:0].INTLINE is other than FFh.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
1	INTB	0	—	R	Indicates that an INTB interrupt on the PCIEC is asserted. This bit is asserted by an Assert INTB message; while this bit is deasserted by a Deassert INTB message. Asserting this bit generates an interrupt to the INTC while PCICONF15[7:0].INTLINE is other than FFh.
0	INTA	0	—	R	Indicates that an INTA interrupt on the PCIEC is asserted. This bit is asserted by an Assert INTB message; while this bit is deasserted by a Deassert INTB message. Asserting this bit generates an interrupt to INTC while PCICONF15[7:0].INTLINE is other than FFh.

(14) Receive Message Register (PCIERMSGR)

PCIERMSGR indicates the message reception status. For details, refer to section 13.5.8, Message Transmission and Reception.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PME_TO_ACK	PME_TOFF	PM_PME	PA_ASN	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R/W1C	R/W1C	R/W1C	R/W1C	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH 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	PME_TO_ACK	0	—	R/W1C	Indicates that a PME_TO_Ack message is received.
10	PME_TOFF	0	—	R/W1C	Indicates that a PME_Turn_Off message is received.
9	PM_PME	0	—	R/W1C	Indicates that a PM_PME message is received.
8	PA_ASN	0	—	R/W1C	Indicates that a PA_Active_State_Nak message is received.
7 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(15) Receive Message Interrupt Enable Register (PCIERMSGIER)

PCIERMSGIER specifies interrupt generation when a message is received. For details, refer to section 13.5.8, Message Transmission and Reception.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PME_TO_ACKE	PME_TOFFE	PM_PMEE	PA_ASNE	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH 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	PME_TO_ACKE	0	—	R/W	Enables interrupt generation when a PME_TO_Ack message is received.
10	PME_TOFFE	0	—	R/W	Enables interrupt generation when a PME_Turn_Off message is received.
9	PM_PMEE	0	—	R/W	Enables interrupt generation when a PM_PME message is received.
8	PA_ASNE	0	—	R/W	Enables interrupt generation when a PA_Active_State_Nak message is received.
7 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(16) Reset Control Register 0 (PCIERSTR0)

PCIERSTR0 is reserved. The initial value of this register should not be changed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	rstpt2	rstpt1	rstpt0	—	—	purgepc	rstpc0	—	—	—	rstmp0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	rstdc0	—	—	—	—	—	—	rstcb1	rstcb0	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	—	All 0	—	R	Reserved. The initial value should not be changed.

(17) Reset Control Register 1 (PCIERSTR1)

PCIERSTR1 is reserved. The initial value of this register should not be changed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	rstsi0	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	rstrr0	—	rsgrg2	rstrg1	rstrg0	—	—	—	rstre0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R	R	R	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	—	All 0	—	R	Reserved. The initial value should not be changed.

(18) Reset Control Register 2 (PCIERSTR2)

PCIERSTR2 is reserved. The initial value of this register should not be changed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	rstdxp	rstd0p	—	—	rstrxr	rstr0r	—	—	rstrxc	rstr0c
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	rsttr0	—	—	—	rstte0	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	—	All 0	—	R	Reserved. The initial value should not be changed.

(19) Reset Control Register 3 (PCIERSTR3)

PCIERSTR3 is reserved. The initial value of this register should not be changed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	rststt0	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	rstdxp	rstd0p	—	—	rstdx0	rstdc0	—	—	rstdxt	rstd0t
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	—	All 0	—	R	Reserved. The initial value should not be changed.

(20) Software Reset Control Register (PCIESRSTR)

PCIESRSTR is used to initialize the PCIEC module under software control. For details of software reset, refer to section 13.5.15, Software Reset.

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 1	—	All 0	—	R	Reserved. The initial value should not be changed.
0	SRST	0	—	R/W	This bit applies a software reset. Write 1 to this bit to initialize the PCIEC module. After that, be sure to write 0 to this bit to complete initialization. Only the bridge can be initialized through this bit. To initialize all blocks in the PCIEC by software, refer to section 13.5.15, Software Reset.

13.4.2 Physical layer control register

Physical Layer control registers are used for access control to the register space of physical layer control registers.

For details of the procedure of using physical layer control registers and this register space, refer to section 13.5.14, Access to Physical Layer Control Registers.

(1) Physical Layer Control Register (PCIEPHYCTLR)

This register controls the PHY control bus.

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	PHYRST	0	—	R/W	Specifies initialization of the PHY control bus. To perform initialization, write 1 to this bit and then clear it to 0. 0: Do not initialize the PHY control bus. (Normal state). 1: Initialize the PHY control bus.
30 to 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
0	PHYCKE	0	—	R/W	Specifies clock supply to the access bus to the PHY control bus. This bit should be set to enable clock supply before accessing the physical layer register space. 0: Clock is not supplied to the PHY control bus. 1: Clock is supplied to the PHY control bus.

(2) Physical Layer Address Register (PCIEPHYADRR)

This register controls the PHY control bus.

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

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

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 25	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
24	PHYACK	0	—	R	Indicates the response from the PHY control bus. 0: No access, or access in progress 1: Access has completed.
23 to 18	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
17, 16	PHYCMD	00	—	R/W	Specifies a PHY control bus command. 00: Idle 01: Write 10: Read 11: Reserved (Do not set)
15 to 12	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	PHYLANE	All 0	—	R/W	Specifies a PHY control bus lane. Bit 0 to 3 of this field corresponds to the lane 0 to 3, respectively. The access is performed to the lane, whose corresponding bit in this field is 1. In the case of write access, more than 0 bits can be set to 1. In the case of read access, only one bit of this field must be set to 1.
7 to 0	PHYADDR	All 0	—	R/W	Specifies a PHY control bus address Indicates the address of physical layer register, to which read or write access is performed.

(3) Physical layer data input register (PCIEPHYDINR)

This register controls the PHY control bus.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PHYDIN															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	PHYDIN	All 0	—	R	Specifies the write data to be sent to the PHY control bus.

(4) Physical layer data output register (PCIEPHYDOUTR)

This register controls the PHY control bus.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PHYDOUT															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PHYDOUT															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	PHYDOUT	All 0	—	R/W	Indicates the read data from the PHY control bus.

(5) Physical Layer Status Register (PCIEPHYSR)

This register indicates the physical layer status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PHYRDY
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PHYRDY	0	—	R	Indicates physical layer is ready or not. 0: Physical layer is not ready. 1: Physical layer is ready.

13.4.3 PCIEC Transfer Control Registers

Transfer Control Registers control bridge operation between the SuperHyway bus and PCI Express. These registers control target transfer and PIO transfer.

These registers are accessible via the SuperHyway bus. Access from the PCI Express side by target transfer is allowed for the MSIF register (PCIEMSIFR), local (SuperHyway) address registers 0 to 5 (PCIELAR0 to PCIELAR5), and local (SuperHyway) address mask registers 0 to 5 (PCIELAMR0 to PCIELAMR5). When other registers are accessed from the PCIEC by target transfer, the PCIEC handles the access as a completer abort and executes error processing.

(1) Transfer Control Register (PCIETCTLR)

PCIETCTLR controls PCIEC transfer. Set each transfer control register, set the CFINIT bit in PCIETCTLR to 1, and then start initialization of the PCIEC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	CPLTYPE	DLDOWN	—	—	CFINIT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
SH-R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH 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, 12	—	All 0	—	R/W	Reserved. Only 0 should be written to these bits.
11 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	PCI R/W	SH R/W	Description
4	CPLTYPE	0	—	R/W	Specifies the type of Completion to be sent in response to the read request received from the PCI Express by target transfer. 0: Completion with long data length 1: Completion with short data length (128-byte boundary) When this bit is set to 0, the PCIEC sends a Completion with the largest possible data length. When the read request data is larger than MPS, the PCIEC sends multiple Completions separated at MPS boundaries. If the read request data is not larger than MPS, the PCIEC does not send multiple Completions. When this bit is set to 1, the PCIEC divides a Completion for a read request that is 128 bytes or larger and smaller than MPS into multiple Completions at 128-byte boundaries and sends them.
3	DL_Down	1	—	R	Indicates whether connection in the data link layer is lost. This bit is set to 1 after cancellation of a reset. This bit is cleared to 0 when a connection with the target device is established, and set to 1 when a connection in the data link layer is lost due to communication failure.
2, 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	CFINIT	0	—	R/W	Indicates end of initialization of the PCIE internal registers. Set an appropriate value to the following PCIEC registers, and then set this field to 1. PCIELAR0 to PCIELAR5 PCIELAMR0 to PCIELAMR5 When this bit is 1, the above registers cannot be modified. Setting this bit to 1 starts establishment of link with the PCI Express connection destination. 0: Initialization is in progress. 1: Initialization ends.

(2) Transfer Status Register (PCIETSTR)

PCIETSTR indicates the PCIEC transfer status. The PCIEC communication is enabled when this register indicates the transmission enable state after initialization by the PCIETCTLR register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DLLACT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DLLACT	0	—	R	Indicates that the data link layer is active.

(3) Interrupt Register (PCIEINTR)

PCIEINTR indicates the interrupt status in the INTC generated from this module. For details, refer to section 13.5.11, Interrupt Request to the INTC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	INT_RX_ERP	INT_RX_CTRL	INT_TX_CTRL	INT_RX_VCX			INT_TX_VCX			INT_RX_VCX_ERR	INT_RX_VC0_ERR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R/W1C	R/W1C	R/W1C	R	R	R	R	R	R	R/W1C	R/W1C
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTTL	INTDL	INTMAC	INTPM	INT_RX_VC0			INT_TX_VC0			INT_PCIMES	INT_PCIPOWER	INT_PCICERR	INT_PCINFERR	INT_PCIFERR	INT_PCISERR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W1C	R/W1C	R/W1C	R/W1C	R	R	R	R	R	R	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 27	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
26	INT_RX_ERP	0	—	R/W1C	Indicates that an error packet receive interrupt has occurred.
25	INT_RX_CTRL	0	—	R/W1C	Indicates that a receive control interrupt has occurred.
24	INT_TX_CTRL	0	—	R/W1C	Indicates that a transmission control interrupt has occurred.
23 to 21	INT_RX_VCX	000	—	R	Indicates that a VCX receive interrupt has occurred. Bit[2]: Posted request receive buffer is full Bit[1]: Non-Posted request receive buffer is full Bit[0]: CPL receive buffer is full

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
20 to 18	INT_TX_VCX	000	—	R	Indicates that a VCX transmit interrupt has occurred Bit[2]: Posted request receive buffer is full Bit[1]: Non-Posted request receive buffer is full Bit[0]: CPL receive buffer is full
17	INT_RX_ VCX_ERR	0	—	R/W1C	Indicates that a VCX packet receive error interrupt has occurred.
16	INT_RX_ VC0_ERR	0	—	R/W1C	Indicates that a VC0 packet receive error interrupt has occurred.
15	INTTL	0	—	R/W1C	Indicates that a TL interrupt has occurred.
14	INTDL	0	—	R/W1C	Indicates that a DL interrupt has occurred.
13	INTMAC	0	—	R/W1C	Indicates that a MAC interrupt has occurred.
12	INTPM	0	—	R/W1C	Indicates that a PM interrupt has occurred.
11 to 9	INT_RX_VC0	000	—	R	Indicates that a VC0 receive interrupt has occurred. Bit[2]: Posted request receive buffer is full Bit[1]: Non-Posted request receive buffer is full Bit[0]: CPL receive buffer is full
8 to 6	INT_TX_VC0	000	—	R	Indicates that a VC0 transmit interrupt has occurred. Bit[2]: Posted request receive buffer is full. Bit[1]: Non-Posted request receive buffer is full Bit[0]: CPL receive buffer is full
5	INT_PCIMES	0	—	R/W1C	Indicates that the PCIEC has received a message. Note that this interrupt does not occur if an interrupt for each message is disabled. When a message is received, the message processing should be performed by software.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
4	INT_ PCIPOWER	0	—	R/W1C	Indicates that a power-down sequence interrupt has occurred.
3	INT_PCICERR	0	—	R/W1C	Indicates that a correctable error interrupt has occurred.
2	INT_PCINFERR	0	—	R/W1C	Indicates that a nonfatal error interrupt has occurred.
1	INT_PCIFERR	0	—	R/W1C	Indicates that a fatal error interrupt has occurred.
0	INT_PCISERR	0	—	R/W1C	Indicates that a system error has occurred. When a system error has occurred, check the related registers and perform the error recovery processing by software. A system error is as follows. • CRS (Configuration Retry Status) has been received. • Connection in the data link layer has been lost. (dl_down)

(4) Interrupt Enable Register (PCIEINTER)

PCIEINTER enables an interrupt request from this module to the INTC. For details, refer to section 13.5.11, Interrupt Request to the INTC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	INT_RX_ERPE	INT_RX_CTRLE	INT_TX_CTRLE	INT_RX_VCXE			INT_TX_VCXE			INT_RX_VCX_ERRE	INT_RX_VC0_ERRE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTTLE	INTDLE	INTMACE	INTPME	INT_RX_VC0E			INT_TX_VC0E			INT_PCIMESE	INT_PCIPOWERE	INT_PCICERRE	INT_PCINFERRE	INT_PCIFERRE	INT_PCISERRE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 27	—	All 0	—	R/W	Reserved These bits are always read as 0. The write value should always be 0.
26	INT_RX_ERPE	0	—	R/W	Enables the generation of an error packet receive interrupt.
25	INT_RX_CTRLE	0	—	R/W	Enables the generation of a receive control interrupt.
24	INT_TX_CTRLE	0	—	R/W	Enables the generation of a transmit control interrupt.
23 to 21	INT_RX_VCXE	000	—	R/W	Enables the generation of a VCX receive interrupt. Bit[2]: Posted receive full interrupt Bit[1]: Non-posted receive full interrupt Bit[0]: CPL receive full interrupt

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
20 to 18	INT_TX_VCXE	000	—	R/W	Enables the generation of a VCX transmit interrupt. Bit[2]: Posted receive full interrupt Bit[1]: Non-posted receive full interrupt Bit[0]: CPL receive full interrupt
17	INT_RX_VCX_ERRE	0	—	R/W	Enables the generation of a VCX packet receive error interrupt.
16	INT_RX_VC0_ERRE	0	—	R/W	Enables the generation of a VC0 packet receive error interrupt.
15	INTTLE	0	—	R/W	Enables the generation of a TL interrupt.
14	INTDLE	0	—	R/W	Enables the generation of a DL interrupt.
13	INTMACE	0	—	R/W	Enables the generation of a MAC interrupt.
12	INTPME	0	—	R/W	Enables the generation of a PM interrupt.
11 to 9	INT_RX_VC0E	000	—	R/W	Enables the generation of a VC0 receive interrupt. Bit[2]: Posted receive full interrupt Bit[1]: Non-posted receive full interrupt Bit[0]: CPL receive full interrupt
8 to 6	INT_TX_VC0EE	000	—	R/W	Enables the generation of a VC0 transmit interrupt. Bit[2]: Posted receive full interrupt Bit[1]: Non-posted receive full interrupt Bit[0]: CPL receive full interrupt
5	INT_PCIMESE	0	—	R/W	The PCIEC enables the generation of a message receive interrupt. Note that this interrupt does not occur if an interrupt for each message is disabled. When a message is received, the message processing should be performed by software.
4	INT_PCIPOWERE	0	—	R/W	Enables the generation of a power-down sequence interrupt.
3	INT_PCICERRE	0	—	R/W	Enables the generation of a correctable error interrupt.
2	INT_PCINFERRE	0	—	R/W	Enables the generation of a nonfatal error interrupt.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
1	INT_PCIFERRE	0	—	R/W	Enables the generation of a fatal error interrupt.
0	INT_PCISERRE	0	—	R/W	Enables the generation of a system error interrupt.

(5) Error Header 0 Register (PCIEEH0R)

PCIEEH0R indicates a header of the packet where an error has occurred when a PCIEC error has generated. For details, refer to section 13.5.13, Error Processing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EH0															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EH0															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	EH0	All 0	—	R	These bits store the header (DW0) of a packet when an error has occurred in the received PCI packet.

(6) Error Header 1 Register (PCIEEH1R)

PCIEEH0R indicates a header of the packet where an error has occurred when a PCIEC error has generated. For details, refer to section 13.5.13, Error Processing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EH1															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EH1															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	EH1	All 0	—	R	These bits store the header (DW1) of a packet when an error has occurred in the received PCI packet.

(7) Error Header 2 Register (PCIEEH2R)

PCIEEH0R indicates a header of the packet where an error has occurred when a PCIEC error has generated. For details, refer to section 13.5.13, Error Processing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EH2															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EH2															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	EH2	All 0	—	R	These bits store the header (DW2) of a packet when an error has occurred in the received PCI packet.

(8) Error Header 3 Register (PCIEEH3R)

PCIEEH0R indicates a header of the packet where an error has occurred when a PCIEC error has generated. For details, refer to section 13.5.13, Error Processing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EH3															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EH3															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	EH3	All 0	—	R	These bits store the header (DW3) of a packet when an error has occurred in the received PCI packet.

(9) Error Source Register (PCIEERRFR)

PCIEERRFR indicates the source of the error generated in the PCIEC. For details, refer to section 13.5.13, Error Processing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Completer Abort Detected
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W1C
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	UNEXPECTED COMPLETION	—	—	—	CPL TIMEOUT	—	—	RECEIVE CACPL	RECEIVE URCPL	—	—	SEND CACPL	SEND URCPL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R/W1C	R	R	R	R/W1C	R	R	R/W1C	R/W1C	R	R	R/W1C	R/W1C
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH 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	Completer Abort Detected	0	—	R/W1C	Indicates that a completer abort has been detected.
15 to 13	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
12	UNEXPECTED COMPLETION	0	—	R/W1C	Indicates that an unexpected Completion has been received.
11 to 9	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
8	CPLTIMEOUT	0	—	R/W1C	Indicates that a Completion timeout has occurred.
7, 6	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
5	RECEIVE CACPL	0	—	R/W1C	Indicates that a Completion with CA (completer abort) status has been received.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
4	RECEIVE URCPL	0	—	R/W1C	Indicates that a Completion with UR (unsupported request) status has been received.
3, 2	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
1	SENDCACPL	0	—	R/W1C	Indicates that a Completion with CA (completer abort) status has been transmitted.
0	SENDURCPL	0	—	R/W1C	Indicates that a Completion with UR (unsupported request) status has been transmitted.

(10) Error Interrupt Register (PCIEERRFER)

PCIEERRFER specifies for each error source whether to generate an INT_PCISERR interrupt when the corresponding error occurs in the PCIEC. For details, refer to section 13.5.13, Error Processing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Completer Abort DetectedE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	UNEXPECTED COMPLETIONE	—	—	—	CPL TIMEOUTE	—	—	RECEIVE CACPLE	RECEIVE URCPLE	—	—	SEND CACPLE	SEND URCPLE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W	R	R	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH 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	Completer Abort DetectedE	0	—	R/W	Enables the generation of an INT_PCISERR interrupt when a completer abort is detected.
15 to 13	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
12	UNEXPECTED COMPLETIONE	0	—	R/W	Enables the generation of an INT_PCISERR interrupt when an unexpected Completion is received.
11 to 9	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
8	CPLTIMEOUTE	0	—	R/W	Enables the generation of an INT_PCISERR interrupt when a Completion timeout occurs.
7, 6	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
5	RECEIVE CACPLE	0	—	R/W	Enables the generation of an INT_PCISERR interrupt when a Completion with CA (completer abort) status is received.
4	RECEIVE URCPLE	0	—	R/W	Enables the generation of an INT_PCISERR interrupt when a Completion with UR (unsupported request) status is received.
3, 2	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
1	SENDCACPLE	0	—	R/W	Enables the generation of an INT_PCISERR interrupt when a Completion with CA (completer abort) status is transmitted.
0	SENDURCPLE	0	—	R/W	Enables the generation of an INT_PCISERR interrupt when a Completion with UR (unsupported request) status is transmitted.

(11) Error Source Register 2 (PCIEERRFR2)

PCIEERRFR2 indicates the source of the error generated in the PCIEC data link layer. For details, refer to section 13.5.13, Error Processing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	Data Link Layer Protocol Error	Replay Timeout	Replay Number Rollover	BAD TLP	BAD DLLP	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Receiver Error	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W1C	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31, 30	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
29	Data LinkLayer Protocol Error	0	—	R/W1C	Indicates that a Data Link Layer Protocol Error has occurred.
28	ReplayTimeout	0	—	R/W1C	Indicates that a Replay Timeout has occurred.
27	Replay Number Rollover	0	—	R/W1C	Indicates that a Replay Number Rollover has occurred.
26	BADTLP	0	—	R/W1C	Indicates that a BAD TLP has been detected.
25	BADDLLP	0	—	R/W1C	Indicates that a BAD DLLP has been detected.
24 to 16	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
15	Receiver Error	0	—	R/W1C	Indicates that a Receiver Error has been detected. The following errors are handled as Receiver Errors and indicated in this bit. • 8b/10b error • Disparity error • Elastic buffer overflow • Elastic buffer underflow
14 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(12) MSI Register (PCIEMSIR)

PCIEMSIR is used to generate an MSI interrupt. Note that the MSI interrupts can be used only when the MSI interrupts are enabled in Endpoints by the Root Port. For details, refer to section 13.5.10, MSI Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	MSIAST				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 5	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	MSIAST	00000	—	R/W	These bits generate MSI interrupts. This field is valid only at Endpoints when the MSI is valid. This field is invalid when the MSICAP0[16].MSI Enable bit is cleared to 0. Writing data to this field sets the corresponding MESsage Pending bit in MSICAP5. Then, MSI interrupts whose corresponding MESsage Mask bit in MSICAP4 has been cleared to 0 are generated from the smaller number to the larger. The maximum value of interrupt source number to be written in this field is the number of interrupt sources specified by MSICAP0[22:20].Multiple MESage Enable – 1. This field is always read as 0. Do not write to these bits when a connection with the destination is not established.

(13) MSIF Register (PCIEMSIFR)

PCIEMSIFR indicates the MSI interrupt generation status. Note that the MSI interrupts can be used only when the MSI interrupts are enabled in Endpoints by the Root Port. For details, refer to section 13.5.10, MSI Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSIF															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSIF															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	MSIF	All 0	R/W*	R/W1C	These bits indicate that MSI interrupts have occurred. This field is valid only when the device is used as a Root Port and the MSI is valid. Writing data in this field from the PCI is recognized as an MSI interrupt and then a bit obtained by adding Bit[5:0] and Bit[12:8] of write data is set to 1. Reading this field from the PCI is handled as normal read processing. Writing this field from the SuperHyway is recognized as an interrupt cancellation and then clears the bit that is set to 1 to 0. Reading this field from the SuperHyway is handled as normal read processing. Note: * Accessing this field from the PCI is handled by the MSI specific method as described above.

(14) Power-Down Control Register (PCIEPWRCTLR)

PCIEPWRCTLR controls the power-down sequence. For details, refer to section 13.5.12, Power Management.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCPL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RCPL	0	—	R/W	Cancels the pending completion transmission to perform power-down sequence. Writing 1 to this bit transmits the pending completion to perform power-down sequence. This bit is always read as 0. For details, refer to descriptions related to power-down sequence.

(15) Packet Connection Control Register (PCIEPCCTLR)

PCIEPCCTLR controls packet connection in PIO transfer. For details, refer to section 13.5.5 (8) Packet Connection.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STPCNCT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	STPCNCT	0	—	R/W	Stops packet connection in PIO transfer. Write 1 to this bit to stop packet connection. When this bit is written to, the data that was previously written is sent as a PCIE packet. This bit is always read as 0.

(16) Local (SuperHyway) Address Registers 0 to 5 (PCIELAR0 to PCIELAR5)

PCIELAR0 to PCIELAR5 specify local bus (SuperHyway) addresses during target transfer (transfer from PCIEC to SuperHyway). For details, refer to section 13.5.6, Target Transfers (Data Transfer from External Device to PCIEC).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAR								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 8	LARn	All 0	—	R/W	These bits indicate a local (SuperHyway) address during transaction transfer from PCI to the SuperHyway. When data is transferred from a space obtained by BARn of the PCI Express to an SuperHyway space, an address indicated in this field is used as upper bits of an address. In this case, lower bits of the PCI Express packet address is used as lower bits of the address, and the upper and lower bit boundaries are specified by LAMRn. When PCIETCTLR.CFINIT is set to 1, this field cannot be written to. In a Root Port, these bits in PEICLAR2 to PEICLAR5 are invalid and cannot be written to.
7 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(17) Local (SuperHyway) Address Mask Registers 0 to 5 (PCIELAMR0 to PCIELAMR5)

These registers specify the size of local bus (SuperHyway) during target transfer (transfer from PCIEC to SuperHyway). For details, refer to section 13.5.6, Target Transfers (Data Transfer from External Device to PCIEC).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	LAMR												
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAMR								SPCSEL				—	—	—	LARE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 29	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
28 to 8	LAMRn	All 0	—	R/W	These bits mask a local address (SuperHyway address) during transaction transfer from the PCI to the SuperHyway. When a packet to be accessed to the PCI Express area obtained by BARn is converted to the SuperHyway transaction, LARn bits and an address on the PCI packet are used as upper and lower bits of an address, respectively. In this case, the upper and lower bit boundaries are specified by LAMRn. In other words, a packet address addr[31:0] to be generated is as follows. Addr[31:30]: LARn[31:30] Addr[29:20]: LAR[29:20] when the corresponding bits in LAMRn[29:20] are cleared to 0; PCI-Addr[29:20] when the corresponding bits in LAMRn[29:20] are set to 1. PCI-Addr[29:20] is as follows. Addr[19:0]: PCI-Addr[19:0] These bits also specify the size of a space obtained in the PCIE space. B'0 0000 0000 0000 0000 0000: 256 bytes B'0 0000 0000 0000 0000 0001: 512 bytes B'0 0000 0000 0000 0000 0011: 1 Kbyte B'0 0000 0000 0000 0000 0111: 2 Kbytes B'0 0000 0000 0000 0000 1111: 4 Kbytes B'0 0000 0000 0000 0001 1111: 8 Kbytes B'0 0000 0000 0000 0011 1111: 16 Kbytes B'0 0000 0000 0000 0111 1111: 32 Kbytes B'0 0000 0000 0000 1111 1111: 64 Kbytes B'0 0000 0000 0001 1111 1111: 128 Kbytes B'0 0000 0000 0011 1111 1111: 256 Kbytes B'0 0000 0000 0111 1111 1111: 512 Kbytes B'0 0000 0000 1111 1111 1111: 1 Mbyte B'0 0000 0001 1111 1111 1111: 2 Mbytes B'0 0000 0011 1111 1111 1111: 4 Mbytes B'0 0000 0111 1111 1111 1111: 8 Mbytes B'0 0000 1111 1111 1111 1111: 16 Mbytes B'0 0001 1111 1111 1111 1111: 32 Mbytes B'0 0011 1111 1111 1111 1111: 64 Mbytes B'0 0111 1111 1111 1111 1111: 128 Mbytes B'0 1111 1111 1111 1111 1111: 256 Mbytes B'1 1111 1111 1111 1111 1111: 512 Mbytes Values other than above should not be set. When PCIETCTLR.CFINIT is set to 1, this field cannot be written to. In a Root Port, these bits in PCIELAMR2 to PCIELAMR5 are invalid and cannot be written to.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
7 to 4	SPCSEL	All 0	—	R/W	These bits specify the type of PCI space to be obtained for transaction transfer from PCI to the SuperHyway. To obtain memory: SPCSEL[0]: Clear to 0. SPCSEL[2:1]: Specify the type of memory to be obtained. B'00: Memory is obtained in the 32-bit address space. B'01: Memory is obtained in the area of 1 Mbyte or smaller in the 32-bit address space. B'10: Memory is obtained in the 64-bit address space. B'11: Reserved and setting prohibited SPCSEL[3]: Specify the memory prefetch. B'0: Memory which cannot be prefetched is obtained. B'1: Memory which can be prefetched is obtained. When a 64-bit address is used in PCIELAMRn, PCIELAMRn+1 cannot be used as independent space since PCIELAMRn+1 is used as an upper address of PCIELAMRn. PCIELAMRn+1 should be 0. To obtain IO space: SPCSEL[3:0]: Set to 4'b0001. When PCIETCTLR.CFINIT is set to 1, this field cannot be written to. When this module is used as a Root Port, these bits in PCIELAMR2 to PCIELAMR5 are invalid and cannot be written to.
3 to 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
0	LARE	0	—	R/W	Enables local address. When PCIELAMRn.LARE is set to 1, transaction transfer in the area specified by PCIELARn/PCIELAMRn is enabled. To enable address conversion by LARn or LAMRn, this bit should be set to 1. When PCIETCTLR.CFINIT is set to 1, this field cannot be written to. When this module is used as a Root Port, these bits in PCIELAMR2 to PCIELAMR5 are invalid and cannot be written to.

(18) PCI Address Lower Register 0 to 3 (PCIEPALR0 to PCIEPALR3)

PCIEPALR0 to PCIEPALR3 specify the lower 32 bits of a PCIEC address during PIO transfer (transfer from SuperHyway to PCIEC). For details, refer to section 13.5.5, PIO Transfers (Data Transfer from PCIEC to External Device).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PAL														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 18	PAL	All 0	—	R/W	These bits specify a PCIEC address (lower).
17 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(19) PCI Address Upper Register 0 to 3 (PCIEPAHR0 to PCIEPAHR3)

PCIEPAHR0 to PCIEPAHR3 specify the upper 32 bits of a PCIEC address during PIO transfer (from SuperHyway to PCIEC). For details, refer to section 13.5.5, PIO Transfers (Data Transfer from PCIEC to External Device).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PAH															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PAH															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	PAH	All 0	—	R/W	These bits specify a PCIEC address (upper).

(20) PCI Address Mask Registers 0 to 3 (PCIEPAMR0 to PCIEPAMR3)

PCIEPAMR0 to PCIEPAMR3 specify an address size in the PCIEC during PIO transfer (transfer from SuperHyway to PCIEC). For details, refer to section 13.5.5, PIO Transfers (Data Transfer from PCIEC to External Device).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	PAM											—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 29	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 18	PAM	All 0	—	R/W	These bits mask a PCI address. When a packet to be accessed to the SuperHyway space is converted to the PCI Express packet, PCIEPALRn bits and an address on the SuperHyway packet are used as upper and lower bits of an address, respectively. In this case, the upper and lower bit boundaries are specified by PCIEPAMRn. In other words, a packet address addr[31:0] on the SuperHyway to be generated is as follows. Addr[31:30]: PCIEPALRn [31:29] Addr[29:20]: PCIEPALRn [28:18] when the corresponding bits in PCIEPAMRn [28:18] are cleared to 0; SuperHyway-Addr[28:18] when the corresponding bits in PCIEPAMRn [28:18] are set to 1. SuperHyway-Addr[28:18] is as follows. Addr[17:0] : SuperHyway-Addr[17:0] PCIEPAHRn is used as the upper 32 bits of a PCI address. The size of transaction transfer destination space in the PCIEC is as follows according to this field setting. 0 0000 0000 00: 256 Kbytes 0 0000 0000 01: 512 Kbytes 0 0000 0000 11: 1 Mbyte 0 0000 0001 11: 2 Mbytes 0 0000 0011 11: 4 Mbytes 0 0000 0111 11: 8 Mbytes 0 0000 1111 11: 16 Mbytes 0 0001 1111 11: 32 Mbytes 0 0011 1111 11: 64 Mbytes 0 0111 1111 11: 128 Mbytes 0 1111 1111 11: 256 Mbytes 1 1111 1111 11: 512 Mbytes Values other than above should not be specified.
17 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
					should always be 0.

(21) PCI Conversion Control Registers 0 to 3 (PCIEPTCTLR0 to PCIEPTCTLR3)

PCIEPTCTLR0 to PCIEPTCTLR3 specify the attribute of a packet issued to the PCIEC during PIO transfer (transfer from SuperHyway to PCIEC). For details, refer to section 13.5.5, PIO Transfers (Data Transfer from PCIEC to External Device).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PARE	—	—	CONNECT	MAX PACKET SIZE				—	TC			—	—	—	—
Initial value:	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	LOCK	—	—	—	SPC	—	—	—	—	—	EP	ATTR	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	PARE	0	—	R/W	Indicates that PAR is valid.
30, 29	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
28	CONNECT	0	—	R/W	Specifies packet connection in PIO transfer. When this bit is set to 1, the PCIE combines multiple PIO transfers for write access to consecutive PCI area addresses, and generates a PCIe packet longer than the usual access length from the internal bus. This packet connection into a long packet improves the data transfer efficiency. Packets are connected when the following conditions are satisfied. Packet connection is enabled. Packets are connected when this CONNECT bit is set to 1. Note that packets are always connected in inter-lane transfer of the PCIEC regardless of the CONNECT bit setting. All transfers are write access. Packets are not connected for read requests. The access size is four bytes or larger. Packets are not connected for 1-byte or 2-byte access. Packets are issued to the memory space. Packets for the I/O space are not connected. All accesses are issued from one VC. Packets are not connected when requests are issued from multiple VCs. Successive accesses are issued to consecutive addresses. Packets are connected when the next request address is the current request address + size. The resultant address does not go beyond a 4-Kbyte boundary. The PCISIG standard prohibits generation of a packet that goes beyond a 4-Kbyte boundary. Data locations are consecutive. Packets are connected only when the locations of the enabled bytes are consecutive. Packets are not connected if the enabled bytes are not consecutive in the result of packet connection. The size of the connected packet does not exceed the maximum size. Packets are connected when the connected packet size does not exceed the maximum packet length specified by MAX_PACKET_SIZE or MPS.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
27 to 24	MAX PACKET SIZE	1010	—	R/W	These bits specify the maximum packet size for packet connection in PIO transfer. This value is valid when the PCIEPTCTLR0-5[27].CONNECT bit is set to 1; that is, packet connection is enabled. 0000: 4 bytes 0001: 8 bytes 0010: 16 bytes 0011: 32 bytes 0100: 64 bytes 0101: 128 bytes 0110: 256 bytes 0111: 512 bytes 1000: 1024 bytes 1001: 2048 bytes 1010: 4096 bytes If a size larger than the maximum payload size (MPS) is specified here, the MPS is used as the maximum packet length. Even when the data length in a request from the internal bus is larger than this maximum packet length, a single request is never divided and a packet is generated with the data length specified by the request. This size is also applied to the inter-lane transfer of the PCIE as the maximum packet length.
23	—	0	—	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 20	TC	000	—	R/W	These bits specify the traffic class (TC) of the PCIE packet in the transfer destination.
19 to 13	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
12	LOCK	0	—	R/W	Locks or unlocks the packet. (An Endpoint cannot issue a lock request.)
11 to 9	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
8	SPC	0	—	R/W	Specifies the space in the transfer destination. 0: Memory 1: IO (An Endpoint cannot issue a request to the IO space.)
7 to 3	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
2	EP	0	—	R/W	Specifies the EP of the transmit packet. If this bit is set to 1, the Poisoned attribute of the PCIe packet to be transmitted is set to 1.
1, 0	ATTR	00	—	R/W	These bits specify the attribute of a transmit packet. These bits specify the ATTR attribute of the PCIe packet to be transmitted as follows. ATTR[0]: No Snoop ATTR[1]: Relaxed Ordering Snoop is disabled in transfer destination by setting No Snoop bit in ATTR[0] to 1. (While VCCAP4[15].ReJeCT SNoop Transactions is set to 1 in the configuration register of transfer destination device, any request is processed as unsupported request unless No Snoop bit in ATTR[0] is set to 1.) Ordering in the transfer destination can be relaxed by setting Relaxed Ordering in ATTR[1] to 1.

13.4.4 PCIE-DMAC Control Registers

PCIE-DMAC Control Registers control DMA-Controller, which is implemented in the PCIEC module for the data-transfer between the SuperHyway bus and PCI Express.

Those registers are accessible via the SuperHyway bus. When a register is accessed from the PCI Express by target transfer, the PCIEC handles the access as a completer abort and executes error processing.

(1) PCI DMAC DMA Operation Register (PCIEDMAOR)

PCIEDMAOR controls the DMAC. For details, refer to section 13.5.7, DMA Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMAE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ABT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	DMAE	0	—	R/W	Specifies the DMA. To use the DMAC function, set this bit to 1.
30 to 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ABT	0	—	R/W	Specifies an arbitration type between channels. 0: CH0 > CH1 > CH2 > CH3 1: Round-robin mode

(2) PCI DMAC PCI Address Lower Registers 0 to 3 (PCIEDMPALR0 to PCIEDMPALR3)

PCIEDMPALR0 to PCIEDMPALR3 specify the lower 32 bits of a PCI address during DMA transfer. For details, refer to section 13.5.7, DMA Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PADRL															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PADRL														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 2	PADRL	All 0	—	R/W	These bits specify the lower 32 bits of a PCI address during DMA transfer. In DMA transfer, only 4-byte boundary can be specified. This field specifies bits excluding lower 2 bits.
1, 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(3) PCI DMAC PCI Address Upper Registers 0 to 3 (PCIEDMPAHR0 to PCIEDMPAHR3)

PCIEDMPAHR0 to PCIEDMPAHR3 specify the upper 32 bits of a PCI address during DMA transfer. For details, refer to section 13.5.7, DMA Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PADRH															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PADRH															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	PADRH	All 0	—	R/W	These bits specify the upper 32 bits of a PCI address during DMA transfer.

(4) PCI DMAC SuperHyway Address Lower Registers 0 to 3 (PCIEDMSALR0 to PCIEDMSALR3)

PCIEDMSALR0 to PCIEDMSALR3 specify a local bus (SuperHyway) address during DMA transfer. For details, refer to section 13.5.7, DMA Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SADR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SADR														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 2	SADR	All 0	—	R/W	These bits specify an SuperHyway address during DMA transfer. In DMA transfer, only 4-byte boundary can be specified. This filed specifies bits excluding lower 2 bits.
1, 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(5) PCI DMAC Byte Cont Registers 0 to 3 (PCIEDMBCNTR0 to PCIEDMBCNTR3)

PCIEDMBCNTR0 to PCIEDMBCNTR3 specify the byte count in DMA transfer. For details, refer to section 13.5.7, DMA Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	BCNT												
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BCNT														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 29	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 2	BCNT	All 0	—	R/W	These bits specify the transfer byte count. Note that only a multiple of four can be specified as transfer byte count. A quarter of the target value (a value excluding the lower 2 bits) should be specified in this field. If these bits are specified as 0, 2^29 bytes are transferred.
1, 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(6) PCI DMAC Stride Count Registers 0 to 3 (PCIEDMSBCNTR0 to PCIEDMSBCNTR3)

PCIEDMSBCNTR0 to PCIEDMSBCNTR3 specify the stride count in DMA transfer. For details, refer to section 13.5.7, DMA Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SBCINI														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SBCNT														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 18	SBCINI	All 0	—	R/W	These bits specify the initial value of the stride counter. These bits specify the initial value of the byte count of data to be transferred as a block in stride gather/scatter transfer. Note that only a multiple of four can be specified as the initial value of the stride counter. A quarter of the target value (a value excluding the lower 2 bits) should be specified in this field.
17, 16	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
15 to 2	SBCNT	All 0	—	R/W	These bits specify the stride counter which indicates the byte count of data to be transferred as a block in stride gather/scatter transfer. During data transfer, these bits indicate the number of remaining bytes to be transferred. When BCNT ≠ 0 and SBCNT = 0, data transfer is continued by loading the SBINI value. Note that only a multiple of four can be specified in this field. A quarter of the target value (a value excluding the lower 2 bits) should be specified in this field.
1, 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(7) PCI DMAC Stride Registers 0 to 3 (PCIEDMSTRR0 to PCIEDMSTRR3)

PCIEDMSTRR0 to PCIEDMSTRR3 specify the stride width in DMA transfer. For details, refer to section 13.5.7, DMA Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SS														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PS														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 18	SS	All 0	—	R/W	These bits specify the stride width of the SuperHyway address. Note that only a multiple of four can be specified as the stride width of the SuperHyway address. A quarter of the target value (a value excluding the lower 2 bits) should be specified in this field. These bits should be cleared to 0 if stride is not performed in the SuperHyway.
17, 16	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 2	PS	All 0	—	R/W	These bits specify the stride width of the PCI address. Note that only a multiple of four can be specified as the stride width of the PCI address. A quarter of the target value (a value excluding the lower 2 bits) should be specified in this field. These bits should be cleared to 0 if stride is not performed in the PCI.
1, 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(8) PCI DMAC Command Chain Address Registers 0 to 3 (PCIEDMCCAR0 to PCIEDMCCAR3)

PCIEDMCCAR0 to PCIEDMCCAR3 specify a command chain address in DMA transfer. For details, refer to section 13.5.7, DMA Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCA											—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 5	CCA	All 0	—	R/W	These bits specify an address of command chain to be executed next. Note that only 32-byte boundary can be specified as a command chain address. This field specifies a value excluding the lower 5 bits.
4 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(9) PCI DMAC Channel Control Registers 0 to 3 (PCIEDMCHCR0 to PCIEDMCHCR3)

PCIEDMCHCR0 to PCIEDMCHCR3 control DMA transfer channels. For details, refer to section 13.5.7, DMA Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CHE	DIR	CCRE	—	—	—	PASRE	SASRE	—	—	ATTR		—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	TC			Burst	—	—	—	—	—	Burst-Length		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	CHE	0	—	R/W	Enables or disables a channel. If this bit is set to 1, the data transfer of the corresponding channel is started. Note, however, that the data transfer is not performed while a bit indicating a transfer end (TE) or a transfer error (PE/SE) is set to 1. If this bit is cleared to 0 during DMA transfer, the data transfer can be forcibly aborted. After the data transfer is aborted, be sure to initialize the registers in the aborted channel. This bit is not cleared to 0 by transfer completion or suspension. 0: Disables a data transfer. 1: Enables a data transfer.
30	DIR	0	—	R/W	Specifies the transfer direction. 0: PCI to SuperHyway 1: SuperHyway to PCI

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
29	CCRE	0	—	R/W	Enables or disables a command chain. If a data transfer is requested while this bit is set to 1, the data transfer is performed by reading a command from an address specified by DCCAR. 0: Disables a command chain. 1: Enables a command chain.
28 to 26	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
25	PASRE	0	—	R/W	Enables or disables the PCI address stride registers. 0: Disables PCIEDMSTRR0-3.PS. 1: Enables PCIEDMSTRR0-3.PS.
24	SASRE	0	—	R/W	Enables or disables the SuperHyway address stride registers. 0: Disables PCIEDMSTRR0-3.SS. 1: Enables PCIEDMSTRR0-3.SS.
23, 22	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
21, 20	ATTR	00	—	R/W	These bits specify the ATTR attribute of a transmit PCIe packet. Each bit specifies an attribute as follows. ATTR[0]: No Snoop ATTR[1]: Relaxed Ordering Setting the ATTR[0].No Snoop bit to 1 stops the snoop operation at the destination. When the EXPCAP2[11].Enable No Snoop bit is 0, a transfer with the ATTR[0] bit set to 0 is prohibited; if attempted, the DMA transfer ends with an error. When the VCCAP4[15].ReJeCT SNoop Transactions bit of the configuration register in the destination device is set to 1, a transfer with the ATTR[0] bit set to 0 is handled as an unsupported request in the destination device. Setting the ATTR[1].Relaxed Ordering bit to 1 relaxes the ordering in the destination. When the EXPCAP2[4].Enable Relax Ordering bit is 0, a transfer with the ATTR[1] bit set to 1 is prohibited; if attempted, the DMA transfer ends with an error.
19 to 12	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 9	TC	000	—	R/W	These bits specify the TC (traffic Class) of the transaction to the PCI.
8	Burst	0	—	R/W	Enables the burst length setting for DMA transfer. When this bit is set to 1, channel arbitration is controlled so that channels are switched after transfer of data for the size specified by the Burst-Length bits in this register. When this bit is 0, channels are switched after a DMA transfer is completed in the selected channel.
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	PCI R/W	SH R/W	Description
2 to 0	Burst-Length	000	—	R/W	These bits specify the burst length for DMA transfer. When multiple channels are enabled, they are switched after transfer of data for the size specified by these bits. This setting is valid when the Burst bit is set to 1. 000: 32 bytes 001: 64 bytes 010: 128 bytes 011: 256 bytes 100: 512 bytes 101: 1024 bytes 110: 2048 bytes 111: 4096 bytes

(10) PCI DMAC Channel Status Registers 0 to 3 (PCIEDMCHSR0 to PCIEDMCHSR3)

PCIEDMCHSR0 to PCIEDMCHSR3 indicates the channel status in DMA transfer. For details, refer to section 13.5.7, DMA Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PEE	—	SEE	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R/W	R	R/W	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PE	—	SE	—	—	—	—	—	IE	—	—	TE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R/W1C	R	R/W1C	R	R	R	R	R	R/W	R	R	R/W1C
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH 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	PEE	0	—	R/W	Enables or disables an interrupt caused by a transfer error in the PCI. 0: Disables an interrupt. 1: Enables an interrupt.
26	—	0	—	R	Reserved This bit is always read as 0. The write value should always be 0.
25	SEE	0	—	R/W	Enables or disables an interrupt caused by a transfer error in the SuperHyway. 0: Disables an interrupt. 1: Enables an interrupt.
24 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	PCI R/W	SH R/W	Description
11	PE	0	—	R/W1C	Indicates that a transfer error has occurred in the PCI. Sources of this error are as follows. A DMA transfer is started while a connection has not been established yet. A DMA transfer using VCX is started while VCX is invalid. A DMA transfer is started by specifying TC which is not mapped in the VC0(X). A DMA transfer is started when Link-IP is an Endpoint and when PCICONF1[2] bus master enable is deasserted. A DMA transfer is started by specifying an attribute which is not allowed in the configuration registers.
10	—	0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
9	SE	0	—	R/W1C	Indicates that a transfer error has occurred in the SuperHyway.
8 to 4	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
3	IE	0	—	R/W	Enables or disables an interrupt. If the TE bit is set to 1 while this bit is set to 1, an interrupt is requested. 0: Disables an interrupt request. 1: Enables an interrupt request.
2, 1	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
0	TE	0	—	R/W1C	Indicates the transfer end flag. If a data transfer is completed after PCIEDMBCNTR0 to PCIEDMBCNTR3 are cleared to 0, this bit is set to 1. If a data transfer is terminated by a transfer error, or if a data transfer is forcibly terminated by clearing the PCIEDMCHCR.CHE bit to 0, this bit will not be set to 1. In addition, if the next transfer is specified by using a command chain (CCRE = 1), this bit will not be set to 1 even if the current data transfer is completed normally. While this bit is set to 1, a data transfer is not performed even if the PCIEDMCHCR.CHE bit is set to 1. 0: Indicates that data is being transferred or that a data transfer is terminated. 1: Indicates that a data transfer is completed (by PCIEDMBCNTR0 to PCIEDMBCNTR3 = 0)

13.4.5 Configuration Registers

Configuration registers are defined in the PCI-SIG Specification, and incorporated in this module. For the outline of the configuration registers incorporated in this module, refer to section 13.5.1, (3) Configuration Registers, and section 13.5.1, (4) Capability Structure.

These registers are accessible via the internal bus (SuperHyway) and PCI Express. When accessed from the PCI Express, configuration read/write accesses should be used. These registers cannot be accessed by memory read/write or I/O read/write in target transfer. When an access is attempted, the PCIEC handles the access as a completer abort and executes error processing.

(1) PCI Configuration Register 0 (PCIEPCICONF0)

PCIEPCICONF0 corresponds to the vendor-ID register and device-ID register defined in the PCI Express Specification.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Device ID															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Vendor ID															
Initial value:	0	0	0	1	1	0	0	1	0	0	0	1	0	0	1	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 16	Device ID	H'0010	R	R	These bits indicate the device ID. The assigned device ID will be indicated.
15 to 0	Vendor ID	H'1912	R	R	These bits indicate the vendor ID. The PCI device vendor will be indicated. H'1912 is the ID assigned to Renesas Electronics Corp.

(2) PCI Configuration Register 1 (PCIEPCICONF1)

PCIEPCICONF1 corresponds to the command register and status register defined in the PCI Express Specification.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Detected Parity Error	Signaled System Error	Received Master Abort	Received Target Abort	Signaled Target Abort	—	—	Master Data Parity Error	—	—	—	CAPaility List	INTerrupt Status	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
SH-R/W:	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R	R	R/W1C	R	R	R	R	R	R	R	R
PCI-R/W:	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R	R	R/W1C	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	INTerrupt DISable	—	SERR Enable	—	Parity Error ReSponse	—	—	—	Bus Master Enable	Memory Space Enable	IO Space Enable
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	Detected Parity Error	0	R/W1C	R/W1C	Indicates that a parity error has been detected. This bit is set to 1 when a Poisoned TLP (Request or Completion) has been received. (This bit is set to 1 regardless of the value of PCICONF0[6].Parity Error Response.)
30	Signaled System Error	0	R/W1C	R/W1C	Indicates that a system error has been detected. Root Port: This bit is set to 1, if a FATAL/NON_FATAL error is detected while PCICONF1[8].SERR Enable = 1 or if a FATAL/NON_FATAL error message is received while PCICONF15[17].SERR Enable = 1. Endpoint: This bit is set to 1, if a FATAL/NON_FATAL error is detected and if a ERR_FATAL/ERR_NONFATAL message is sent while PCICONF1[8].SERR Enable = 1.
29	Received Master Abort	0	R/W1C	R/W1C	Indicates that a master abort has been received. This bit is set to 1 if a completion with the Unsupported Request Completion Status has been received.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
28	Received Target Abort	0	R/W1C	R/W1C	Indicates that a target abort has been received. This bit is set to 1 if a completion with the Completer Abort Completion Status has been received.
27	Signaled Target Abort	0	R/W1C	R/W1C	Indicates that a target abort has been transmitted. This bit is set to 1 when a completion with the Completer Abort Completion Status has been transmitted.
26, 25	DEVSEL Timing	All 0	R	R	These bits are always read as 0. The write value should always be 0. These bits do not affect the PCI Express.
24	Master Data Parity Error	0	R/W1C	R/W1C	Indicates that a parity error has occurred. This bit is set to 1 by the following operations while PCICONF1[6].Parity Error Response is set to 1. 1. The requester receives a Poisoned completion. 2. The requester transmits a Poisoned request. This bit is not set to 1 if PCICONF1[6].PERS is cleared to 0.
23	Fast Back to Back Transaction CAPable	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
22	Reserved	0	R	R	Reserved This bit is always read as 0. The write value should always be 0.
21	66Mhz CAPABLE	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
20	CAPaility List	1	R	R	Indicates that the Expansion Capability List exists in the PCI compatible configuration space. This field is set to 1 because this module has the Expansion Capability List.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
19	INTerrupt Status	0	R	R	Indicates that an INTx message is pending. In an Endpoint, this bit is set to 1 when an INTx interrupt has occurred; while it is cleared to 0 when an INTx interrupt has been cancelled. An INTx interrupt cannot be generated while this bit is set to 1. In addition, if PCICONF1[10].INTerrupt DISable is set to 1 while this bit is set to 1, a Deassert INTx message is sent and this bit is cleared to 0.
18 to 11	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
10	INTerrupt DISable	0	R/W	R/W	Disables an INTx message transmission. Setting this bit to 1 disables an INTx message transmission. If an INTx interrupt has been asserted by INTx emulation (PCICONF1[19].INTerrupt Status = 1), a Deassert INTx message is sent when this bit is set to 1 and the interrupt is then deasserted. (PCICONF1[19].INTerrupt Status is cleared to 0.)
9	Fast Back to Back Transaction Enable	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
8	SERR Enable	0	R/W	R/W	Enables the system error report. Root Port: If a NonFatal/Fatal is detected while this bit is set to 1 and while the following signal is set to 1, the corresponding interrupt will be generated. EXPCAP7[2].System ERRor on Fatal Error Enable EXPCAP7[1].System ERRor on Non Fatal Error Enable A NonFatal/Fatal error will be detected both by detecting an error in a Root Port and by receiving an error message from an Endpoint. The above interrupt can also be generated by setting the following register to 1. EXPCAP2[2].Fatal Error RePorting Enable EXPCAP2[1].Non Fatal Error RePorting Enable Endpoint: If a NonFatal/Fatal is detected while this bit is set to 1, the hardware transmits an ERR_NONFATAL/ERR_FATAL message. The above message can also be sent by setting the following register to 1. EXPCAP2[2].Fatal Error RePorting Enable EXPCAP2[1].Non Fatal Error RePorting Enable
7	IDSel Stepping/Wait Cycle Control	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
6	Parity Error ReSponse	0	R/W	R/W	Specifies the response for a parity error. The PCICONF1[24].Master Data Parity Error bit is set to1 by the following operations while this bit is set to 1. 1. The requester receives a Poisoned completion. 2. The requester transmits a Poisoned request.
5	VGA Plate Snoop	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
4	Memory Write and Invalidate	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
3	SCE	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
2	Bus Master Enable	0	R/W	R/W	Enables a memory/IO request issue. Root Port: While this bit is set to 1, a memory/IO request can be received. While this bit is cleared to 0, a received request is handled as an unsupported request. Endpoint: While this bit is set to 1, a memory/IO request can be issued. While this bit is cleared to 0, any memory/IO request cannot be issued.
1	Memory Space Enable	0	R/W	R/W	Enables an access to memory space. While this bit is cleared to 0, an access to the memory space is processed as an Unsupported Request.
0	IO Space Enable	0	R/W	R/W	Enables an access to IO space. While this bit is cleared to 0, an access to the IO space is processed as an Unsupported Request.

(3) PCI Configuration Register 2 (PCIEPCICONF2)

PCIEPCICONF2 corresponds to the revision ID register and class code register defined in the PCI Express Specification.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Class Code								Sub-Class Code							
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Prog IF								Revision ID							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	Class Code	H'FF	R	R	These bits indicate the class code. The value set in the Class Code Set field of PCIEIDSETR1 is reflected on these bits.
23 to 16	Sub-Class Code	H'00	R	R	These bits indicate the sub-class code. The value set in the Sub Class Code Set field of PCIEIDSETR1 is reflected on these bits.
15 to 8	Prog IF	H'00	R	R	These bits indicate the programming IF. The value set in the PROG IF Set field of PCIEIDSETR1 is reflected on these bits.
7 to 0	Revision ID	H'00	R	R	These bits indicate the revision ID. The value set in the Rev ID Set field of PCIEIDSETR1 is reflected on these bits.

(4) PCI Configuration Register 3 (PCIEPCICONF3)

PCIEPCICONF3 corresponds to the cache line register, master latency timer register, header type register, and BIST register defined in the PCI Express Specification.

This module does not support the BIST register. In addition, this module does not support functions of the master latency timer register and cache line register which have no meaning for the PCI Express.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BiST CAPable	Start BiST	—	—	BiST completion CODE				Single Function / Multi Function	HeaDer TYPE						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
SH-R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MASter Latency Timer								Cache Line Size							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	BiST CAPable	0	R	R	Indicates the BIST function support status. 1: BIST function supported. 0: BIST function not supported. This module does not support the BIST function.
30	Start BiST	0	R	R	Indicates the BIST function execution status. 1: BIST function execution in progress. 0: BIST function ends. This module does not support the BIST function.
29, 28	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
27 to 24	BiST completion CODE	H'0	R	R	These bits indicate the BIST completion status code. H'1 to H'F: Failure detected. H'0: BIST has passed. This module does not support the BIST function.
23	Single Function / Multi Function	0	R	R	Indicates whether single function device or multi function device is used. 1: Multi function device 0: Single function device This module supports only single function.
22 to 16	HeaDer TYPE	Root Port: H'01 Endpoint: H'00	R	R/W	These bits specify a layout of the configuration registers. H'00: Type00 layout H'01: Type01 layout H'02: Type02 layout (setting prohibited) This field is set according to the mode pin setting at initialization.
15 to 8	MASter Latency Timer	H'00	R	R	These bits are always read as 0. The write value should always be 0. These bits do not affect the PCI Express.
7 to 0	Cache Line Size	H'00	R	R	These bits are always read as 0. The write value should always be 0. These bits do not affect the PCI Express.

(5) PCI Configuration Register 4 (PCIEPCICONF4)

PCIEPCICONF4 corresponds to the base address register 0 defined in the PCI Express Specification.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAR0															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAR0															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	BAR0	—	R/W*	R/W*	These bits indicate the value of Base Address Register 0. When used as a Root Port, set a value to these bits by software during the configuration cycle. When used as an Endpoint, a value is set to these bits by a configuration request from the Root Port during the configuration cycle. Note: * The initial value and R/W attribute to the peripheral bus are set according to the PCILAMR2 value when the CFINIT bit is set to 1.

(6) PCI Configuration Register 5 (PCIEPCICONF5)

PCIEPCICONF4 corresponds to the base address register 1 defined in the PCI Express Specification.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAR1															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAR1															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	BAR1	—	R/W*	R/W*	These bits indicate the value of Base Address Register 1. When used as a Root Port, set a value to these bits by software during the configuration cycle. When used as an Endpoint, a value is set to these bits by a configuration request from the Root Port during the configuration cycle. Note: * The initial value and R/W attribute to the peripheral bus are set according to the PCILAMR2 value when the CFINIT bit is set to 1.

(7) PCI Configuration Register 6 (PCIEPCICONF6)

If the PCIEC is used as an Endpoint, this register corresponds to the base address register 2 defined in the PCI Express Specification; while if the PCIEC is used as an Root Port, this register corresponds to the primary bus number register, secondary bus number register, subordinate bus number register, and secondary latency timer register.

Note that this module does not support the function of the secondary latency timer register which has no meaning for the PCI Express.

The primary bus number register, secondary bus number register, and subordinate bus number register should be set by software.

When the PCIEC is used as an Endpoint:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAR2															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAR2															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	BAR2	—	R/W*	R/W*	These bits indicate the value of Base Address Register 2. A value is set to these bits by a configuration request from the Root Port during the configuration cycle. Note: * The initial value and R/W attribute to the peripheral bus are set according to the PCILAMR2 value when the CFINIT bit is set to 1.

When the PCIEC is used as a Root Port:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SECondary LATency Timer								SUBordinate BUS NUMber							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SECondary BUS NUMber								PRIMary BUS NUMber							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	SECondary LATency Timer	H'00	R	R	These bits are always read as 0. The write value should always be 0. These bits do not affect the PCI Express.
23 to 16	SUBordinate BUS NUMber	H'00	R/W	R/W	These bits indicate the maximum number of device to be connected to the downstream link.
15 to 8	SECondary BUS NUMber	H'00	R/W	R/W	These bits indicate number of the bus directly connected to the secondary IF.
7 to 0	PRIMary BUS NUMber	H'00	R/W	R/W	These bits indicate number of the bus directly connected to the primary IF.

(8) PCI Configuration Register 7 (PCIEPCICONF7)

If the PCIEC is used as an Endpoint, this register corresponds to the base address register 3 defined in the PCI Express Specification; while if the PCIEC is used as an Root Port, this register corresponds to the I/O base register, I/O limit register, and secondary status register.

Note that this module does not support the function of the I/O base register and I/O limit register.

When the PCIEC is used as an Endpoint:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAR3															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAR3															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	BAR3	—	R/W*	R/W*	These bits indicate the value of Base Address Register 3. A value is set to these bits by a configuration request from the Root Port during the configuration cycle. Note: * The initial value and R/W attribute to the peripheral bus are set according to the PCILAMR3 value when the CFINIT bit is set to 1.

When the PCIEC is used as a Root Port:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Detected Parity Error	Received System Error	Received Master Abort	Received Target Abort	Signaled Target Abort	—	—	Master Data Parity Error	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R	R	R/W1C	R	R	R	R	R	R	R	R
PCI-R/W:	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R	R	R/W1C	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IOLiMiT End Address				IOLiMiT TYPE				IOBaSe End ADdress				IOBaSe TYPE			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	Detected Parity Error	0	R/W1C	R/W1C	Indicates that a parity error is detected at the link destination. This bit is set to 1 when a Poisoned TLP is transmitted. (This bit is set to 1 irrespective of the value of the PCICONF15[16].Parity Error Response.)
30	Received System Error	0	R/W1C	R/W1C	Indicates that a system error is detected at the link destination. This bit is set to 1 when an ERR_FATAL/ERR_NONFATAL message is received.
29	Received Master Abort	0	R/W1C	R/W1C	Indicates that a master abort is received at the link destination. This bit is set to 1 when a completion with Unsupported Request Completion Status is transmitted.
28	Received Target Abort	0	R/W1C	R/W1C	Indicates that a target abort is received at the link destination. This bit is set to 1 when a completion with Completer Abort Completion Status is transmitted.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
27	Signaled Target Abort	0	R/W1C	R/W1C	Indicates that a target abort is transmitted at the link destination. This bit is set to 1 when a completion with Completer Abort Completion Status is received.
26, 25	DEVSEL Timing	00	R	R	These bits are always read as 0. The write value should always be 0. These bits do not affect the PCI Express.
24	Master Data Parity Error	0	R/W1C	R/W1C	This bit is set if one of the following conditions is satisfied while PCICONF15[16].Parity Error Response = 1. 1. The completer transmits a completion with poisoned attribute. 2. The completer receives a write request with poisoned attribute.
23	Fast Back to Back Transaction CAPable	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
22	Reserved	0	R	R	Reserved This bit is always read as 0. The write value should always be 0.
21	66Mhz CAPable	0	R	R	These bits are fixed to 0 and do not apply to the PCI-Express.
20 to 16	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 12	IOLiMiT End Address	H'0	R	R	These bits specify Address [15:12] of the upper limit address of an IO transaction to be transferred to the primary. (Not supported)
11 to 8	IOLiMiT TYPE	H'0	R	R	These bits specify the upper-limit address decode type of an IO transaction to be transferred to the primary. H'0: 16-bit I/O address H'1: 32-bit I/O address (Not supported)

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
7 to 4	IOBaSe End ADdress	H'0	R	R	These bits specify Address [15:12] of the base address of an IO transaction to be transferred to the primary. (Not supported)
3 to 0	IOBaSe TYPE	H'0	R	R	These bits specify the base address decode type of an IO transaction to be transferred to the primary. H'0: 16-bit I/O address H'1: 32-bit I/O address (Not supported)

(9) PCI Configuration Register 8 (PCIEPCICONF8)

If the PCIEC is used as an Endpoint, this register corresponds to the base address register 4 defined in the PCI Express Specification; while if the PCIEC is used as an Root Port, this register corresponds to the memory base register and memory limit register.

Note that this module does not support the function of the memory base register and memory limit register.

When the PCIEC is used as an Endpoint:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAR4															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAR4															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	BAR4	—	R/W*	R/W*	These bits indicate the value of Base Address Register 4. A value is set to these bits by a configuration request from the Root Port during the configuration cycle. Note: * The initial value and R/W attribute to the peripheral bus are set according to the PCILAMR4 value when the CFINIT bit is set to 1.

When the PCIEC is used as a Root Port:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MEMory LiMiT												—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MEMory BASE												—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 20	MEMory LiMiT	H'000	R	R	These bits specify the upper limit address of the memory mapped IO to be transferred between the primary IF and the secondary IF. This field corresponds to Address [31:20]. (Not supported)
19 to 16	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
15 to 4	MEMory BASE	H'000	R	R	These bits specify the base address of the memory mapped IO to be transferred between the primary IF and the secondary IF. This field corresponds to Address [31:20]. (Not supported)
3 to 0	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.

(10) PCI Configuration Register 9 (PCIEPCICONF9)

If the PCIEC is used as an Endpoint, this register corresponds to the base address register 5 defined in the PCI Express Specification; while if the PCIEC is used as an Root Port, this register corresponds to the prefetchable memory base register and prefetchable memory limit register.

Note that this module does not support the function of the prefetchable memory base register and prefetchable memory limit register.

When the PCIEC is used as an Endpoint:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAR5															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAR5															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	BAR5	—	R/W*	R/W*	These bits indicate the value of Base Address Register 5. A value is set to these bits by a configuration request from the Root Port during the configuration cycle. Note: * The initial value and R/W attribute to the peripheral bus are set according to the PCILAMR5 value when the CFINIT bit is set to 1.

When the PCIEC is used as a Root Port:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PRefetchable MEMory LiMiT												Prefetchable Memory address decode TYPE			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Prefetchable MEMory BASE												PPRefetchable Memory BaSe TYPE			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 20	PRefetchable MEMory LiMiT	H'000	R	R	These bits specify the upper limit address of the prefetchable memory to be transferred between the primary IF and the secondary IF. This field corresponds to Address [31:20]. (Not supported)
19 to 16	Prefetchable Memory address decode TYPE	H'0	R	R	These bits specify the upper limit address decode format of prefetchable memory to be transferred between the primary IF and the secondary IF. H'0: 32-bit address H'1: 64-bit address (Not supported)

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
15 to 4	PRefetchable MEMory BASE	H'000	R	R	These bits specify the base address of the prefetchable memory to be transferred between the primary IF and the secondary IF. This field corresponds to Address [31:20]. (Not supported)
3 to 0	PPRefetchable Memory BaSe TYPE	H'0	R	R	These bits specify the base address decode format of prefetchable memory to be transferred between the primary IF and the secondary IF. H'0: 32-bit address H'1: 64-bit address (Not supported)

(11) PCI Configuration Register 10 (PCIEPCICONF10)

If the PCIEC is used as an Endpoint, this register corresponds to the card bus CIS pointer register defined in the PCI Express Specification; while if the PCIEC is used as a Root Port, this register corresponds to the prefetchable base register (upper 32 bits).

Note that this module does not support the function of the card bus CIS pointer register and prefetchable memory base register.

When the PCIEC is used as an Endpoint:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Card Bus CisPointer															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Card Bus CisPointer															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	Card Bus CisPointer	H'00000000	R	R	These bits indicate the value of the card bus CIS pointer. CISSETR value is set in this field. This device does not support the card bus. (Not supported)

When the PCIEC is used as a Root Port:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Prefetchable memory BaSe UPper32															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Prefetchable memory BaSe UPper32															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	PRefetchable memory BaSe UPper32	H'00000000	R	R	These bits specify the upper limit address of the prefetchable memory to be transferred between the primary IF and the secondary IF. This field corresponds to Address [63:32]. This field is valid when PCICONF9.PMBSTYPE = H'1 (64-bit address decode format). (Not supported)

(12) PCI Configuration Register 11 (PCIEPCICONF11)

If the PCIEC is used as an Endpoint, this register corresponds to the subsystem ID register defined in the PCI Express Specification; while if the PCIEC is used as a Root Port, this register corresponds to the prefetchable limit register (upper 32 bits).

Note that this module does not support the function of the prefetchable limit register.

When the PCIEC is used as an Endpoint:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Sub System ID															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Sub System VenDor ID															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 16	Sub System ID	H'00000000	R	R	These bits indicate the Sub System ID. IDSETR2.SSIDSet value is set.
15 to 0	Sub System VenDor ID	H'00000000	R	R	These bits indicate the Sub System Vendor ID. IDSETR2.SSVIDS value is set.

When the PCIEC is used as a Root Port:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Prefetchable memory LiMit UPper32															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Prefetchable memory LiMit UPper32															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	PRefetchable memory LiMit UPper32	H'00000000	R	R	These bits specify the base address of the prefetchable memory to be transferred between the primary IF and the secondary IF. This field corresponds to Address [63:32]. This field is valid when PCICONF9. PMLMTTYPE = H'1 (64-bit address decode format). (Not supported)

(13) PCI Configuration Register 12 (PCIEPCICONF12)

If the PCIEC is used as an Endpoint, this register corresponds to the subsystem vendor ID register defined in the PCI Express Specification; while if the PCIEC is used as a Root Port, this register corresponds to the I/O base register (upper 16 bits) and I/O limit register (lower 16 bits).

Note that this module does not support the function of the I/O base register and I/O limit register.

When the PCIEC is used as an Endpoint:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Expansion ROM BAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Expansion ROM BAR					—	—	—	—	—	—	—	—	—	—	Expansion ROM Enable
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 11	Expansion ROM BAR	H'000000	R	R	These bits indicate the value of the expansion ROM base address register. The initial value and R/W attribute to the peripheral bus are set according to the EROMSETR value. This device does not support expansion ROM. (Not supported)
10 to 1	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
0	Expansion ROM Enable	0	R	R	Indicates that accessing expansion ROM is enabled. This device does not support expansion ROM. (Not supported)

When the PCIEC is used as a Root Port:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IOLiMiT UPper Address															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IO Base Upper Address															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 16	IOLiMiT UPper ADdress	H'0000	R	R	These bits specify Address [31:16] of the upper limit address of an IO transaction to be transferred to the primary IF. This field is valid when PCICONF7. IOLMTTYPE = 1 (32-bit address decode). (Not supported)
15 to 0	IO BaSe Upper Address	H'0000	R	R	These bits specify Address [31:16] of the base address of an IO transaction to be transferred to the primary IF. This field is valid when PCICONF7. IOBSTYPE = 1 (32-bit address decode). (Not supported)

(14) PCI Configuration Register 13 (PCIEPCICONF13)

PCIEPCICONF13 corresponds to the capability pointer register defined in the PCI Express Specification.

When the PCIEC is used as an Endpoint:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CAPabilities Pointer							
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 8	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	CAPabilities Pointer	H'40	R	R/W	Pointer to the expansion capability list. This IP has expansion capability list, and indicates the pointer H'40 of the PCI PM capability.

When the PCIEC is used as a Root Port:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CAPabilities Pointer							
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 8	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	CAPabilities Pointer	H'40	R	R/W	Pointer to the expansion capability list. This IP has expansion capability list, and indicates the pointer H'40 of the PCI PM capability.

(15) PCI Configuration Register 14 (PCIEPCICONF14)

If the PCIEC is used as an Endpoint, this register corresponds to the expansion ROM base address register defined in the PCI Express Specification.

Note that this module does not support the function of the expansion ROM base address register.

When the PCIEC is used as an Endpoint:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.

When the PCIEC is used as a Root Port:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Expansion ROM BAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Expansion ROM BAR					—	—	—	—	—	—	—	—	—	—	Expansion ROM Enable
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 11	Expansion ROM BAR	H'000000	R	R	These bits indicate the value of the expansion ROM base address register. The initial value and R/W attribute to the peripheral bus are set according to the EROMSETR value. This device does not support expansion ROM. (Not supported)
10 to 1	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
0	Expansion ROM Enable	0	R	R	Indicates that accessing expansion ROM is enabled. This device does not support expansion ROM. (Not supported)

(16) PCI Configuration Register 15 (PCIEPCICONF15)

If the PCIEC is used as an Endpoint, this register corresponds to the interrupt line register, interrupt pin register, minimum grant register, and maximum latency register defined in the PCI Express Specification; while if the PCIEC is used as a Root Port, this register corresponds to the interrupt line register, interrupt pin register, minimum grant register, maximum latency register, and bridge control register.

Note that this module does not support the function of the minimum grant register and maximum latency register which have no meaning for the PCI Express.

When the PCIEC is used as an Endpoint:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MAX LAT								MINGNT							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTerrupt PIN								INTerrupt LINE							
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	MAX LAT	H'00	R	R	These bits are always read as 0. The write value should always be 0. These bits do not affect the PCI Express.
23 to 16	MINGNT	H'00	R	R	These bits are always read as 0. The write value should always be 0. These bits do not affect the PCI Express.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
15 to 8	INTerrupt PIN	H'00	R	R/W	These bits specify a type of INTx to be used. To output an INTx interrupt from this module, an interrupt to be specified by this register is used. H'00: PCI interrupt pin is not used. H'01: INTA H'02: INTB H'03: INTC H'04: INTD H'05 to H'FF: Reserved (Setting prohibited)
7 to 0	INTerrupt LINE	H'FF	R/W	R/W	These bits specify an interrupt path for an INTx interrupt. H'00 to H'FE: Notifies the INTC of an INTx interrupt. H'FF: Does not notify the INTC of an INTx interrupt. For an Endpoint, this field has no meaning.

When the PCIEC is used as a Root Port:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	Discard Timer SERR Enable	Discard Timer STATUS	SECondary Discard Timer	Primary Discard Timer	Fast Back to Back Transaction CAPable	SECondary Bus ReSeT	Master ABort MoDe	VGA 16bit DECode	VGA Enable	ISA Enable	SERR Enable	Parity Error ReSponse
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTerrupt PIN								INTerrupt LINE							
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 28	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
27	Discard Timer SERR Enable	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
26	Discard Timer STATUS	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
25	SECondary Discard Timer	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
24	Primary Discard Timer	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
23	Fast Back to Back Transaction CAPable	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
22	SECondary Bus ReSeT	0	R/W	R/W	Resets the secondary IF. Setting this bit to 1 makes a transition to the LTSSM HotReset state. Clearing this bit cancels the LTSSM HotReset state.
21	Master Abort MoDe	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
20	VGA 16bit DECode	0	R	R	Enables VGA IO decoding. (Not supported)
19	VGA Enable	0	R	R	Enables VGA address conversion. (Not supported)
18	ISA Enable	0	R	R	Enables ISA IO address conversion. (Not supported)

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
17	SERR Enable	0	R/W	R/W	Enables ERR_FATAL/ERR_NONFATAL/ERR_COR message transfer to the primary IF. 1: Enables error message transfer. 0: Disables error message transfer. When this bit is 0, error notification according to ERR_FATAL/ERR_NONFATAL/ERR_ COR message from Endpoint does not processed by the Root Port.
16	Parity Error ReSponse	0	R/W	R/W	Specifies a parity error response. When this bit is 1, the PCICONF7[24].Master Data Parity Error bit is set by the following operation. 1. The completer transmits a completion with Poisoned attribute. 2. The completer receives a write request with Poisoned attribute.
15 to 8	INTerrupt PIN	H'00	R	R/W	These bits specify a type of INTx to be used. H'00: PCI interrupt pin is not used. H'01: INTA. H'02: INTB. H'03: INTC. H'04: INTD. H'05 to H'FF: Reserved (Setting prohibited) When this module is used as a Root Port, an INTx interrupt cannot be generated. 00h should be specified.
7 to 0	INTerrupt LINE	H'FF	R/W	R/W	These bits specify an interrupt path for an INTx interrupt. H'00 to H'FE: Notifies the INTC of an INTx interrupt. H'FF: Does not notify the INTC of an INTx interrupt. When this module is used as a Root Port, specifying a value other than H'FF enables to notify the INTC of an INTx interrupt.

(17) PCI Power Management Capability Register 0 (PCIEPMCAP0)

PCIEPMCAP0 corresponds to the power management capability register in the power management capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PME SuPport D3COLD	PME SuPport D3HOT	PME SuPport D2	PME SuPport D1	PME SuPport D0	D2 SuPport	D1 SuPport	AUX CURrent			Device Specific Initialization	—	PME Clock	PCI PM VERsion		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NEXT CAPability Pointer								CAPability ID							
Initial value:	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	PME SuPport D3COLD	0	R	R/W	Indicates the PME generation support status in the D3Cold state. 1: Supports PME generation in the D3Cold state. 0: Does not support PME generation in the D3Cold state. Root Port: Set this bit to 1 at initialization. Endpoint: Set the device function at initialization.
30	PME SuPport D3HOT	0	R	R/W	Indicates the PME generation support status in the D3HOT state. 1: Supports PME generation in the D3HOT state. 0: Does not support PME generation in the D3HOT state. Root Port: Set this bit to 1 at initialization. Endpoint: Set the device function at initialization.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
29	PME SuPport D2	0	R	R/W	Indicates the PME generation support status in the D2 state. 1: Supports PME generation in the D2 state. 0: Does not support PME generation in the D2 state. Set the device function at initialization.
28	PME SuPport D1	0	R	R/W	Indicates the PME generation support status in the D1 state. 1: Supports PME generation in the D1 state. 0: Does not support PME generation in the D1 state. Set the device function at initialization.
27	PME SuPport D0	0	R	R/W	Indicates the PME generation support status in the D0 state. 1: Supports PME generation in the D0 state. 0: Does not support PME generation in the D0 state. Set the device function at initialization.
26	D2 SuPport	0	R	R/W	Indicates the D2 power management state support status. 1: Supports the D2 state. 0: Does not support the D2 state. Set the device function at initialization.
25	D1 SuPport	0	R	R/W	Indicates the D1 power management state support status. 1: Supports the D1 state. 0: Does not support the D1 state. Set the device function at initialization.
24 to 22	AUX CURrent	000	R	R/W	These bits indicate the current of 3.3-V auxiliary power supply. Set at initialization. 111: 375 mA 110: 320 mA 101: 270 mA 100: 220 mA 011: 160 mA 010: 100 mA 001: 55 mA 000: 0 mA (without auxiliary power supply)

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
21	Device Specific Initialization	0	R	R/W	Indicates whether the device specific initialization sequence is necessary or not after D0 initialization. 1: Device specific initialization sequence is necessary. 0: Device specific initialization sequence is not necessary. Set at initialization.
20	Reserved	0	R	R	Reserved This bit is always read as 0. The write value should always be 0.
19	PME Clock	0	R	R	This bit is always read as 0. The write value should always be 0. This bit does not affect the PCI Express.
18 to 16	PCI PM VERsion	H'3	R	R/W	These bits indicate the corresponding PCI PM Interface Specification version. This device corresponds to the PCI PM Interface Specification Ver1.2.
15 to 8	NEXT CAPability Pointer	H'50	R	R/W	Pointer to the expansion capability list.
7 to 0	CAPability ID	H'01	R	R/W	Capability list ID. These bits indicate the Power Management Capability (H'01).

(18) PCI Power Management Capability Register 1 (PCIEPMCAP1)

PCIEPMCAP1 corresponds to the power management status/command register in the power management capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA								Bus Power Clock Control Enable	B2B3 SuPport	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PME Status	DaTa SCALE		DaTa SELect				PME Enable	—	—	—	—	NO Soft ReSet	—	PoWer STATE	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R	R/W	R/W
PCI-R/W:	R/W1C	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	DATA	H'00	R	R/W	These bits indicate the value selected by PMCAP1.DTSEL. When PMCAP1.DTSEL is modified, write the specified value.
23	Bus Power Clock Control Enable	0	R	R/W	Enables the PCI bus power/clock control function. 1: Enables the PCI bus power/clock control function. 0: Disables the PCI bus power/clock control function. Set the device function at initialization.
22	B2B3 SuPport	0	R	R/W	Indicates the bus status in the D3Hot state. This field is valid when PMCAP1.BPCC_EN = 1. 1: Clocks supplied to the secondary IF stops in the D3Hot state. (B2) 0: Clocks supplied to the secondary IF does not stop in the D3Hot state. (B3) Set the device function at initialization.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
21 to 16	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
15	PME Status	0	R/W1C	R/W	This field is valid when the PME is supported. Indicates the PME generation status. 1: A PME message is being transmitted. 0: A PME message is not transmitted. Set this bit to 1 when a PME message is transmitted. Clear this bit to 0 on receiving a configuration write request to write 1 to this field. Note: When PME generation is enabled in the D3Cold state, this field becomes R/W1CS.
14, 13	DaTa SCALE	00	R	R/W	These bits indicate the scale to indicate the value selected by PMCAP1.DTSEL to PMCAP1.DATA. When PMCAP1.DTSEL is modified, write the specified value.
12 to 9	DaTa SELect	H'0	R/W	R/W	These bits specify the value and unit to be indicated to PMCAP1.DATA and PMCAP1.SCALE. Update this value on receiving a configuration write request for this field.
8	PME Enable	0	R/W	R/W	Enables PME message transmission. 1: Enables PME message transmission. 0: Disables PME message transmission. When the device supports the PME message transmit function, the value should be modified by receiving a configuration write request to this field. When the device does not support the PME message transmit function, this bit should always be 0. Note: When PME generation is enabled in the D3Cold state, this field becomes R/WS.
7 to 4	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
3	NO Soft ReSet	0	R	R/W	Indicates whether internal reset is performed or not when the device makes a transition from D3Hot state to D0 state. 1: Internal reset is not performed when the device makes a transition from D3Hot state to D0 state. 0: Internal reset is performed when the device makes a transition from D3Hot state to D0 state. Set the device function at initialization.
2	Reserved	0	R	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	PoWer STATE	00	R/W	R/W	These bits indicate the power state. 00: D0 01: D1 10: D2 11: D3Hot When the value of this field is modified by receiving a configuration write request, an INT_PCIPOWER interrupt is generated. Power-down sequence specified by software should be performed after interrupt detection.

(19) MSI Capability Register 0 (PCIEMSICAP0)

PCIEMSICAP0 configures the MSI capability structure. For details, refer to section 13.5.10, MSI Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	Per Vector MasKing	64bit ADdress CAPable	Multiple MESsage Enable			Multiple MESsage CAPable			MSI Enable
Initial value:	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NEXT CAPability Pointer								CAPability ID							
Initial value:	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 25	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
24	Per Vector MasKing	1	R	R/W	Specifies whether the Per Vector Masking function is supported or not. 1: Supports the Per Vector Masking function. 0: Does not support the Per Vector Masking function. This module supports the Per Vector Masking function.
23	64bit Address CAPable	1	R	R/W	Specifies whether the 64-bit address message transmit function is supported or not. 1: A 64-bit address message can be transmitted. 0: A 64-bit address message cannot be transmitted. This module can transmit a 64-bit address message.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
22 to 20	Multiple MESsage Enable	000	R/W	R/W	Indicates transmittable number of interrupt vectors. 000: 1 vector 001: 2 vectors 010: 4 vectors 011: 8 vectors 100: 16 vectors 101: 32 vectors 110: Reserved 111: Reserved The number of vectors to be assigned to this module is determined and written to this field during a configuration cycle. The maximum number of vectors to be assigned is the number of vectors required by MSICAP0[19:17].Multiple Message Capable; while the minimum number of vectors is 1.
19 to 17	Multiple MESsage CAPable	000	R	R/W	Specifies the number of interrupt vectors to be transferred from the device. 000: 1 vector 001: 2 vectors 010: 4 vectors 011: 8 vectors 100: 16 vectors 101: 32 vectors 110: Reserved 111: Reserved The number of interrupts to be used should be set at initialization. This register is read out during a configuration cycle, and the number of vectors to be assigned to this module is determined and written to MSICAP0[22:20].Multiple Message Enable field.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
16	MSI Enable	0	R/W	R/W	Enables the MSI function. 1: Enables the MSI function. 0: Disable the MSI function. When this bit is cleared to 0, the MSI function is invalid. INTx interrupts should be used. When this bit is set to 1, the MSI function is valid. INTx interrupts cannot be used. The value of this bit is set during a configuration cycle. The value should not be modified by other software.
15 to 8	NEXT CAPability Pointer	H'70	R	R/W	Pointer to the expansion capability list. Indicates the PCI Express Capability pointer H'70.
7 to 0	CAPability ID	H'05	R	R/W	Capability List ID Indicates MSI Capability ID (H'05).

(20) MSI Capability Register 1 (PCIEMSICAP1)

PCIEMSICAP1 configures the MSI capability structure. For details, refer to section 13.5.10, MSI Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Lower MESsage Address															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Lower MESsage Address														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 2	Lower MESsage Address	H'00000000	R/W	R/W	These bits indicate the lower address [31:2] of an MSI message during an MSI transfer. This field value is set during a configuration cycle and it is valid when the MSICAP0[16].MSI Enable is set to 1. When an MSI interrupt occurs, this field value is used as the lower 30 bits of an address. If this device is used as a Root Port, the configuration program should set a conversion register so that the PCI can access the PCIEMSIFR register in this device and set a PCI address to the field in the register corresponding to the Lower or Upper Message Address of an Endpoint to access PCIEMSIFR in this device.
1, 0	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.

(21) MSI Capability Register 2 (PCIEMSICAP2)

PCIEMSICAP2 configures the MSI capability structure. For details, refer to section 13.5.10, MSI Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Upper MESsage Address															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Upper MESsage Address															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	Upper MESsage Address	H'00000000	R/W	R/W	These bits indicate the upper address [63.32] of an MSI message during an MSI transfer. This field value is set during a configuration cycle and it is valid when the MSICAP0[16].MSI Enable is set to 1. When an MSI interrupt occurs, this field value is used as the upper 32 bits of an address. If this device is used as a Root Port, the configuration program should set a conversion register so that the PCI can access the PCIEMSIFR register in this device and set a PCI address to the field in the register corresponding to the Lower or Upper Message Address of an Endpoint to access PCIEMSIFR in this device.

(22) MSI Capability Register 3 (PCIEMSICAP3)

PCIEMSICAP3 configures the MSI capability structure. For details, refer to section 13.5.10, MSI Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MESsage DATA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 16	Reserved	All 0	R	R	Reserved Thess bits are always read as 0. The write value should always be 0.
15 to 0	MESsage DATA	H'0000	R/W	R/W	These bits indicate the MSI message data during an MSI transfer. This field value is set during a configuration cycle and it is valid when the MSICAP0[16].MSI Enable is set to 1. When an MSI interrupt occurs, this field value is used as the lower 16 bits of data. (Upper 16 bits are cleared to 0.) If multiple interrupt vectors have been assigned to this device by the MSICAP0[22:20].Multiple MESsage Enable, transfer data items corresponding to the number of vectors can be modified. If 2^n of vectors are assigned to this device, lower n bits of data indicated by this register can be modified. If this device is used as a Root Port, the configuration program should set the shift amount to Bit[12:8] in the corresponding field of the register in an Endpoint, clear Bit[4:0] to 0, and write 0s to other bits. In this case, shift amount indicates the bit count of the start bit in PCIEMSIFR[31:0] to be assigned to this device. This device can use a maximum of 32 interrupts as MSI interrupts.

(23) MSI Capability Register 4 (PCIEMSICAP4)

PCIEMSICAP4 configures the MSI capability structure. For details, refer to section 13.5.10, MSI Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MESsage MASK															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MESsage MASK															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	MESsage MASK	H'00000000	R/W	R/W	These bits specify whether the MSI transmission is masked or not. Setting 1 to the bit N of this field suppresses the MSI interrupt generation caused by factor N. MESMASK[N] = 1: Masks MSI transmission of interrupt vector [N]. MESMASK[N] = 0: Does not mask MSI transmission of interrupt vector [N].

(24) MSI Capability Register 5 (PCIEMSICAP5)

PCIEMSICAP5 configures the MSI capability structure. For details, refer to section 13.5.10, MSI Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MESsage PENDing															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MESsage PENDing															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	MESsage PENDing	H'00000000	R	R/W	These bits indicate the pending status of an MSI transfer. Setting bit N in this field request the hardware to generate an MSI interrupt whose interrupt source = N. The hardware then generates an MSI interrupt sequentially from smaller source number to larger among the MSI interrupts where corresponding MSICAP4[31:0].MESage MASK bits are cleared to 0. MESMASK[N] = 1: Indicates that the MSI transfer of an interrupt vector [N] is pending. MESMASK[N] = 0: Indicates that the MSI transfer of an interrupt vector [N] is not pending.

(25) PCIE Capability Register 0 (PCIEEXPCAP0)

PCIEEXPCAP0 corresponds to the PCI Express capability register and PCI Express capability list register in the PCI Express capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	INTerrupt MESsage NUMber					SLoT IMPlemented	DEVice PorT TYPE				CAPability VERsion			
Initial value:	0	0	0	0	0	0	0	0	0	1/0	0	0	0	0	0	1
SH-R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NEXT CAPability Pointer								CAPability ID							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31, 30	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
29 to 25	INTerrupt MESsage NUMber	00000	R	R/W	These bits specify an MSI offset used for the HotPlug MSI or PME. An offset to be set should be within the number of interrupt vectors set by MSICAP0.Multiple Message Enable. In the initial status, 0 is used as an offset.
24	SLoT IMPlemented	0	R	R/W	Specifies whether or not link is connected to the PCI-Express slot. 1: Slot is connected. 0: Slot is not connected. Root Port: When the slot is connected, 1 should be set at initialization. Endpoint: This field is invalid. The initial value should not be changed.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
23 to 20	DEVice PorT TYPE	Root Port: H'4 Endpoint: H'0	R	R/W	Specifies the device type. H'0: PCI Express Endpoint (Initial value) H'1: (Legacy PCI Express Endpoint) Setting prohibited H'4: Root Port of PCI Express Root Comple H'5: (Upstream Port of PCI Express Switch) Setting prohibited H'6: (Downstream Port of PCI Express Switch) Setting prohibited H'7: (PCI Express to PCI/PCI-X Bridge) Setting prohibited H'8: (PCI/PCI-X to PCI Express Bridge) Setting prohibited H'9: (Root Complex Integrated Endpoint Device) Setting prohibited H'A: (Root Complex Event Collector) Setting prohibited Root Port: Set H'4 to these bits. Endpoint: Set H'0 to these bits. The value corresponding to the Root Port/Endpoint is set by the mode pin at initialization.
19 to 16	CAPability VERsion	H'1	R	R	These bits indicate the capability version.
15 to 8	NEXT CAPability Pointer	H'00	R	R/W	Pointer to the expansion capability list. Indicates H'00 of End Of List.
7 to 0	CAPability ID	H'10	R	R/W	Capability list ID. Indicates PCI Express Capability ID (H'10).

(26) PCIE Capability Register 1 (PCIEEXPCAP1)

PCIEEXPCAP1 corresponds to device capability register in the PCI Express capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	CAPtured Slot Power Limit Scale		CAPtured Slot Power Limit Value								—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Role-Based Error Reporting	—	—	—	Endpoint L1 Acceptable Latency			Endpoint L0s Acceptable Latency			Extended TAG Field SUPported	Phantom Function SUPporTeD		Max Payload Size SUPporTeD		
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1/0	0/1
SH-R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 28	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
27, 26	CAPtured Slot Power Limit Scale	00	R	R/W	These bits indicate the scale of the slot power limit value. 00: 1.0× 01: 0.1× 10: 0.01× 11: 0.001× Root Port: Setting is not needed. Endpoint: The value specified by the Set_Slot_Power_Limit message is stored from Upstream.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
25 to 18	CAPtured Slot Power Limit Value	H'00	R	R/W	These bits in conjunction with EXPCAP1.CAPSLPLSC indicate the power control limit (watt) supplied from the slot. Root Port: Setting is not needed. Endpoint: The value specified by the Set_Slot_Power_Limit message is stored from Upstream.
17, 16	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
15	Role-Based Error Reporting	1	R	R	This bit is always read as 1.
14 to 12	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 9	Endpoint L1 Acceptable Latency	000	R	R/W	These bits specify the device acceptable transition latency from L1 to L0. If the PCIE is used as an Endpoint, the device support status should be set at initialization. If the PCIE is used as a Root Port, the initial value should not be changed. 000: L1 to L0 transition completes within 1 μs. 001: L1 to L0 transition completes within 2 μs. 010: L1 to L0 transition completes within 4 μs. 011: L1 to L0 transition completes within 8 μs. 100: L1 to L0 transition completes within 16 μs. 101: L1 to L0 transition completes within 32 μs. 110: L1 to L0 transition completes within 64 μs. 111: No restriction This field is used by power management software.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
8 to 6	Endpoint L0s Acceptable Latency	000	R	R/W	These bits specify the device acceptable transition latency from L0s to L0. If the PCIE is used as an Endpoint, the device support status should be set at initialization. If the PCIE is used as a Root Port, the initial value should not be changed. 000: L0s to L0 transition completes within 64 μs. 001: L0s to L0 transition completes within 128 μs. 010: L0s to L0 transition completes within 256 μs. 011: L0s to L0 transition completes within 512 μs. 100: L0s to L0 transition completes within 1 μs. 101: L0s to L0 transition completes within 2 μs. 110: L0s to L0 transition completes within 4 μs. 111: No restriction This field is used by power management software.
5	Extended TAG Field SUPported	0	R	R	Indicates the TAGID size which is supported as a requester ID. 0: 5-bit TAG 1: Extended 8-bit TAG. This module does not support the extended 8-bit TAGID function. Fixed to 0.
4, 3	Phantom Function SUPporTeD	00	R	R	These bits indicate the support status of phantom function which is supported for transaction ID expansion. 00: The phantom function is not supported. 01: The MSB of the requester ID function number can be used for the phantom function. 10: The upper 2 bits of the requester ID function number can be used for the phantom function. 11: All 3 bits of the requester ID function number can be used for the phantom function. This module does not support the phantom function. These bits are fixed to 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
2 to 0	Max Payload Size SUPporTeD	PCIEC0: 010 PCIEC1/ PCIEC2: 001	R	R/W	These bits indicate the maximum payload size supported by the device. 000: 128 bytes 001: 256 bytes 010: 512 bytes 011: 1024 bytes 100: 2048 bytes 101: 4096 bytes 110: Reserved 111: Reserved This device supports the following payload sizes: PCIEC0: 1024 bytes PCIEC1: 512 bytes PCIEC2: 512 bytes The initial value should not be changed.

(27) PCIE Capability Register 2 (PCIEEXPCAP2)

PCIEEXPCAP2 corresponds to the device capability register and device control register in the PCI Express capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	Transaction PenDing	AUX Power DeTeCteD	Unsupported Request DeTeCteD	Fatal Error DeTeCteD	Non Fatal Error DeTeCteD	Correctable Error DeTeCteD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W1C	R/W1C	R/W1C	R/W1C
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W1C	R/W1C	R/W1C	R/W1C

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Bridge Configuration Retry Enable	Max Read REQuest SiZE			ENable No SNooP	AUX Power PM Enable	Phantom Function Enable	Extended TAG Enable	Max Payload Size			ENabled ReLax Odering	Unsupported Request RePorting Enable	Fatal Error RePorting Enable	Non Fatal Error RePorting Enable	Correctable Error RePorting Enable
Initial value:	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
SH-R/W:	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 22	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
21	Transaction PenDing	0	R	R	Indicates that there are non posted requests without completion. 1: Non posted request processing is pending. 0: Non posted request processing is not pending.
20	AUX Power DeTeCteD	0	R	R/W	Indicates whether an AUX Power is detected or not. 1: An AUX Power is detected. 0: An AUX Power is not detected. The device detection status should be reflected.
19	Unsupported Request DeTeCteD	0	R/W1C	R/W1C	Indicates whether an Unsupported Request is received or not. 1: An Unsupported Request is received. 0: An Unsupported Request is not received. 1 is set when an Unsupported Request is received.
18	Fatal Error DeTeCteD	0	R/W1C	R/W1C	Indicates whether a Fatal Error is detected or not. 1: A Fatal Error is detected. 0: A Fatal Error is not detected. 1 is set when A Fatal Error is detected.
17	Non Fatal Error DeTeCteD	0	R/W1C	R/W1C	Indicates whether a Non Fatal Error is detected or not. 1: A Non Fatal Error is detected. 0: A Non Fatal Error is not detected. 1 is set when a Non Fatal Error is detected.
16	Correctable Error DeTeCteD	0	R/W1C	R/W1C	Indicates whether a Correctable Error is detected or not. 1: A Correctable Error is detected. 0: A Correctable Error is not detected. 1 is set when a Correctable Error is detected.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
15	Bridge Configuration Retry Enable	0	R	R	Fixed to 0. This field is valid only for the PCI Express/PCI and PCI-X bridge device.
14 to 12	Max Read REQuest SiZE	000	R/W	R/W	These bits specify the maximum payload size when a read request is issued. 000: Max. 128 bytes (Initial value) 001: Max. 256 bytes 010: Max. 512 bytes 011: Max. 1024 bytes 100: Max. 2048 bytes 101: Max. 4096 bytes 110: Reserved 111: Reserved
11	ENable No SNooP	1	R/W	R/W	Enables or disables issuance of a No Snoop attribute transaction. 0: Disables issuance of a No Snoop attribute transaction. 1: Enables issuance of a No Snoop attribute transaction. When this bit is cleared to 0, a No Snoop transaction cannot be issued.
10	AUX Power PM Enable	0	R/W	R/W	Enables or disables an AUX Power. 1: Enables an AUX Power. 0: Disables an AUX Power.
9	Phantom Function Enable	0	R	R	Enables or disables the phantom function. 1: Enables the phantom function. 0: Disables the phantom function. This bit is always cleared to 0 because this module does not support the phantom function.
8	Extended TAG Enable	0	R	R	Enables or disables the extended 8-bit TAGID function. 0: Disables the extended 8-bit TAGID function. 1: Enables the extended 8-bit TAGID function. This bit is always cleared to 0 because this module does not support the extended 8-bit TAGID function.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
7 to 5	Max Payload Size	000	R/W	R/W	These bits indicate the maximum payload size which can be used by the device. 000: 128 bytes 001: 256 bytes 010: 512 bytes 011: 1024 bytes 100: 2048 bytes 101: 4096 bytes 110: Reserved 111: Reserved The minimum value of the maximum payload size of the device in the system is set during a configuration cycle.
4	ENabled ReLax Odering	1	R/W	R/W	Enables or disables issuance of a Relaxed Ordering attribute transaction of the device. 1: Enables issuance of a Relaxed Ordering attribute transaction. 0: Disables issuance of a Relaxed Ordering attribute transaction. When this bit is cleared to 0, a Relaxed Ordering transaction cannot be issued.
3	Unsupported Request RePorting Enable	0	R/W	R/W	Specifies an error report when an Unsupported Request is detected. Root Port: When this bit is set to 1, a Fatal/Nonfatal error interrupt is generated by detecting an Unsupported Request. Endpoint: When this bit is set to 1, a Fatal/Nonfatal error message is transmitted by detecting an Unsupported Request. To generate an interrupt or a transmit message, other registers should also be set.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
2	Fatal Error RePorting Enable	0	R/W	R/W	Enables a Fatal Error report. Root Port: When this bit is set to 1, an interrupt is generated by detecting a Fatal Error. Endpoint: When this bit is set to 1, an error message is transmitted by detecting a Fatal Error.
1	Non Fatal Error RePorting Enable	0	R/W	R/W	Enables Non Fatal Error report. Root Port: When this bit is set to 1, an interrupt is generated by detecting a Non Fatal Error. Endpoint: When this bit is set to 1, an error message is transmitted by detecting a Non Fatal Error.
0	Correctable Error RePorting Enable	0	R/W	R/W	Enables a Correctable Error report. Root Port: When this bit is set to 1, an interrupt is generated by detecting a Correctable Error. Endpoint: When this bit is set to 1, an error message is transmitted by detecting a Correctable Error.

(28) PCIE Capability Register 3 (PCIEEXPCAP3)

PCIEEXPCAP3 corresponds to link capability register in the PCI Express capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Port NUMber								—	—	—	Data Link Layer ACTive RePorting CAPable	Supprize DowN Error RePorting CaPable	CLocK Power Management	L1 Exit LATency	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L1 Exit LATency	L0s Exit LATency			ASPM SuPported		MAXimum LinK WiDth						MAXimum LinK SPEED			
Initial value:	0	0	0	0	0	1	0	0	0	1/0	0	0/1	0	0	0	1
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	Port NUMber	H'00	R	R/W	These bits indicate the PCI Express Port number. The Port number should be set at initialization.
23 to 21	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
20	Data Link Layer ACTive RePorting CAPable	0	R	R/W	Indicates whether or not the Data Link Layer ACTive State can be indicated by EXPCAP4.DLLACT. 1: Data Link Layer Active State can be indicated. 0: Data Link Layer Active State cannot be indicated. Root Port: To support this function, set this bit to 1 at initialization. Endpoint: The initial value should not be changed.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
19	Supprize DowN Error RePorting CaPable	0	R	R	Indicates whether the Surprise Down error detection function is supported. 1: The Surprise Down error detection function supported. 0: The Surprise Down error detection function not supported. This device does not support the Surprise Down error function. Fixed to 0.
18	CLocK Power Management	0	R	R	Enables or disables clock stop when the link is in the L1, L2Ready, L3Ready state. 1: Enables clock stop when the link is in the L1, L2Ready, L3Ready state. 0: Disables clock stop when the link is in the L1, L2Ready, L3Ready state. Set the device function at initialization. Note: This field can be set with device which conforms to form factor supporting Clock Request Capability.
17 to 15	L1 Exit LATency	100	R	R	These bits specify the transition latency from L1 to L0 of the device. 000: L1 to L0 transition completes within the time less than 1 μs. 001: L1 to L0 transition completes within the time between 1 us or more and less than 2 μs. 010: L1 to L0 transition completes within the time between 2 us or more and less than 4 μs. 011: L1 to L0 transition completes within the time between 4 us or more and less than 8 μs. 100: L1 to L0 transition completes within the time between 8 us or more and less than 16 μs. 101: L1 to L0 transition completes within the time between 16 us or more and less than 32 μs. 110: L1 to L0 transition completes within the time between 32 μs or more and 64 μs or less. 111: L1 to L0 transition completes within the time exceeding 64 μs. L1 to L0 transition in this module requires 10.9 μs or less.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
14 to 12	L0s Exit LATency	000	R	R	These bits specify the transition latency from L0s to L0 of the device. 000: L0s to L0 transition completes within the time less than 64 μs. 001: L0s to L0 transition completes within the time between 64 μs or more and less than 128 μs. 010: L0s to L0 transition completes within the time between 128 μs or more and less than 256 μs. 011: L0s to L0 transition completes within the time between 256 μs or more and less than 512 μs. 100: L0s to L0 transition completes within the time between 512 μs or more and less than 1 μs. 101: L0s to L0 transition completes within the time between 1 μs or more and less than 2 μs. 110: L0s to L0 transition completes within the time between 2 μs or more and 4 μs or less. 111: L0s to L0 transition completes within the time exceeding 4 μs. L0s to L0 transition in this module requires less than 64 μs.
11, 10	ASPM SuPported	01	R	R	These bits indicate the ASPM support status. 00: Reserved 01: Supports L0s transition. 10: Reserved 11: Reserved This module supports the L0s transition.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
9 to 4	MAXimum LinK WiDth	PCIEC0: 000100 PCIEC1/ PCIEC2: 000001	R	R	These bits indicate the maximum link width. 000000: Reserved 000001: ×1 000010: ×2 000100: ×4 001000: ×8 001100: ×12 010000: ×16 100000: ×32 The maximum link width of this module is as follows: PCIEC0: ×4 PCIEC1/PCIEC2: ×1
3 to 0	MAXimum LinK SPEED	H'1	R	R	These bits indicate the maximum link speed. H'1: 2.5 Gb/s Other than above: Reserved Maximum speed is 2.5 Gb/s.

(29) PCIE Capability Register 4 (PCIEEXPCAP4)

PCIEEXPCAP4 corresponds to the link status register and link control register in the PCI Express capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	Data Link Layer ACTive	Slot CLocK ConFiG	LinK Training	—	NeGotiated LinK WIDTH						LinK SPPED			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
SH-R/W:	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	Enable CLocK Power Management	Extended Sync	Common CLocK ConFiG	ReTRain LinK	LinK DISable	Read Completion Boundary	—	ASPM ConTRroL	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1/0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/ R/W	R	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/ R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31, 30	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
29	Data Link Layer ACTive	0	R	R	Indicates whether or not DLCMSM is in the DL_ACTIVE state. This field is valid only when EXCAP3[20].Data Link Layer ACTive RePorting CAPable is set to 1. 1: DLCMSM is in the DL_ACTIVE state. 0: DLCMSM is not in the DL_ACTIVE state.
28	SLot CLocK ConFiG	0	R	R/W	Indicates whether the same reference clock as the clock which is supplied from the platform to the connector is used. If the same reference clock as the clock to be supplied to the connector is used, set this field to 1. 1: The same reference clock is used. 0: The same reference clock is not used.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
27	LinK Training	0	R	R	Indicates that the MAC LTSSM is being link training.
26	Reserved	0	R	R	Reserved This bit is always read as 0. The write value should always be 0.
25 to 20	NeGotiated LinK WIDTH	000001	R	R	Indicates link width adjusted between links. 000001: ×1 000010: ×2 000100: ×4 001000: ×8 001100: ×12 010000: ×16 100000: ×32
19 to 16	LinK SPPED	H'1	R	R	Indicates the link speed adjusted between links. Only 2.5 Gb/s (H'1) is valid.
15 to 9	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ENable CLocK Power Management	0	R	R	Enables the Clock Power Management function. The value should be modified when a configuration write request to this field is received. Note: This field is valid only when EXCAP3.Clock Power Management is set to 1. When EXCAP3.Clock Power Management is set to 0, this field should be fixed to 0.
7	Extended Sync	0	R/W	R/W	Specifies the extended sync. Setting this bit to 1 transmits additional transmit order set when the MAC LTSSM makes a transition from L0s/Recovery state.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
6	Common CLocK ConFiG	0	R/W	R/W	Specifies that the common clock is used for both link edges. If the same reference clock is used at both link edges, this field should be set to 1. The value of this field is used for calculation of transition latency from L0s/L1. When the value of this field is modified, the link retraining (writing 1 to EXPCAP4[5].Retrain Link) should then be performed by software.
5	ReTRain LinK	0	R/W	R/W	Performs link retraining. Writing 1 to this field performs link training again. This bit is always read as 0. This field is valid only when this module is used as a Root Port.
4	LinK DISable	0	R/W	R/W	Disables link. Writing 1 to this field disables link irrespective of the link status. This field is valid only when this module is used as a Root Port.
3	Read Completion Boundary	Root Port: 1 End-point: 0	Root Port: R End-point: R/W	Root Port: R End-point: R/W	Indicates the read completion boundary. 0: 64-byte boundary 1: 128-byte boundary Root Port: 1 which indicates a 128-byte boundary is set. Endpoint: The read completion boundary supported by the Root Port is set at initialization.
2	Reserved	0	R	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	ASPM ConTRroL	00	R/W	R/W	These bits specify the ASPM support level. The ASPM level should be specified. 00: Disabled 01: L0s transition enabled. 10: L1 transition enabled (Setting prohibited) 11: L0s/L1 transition enabled (Setting prohibited)

(30) PCIE Capability Register 5 (PCIEEXPCAP5)

PCIEEXPCAP5 corresponds to the slot capability register in the PCI Express capability structure. This module does not support the slot capability register function.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PHYsical SLot NUMber													NO Command Complete SuPporT	ELEctro-mechanical INTLOCK PReSeNT	SLot Power Limit Scale
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SLot Power Limit Scale	SLot Power Limit Value								HOT PLuG CAPable	HOT PLuG SurPRise	PoWer INDicator PReSent	ATtention INDicator PReSent	MRL SENsor PReSent	PoWer CONtroller PReSent	ATtention BuTtoN PReSent
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 19	PHYsical SLot NUMber	H'0000	R	R/W	These bits indicate a physical slot number to be connected to the device. Device status should be set at initialization.
18	NO Command Complete SuPporT	0	R	R/W	Indicates whether or not the completion is notified when the EXPCAP6 register completes the HOT Plug command execution. 1: The completion is notified at completion of HOT Plug command. 0: The completion is not notified at completion of HOT Plug command. Device state should be set at initialization.
17	ELEctromechanical INTLOCK PReSeNT	0	R	R/W	Specifies whether the electromechanical interlock exists or not. If the electromechanical interlock for the slot exists on the chassis, set 1 to this field at initialization.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
16, 15	SLot Power Limit Scale	00	R	R/W	These bits indicate the slot power limit scale. 00: 1.0× 01: 0.1× 10: 0.01× 11: 0.001× A configuration write request to this field outputs a Set_Slot_Power_Limit message. This field is valid when EXPCAP0[24].SLoT IMPlemented is set to1. Device state should be set at initialization.
14 to 7	SLot Power Limit Value	H'00	R	R/W	These bits indicate the upper limit of slot power to be supplied by slot power supply (watt). A configuration write request to this field outputs a Set_Slot_Power_Limit message. This field is valid when EXPCAP0[24].SLoT IMPlemented is set to1. Set device state at initialization.
6	HOT PLuG CAPable	0	R	R/W	Specifies whether Hot Plug for this slot is supported or not. To support Hot Plug, set 1 to this field at initialization.
5	HOT PLuG SurPRise	0	R	R/W	Indicates whether the Hot Plug Surprise is supported or not. If an adaptor to be inserted in this slot can be removed without notice, set this field to 1 at initialization.
4	PoWer INDicator PReSent	0	R	R/W	Indicates whether the power indicator is supported or not. If the power indicator for this slot exists on the chassis, set this field to 1 at initialization.
3	ATtention INDicator PReSent	0	R	R/W	Indicates whether the attention indicator is supported or not. If the attention indicator for this slot exists on the chassis, set this field to 1 at initialization.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
2	MRL SENsor PReSent	0	R	R/W	Indicates whether the MRL sensor is supported or not. If the MRL sensor for this slot exists on the chassis, set this field to 1 at initialization.
1	PoWer CONtroller PReSent	0	R	R/W	Indicates whether the power controller is supported or not. If the power controller for this slot exists on the chassis, set this field to 1 at initialization.
0	ATtention BuTtoN PReSent	0	R	R/W	Indicates whether the attention button is supported or not. If the attention button for this slot exists on the chassis, set this field to 1 at initialization.

(31) PCIE Capability Register 6 (PCIEEXPCAP6)

PCIEEXPCAP6 corresponds to the slot status register and slot control register in the PCI Express capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	Data Link Layer STate CHanGed	ELectro MEChanical interlock STatuS	PResence DETect State	MRL SENsor State	COMmand ComPLeted	PResence DETect CHanGed	MRL SENsor CHanGed	PoWer FAuLt DETect	ATtension BuTton PReSsed
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R/W1C	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R/W1C	R	R	R	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	Data Link Layer STate CHanGe Enable	ELectro MECHanical interlock CONtrol	PoWer CONtroller CONtrol	PoWer INDicator CONtrol		attention INDicator CONtrol		HOT PLuG INTerrupt Enable	COMmand COMpleted INTerrupt Enable	PResence DETect CHanGed Enable	MRL SENsor CHanGed Enable	PoWer FAuLt DETect Enable	ATtension BuTton PReSsed ENable
Initial value:	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
SH-R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 25	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
24	Data Link Layer ST CHanGed	0	R/W1C	R/W1C	Indicates that EXPCAP4.DLLACT changes. When EXPCAP4.DLLACT is modified, 1 is set.
23	ELectro MEChanical interlock STatuS	0	R	R/W	Specifies the Electromechanical Interlock state. 1: The Electromechanical Interlock is locked. 0: The Electromechanical Interlock is not locked. The Electromechanical Interlock state should be reflected.
22	PResence DETect State	0	R	R/W	Specifies whether the adaptor is inserted into the slot or not. 1: Adaptor is inserted into the slot. 0: The slot is empty. The slot state should be reflected.
21	MRL SENsor State	0	R	R/W	Indicates the MRL sensor state. 1: The MRL is closed. 0: The MRL is open. The MRL sensor state should be reflected.
20	COMmand ComPLeted	0	R/W1C	R/W	Indicates that HOTPLUG command completion when EXPCAP5.NOCCSP[18].NO Command Complete SuPporT = 0. 1: HOTPLUG command execution completes. 0: HOTPLUG command execution is in progress. This bit should be set to 1 when the HOTPLUG command execution completes.
19	PResence DETect CHanGed	0	R/W1C	R/W	Indicates that EXPCAP6[22].Presence Detect State changes. This bit is set to 1 when PResence DETect State changes, and cleared by writing 1 to this field.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
18	MRL SENsor CHanGed	0	R/W1C	R/W	Indicates that EXPCAP6[21].MRL Sensor State changes. This bit is set to 1 when MRL SENsor STate changes, and cleared by writing 1 to this field.
17	PoWer FAuLt DETect	0	R/W1C	R/W	Indicates that Power Fault is detected. This bit should be set to 1 when Power Fault is detected, and cleared to 0 when configuration write request to write 1 to this field is received.
16	ATtension BuTton PReSsed	0	R/W1C	R/W	Indicates that attention button is pressed. This bit should be set to 1 when attention button is pressed, and cleared to 0 when configuration write request to write 1 to this field is received.
15 to 13	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
12	Data Link Layer STate CHanGe Enable	0	R/W	R/W	Enables to set PCIEXP6.DLLSTCHG.
11	ELectro MECHanical interlock CONtrol	0	R/W	R/W	Toggles the Electromechanical InterLock state.
10	PoWer CONtroller CONtrol	0	R/W	R/W	Controls the Power controller. 1: Power On 0: Power Off
9, 8	PoWer INDicator CONtrol	11	R/W	R/W	Controls the Power Indicator. 00: Reserved 01: ON 10: Blink 11: OFF
7, 6	attention INDicator CONtrol	11	R/W	R/W	Controls the Attention Indicator. 00: Reserved 01: ON 10: Blink 11: OFF

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
5	HOT PLuG INTerrupt Enable	0	R/W	R/W	Enables HOTPLUG interrupt generation.
4	COMmand COMpleted INTerrupt Enable	0	R/W	R/W	Enables the generation of a Command Complete interrupt. Generates a Command Complete interrupt.
3	PResence DETect CHanGed Enable	0	R/W	R/W	Enables the generation of a PResence DETect CHanGed interrupt. A PResence DETect CHanGed interrupt is generated when EXPCAP6[5].Hot Plug Interrupt Enable is 1 and this field is 1.
2	MRL SENsor CHanGed Enable	0	R/W	R/W	Enables generation of a MRL SENsor CHanGed interrupt. An MRL SENsor CHanGed interrupt is generated when EXPCAP6[5].Hot Plug Interrupt Enable is 1 and this field is 1.
1	PoWer FAuLt DETect Enable	0	R/W	R/W	Enables generation of a PoWer FAuLt DETect interrupt. A PoWer FAuLt DETect interrupt is generated when EXPCAP6[5].Hot Plug Interrupt Enable is 1 and this field is 1.
0	ATtension BuTton PReSsed ENable	0	R/W	R/W	Enables generation of an ATtension BuTton PReSsed interrupt. An ATtension BuTton PReSsed interrupt is generated when EXPCAP6[5].Hot Plug Interrupt Enable is 1 and this field is 1.

(32) PCIE Capability Register 7 (PCIEEXPCAP7)

PCIEEXPCAP7 corresponds to the root capability register and root control register in the PCI Express capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CRS software VISibility
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	CRS software VISibility Enable	PME INTerrupt Enable	System ERRor on Fatal Error Enable	System ERRor on Non Fatal Error Enable	System ERRor on Correctable Error Enable
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 17	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
16	CRS software VISibility	1	R	R	Indicates whether the function to notify software of CRS (Configuration Retry Status) reception is supported or not. This module has a mechanism of notifying software of CRS reception. This field has meaning only when this module is used as a Root Port.
15 to 5	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
4	CRS software VISibility Enable	0	R/W	R/W	Asserts a System Error when the CRS (Configuration Retry Status) is received. When this bit is set to 1, INT_PCIESERR is asserted when CRS is received.
3	PME INTerrupt Enable	0	R/W	R/W	Enables interrupt generation for the INTC when a PME message is received. This bit should be set to 1 to generate a PME message interrupt.
2	System ERRor on Fatal Error ENable	0	R/W	R/W	Enables interrupt generation for the INTC through Fatal Error detection. An error is detected in a Root Port or by receiving an error message. This field is valid only when this module is used as a Root Port.
1	System ERRor on Non Fatal Error Enable	0	R/W	R/W	Enables interrupt generation for the INTC through NonFatal Error detection. An error is detected in a Root Port or by receiving error message. This field is valid only when this module is used as a Root Port.
0	System ERRor on Correctable Error Enable	0	R/W	R/W	Enables interrupt generation for the INTC through Correctable Error detection. An error is detected in a Root Port or by receiving error message. This field is valid only when this module is used as a Root Port.

(33) PCIE Capability Register 8 (PCIEEXPCAP8)

PCIEEXPCAP8 corresponds to the root status register in the PCI Express capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PME PenDing	PME Status
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W1C

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PME REQuester ID															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 18	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
17	PME PenDing	0	R	R/W	Indicates that other PME is pending while Status = 1. This bit should be set to 1 if a PME is received while PME Status = 1. If the PME Status is cleared by the upper software while PME PenDing = 1 and PME Status = 1, the PME Status should be set again and the requester ID which is pending should be set to the PME REQuester ID.
16	PME Status	0	R/W1C	R/W (R/W1C)	Indicates that a PME has been received from the requester indicated by PMEREQID. This bit should be set to 1 by the PME reception. This bit should be cleared to 0 if 1 is written to this bit by the upper software.
15 to 0	PME REQuester ID	H'0000	R	R/W	These bits indicate the requester ID which has issued a PME. When a PME is received, a requester ID should be set in this field.

(34) VC Capability Register 0 (PCIEVCCAP0)

PCIEVCCAP0 corresponds to the virtual channel expansion capability header in the virtual channel capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NEXT CAPability												CAPability VERsion			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CAPability ID															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 20	NEXT CAPability	H'000	R	R/W	Pointer to Capability List. These bits indicate 0 which stands for the end of the list. When Device Serial Number Capability is used, set H'1B0 to these bits.
19 to 16	CAPability VERsion	H'1	R	R	These bits indicate the Capability version.
15 to 0	CAPability ID	H'0002	R	R	These bits indicate the Capability ID.

(35) VC Capability Register 1 (PCIEVCCAP1)

PCIEVCCAP1 corresponds to the port VC capability register 1 in the virtual channel capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PorT ARBitoration TaBLe entry SIZE		REFerence CLocK		—	Low Priority EXTended VC count			—	EXTended VC count		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 12	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
11, 10	PorT ARBitoration TaBLe entry SIZE	00	R	R	These bits indicate the entry size of the port arbitration table. This module does not support port arbitration.
9, 8	REFerence CLocK	00	R	R	These bits indicate ReferenceClock in the TimeBase WRR port arbitration. This module does not support port arbitration.
7	Reserved	0	R	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	Low Priority EXTended VC count	000	R	R	These bits indicate the Low PriorityVC count other than default VC0. This module supports only VC0 as Low Priority VC.
3	Reserved	0	R	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
2 to 0	EXTended VC count	001	R	R	These bits indicate the VC count other than default VC0. Though this module has registers corresponding to VC0 and VCX, only VC0 is supported (VCX cannot be used).

(36) VC Capability Register 2 (PCIEVCCAP2)

PCIEVCCAP2 corresponds to the port VC capability register 2 in the virtual channel capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC ARBitration TaBLe OffSET								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	VC ARBitration WRR 128 phase	VC ARBitration WRR 64 phase	VC ARBitration WRR 32phase	VC ARBitration hardware FIXed
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	VC ARBitration TaBLe OffSET	H'00	R	R	These bits indicate the VC arbitration position. This IP does not perform the VC arbitration by means of a table.
23 to 4	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
3	VC ARBitration WRR 128 phase	0	R	R	Indicates that the VC arbitration for Low Priority Group VC is performed by means of Weighted Round Robin 128 Phase. (Not supported)

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
2	VC ARBitration WRR 64 phase	0	R	R	Indicates that the VC arbitration for Low Priority Group VC is performed by means of Weighted Round Robin 64 Phase. (Not supported)
1	VC ARBitration WRR 32phase	0	R	R	Indicates that the VC arbitration for Low Priority Group VC is performed by means of Weighted Round Robin 32 Phase. (Not supported)
0	VC ARBitration hardware FIXed	0	R	R	Indicates that the VC arbitration for Low Priority Group VC is performed by means of HardWareFix method. (Not supported)

(37) VC Capability Register 3 (PCIEVCCAP3)

PCIEVCCAP3 corresponds to the port VC status register and port VC control register in the virtual channel capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VC ARBitration TaBLe Status
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	VC ARBitration SELect			Load VC ARBitration TaBLe
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 17	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
16	VC ARBitration TaBLe Status	0	R	R	Indicates the VC arbitration table status. (Not supported)
15 to 4	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 1	VC ARBitration SELect	000	R	R	These bits select a VC arbitration method for Low Priority Group VC. (Not supported)
0	Load VC ARBitration TaBLe	0	R	R	Loads the VC arbitration table. (Not supported)

(38) VC Capability Register 4 (PCIEVCCAP4)

PCIEVCCAP4 corresponds to the VC resource capability register in the virtual channel capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PorT ARBitration TaBLe OffSET								—	MAXimum TiME SLOT						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ReJeCT Snoop Transactions	—	—	—	—	—	—	—	—	—	PorT ARBitration WRR256	PorT ARBitration Time Base WRR128	PorT ARBitration WRR128	PorT ARBitration WRR64	PorT ARBitration WRR32	PorT ARBitration hardware FIXed
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	PorT ARBitration TaBLe OffSET	H'00	R	R	These bits indicate an offset address to the port arbitration table. VC0 does not have the port arbitration table. (Not supported)
23	Reserved	0	R	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 16	MAXimum TiME SLOT	H'00	R	R	Indicates the maximum time slot during Time Based WRR port arbitration. VC0 does not perform WRR port arbitration. (Not supported)

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
15	ReJeCT SNoop Transactions	0	R	R/W	Indicates rejection of SNOOP transaction. When this bit is set to 1, TLP without No Snoop attribute is handled as Unsupported Request. This is valid only for a Root Port. Root Port: VC0 state should be set at initialization. Endpoint: This bit is invalid. The write value should always be 0.
14 to 6	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PorT ARBitration WRR256	0	R	R	Indicates that port arbitration is performed by means of Weighted Round Robin 256-phase. VC0 does not support port arbitration.
4	PorT ARBitration Time Base WRR128	0	R	R	Indicates that port arbitration is performed by means of Time Base Weighted Round Robin 128-phase. VC0 does not support port arbitration.
3	PorT ARBitration WRR128	0	R	R	Indicates that port arbitration is performed by means of Weighted Round Robin 128-phase. VC0 does not support port arbitration.
2	PorT ARBitration WRR64	0	R	R	Indicates that port arbitration is performed by means of Weighted Round Robin 64-phase. VC0 does not support port arbitration.
1	PorT ARBitration WRR32	0	R	R	Indicates that port arbitration is by means of Weighted Round Robin 32-phase. VC0 does not support port arbitration.
0	PorT ARBitration hardware FIXed	0	R	R	Indicates that port arbitration is performed by means of HardWareFix method. VC0 does not support port arbitration.

(39) VC Capability Register 5 (PCIEVCCAP5)

PCIEVCCAP5 corresponds to the VC resource control register in the virtual channel capability structure, and controls VC0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC Enable	—	—	—	—	VCID			—	—	—	—	PorT ARBitration SELect			Load Port ARBitration TaBLe
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
SH-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
PCI-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	VC ENable	1	R	R	Indicates whether or not VC is enabled. VC0 is always enabled.
30 to 27	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 24	VCID	000	R	R	These bits indicate the VC ID. VC0 is always 0.
23 to 20	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 17	PorT ARBitration SELect	000	R	R	These bits select the port arbitration method. This device does not perform VC arbitration. (Not supported)
16	Load Port ARBitration TaBLe	0	R	R	Loads a port arbitration table. This device does not perform VC arbitration. (Not supported)

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
15 to 8	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TC7	1	R/W	R/W	Maps TC7 to VC0. In the initial state, all TCs are mapped to VC0.
6	TC6	1	R/W	R/W	Maps TC6 to VC0. In the initial state, all TCs are mapped to VC0.
5	TC5	1	R/W	R/W	Maps TC5 to VC0. In the initial state, all TCs are mapped to VC0.
4	TC4	1	R/W	R/W	Maps TC4 to VC0. In the initial state, all TCs are mapped to VC0.
3	TC3	1	R/W	R/W	Maps TC3 to VC0. In the initial state, all TCs are mapped to VC0.
2	TC2	1	R/W	R/W	Maps TC2 to VC0. In the initial state, all TCs are mapped to VC0.
1	TC1	1	R/W	R/W	Maps TC1 to VC0. In the initial state, all TCs are mapped to VC0.
0	TC0	1	R	R	Maps TC0 to VC0. TC0 is always mapped to VC0.

(40) VC Capability Register 6 (PCIEVCCAP6)

PCIEVCCAP6 corresponds to the VC resource status register in the virtual channel capability structure.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VC NeGotiation PenDing	PorT ARBitration TaBLe Status
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 18	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
17	VC NeGotiation PenDing	1	R	R	Indicates that VC0 Negotiation is in progress.
16	PorT ARBitration TaBLe Status	0	R	R	Indicates the port arbitration table status. This module does not support port arbitration by means of table.
15 to 0	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.

(41) VC Capability Register 7 (PCIEVCCAP7)

PCIEVCCAP7 corresponds to the port arbitration table in the virtual channel capability structure, and sets VCX.

Since this module does not support VCX, this register is invalid. Do not write any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PorT ARBitration TaBLe OffSET								—	MAXimum TiMe SLOT						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ReJeCT Snoop Transac-tions	—	—	—	—	—	—	—	—	—	PorT ARBitration WRR256	PorT ARBitration Time Base WRR128	PorT ARBitration WRR128	PorT ARBitration WRR64	PorT ARBitration WRR32	PorT ARBitration hardware FIXed
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	PorT ARBitration TaBLe OffSET	H'00	R	R	These bits indicate an offset address to the port arbitration table. VCX does not have port arbitration table. (Not supported)
23	Reserved	0	R	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 16	MAXimum TiMe SLOT	H'00	R	R	Indicates the maximum time slot during Time Based WRR port arbitration. VCX does not perform WRR port arbitration. (Not supported)

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
15	ReJeCT SNoop Transactions	0	R	R/W	Indicates rejection of SNOOP transaction. When this bit is set to 1, TLP without No Snoop attribute is handled as Unsupported Request. Valid only for a Root Port. Root Port: Set VCX state at initialization. Endpoint: This bit is invalid. The write value should always be 0.
14 to 6	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PorT ARBitration WRR256	0	R	R	Indicates that port arbitration is performed by means of Weighted Round Robin 256-phase. VCX does not support port arbitration.
4	PorT ARBitration Time Base WRR128	0	R	R	Indicates that port arbitration is performed by means of Time Base Weighted Round Robin 128-phase. VCX does not support port arbitration.
3	PorT ARBitration WRR128	0	R	R	Indicates that port arbitration is performed by means of Weighted Round Robin 128-phase. VCX does not support port arbitration.
2	PorT ARBitration WRR64	0	R	R	Indicates that port arbitration is performed by means of Weighted Round Robin 64-phase. VCX does not support port arbitration.
1	PorT ARBitration WRR32	0	R	R	Indicates that port arbitration is performed by means of Weighted Round Robin 32-phase. VCX does not support port arbitration.
0	PorT ARBitration hardware FIXed	0	R	R	Indicates that port arbitration is performed by means of HardWareFix method. VCX does not support port arbitration.

(42) VC Capability Register 8 (PCIEVCCAP8)

PCIEVCCAP8 corresponds to the VC resource control register in the virtual channel capability structure, and controls VCX.

Since this module does not support VCX, this register is invalid. Do not write any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC Enable	—	—	—	—	VCID			—	—	—	—	PorT ARBitration SELect			Load Port ARBitration TaBLe
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
PCI-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	VC ENable	0	R	R	Indicates whether or not VCX is enabled. In the initial status, VCX is not enabled. Since VCX cannot be used in this module, this bit cannot be set to 1.
30 to 27	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 24	VCID	000	R/W	R/W	These bits indicate the VC ID. VCX ID should be set.
23 to 20	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 17	PorT ARBitration SELect	000	R	R	These bits select the port arbitration method. This device does not perform VC arbitration. (Not supported)

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
16	Load Port ARBitration TaBLe	0	R	R	Loads a port arbitration table. This device does not perform VC arbitration. (Not supported)
15 to 8	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TC7	0	R/W	R/W	Maps TC7 to VCX. In the initial status, all TCs are mapped to VC0.
6	TC6	0	R/W	R/W	Maps TC6 to VCX. In the initial status, all TCs are mapped to VC0.
5	TC5	0	R/W	R/W	Maps TC5 to VCX. In the initial status, all TCs are mapped to VC0.
4	TC4	0	R/W	R/W	Maps TC4 to VCX. In the initial status, all TCs are mapped to VC0.
3	TC3	0	R/W	R/W	Maps TC3 to VCX. In the initial status, all TCs are mapped to VC0.
2	TC2	0	R/W	R/W	Maps TC2 to VCX. In the initial status, all TCs are mapped to VC0.
1	TC1	0	R/W	R/W	Maps TC1 to VCX. In the initial status, all TCs are mapped to VC0.
0	TC0	0	R	R	Maps TC0 to VCX. TC0 is always mapped to VC0.

(43) VC Capability Register 9 (PCIEVCCAP9)

PCIEVCCAP9 corresponds to the port arbitration table in the virtual channel capability structure, and sets VCX.

Since this module does not support VCX, this register is invalid. Do not write any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VC NeGotiation PenDing	PorT ARBitration TaBLe Status
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 18	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.
17	VC NeGotiation PenDing	1	R	R	Indicates that VCX Negotiation is in progress.
16	PorT ARBitration TaBLe Status	0	R	R	Indicates the port arbitration table status. This module does not support VC arbitration by means of table. (Not supported)
15 to 0	Reserved	All 0	R	R	Reserved These bits are always read as 0. The write value should always be 0.

(44) Device Serial Number Capability Register 0 (PCIENUMCAP0)

PCIENUMCAP0 corresponds to the expansion capability header in the device serial number capability structure. This device supports the device serial number capability structure but does not assign a serial number. In order to use the device serial number capability structure, the serial number needs to be assigned by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NEXT CAPPability												CAPability VERsion			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CAPability ID															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 20	NEXT CAPPability	H'000	R	R/W	Pointer to the capability list. These bits indicate H'000 of End Of List.
19 to 16	CAPability VERsion	H'1	R	R	These bits indicate the Capability version.
15 to 0	CAPability ID	H'0003	R	R	These bits indicate the Capability ID.

(45) Device Serial Number Capability Register 1 (PCIENUMCAP1)

PCIENUMCAP1 corresponds to the serial number register (lower DW) in the device serial number capability structure. This device supports the device serial number capability structure but does not assign a serial number. In order to use the device serial number capability structure, the serial number needs to be assigned by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEVice SERial NUMber 1ST															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEVice SERial NUMber 1ST															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	DEVice SERial NUMber 1ST	H'00000000	R	R	These bits indicate the device serial number. The value in the DSERSETR0 field is reflected.

(46) Device Serial Number Capability Register 2 (PCIENUMCAP2)

PCIENUMCAP2 corresponds to the serial number register (upper DW) in the device serial number capability structure. This device supports the device serial number capability structure but does not assign a serial number. In order to use the device serial number capability structure, the serial number needs to be assigned by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEVice SERial NUMber 2ND															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEVice SERial NUMber 2ND															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	DEVice SERial NUMber 2ND	H'00000000	R	R	These bits indicate the device serial number. The value in the DSERSETR1 field is reflected.

13.4.6 PCI Express Control System Control Registers

Registers described in this section controls the control system in the PCI Express controller. Typically, it is possible to perform PCI Express operation without changing the initial value. To modify parameter details to control the PCI Express by software, registers in this section should be set.

These registers are accessible only via the SuperHyway bus. When these registers are accessed from the PCI Express by target transfer, the PCIEC handles the access as a completer abort and executes error processing.

(1) ID Setting Register 1 (PCIEIDSETR1)

PCIEIDSETR1 specifies the values of a class code, sub-class code, programming interface, and revision ID. Setting the values to PCIEIDSETR1 before PCI Express initialization allows setting of the class code, sub-class code, programming interface, and revision ID.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Class Code Set								SubClassCodeSet							
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROGIFset								Rev ID Set							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	Class Code Set	H'FF	—	R/W	Write Class Code to these bits at initialization. This field value is reflected on the Class Code field in PCICONF2.
23 to 16	Sub Class Code Set	H'00	—	R/W	Write Sub Class Code to these bits at initialization. This field value is reflected on the Sub Class Code field in PCICONF2.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
15 to 8	PROG IF Set	H'00	—	R/W	Write Programming IF Code to these bits at initialization. This field value is reflected on the PROG IF field in PCICONF2.
7 to 0	Rev ID Set	H'00	—	R/W	Write Revision ID to these bits at initialization. This field value is reflected on the Revision ID field in PCICONF2.

(2) ID Setting Register 2 (PCIEIDSETR2)

PCIEIDSETR2 specifies the values of the sub system ID and sub system vendor ID.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Sub System ID Set															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Sub System Vendor ID Set															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 16	Sub System ID Set	H'0000	—	R/W	These bits specify the Sub System ID. The sub system device ID assigned to the PCI device vendor should be specified at initialization. The value in this field is reflected on the PCICONF11.Subsystem ID.
15 to 0	Sub System Vendor ID Set	H'0000	—	R/W	These bits specify the Sub System vendor ID. The sub system vendor ID assigned to the PCI device vendor should be specified at initialization. The value in this field is reflected on the PCICONF11.Sub System Vendor ID.

(3) Device Serial Number Setting Register 0 (PCIEDSERSETR0)

PCIEDSERSETR0 specifies the device serial number. 0 is written to PCIEDSERSETR0 at initialization. In order to use the device serial number, set a value to this register before initialization.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DSERSET0															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSERSET0															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	DSERSET0	H'00000000	—	R/W	These bits specify the 1stDW of the device serial number. To use the device serial number, write the serial number to these bits before the CFINIT bit is set to 1.

(4) Device Serial Number Setting Register 1 (PCIEDSERSETR1)

PCIEDSERSETR1 specifies the device serial number. 0 is written to PCIEDSERSETR1 at initialization. In order to use the device serial number, set a value to this register before initialization.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DSERSET1															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSERSET1															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	DSERSET1	H'00000000	—	R/W	These bits specify the 2ndDW of device serial number. To use the device serial number, write the serial number to these bits before the CFINIT bit is set to 1.

(5) TL Status Register (PCIETLSR)

PCIETLSR indicates the transaction layer status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TRPD	TRPDSET	TRPDCLR	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R/W1C	R/W1C	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	TRansaction PenDing	0	—	R	Indicates Transaction Pending. This bit is set to 1 when a Non Posted request is generated, and cleared to 0 when a corresponding Completion is received or Completion Timeout occurs. 0: A Non Posted request is not Pending. 1: A Non Posted request is Pending.
30	TRansaction PenDing SET	0	—	R/W1C	Indicates that Transaction Pending is set. This bit is set to 1 when TLSR.TRPD is set to 1 from 0, and cleared to 0 when 1 is written to TLSR.TRPD. 0: TLSR.TRPD does not change. 1: TLSR.TRPD changes from 0 to 1.
29	TRansaction PenDing CLeaR	0	—	R/W1C	Indicates that Transaction Pending is cleared. This bit is set to 1 when TLSR.TRPD is cleared to 0 from 1, and cleared to 0 when 1 is written to TLSR.TRPD. 0: TLSR.TRPD does not change. 1: TLSR.TRPD changes from 0 to 1.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
28 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(6) TL Control Register (PCIETLCTLR)

PCIETLCTLR controls the transaction layer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Bus Number								Device Number					Function Number		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	Completion Timeout Time						—	—	—	—	—	—	—	—
Initial value:	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 24	Bus Number	H'00	—	R/W	These bits indicate the bus number. When a Type0 Configuration Write request is received, the bus number specified in the request is stored in this field.
23 to 19	Device Number	00000	—	R/W	These bits indicate the device number. When a Type0 Configuration Write request is received, the device number specified in the request is stored in this field.
18 to 16	Function Number	000	—	R/W	These bits indicate the function number. When a Type0 Configuration Write request is received, the function number specified in the request is stored in this field.
15, 14	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
13 to 8	Completion Timeout Time	110010	—	R/W	These bits specify Completion Timer Timeout time. (unit: ms) Completion Timeout occurs when a Completion cannot be received within the time specified by this field. Set the time between 10 and 50 at initialization.
7 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(7) DL Status Register (PCIEDLSR)

PCIEDLSR indicates the data link layer status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DLLACT	DLDN	DLLPE	RPTO	RPNRO	BADTLP	BADDLLP	—	—	—	—	—	—	—	VCXNGPD	VC0NGPD
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
SH-R/W:	R	R	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	Data Link Layer ACTive	0	—	R	Indicates whether Data Link Control and Management State Machine is in the DL_Active state or not. 0: DLCMSM is not in the DL_Active state. 1: DLCMSM is in the DL_Active state.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
30	DL DowN	1	—	R	Indicates whether Data Link Control and Management State Machine is in the DL Down state or not. This bit is read as 1 when DL communication between links has not been established yet. 0: DLCMSM is not in the DL Down state. 1: DLCMSM is in the DL Down state.
29	Data Link Layer Protocol Error	0	—	R/W1C	Indicates whether Data Link Layer Protocol Error occurs or not. This bit is set to 1 when Data Link Layer protocol error occurs, and cleared to 0 by writing 1 to it. 0: Data Link Layer Protocol Error does not occur. 1: Data Link Layer Protocol Error occurs.
28	RePlay TimerOut	0	—	R/W1C	Indicates whether Replay TimeOut occurs or not. This bit is set to 1 when Replay Timeout occurs, and cleared to 0 by writing 1 to it. 0: Replay TimeOut does not occur. 1: Replay TimeOut occurs.
27	RePlay Number Roll Over	0	—	R/W1C	Indicates whether Replay Number Roll Over occurs or not. This bit is set to 1 when Replay Number Roll Over occurs, and cleared to 0 by writing 1 to it. 0: Replay Number Roll Over does not occur. 1: Replay Number Roll Over occurs.
26	BAD TLP	0	—	R/W1C	Indicates whether BAD TLP is detected or not. This bit is set to 1 when BAD TLP is detected, and cleared to 0 by writing 1 to it. 0: BAD TLP is not detected. 1: BAD TLP is detected.
25	BAD DLLP	0	—	R/W1C	Indicates whether BAD DLLP is detected or not. This bit is set to 1 when BAD DLLP is detected, and cleared to 0 by writing 1 to it. 0: BAD DLLP is not detected. 1: BAD DLLP is detected.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
24 to 18	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
17	VCX NeGotiation PenDing	1	—	R	Indicates the VCX NeGotiation PenDing status. When VCX is enabled, DL performs the VCX Negotiation processing. This bit indicates that VCX Negotiation is in progress, and VCX cannot be used. Start the use of VCX after confirming that this bit is not set. 0: The VCX Negotiation processing is not performed. 1: The VCX Negotiation processing is in progress.
16	VC0 NeGotiation PenDing	1	—	R	Indicates the VC0 Negotiation Pending status. VC0 is always enabled, and VC0 Negotiation is automatically performed after initialization. This bit indicates that VC0 Negotiation is in progress or disabled, and VC0 cannot be used. 0: The VC0 Negotiation processing is not performed. 1: The VC0 Negotiation processing is in progress.
15 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(8) DL Control Register (PCIEDLCTLR)

PCIEDLCTLR controls the data link layers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(9) MAC Status Register (PCIEMACSR)

PCIEMACSR indicates the MAC layer status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RVERR	LKTR	—	—	DISST	HOTRSTST	LKWIDTH						LKSPEED			
Initial value:	0	0	0	0	0	0	0	0	0	1/0	0	0/1	0	0	0	1
SH-R/W:	R/W1C	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	ReceiVer ERRor	0	—	R/W1C	Indicates whether a Receiver Error occurs or not. This bit is set when the MAC detects the following errors. — 8b/10b error — Disparity error — Elastic buffer overflow — Elastic buffer underflow — Inter-lane desqueue error 0: A Receiver Error is not detected. 1: A Receiver Error is detected.
30	LinK Trining	0	—	R	Indicates whether Link Training execution is in progress or not while MAC LTSSM is in the Recover or Configuration state. 0: Link Training execution is not in progress. 1: Link Training execution is in progress.
29, 28	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
27	DISabled State	0	—	R	Indicates whether or not MAC LTSSM is in the Disabled state. Note: This bit is set only when link disable is requested (when EXPCAP4.LKDIS is set to 1). Accordingly, this bit is always read as 0 when this module is used as Endpoint. 0: MAC LTSSM is not in the Disabled state. 1: MAC LTSSM is in the Disabled state.
26	HOT ReSeT State	0	—	R	Indicates whether or not MAC LTSSM is in the Hot Reset state. 0: MAC LTSSM is not in the Hot Reset state. 1: MAC LTSSM is in the Hot Reset state.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
25 to 20	LinK WIDTH	PCIEC0: 000100 PCIEC1/ PCIEC2: 000001	—	R	These bits indicate the link width established by MAC configuration. When DLSR.DLLACT = 1, this field is valid. This IP supports link width of ×1 and ×4, and takes either ×1 or ×4. 000001: ×1 (000010: ×2) 000100: ×4 The initial value of these bits is as follows: PCIEC0: ×4 PCIEC1/PCIEC2: ×1
19 to 16	LinK SPEED	0001	—	R	These bits indicate the link speed established by MAC configuration. When DLSR.DLLACT = 1, this field is valid. This IP supports link width of 2.5 Gb/s. 0001: 2.5 Gb/s
15 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(10) MAC Control Register (PCIEMACCTLR)

PCIEMACCTLR controls the MAC layer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LTSMDIS	LKRCFG	—	REDET	SCRDIS	—	—	—	NFTS							
Initial value:	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
SH-R/W:	R/W	R/W	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	LTSsM DISable	1	—	R/W	Disables the MAC LTSSM operation. While this bit is set to 1, the LTSSM does not operate. 0: Enables the MAC LTSSM operation. 1: Disables the MAC LTSSM operation. If PCIETCTLR.CFINIT is set to1, this field is cleared to 0 to start the MAC LTSSM operation.
30	LinK ReConFiG	0	—	R/W	Performs the re-configuration processing of MAC LTSSM. When this bit is set to 1, the MAC LTSSM first enters the Recovery state and then enters the Configuration state. This bit should be set while Link_up = 1. This bit is cleared to 0 when the re-configuration is completed. 0: Does not perform the re-configuration. 1: Performs the re-configuration.
29	—	0	—	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
28	ReDEtecT	0	—	R/W	Requests a transition to the Detect state (initial state) to the MAC LTSSM. If this bit is set to 1, the MAC LTSSM immediately makes a transition to the Detect state. This bit is cleared to 0 one clock after it is set to 1. 0: The MAC LTSSM does not make a transition to the Detect state. 1: The MAC LTSSM makes a transition to the Detect state.
27	SCRamble DISable	0	—	R/W	Disables the MAC data scrambling function. Note: This bit should be set while the MACCTLR.LTSMDIS bit is set to 1. If this bit is modified while the MACCTLR.LTSMDIS bit is cleared to 0, the operation is regarded as undefined. 0: Enables the data scrambling function. 1: Disables the data scrambling function.
26 to 24	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	Number FTS	H'FF	—	R/W	These bits specify the number of Fast Training Sequences to be transferred when the MAC returns from L0 to L0s. The maximum number of FTSs to be specified is 255 (initial value) and the minimum value of that is 6. Data smaller than 6 should not be set.
15 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(11) PM Status Register (PCIEPMSR)

PCIEPMSR indicates the power management status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	L1FAEG	—	—	—	L2FAEG	—	—	—	PMEL1RX	—	—	—	—	PMSTATE		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W1C	R	R	R	R/W1C	R	R	R	R/W1C	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	L1FAllEdGe	0	—	R/W1C	Indicates the L1 Fall Edge. This bit is set to 1 when the L1 startup sequence initiated by writing 1 to PMCTLR.L1IATN is suspended or completed. This bit is cleared to 0 by writing 1 to it. Writing 0 to this bit is invalid. 0: Indicates that the L1 startup sequence is not completed or suspended. 1: Indicates that the L1 startup sequence is completed or suspended.
30 to 28	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
27	L2FAllEdGe	0	—	R/W1C	This bit is set to 1 when the L2 startup sequence initiated by writing 1 to PMCTLR.L2IATN is suspended or completed. This bit is cleared to 0 by writing 1 to it. Writing 0 to this bit is invalid. 0: Indicates that the L2 startup sequence is not completed or suspended. 1: Indicates that the L2 startup sequence is completed or suspended.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
26 to 24	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
23	PMEnterL1RX	0	—	R/W1C	This bit is set to 1 when PM_ENTER_L1 DLLP is received. This bit is cleared to 0 when 1 is written to it. Writing 0 to this bit is invalid. 0: Indicates that PM_ENTER_L1 DLLP is not received. 1: Indicates that PM_ENTER_L1 DLLP is received.
22 to 19	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
18 to 16	PMSTATE	000	—	R	These bits indicate the state of PowerManagemnet state machine. 000: LDn state 001: L0 state 011: L1 state 010: L2 state 100: L0s state Others : Reserved
15 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(12) PM Control Register (PCIEPMCTLR)

PCIEPMCTLR controls power management.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	L1IATN	—	—	—	L2IATN	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	L1InitiATioN	0	—	R/W	Starts L1 transition. Writing 1 initiates the L1 transition. Writing 0 to this bit is invalid. When transition to L1 is in progress, this bit is read as 1; when the PM state makes a transition from L0 to other state, this bit is read as 0. 0: Indicates that the L1 transition sequence is not initiated. 1: Indicates that a transition to the L1 state is in progress.
30 to 28	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
27	L2InitiATioN	0	—	R/W	Starts L2 transition. Writing 1 initiates the L2 transition. Writing 0 to this bit is invalid. When transition to L2 is in progress, this bit is read as 1; when the PM state makes a transition from L0 to other state, this bit is read as 0. 0: Indicates that the L2 transition sequence is not initiated. 1: Indicates that a transition to the L2 state is in progress.
26 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(13) TL Interrupt Mask Register (PCIETLINTENR)

PCIETLINTENR controls the notification of a transaction layer interrupt in the PCI-Express controller to the INTC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TRPDSETEN	TRPDCLREN	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	—	0	—	R	Reserved This bit is always read as 0. The write value should always be 0.
30	TRansaction PenDing SET Enable	0	—	R/W	Enables a Transaction Pending SET interrupt. When this bit is set to 1, an interrupt is generated. 0: A Transaction Pending SET interrupt is not generated. 1: A Transaction pending set interrupt is generated.
29	TRansaction PenDing CLR Enable	0	—	R/W	Enables a Transaction Pending CLeaR interrupt. When this bit is set to 1, an interrupt is generated. 0: A Transaction Pending CLeaR interrupt is not generated. 1: A Transaction Pending CLeaR interrupt is generated.
28 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(14) DL Interrupt Mask Register (PCIEDLINTENR)

PCIEDLINTENR controls the notification of data link layer interrupt in the PCI-Express controller to the INTC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DLLACTEN	DLDNEN	DLLPEEN	RPTOEN	RPNROEN	BADTLPEN	BADDLLPEN	—	—	—	—	—	—	—	VCXNGPDEN	VC0NGPDEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	Data Link Layer ACTive Enable	0	—	R/W	Enables a DL_Active interrupt. 0: A DL_Active interrupt is not generated. 1: A DL_Active interrupt is generated.
30	DL DowNENable	0	—	R/W	Enables a DL Down interrupt. 0: A DL Down interrupt is not generated. 1: A DL down interrupt is generated.
29	Data Link Layer Protocol ErrorENable	0	—	R/W	Enables a Data Link Layer Protocol Error interrupt. 0: A Data Link Layer Protocol Error interrupt is not generated. 1: A Data Link Layer Protocol Error interrupt is generated.
28	RePlay TimerOutENable	0	—	R/W	Enables a Replay TimeOut interrupt. 0: A Replay TimeOut interrupt is not generated. 1: A Replay TimeOut interrupt is generated.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
27	RePlay Number Roll OverENable	0	—	R/W	Enables a Replay Number Roll Over interrupt. 0: A Replay Number Roll Over interrupt is not generated. 1: A Replay Number Roll Over interrupt is generated.
26	BAD TLPENable	0	—	R/W	Enables a BAD TLP interrupt. 0: A BAD TLP interrupt is not generated. 1: A BAD TLP interrupt is generated.
25	BAD DLLP Enable	0	—	R/W	Enables a BAD DLLP interrupt. 0: A BAD DLLP interrupt is not generated. 1: A BAD DLLP interrupt is generated.
24 to 18	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
17	VCX NeGotiation PenDingENable	0	—	R/W	Enables a VCX NeGotiation PenDing interrupt. When this bit is set to 1, an interrupt is generated. 0: A VCX Negotiation interrupt is not generated. 1: A VCX Negotiation interrupt is generated.
16	VC0NeGotiation PenDingENable	0	—	R/W	Enables a VC0 Negotiation Pending interrupt. When this bit is set to 1, an interrupt is generated. 0: A VC0 Negotiation interrupt is not generated. 1: A VC0 Negotiation interrupt is generated.
15 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(15) MAC Interrupt Mask Register (PCIEMACINTENR)

PCIEMACINTENR controls the notification of MAC layer interrupt in the PCI-Express controller to the INTC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RVERREN	LKTREN	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	ReceiVer ERRor Enable	0	—	R/W	Enables a ReceiverError interrupt. 0: Enables a Receiver Error interrupt. 1: Disables a Receiver Error interrupt.
30	LinK TRining Enable	0	—	R/W	Enables Link Training interrupt. 0: Enables Link Training interrupt. 1: Disables Link Training interrupt.
29 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(16) PM Interrupt Mask Register (PCIEPMINTENR)

PCIEPMINTENR controls the notification of power management interrupt in the PCI-Express controller to the INTC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	L1FAEGEN	—	—	—	L2FAEGEN	—	—	—	PMEL1RXEN	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
31	L1FAllEdGeENable	0	—	R/W	Enables an L1FAEG interrupt. 0: Disables an L1FAllEdGe interrupt. 1: Enables an L1FAllEdGe interrupt.
30 to 28	—	All 0	—	R	Reserved
27	L2FAllEdGeENable	0	—	R/W	Enables an L2FAEG interrupt. 0: Disables an L2Falledge interrupt. 1: Enables an L2Falledge interrupt.
26 to 24	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
23	PM_Enter_L1_ RXENable	0	—	R/W	Enables a PMEnterL1RX interrupt. 0: Disables a PMEnterL1RX interrupt. 1: Enables a PMEnterL1RX interrupt.
22 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

13.4.7 PCI Express Transmission System Registers

Registers described in this section control the transmitter in the PCI Express controller.

The communication using the bridge in this module can perform the PCI Express operation without changing the initial value. To modify parameter details to control the PCI Express by software, registers in this section should be set.

Those registers are accessible via the SuperHyway bus. When a register is accessed from the PCI Express by target transfer, the PCIEC handles the access as a completer abort and executes error processing.

The bridge in this module accesses the registers in this section and performs the packet transmission and reception. Registers accessed by the bridge should not be accessed by software during bridge operation.

The registers described in this section are defined as 64-bit registers. To access these registers from software, use the floating-point registers (DRn, n = even number) and transfer instructions (fmov) dedicated to floating-point data and access in 8-byte or 4-byte units. After reset cancellation, 4-byte access is performed even if the floating-point register is used for access. Accordingly, to access a register in 8-byte units, execute the FSCHG instruction to specify 8-byte access. To access in 4-byte units, access the lower four bytes and then the upper four bytes in a row. In write access, when the upper four bytes are written, data is actually written to a 64-bit register; no data is written to the register when only the lower four bytes are written.

(1) Transmit Status Register (PCIETXSR)

PCIETXSR indicates the transmit status.

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	TXEMP	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
63	TXEMP	1	—	R	Indicates the Empty flag of all transmit buffers in VC0 and VCX. The power management is performed by checking the empty status through this bit. This bit is initialized by writing 1 to TXBUFCLR. 0: Indicates that some buffers in VC0 and VCX are not empty. 1: Indicates that all buffers in VC0 and VCX are empty.

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
62 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

(2) Transmit VC0 Status Register (PCIETXVC0SR)

PCIETXVC0SR controls VC0 packet transmission.

This module performs the packet transmission by accessing this register from the bridge. Accordingly, this register should not be accessed by software during bridge operation.

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	TXBUFCLR	—	—	—	—	—	—	—	PTXEMP	—	—	—	NPTXEMP	—	—	—
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
SH-R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	CTXEMP	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	PCI R/W	SH R/W	Description
63	TX BUFfer CLeaR	0	—	R/W	Clears the VC0 transmit buffers. The transmit headers and data buffers of Posted request, Non Posted request, and completion are cleared. This bit is a one-shot trigger bit to which only 1 can be written. This bit is always read as 0. 0: (Initial value) 1: Clears the VC0 transmit buffers.
62 to 56	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
55	Posted TX EMPty	1	—	R	Reserved This bit is always read as 0. The write value should always be 0.
54 to 52	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
51	Posted TX EMPty	1	—	R	Reserved This bit is always read as 0. The write value should always be 0.
50 to 48	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.
47	Completion TX EMPty	1	—	R	Reserved This bit is always read as 0. The write value should always be 0.
46 to 0	—	All 0	—	R	Reserved These bits are always read as 0. The write value should always be 0.

13.4.8 Physical Layer Control Register

Registers described in this section control the physical layer of PCI Express controller.

The registers to control each lane are defined for each lane, and the registers to control whole of lanes is defined for only lane-0.

The physical layer control register is not mapped on the SuperHyway address space. For details to access those register, refer to section 13.5.14 Access to Physical Layer Control Register.

Do not modify the initial value, for the reserved registers and reserved bits.

(1) Physical Layer Control Register (PCIEPLCTLR)

PCIEPLCTLR controls the behavior of the physical layer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	Phy StandBy	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
SH-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI-R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R/W	Reserved.The initial value shold not be changed.
10	—	1	R/W	Reserved.The initial value shold not be changed.
9 to 8	—	All 0	R/W	Reserved.The initial value shold not be changed.
7	Phy StandBy	1	R/W	Enables the PCIEC module. To use the PCIEC module, set this bit to 1.
6 to 0	—	All 0	R/W	Reserved.The initial value shold not be changed.

13.5 Operation

13.5.1 Supported Functions

The following describes the functions supported by the PCI Express controller.

(1) Packet Transmission/Reception

Table 13.7 shows the supported PCI Express packets. This module supports packets which are not prohibited by the standards.

Table 13.7 Supported PCI Express Packets

Packet Type	Root Port		Endpoint	
	Transmission	Reception	Transmission	Reception
Memory read	O	O	O	O
Memory write	O	O	O	O
I/O read	O	O	—	O
I/O write	O	O	—	O
Lock	O	—	—	—
Configuration read	O	O	—	O
Configuration write	O	O	—	O
Message	O*	O*	O*	O*

Legend:
O: Supported by hardware
×: Not supported
—: Prohibited to be used for the PCI Express by the standards.
Note: * The PM-related messages must be processed by software.

(2) Message Transmission/Reception

Table 13.8 shows the supported PCI Express messages.

This module does not support Vendor Defined Messages.

This module can transmit or receive power management related messages, which requires software-based control.

Table 13.8 Supported PCI Express Messages

Message Type	Root Port		Endpoint	
	Transmission	Reception	Transmission	Reception
Assert_INTA	—	O	O	—
Assert_INTB	—	O	O	—
Assert_INTC	—	O	O	—
Assert_INTD	—	O	O	—
Deassert_INTA	—	O	O	—
Deassert_INTB	—	O	O	—
Deassert_INTC	—	O	O	—
Deassert_INTD	—	O	O	—
PME_Active_State_Nak	Δ	—	—	Δ
PM_PME	—	Δ	Δ	—
PME_Turn_Off	Δ	—	—	Δ
PME_To_Ack	—	Δ	Δ	—
ERR_COR	—	O	O	—
ERR_NONFATAL	—	O	O	—
ERR_FATAL	—	O	O	—
Unlock	O	—	—	O
Set_Slot_Power_Limit	O	—	—	O
Vendor_Define Type0	×	×	×	×
Vendor_Define Type1	×	×	×	×

Legend:

O: Supported by hardware

Δ: Can be transmitted/received. Software-based control is required.

—: Prohibited to be used for the PCI Express by the standards.

×: Not supported

(3) Configuration Registers

Table 13.9 shows the supported PCI Express configuration registers. This module does not support registers related to the BIST, switches, and expansion ROM.

Table 13.9 Supported PCI Express Configuration Registers

Configuration Register	PCIEC Register Name	Root Port	Endpoint
Vendor ID register	PCICONF0[15: 0]	O	O
Device ID register	PCICONF0[31:16]	O	O
Command register	PCICONF1[15: 0]	O	O
Status register	PCICONF1[31:16]	O	O
Revision ID register	PCICONF2[7: 0]	O	O
Class code register	PCICONF2[31: 8]	O	O
Cache line size	PCICONF3[7: 0]	—	—
Master latency timer	PCICONF3[15: 8]	—	—
Header type register	PCICONF3[23:16]	O	O
BIST register	PCICONF3[31:24]	×	×
Base address register 0	PCICONF4[31: 0]	O	O
Base address register 1	PCICONF5[31: 0]	O	O
Base address register 2	PCICONF6[31: 0]	—	O
Primary bus number	PCICONF6[7: 0]	O*	—
Secondary bus number	PCICONF6[15: 8]	O*	—
Subordinate bus number	PCICONF6[23:16]	O*	—
Secondary latency timer	PCICONF6[31:24]	—	—
Base address register 3	PCICONF7[31: 0]	—	O
I/O base register	PCICONF7[7: 0]	×	—
I/O limit register	PCICONF7[15: 8]	×	—
Secondary status register	PCICONF7[31:16]	O	—
Base address register 4	PCICONF8[31: 0]	—	O
Memory base	PCICONF8[15: 0]	×	—
Memory limit	PCICONF8[31:16]	×	—
Base address register 5	PCICONF9[31: 0]	—	O
Prefetchable memory base	PCICONF9[15: 0]	×	—

Configuration Register	PCIEC Register Name	Root Port	Endpoint
Prefetchable memory limit	PCICONF9[31:16]	×	—
Card bus CIS pointer register	PCICONF10[31: 0]	—	×
Prefetchable memory base (upper 32 bits)	PCICONF10[31: 0]	×	—
Subsystem ID register	PCICONF11[31: 0]	—	O
Prefetchable memory limit (upper 32 bits)	PCICONF11[31: 0]	×	—
Subsystem vendor ID register	PCICONF12[31: 0]	—	O
I/O base (upper 16 bits)	PCICONF12[15: 0]	×	—
I/O limit (upper 16 bits)	PCICONF12[31:16]	×	—
Capability pointer	PCICONF13[31: 0]	O	O
Expansion ROM base address register	PCICONF14[31:16]	—	×
Interrupt line	PCICONF15[7: 0]	O	O
Interrupt pin	PCICONF15[15: 8]	O	O
Minimum grant	PCICONF15[23:16]	—	—
Maximum latency	PCICONF15[31:24]	—	—
Bridge control register	PCICONF15[31:16]	O	—

Legend:

O: Supported by hardware

×: Not supported

—: Prohibited to be used for the PCI Express by the standards.

Note: * Only registers are mounted by hardware. Register values should be set by software.

(4) Capability Structure

Table 13.10 shows the supported PCI Express capability structures. This module supports these capability structures.

Table 13.10 Supported PCI Express Capability Structures

Capability Structure	Supported /Not Supported	Start Address
PCI power management	O	H'040
MSI	O	H'050
PCI Express	O	H'070
Advanced Error Reporting	×	—
Virtual Channel	O	H'100
Device Serial Number	O*	H'1B0
PCI Express Link Complex Declaration	×	—
PCI Express Root Complex Internal Link Control	×	—
Power Budgeting	×	—
PCI Express Root Complex Event Collector Endpoint Association	×	—
Multi-Function Virtual Channel	×	—
Vendor-Specific	×	—
RCRB Header	×	—

Legend:

O: Supported by hardware

×: Not supported

Note: * This module implement Device Serial Number capability structure as a register, but valid serial number is not specified by the hardware. When using the Device Serial Number capability structure, set the serial number by software. The Device Serial Number capability structure is not included in the capability list chain in the initial state. To use the structure, add it to the capability list chain.

13.5.2 Pin Assignment

This module operates as either a Root Port or an Endpoint defined by the PCI Express standards. This operating mode is specified by mode pins. Either a Root Port or an Endpoint should be specified by setting mode pins as described in table 13.1, Pin Configuration.

Note that this module does not operate as a legacy Endpoint, root complex integrated Endpoint, switch, or root complex invent controller that is defined by the PCI Express standards.

(1) Root Port

A Root Port is a device that controls the entire PCI Express, and one or more Root Ports are required per PCI Express system. This module can operate as a Root Port for which the SH processor acts as a host processor.

The Root Port controls the entire system, such as initializing the PCI Express system when a configuration cycle is generated, receiving error messages, and recovering from an error. In addition, the Root Port can transmit request packets, return Completion packets, and transmit and receive messages.

(2) Endpoint

An Endpoint is a device that performs data communications under the control of a Root Port. Multiple Endpoints can exist in a PCI Express system. This module can operate as an Endpoint.

Upon reception of an initialization by a configuration cycle, the Endpoint detects errors and provides reports to the Root Port. In addition, the Endpoint can transmit request packets, return Completion packets, and transmit and receive messages.

13.5.3 Initialization (PCIEC Module Initialization)

To perform communications by means of this module, it is necessary to set up a bridge function and establish a PCI Express connection. (These settings are not necessary if the bridge function is not used.).

The PCIEC should be initialized in the following procedures:

(1) Setting Up Bridge Function

Set transfer information to the following registers which are used for transfer. For details on the transfer information to be set, refer to sections 13.5.6, Target Transfers (Data Transfer from External Device to PCIEC) and 13.4.3, PCIEC Transfer Control Registers.

- PCIELAR0 to PCIELAR5
- PCIELAMR0 to PCIELAMR5

Transfer information need not be set if the transfer register is not used or is used in its initial value.

(2) Establishing PCI Express Connection

After assigning transfer information to the above transfer control registers, set the PCIETCTLR[0].CFINIT bit to 1, and specify the start of the connection establishment. (The above transfer control registers cannot be changed after CFINIT is set to 1).

Setting the PCIETCTLR[0].CFINIT bit to 1 starts the initialization of the data link layer to prepare for communication with the connection-target PCI Express device.

When the initialization of the data link layer is complete, the DL_Active state is entered, making the system ready for communication by VC0. The initialization is completed when the DL-Active state is confirmed by any of the following methods:

Establishing communication by VC0

- PCIETSTR[0].DLLACT is set to 1.
- VCCAP6[17]. VC NeGotiation PenDing is set to 1.
- INTDL interrupt which indicates DL_Active is generated.

Before generating an INTDL interrupt by DL_Active, the following settings should be performed in advance.

- Set PCIEINTER[14].INTDLE to 1.
- Set DLINTENR[31].Data Link Layer ACTive Enable to 1.

This module does not support the virtual channel (VCX); only VC0 can be used for communication.

13.5.4 Configuration Cycle (PCI Express Initialization)

When using this module as a Root Port, a configuration cycle must be generated to configure the connection-target device. A configuration cycle refers to the process of checking the status of the configuration register of the connection-target Endpoint through the use of configuration access, and assigning values to the Root Port itself and to the configuration register of the Endpoint according to the result of the checking. When the Root Port itself accesses its own configuration register, such access is normally made through the SuperHyway bus.

This section explains how to generate or receive configuration access, and the items to be specified during a configuration cycle.

When this module is used as a Root Port, a configuration access must be performed to generate a configuration cycle, and various initial settings must be performed.

When used as an Endpoint, this module receives a configuration access from a Root Port, and accepts the initialization processing.

(1) Generating Configuration Access

When accessing external device configuration registers by means of a configuration access from this module, the following procedures should be used.

When accessing this module configuration registers, the following procedures should not be used, but the register which has been mapped in the SuperHyway bus address space should be accessed via the SuperHyway bus.

(a) Setting PCIEPAR

Specify the access destination configuration register number, extension register number, and access destination device bus/device/function numbers in PCIEPAR.

(b) Setting PCIEPCTLR

Specify the type of configuration access to be generated and set the access enable bit in PCIEPCTLR.

(c) Accessing PCIEPDR

Generate a configuration read by read-accessing PCIEPDR, and generate a configuration write by write-accessing PCIEPDR. When PCIEPDR is read, the result of the configuration read is read out.

(d) Checking PCIEPCTLR

Check the PCIEPCTLR[16].CRS bit, and verify whether the CRS (Configuration Request Retry Status) has been returned. If this bit is set to 1, it is indicated that the connection-target device has not been activated, and that a correct response to the configuration request has not been made. If this bit is set to 1, write 1 to this bit and resume the processing from step (c) described above.

If a configuration access to a device is successful, this bit for that device need not be checked again.

(2) Receiving Configuration Access

The reception of a configuration access by this module is automatically processed by hardware; therefore, software-based control of it is not required.

Any change in power state by a configuration write access to the PMCAP1[1:0].PowerState field, however, requires software processing. For further details, see section 13.5.12, Power Management.

When a correct configuration write access is received, this module reads the bus number and device number in the received packet, and writes them to TLCTLR[31:24].BusNumber, TLCTLR[23:19].DeviceNumber, and TLCTLR[18:16].FunctionNumber. These values are used as requester IDs for the packets generated by this module.

(3) Set Contents

When using this module as a Root Port, issue a configuration access, and specify the following setting in order to initialize the PCI Express. The following settings should be made in the registers for both Root Ports and Endpoints.

The explanation given below relates to the PCI Express devices that have a single connection target. If the connection target is a switch or bridge, different settings will be required.

(a) Setting MPS (Max Payload Size)

Examine the MPSS (Max Payload Size Supported) in the configuration registers for all PCI Express devices in the PCI Express system, including Root Ports and Endpoints, and determine the smallest value as the system MPS. Assign the determined MPS value to the configuration registers for all devices, including Root Ports and Endpoints.

(b) Setting MRRS (Max Read Request Size)

In this LSI, the MRRS value should be the same as the MPS value. Set the MPS value to the configuration registers for all the PCI-Express devices including Root Ports and Endpoints.

(c) Setting PCI Address Space (Setting BAR)

Allocate the PCI address space for each device according to the PCI Express standards. After the address space allocation, set the result to BAR in each device.

(d) Setting Operating Mode

Set values to the following configuration registers that define the operating mode of the PCI Express. No value needs to be assigned if the registers are used in their initial value. For details on the individual registers, refer to section 13.4.5, Configuration Registers.

After the configuration cycle is completed, the values of these registers should not be changed.

- PCICONF1[10].Interrupt Disable
- PCICONF1[8].SERR Enable
- PCICONF1[6].Parity Error Response
- PCICONF15[17].SERR Enable (only for a Root Port)
- PCICONF15[15:8].Interrupt Pin (only for an Endpoint)
- PCICONF15[7:0].Interrupt Line (only for a Root Port)
- EXPCAP2[11].Enable No Snoop
- EXPCAP2[4].Enable Relaxed Ordering
- EXPCAP2[3].Unsupported Request Reporting Enable
- EXPCAP2[2].Fatal Error Reporting Enable
- EXPCAP2[1].Non Fatal Error Reporting Enable
- EXPCAP2[0].Correctable Error Reporting Enable
- EXPCAP3[20].Data Link Layer Active Reporting Capable (only for a Root Port)
- EXPCAP7[4].CRS Software Visibility Enable
- EXPCAP7[3].PME Interrupt Enable
- EXPCAP7[2].System Error on Fatal Error Enable
- EXPCAP7[1].System Error on Non-Fatal Error Enable
- EXPCAP7[0].System Error on Correctable Error Enable

(e) Setting INTx/MSI Interrupts

Determine the interrupt to be used in the system, INTx or MSI, and set the result to each device.

For details, refer to section 13.5.9, INTx Interrupts, and section 13.5.10, MSI Interrupts.

(f) Setting Master Enable

Set PCICONF1[2].Bus Master Enable, PCICONF1[1].Memory Space Enable, and PCICONF1[0].I/O Space Enable according to a transfer subsequent to the initialization.

When the Root Port receives a request from an Endpoint, first set the Bus Master Enable bit of the Root Port to 1. At the same time, if memory access is to be accepted, set Memory Space Enable to

1; or when I/O access is to be accepted, set I/O Space Enable to 1. Without these settings, the Root Port will not accept requests. In the next step, set the Bus Master Enable bit of the Endpoint to 1. Without this setting, the Endpoint cannot issue requests.

When memory access or I/O access is to be performed to the Endpoint, set the Memory Space Enable and I/O Space Enable of the Endpoint to 1. Without this setting, the Endpoint will not receive requests.

13.5.5 PIO Transfers (Data Transfer from PCIEC to External Device)

This section describes PIO transfers. A PIO transfer refers to the transfer which is performed by accessing the PCIEC module memory space via the internal bus and by generating PCI Express packets.

(1) Overview

A PIO transfer refers to the process by which the CPU and other units generate and transmit PCI Express packets by accessing the PCI memory space through the SuperHyway bus. By PIO transfers, memory read/write and I/O read/write can be performed on external PCI Express devices.

By PIO transfers, the module can easily generate PCI Express packets through access to the PCI memory space. By read access, read packets are generated on the PCI Express; by write access, write packets are generated on the PCI Express.

In a usual PIO transfer, one PCI Express packet is generated per access to the PCI memory space. The data length of the generated PCI Express is equal to the access size to the PCI memory space. For this reason, for 4-byte access by the CPU, only short PCI Express packets with a data length of 4 bytes can be generated, with the result that the data transfer efficiency is diminished when large amount of data are to be transferred.

To transfer large quantities of data, the packet connection function or the DMAC incorporated in this module must be used.

The packet connection function combines write accesses of multiple successive PIO transfers and generates a single PCI Express packet with a large data length. Generating a single long packet reduces the transfer overhead such as transfer of packet headers and improves the data transfer efficiency. For details of the packet connection function, refer to (8), Packet Connection.

The DMAC in the PCIEC enables efficient data transfer using long PCI Express packets without using the CPU. For details of the DMAC in the PCIEC, refer to section 13.5.7, DMA Transfer.

(2) Address Map (SuperHyway Space)

Table 13.11 shows the address map of the SuperHyway space.

The PCIEC consists of three types (or physically eight types) of address areas: the PCI memory area (six types), the control register area, and the configuration register area. PCI Express packets are generated through access to the PCI memory area. The section below describes the mapping between the PCI memory area and the PCI Express address space.

Table 13.11 SuperHyway Space Address Map

Memory Area	PCIEC0 Address		PCIEC1 Address		PCIEC2 Address		Physical Address Size
	29-Bit Address Mode	32-Bit Address Mode	29-Bit Address Mode	32-Bit Address Mode	29-Bit Address Mode	32-Bit Address Mode	
PCI area 0	H'FD00 0000 to H'FD7F FFFF	←	H'FD80 0000 to H'FDFF FFFF	←	H'FC80 0000 to H'FCBF FFFF	←	PCIEC0/1: 8 Mbytes PCIEC2: 4 Mbytes
PCI area 1	Not available	H'C000 0000 to H'DFFF FFFF	Not available	H'A000 0000 to H'BFFF FFFF (only for memory space setting 0 to 4)	Not available	H'8000 0000 to H'9FFF FFFF (only for memory space setting 0 to 4)	512 Mbytes
PCI area 2	H'1000 0000 to H'13FF FFFF (only for memory space setting 1, 2, 5, or 6)	H'1000 0000 to H'13FF FFFF (only for memory space setting 1, 2, 5, or 6)	Not available	H'3000 0000 to H'3FFF FFFF	Not available	H'2000 0000 to H'2FFF FFFF	PCIEC0: 64 Mbytes PCIEC1/2: 256 Mbytes
PCI area 3	H'FE10 0000 to H'FE1F FFFF	←	H'FE30 0000 to H'FE3F FFFF	←	H'FCD0 0000 to H'FCDF FFFF	←	1 Mbytes
Control register area (1)	H'FE00 0000 to H'FE03 FFFF	←	H'FE20 0000 to H'FE23 FFFF	←	H'FCC0 0000 to H'FCC3 FFFF	←	256 Kbytes
Configu-ration registers	H'FE04 0000 to H'FE04 0FFF	←	H'FE24 0000 to H'FE24 0FFF	←	H'FCC4 0000 to H'FCC4 0FFF	←	4 Kbytes
Control register area (2)	H'FE04 1000 to H'FE07 FFFF	←	H'FE24 1000 to H'FE27 FFFF	←	H'FCC4 1000 to H'FCC7 FFFF	←	252 Kbytes
Reserved	H'FE08 0000 to H'FE0F FFFF	←	H'FE28 0000 to H'FE2F FFFF	←	H'FCC8 0000 to H'FCCF FFFF	←	512 Kbytes

(3) Access to PCI Memory Space and PCI I/O Space

Figure 13.2 shows mapping from the SuperHyway address space to PCI address space. As shown in the figure, any access to the PCI area in the SuperHyway address space is mapped to either the PCI address space or PCI I/O space. The PIO transfer control register (to be explained later) specifies the space to which the mapping is to be performed or an address in a specific space where the mapping is to be performed. Access to the PCI memory space or the PCI I/O space can be performed by accessing the space (PCI area) on the SuperHyway that is mapped to the PCI space.

Read packets for the PCI memory space or PCI I/O space are generated from read access to the PCI area; write packets for the PCI memory space or PCI I/O space are generated from write access to the PCI area.

During access to the PCI memory space, the packet length is determined according to the access size to the PCI area. In other words, if the PCI area is accessed in 4-byte access, read/write packets of 4 bytes (1 DW) are generated in the PCI memory space.

Only 4-byte (1 DW) access is allowed in the PCI I/O space. If the PCI area is to be mapped to the PCI I/O space, access to the PCI area should be made with an access size of 4 bytes.

The PIO transfer control register specifies the transfer destination space (selection of PCI memory or I/O space), the start address in each space, the size of transfer destination space, and the attributes of transfer packets.

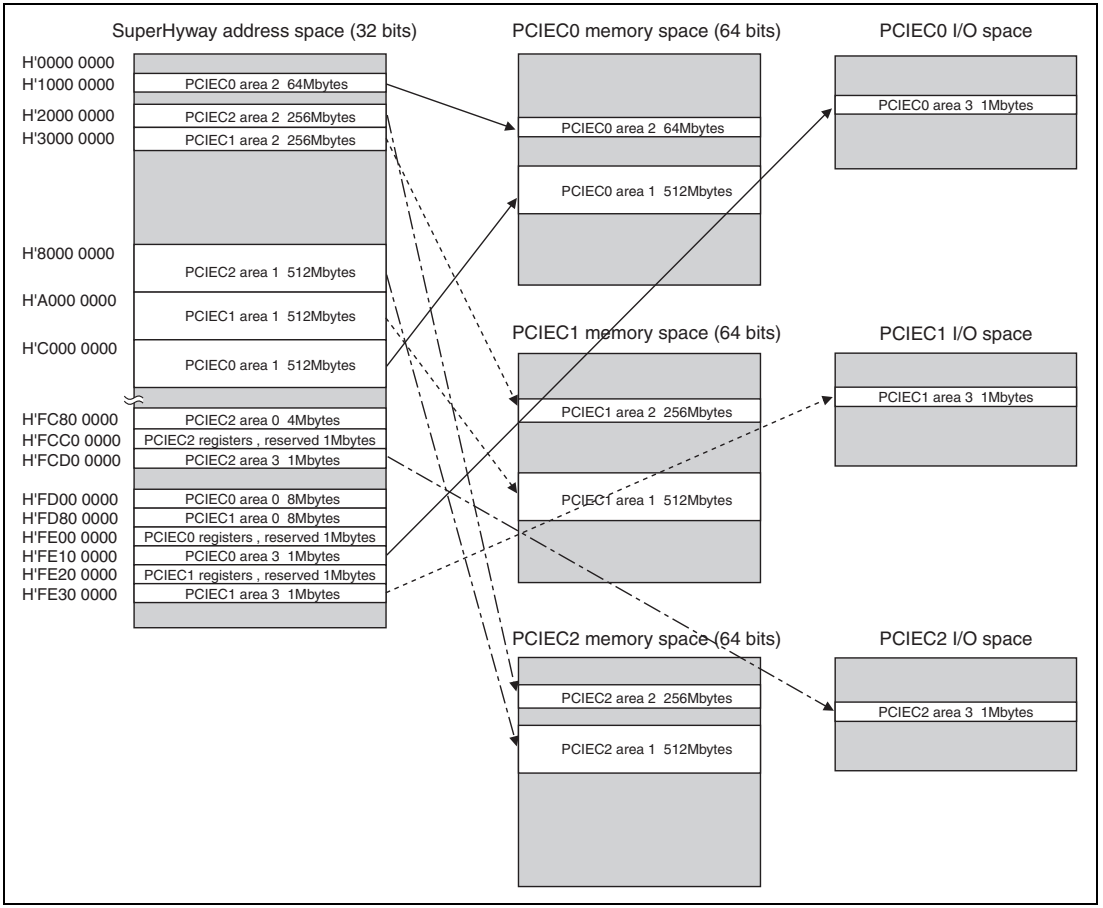


Figure 13.2 Mapping from SuperHyway Address Space to PCI Address Space

(4) Register Settings for PIO Transfers

Table 13.12 shows the transfer control registers used for PIO transfers. Accesses to the PCI areas 0 to 3 are mapped to the PCI memory or IO space depending on these register settings. These register functions are shown in table 13.12. For details of each register, refer to section 13.4.3, PCIEC Transfer Control Registers.

Table 13.12 Transfer Control Registers for PIO Transfers

PCIEPALR0 – PCIEPALR3	Start address of the PCI address space to which the PCI areas 0 to 3 are mapped (lower 32 bits)
PCIEPAHR0 – PCIEPAHR3	Start address of the PCI address space to which the PCI areas 0 to 3 are mapped (upper 32 bits)
PCIEPAMR0 – PCIEPAMR3	Sets size of data in the PCI areas 0 to 3 to be mapped to the PCI address space.
PCIEPTCTLR0 – PCIEPTCTLR3	Sets whether PCI areas 0 to 3 are enabled or disabled. Sets transfer destination space (PCI memory space or PCI I/O space). Sets attributes (Lock, EP, No Snoop, Relax Ordering) for conversion

PCIEPALRn and PCIEPAHRn (n = 0 to 3) specify an address in the PCI Express space to which PCI area n is mapped.

PCIEPAMRn specifies the size of the PCI area. A size larger than the size of the PCI area listed in table 13.11, SuperHyway Space Address Map cannot be specified.

PCIEPTCTLRn specifies whether a given area is enabled/disabled, the space to which a transfer is to be made, and the attributes of packets during the transfer process. Unless it is specified in this register that PCI area n is enabled (default: disabled), any access to the applicable PCI area is nullified. When performing a lock transfer or specifying other attributes, this register must be set before performing access to the PCI area. For details of lock transfer and attribute settings, refer to section 13.5.5 (6), Lock Requests and section 13.5.5 (7), Attributes of PCI Express Packets.

(5) Address Conversion from SuperHyway Bus to PCI

The address to be used during access to the PCI space by access to a PCI area is determined by the address of the accessed PCI area and by the settings of the transfer control register. Address conversion details are performed as described below and as shown in figure 13.3, Address Conversion for PCI Space. (In the figure and the description given below, the number n takes a value 0 to 3, and corresponds to PCI areas 0 to 3).

The lower 16 bits of a PCI address ([17:2]) are generated from the lower bits of the SuperHyway address.

The middle 11 bits of a PCI address ([28:18]) are selected from the corresponding bits of the SuperHyway address or PCIEPALRn, depending on the value of the transfer control register (PCIEPAMRn). (If the corresponding bit in PCIEPAMRn is 1, the SuperHyway address is used; if it is 0, PCIEPALRn is used.)

For the upper 35 bits ([63:29]) of a PCI address, the upper 3 bits of PCIEPAHRn and PCIEPALRn are used.

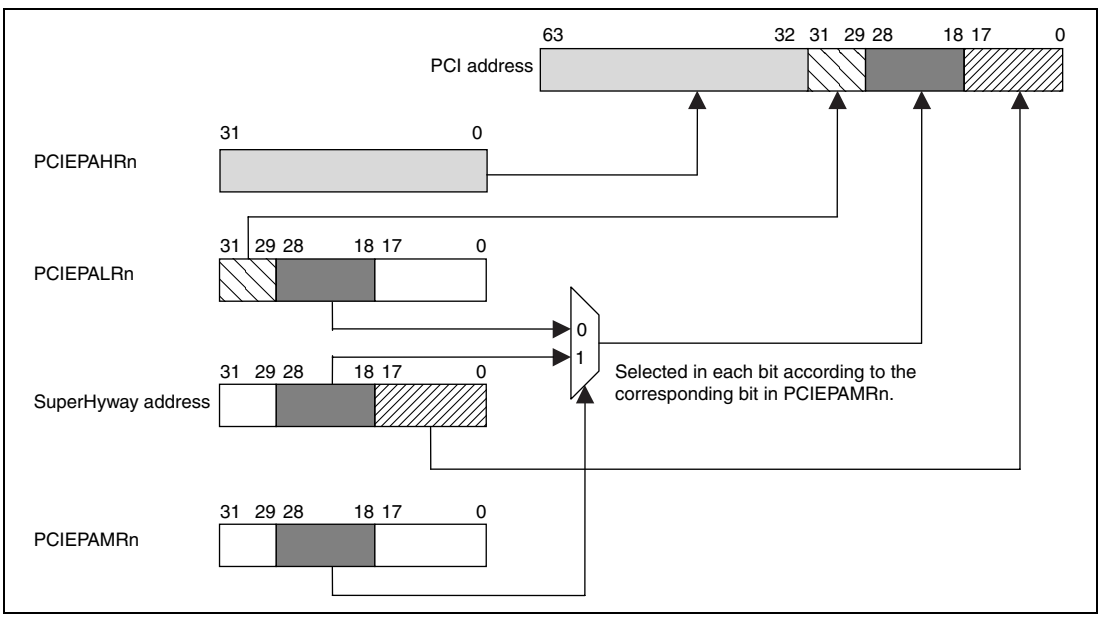


Figure 13.3 Address Conversion for PCI Space

(6) Lock Requests

In PCI Express, the issuance of lock requests, not recommended for use in newly created programs, is allowed only in Root Ports to ensure compatibility with Legacy PCI buses. By means of a lock request, the user can perform lock access to a Legacy-Endpoint or to bridges that provide a relay to a Legacy-Endpoint.

Lock requests can be issued from a PCI Express Root Port to a Legacy-Endpoint. PCI Express Endpoints cannot issue a lock request. Also, Root Ports and PCI Express Endpoints cannot receive a lock request. (If received, such a lock request will be error-handled as an unsupported request.)

In a PCI Express lock transfer, transfers from the time a lock-specified read request is received to the time an Unlock message is sent are performed on a locked basis. This module can perform lock transfers according to the following procedures:

- Set the LOCK bit in the transfer control register (PCIEPTCTLRn) to 1. This setting ensures the PIO transfers using PCI area n issue lock requests.
- Perform read/write access operations that are processed on a locked basis.
- Send an Unlock message (release the locking).
- Clear the LOCK bit in the transfer control register (PCIEPTCTLRn) to 0.

A lock transfer can be used only when this module is operating as a Root Port.

(7) Attributes of PCI Express Packets

Through the setting of the following fields in PCIEPTCTLRn, this module can control the attributes of packets that are issued to the PCI memory or PCI I/O space:

- TC (Traffic Class) / VC (Virtual Channel)

  Specifies TC/VC for access to the PCI space. For TC (Traffic Class) specify 0 to 7; for VC, specify 0 or 1. For usual transfers, TC and VC are used as TC = 0 and VC = 0. When performing a high-priority transfer using a virtual channel, assign a value other than 0 to TC/VC.

  This module does not support the virtual channel. Specify 0 for VC.
- SPC (Space)

  Specifies the space to be accessed.

  Clearing this bit to 0 causes the transmission of packets to the PCI memory space, and setting it to 1 sends packets to the PCI I/O space.
- EP (Poisoned)

  Specifies the EP bit of the transmission packet.

Setting this bit to 1 adds the Poisoned attributes to the packet to be sent. The Poisoned attribute is used when data containing an error is to be transmitted; it is not used in usual transfers.

- ATTR[0].No Snoop

  Specifies the No-Snoop bit of the transmission packet.

  Setting this bit to 1 specifies that cache snooping is not to be performed on the connection-target PCI Express device.

  When the EXPCAP2[11] (Enable No Snoop) bit in the configuration register is 0, no packet can be sent with the No Snoop bit set to 1. If such an attempt is made, it will be handled as an error.

  This option is used when the area to be accessed is not cache-enabled. If the Reject-Snoop bit is set in the configuration register on the connection-target PCI Express device, any reception of packets in which this bit is cleared to 0 is rejected.

- ATTR[1].Relaxed Ordering

  Specifies the Relaxed Ordering bit of the transmission packet.

  This bit specifies the relaxation of packet ordering.

  When the EXPCAP2[4] (Enable Relax Ordering) bit in the configuration register is 0, no packet can be sent with the Relaxed Ordering bit set to 1. If such an attempt is made, it will be handled as an error.

(8) Packet Connection

In the PIO transfer, the packet connection function combines multiple successive accesses to the PCI memory area into a single PCI Express packet with a large data length. Generating a long PCI Express packet reduces the transfer overhead such as transfer of packet headers and improves the efficiency of PCI Express transfer.

Packets are connected in the following cases.

(a) Write access while the PCIEPTCTLR0-3[28].CONNECT bit is set to 1
(b) Write access during inter-lane transfer between PCIEC modules

Packets cannot be connected for read access. Even in case (a) or (b) above, packets are not connected if the packet connection conditions described later are not satisfied.

In case (a), before packet connection, the maximum allowed length of the connected packet must be set in the PCIEPTCTLR0-3[27:24].MAX_PACKET_SIZE bits, and the PCIEPTCTLR0-3[28].CONNECT bit must be set to 1. After that, perform necessary memory access for the PIO transfer whose packets are to be connected. After the access is completed, clear the PCIEPTCTLR0-3[28].CONNECT bit to 0. If this bit is not cleared, no PCI Express packet will be sent until the next PCIEC module access.

When the expected length of the packet after connection is larger than the packet length specified in the PCIEPTCTLR0-3[27:24].MAX_PACKET_SIZE bits, packets are not connected.

In case (b), the PCIEPTCTLR0-3[28].CONNECT bit does not need to be set. In the PCIEC inter-lane transfer, packets are automatically connected. To disable packet connection, set the PCIEPTCTLR0-3[27:24].MAX_PACKET_SIZE bits to 0, that is, specify the maximum length of packet connection as four bytes.

Packet Connection Conditions:

The packet connection function works for successive PIO accesses satisfying the following conditions. When these conditions are satisfied, successive PIO transfers will be connected and a single PCI Express packet will be generated. When these conditions are not satisfied, packets will not be connected but multiple PCI Express packets will be generated for the desired PIO access.

(a) All transfers are write accesses.

Packets are not connected in read accesses.

(b) The access size is four bytes or larger.

One-byte or two-byte PIO accesses are not connected.

(c) Packets are issued to the memory space.

Packets to be issued to the IO space are not connected.

(d) All accesses are issued from the same initiator.

Accesses from different initiators, such as PIO accesses from CPU0 and CPU1, are not connected even if the addresses are consecutive.

(e) Successive accesses are issued to consecutive addresses.

Packets are connected only when the addresses of successive PIO accesses are consecutive.

(f) The access through the PCI Express packet generated by connection does not go beyond a 4-Kbyte boundary.

The PCI Express standard prohibits a data transfer through one request going beyond a 4-Kbyte boundary. In such a case, packets are not connected even when the other conditions are satisfied.

(g) Enabled bytes are consecutive.

In successive PIO accesses, if the bytes enabled in the accesses become non-consecutive after packet connection, packets are not connected.

(h) The size of the packet after connection does not exceed the maximum size.

When the generated packet size exceeds the maximum packet length specified in the PCIEPTCTLR0-3[27:24].MAX_PACKET_SIZE bits or the packet length specified by Max_Payload_Size set in the configuration register, packets are not connected.

13.5.6 Target Transfers (Data Transfers from External Device to PCIEC)

This section explains target transfers. In this context, a target transfer refers to the reception of a PCI Express packet from an external device by this module, and the transfer of data to another module in this LSI via an internal bus.

(1) Overview

A target transfer refers to the process in which an external device accesses this module from a PCI Express packet, generates a request to the SuperHyway bus, and performs transfers by sending packets to another module. By a target transfer, an external device can send memory read/write and I/O read/write packets, and can thus perform read/write operations on another module in this LSI or external memory, such as DRAM, that is connected to this LSI.

In a target transfer, this module can receive packets of any data length less than or equal to the size specified in the MPS (Max Payload Size). If a transfer involving a size greater than the size supported by the SuperHyway bus, this module splits packets and generates requests to multiple internal buses.

(2) Address Map (PCI Express Space)

Figure 13.4 shows PCI space mapping to SuperHyway space.

The assignment of addresses in the PCI Express space is dynamically determined by the Root Port during a configuration cycle, based on the register settings at initialization. In the register settings at initialization, the size of each area and the type of area to be allocated (the types of memory space and I/O space) are specified. If the initialization is completed by setting CFINIT to 1, the contents of the initialization are reflected in BARn (Base Address Register n) or R/W attributes of the configuration register, where n denotes the register number for a BAR, such as n = 0 to 1 for a Root Port and n = 0 to 5 for an Endpoint. During a subsequent configuration cycle, the Root Port references these settings, determines the address map, and sets the results to BARn of the configuration register for each device. The address pointed to by BARn serves as the start address in PCI Express space that is assigned to individual devices.

As an area in which memory space is to be allocated, this module supports either 64-bit or 32-bit PCI address space (the first 4G area of the 64-bit space). When allocating an area in 32-bit address space, one BARn is used; when allocating an area in 64-bit address space, two contiguous BARn registers (BARn+1/BARn) are used. For this reason, a maximum of one 64-bit address space area can be allocated in a Root Port, and a maximum of three 64-bit address space areas in an Endpoint.

In I/O space, areas are allocated using one BAR register.

For details of register settings for target transfers, see (3) Register Settings for Target Transfers.

An access request from the PCI Express to BARn is received by this module and converted into access to the SuperHyway bus. The address to be converted to is specified by PCIELARn. For details of address conversion, see (4) Address Conversion from PCI to SuperHyway bus.

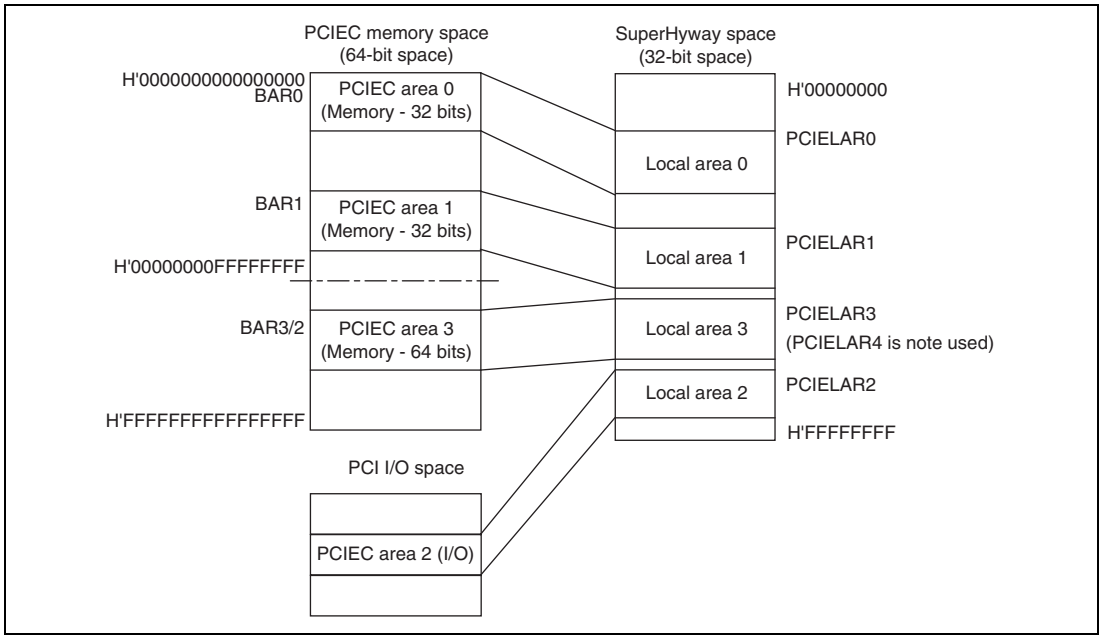


Figure 13.4 PCI Space Mapping to SuperHyway Space

(3) Register Settings for Target Transfers

Table 13.13 shows the transfer control registers for target transfers. These registers control the access to the area to be allocated in PCI space and the access to the internal bus from the allocated area.

This module has six sets of target transfer registers. When used as a Root Port, this module can allocate a maximum of two PCI areas, and when used as an Endpoint, it can allocate a maximum of six PCI areas in the PCI space. As memory space that can be allocated in PCI space, this module supports both 64-bit and 32-bit spaces. When 32-bit space is used, one set of target transfer registers allocate one PCI space, when 64-bit space is used, two sets of transfer registers allocate one PCI space.

For details of these registers, refer to section 13.4.3, PCIEC Transfer Control Registers.

Table 13.13 Transfer Control Registers for Target Transfers

PCIELARLn	Start address of the local bus (SuperHyway) space to which PCI area n will be mapped
PCIELAMRn	Sets size of PCI area n

(n = 0 or 1 for a root point, and n = 0 to 5 for an Endpoint)

PCIELARLn specifies the address on the SuperHyway bus to which area BARn is mapped, where n takes the value 0 or 1 in the case of a Root Port, or 0 to 5 in the case of an Endpoint.

PCIELAMRn specifies the size and type (memory space, I/O space, and so forth) of the PCI area to be allocated in the PCI space, and whether the area is enabled/disabled. A failure to specify the area being enabled by means of this register nullifies the allocation of the area in the PCI space, and no transfers are performed to the internal bus (in the initial setting after a reset, all areas are disabled).

(4) Address Conversion from PCI Address to SuperHyway Bus Address

Figure 13.5 shows decoding of PCI space addresses and figure 13.6 shows address conversion from PCI addresses to SuperHyway addresses.

For a PCI Express packet that has been received, first its address is decoded. The address decoding differs depending on whether the address width of the received packet is 32 or 64 bits. If the address width is 32 bits, the address in the received packed is compared with BARn, and the value of a matching *n* is determined. After that, the corresponding PCIELARn and PCIELAMRn are used to convert the address into the address of the SuperHyway bus. If the address width of the received packet is 64 bits, the 64-bit address obtained by combining BARn+1/BARn is compared with the 64-bit address of the received packet, and the value of a matching *n* is determined. After that, the corresponding PCIELARn and PCIELAMRn are used to convert the address into the address of the SuperHyway bus. The registers PCIELARn+1 and PCIELAMRn+1 are not used in this process.

The lower bits (bits [17:0]) of the SuperHyway bus address after conversion are generated from the lower bits of the received PCI packet. For the middle bits (bits [28:18]), the corresponding bits of the received packet address or PCIELARn are used depending on the PCIELAMRn bit value. For the upper bits (bits [31:29]), bits [31:29] in the PCIELARn are used without modification.

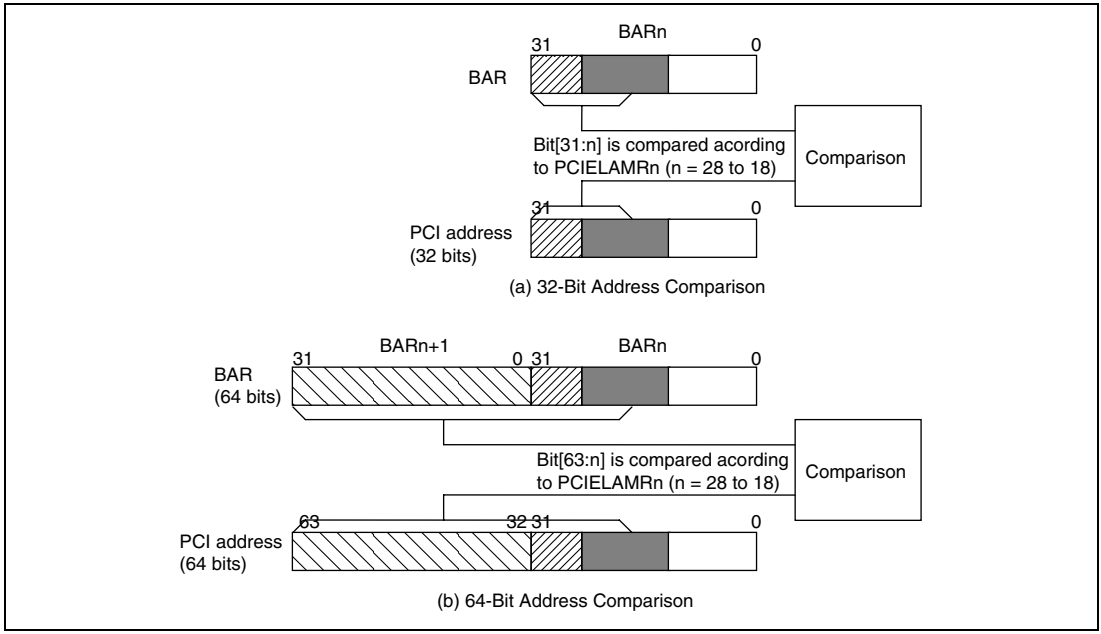


Figure 13.5 PCI Space Address Decoding

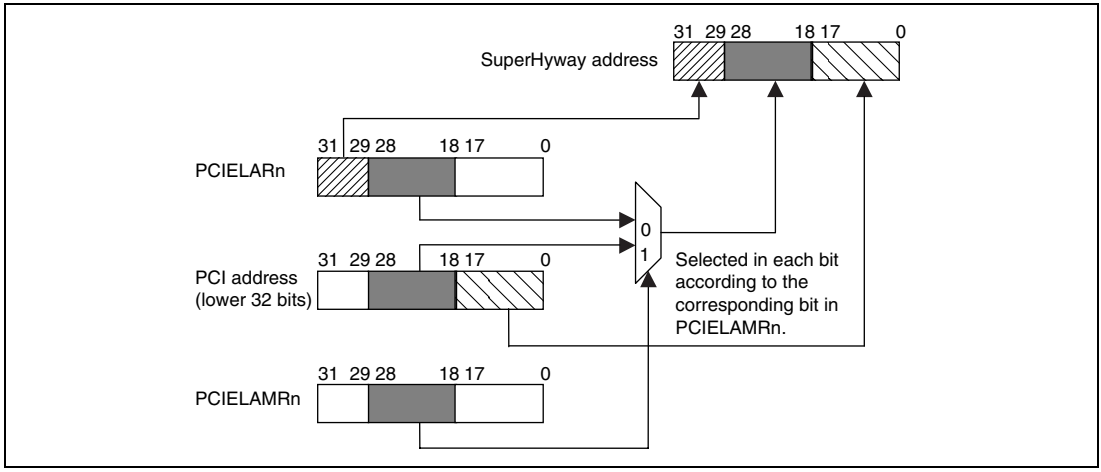


Figure 13.6 Conversion from PCI Address to SuperHyway Address

(5) Accesses from PCI Express to SuperHyway Bus

Internal bus space areas to which the PCI Express can access through this module are $\overline{CS2}$, $\overline{CS3}$, DBSC space , and other PCIEC modules. Here, the other PCIEC modules are PCIEC1 and PCIEC2 when access is made from PCIEC0; PCIEC0 and PCIEC2 when access is made from PCIEC1; and PCIEC0 and PCIEC1 when access is made from PCIEC2.

13.5.7 DMA Transfer

This section describes the DMA transfer using the DMAC (PCIEC-DMAC) incorporated in the PCIEC.

(1) Overview

The PCIEC-DMAC is designed to perform efficient data transfers between the PCI Express and other modules or external memories that are connected via the SuperHyway bus. The PCIEC-DMAC is designed so that it can issue packets with a maximum data length of 128 bytes* to the PCI Express, thus permitting high-speed data transfers that fully exploit the high transfer capacity of the PCI Express.

Note: * The maximum length of the packet data to be issued to the PCI Express is Max Payload Size.

The PCIEC-DMAC supports stride transfers for the transfer of data from non-contiguous regions. The PCIEC-DMAC supports command chains as a function for execution of multiple transfer commands. In a stride transfer, the PCIEC-DMAC supports transfers in which non-contiguous regions serve as a transfer source or destination by providing a function that adds an offset to transfer source/destination addresses after performing a fixed number of transfers. In a command chain, a set of DMAC settings, such as transfer source or destination addresses and transfer sizes, is treated as a command. By providing a function that sequentially reads and executes commands that are stored in memory, the PCIEC-DMAC supports the continuous execution of multiple transfers without the CPU intervention.

(2) Features

- Number of channels: 4 channels
- Address space: PCI Express = 64 bits, SuperHyway bus = 32 bits
- Transfer data length: PCI Express = 4 bytes to 4 Kbytes, SuperHyway bus = 4 bytes to 32 bytes
- Maximum transfer count: 536,870,912 (2^{29})
- Address mode: Dual mode

- Transfer request: Auto-request (started by register control)
- Data transfer: Normal mode (continuous transfer), stride transfer, command chain
- Priority: Either channel priority fixed mode or round-robin mode selectable
- Interrupt request: Interrupts can be requested to the INTC when a data transfer has completed or when an error has occurred.

(3) DMAC Transfer Request

The PCIEC-DMAC supports auto-request mode. The PCIEC-DMAC is started up by writing register from the CPU.

(4) Channel Priority

When receiving simultaneous transfer requests with respect to multiple channels, the PCIEC-DMAC performs transfers according to the specified priority. The priority of channels can be selected from two modes: fixed and round-robin. These modes are selected by the ART bit in the PCIEDMAOR register.

To improve transfer efficiency, the PCIEC-DMAC uses largest possible PCI Express packets for transfer. Once the transmission or reception processing is started in a channel, execution of the processing is not interrupted until the transfer processing for the packet is completed. For this reason, even if a transfer request with a higher priority becomes executable, no channel switching is done until the current packet transmission at that stage is completed; that is, a maximum of a 4-Kbyte transfer is completed.

Channel switching occurs when one set of data transfer in the channel being executed is completed. Completion of one set of data transfer means that the transfers for both the SuperHyway bus and PCI Express are completed at the same time.

(a) Fixed mode

In fixed mode, the channel priority does not change. The priority is fixed as follows.

CH0 > CH1 > CH2 > CH3

(b) Round-robin mode

In round-robin mode, after one set of transfer has completed in a channel, the channel priority is changed so that the completed channel has the lowest priority.

(5) Normal Transfer

In a normal-mode transfer, data is transferred from a specified source address to a specified

destination address. Either of the following transfer direction can be selected: from the PCI to SuperHyway bus or from SuperHyway bus to the PCI.

The PCIEC-DMAC performs normal-mode transfers according to the following procedures. For details on each register specification, refer to section 13.4.4, PCIEC-DMAC Control Registers.

(a) Setting PCIEC-DMAC

Specify PCIEDMAOR the DMA_Enable and arbitration.

(b) Setting Transfer

Set a PCI/SuperHyway address, a byte count, and a transfer termination interrupt.

Specify source/destination addresses in the registers PCIEDMPALRn/ PCIEDMPAHRn, PCIEDMSALRn, and PCIEDMBCNTRn, where n denotes a channel number (0 to 3). The address to be specified does not depend on the direction of transfer. Specify a PCI address in PCIEDMPALRn/ PCIEDMPAHRn, and an SuperHyway bus address in PCIEDMSALRn.

To generate an interrupt when a transfer is completed, specify an interrupt setting in the PCIEDMCHSRn register. If a stride transfer is not to be performed, clear PCIEDMSBCNTRn and PCIEDMSTRRn to 0. If a command chain is not used, clear PCIEDMCCARn to 0.

(c) Starting up the DMAC

In the PCIEDMCHCRn, specify the direction of transfer, and at the same time initiate the transfer process by enabling the channel.

If a stride transfer is not to be performed, clear the PCIEDMCHCRn[24].SARE and PCIEDMCHCRn[25].PARE to 0.

If a command chain is not used, clear the PCIEDMCHCRn[29].CCRE to 0.

(d) Waiting for transfer end

Recognize the end of a DMA transfer by checking that PCIEDMCHSRn[0].TE is set to 1 or by detecting a transfer end interrupt.

(e) Performing end processing

Complete the DMA transfer by clearing PCIEDMCHCRn[31].CHE to 0. Note that PCIEDMCHSRn[0].TE is cleared to 0 by writing 1 to it. If this end processing is not performed, the next DMA transfer cannot be started.

(6) Stride Transfer

The stride transfer involves execution of striding, that is, the addition of an offset to source or destination address, after the transfer of a specific number of bytes,. By providing a striding to the destination address, a scatter transfer can be performed. Similarly, by providing a striding to the source address, a gather transfer can be performed. By providing a striding to both the source and destination addresses, the transfer of non-contiguous regions can be performed.

When performing a stride transfer and specifying transfer settings, set a stride interval (stride counter) to PCIEDMSBCNTRn, and a stride width to PCIEDMSTRRn. When performing a stride transfer only for either the PCI side or SuperHyway side, set the stride interval (SS or PS field in PCIEDMSTRRn) for the non-stride transfer side to 0.

When starting the DMAC, set either PCIEDMCHCRn[24].SARE or PCIEDMCHCRn[25].PARE to 1.

Other settings are the same as in the normal-mode transfer.

(7) Command Chain

A command chain allows this module to continuously execute multiple DMAC commands. A DMAC command refers to a set of information that requests the PCIEC-DMAC transfer; that is, the information specified by PCIEDMPALRn, PCIEDMSALRn, PCIEDMBCNTRn, PCIEDMSBCNTRn, PCIEDMSTRRn, PCIEDMCCARn, and PCIEDMCHCRn. As well as in the PCIE-DMAC control register, this information can also be set in the memory in the format shown in figure 13.7. (The upper 32 bits of the address on the PCI side cannot be specified by a DMAC command. The address specified in the PCIEC-DMAC control registers is always used.) Use of a command chain enables the PCIEC-DMAC to read the next DMAC command after execution of a DMAC command, to write the read command contents to the PCIEC-DMAC control registers, and to execute the read command. To configure a DMAC command chain and perform continuous transfer, specify the next DMAC command in the previous DMAC command.

When a command chain is used, the PCIEC-DMAC first executes the DMAC command specified in the PCIEC-DMAC control registers for the target channel. After execution of this command, the PCIEC-DMAC reads the next DMAC command from the address specified in PCIEDMCCARn, writes the command contents to the PCIEC-DMAC control registers for the corresponding channel, and executes it. If the CCRE bit of the newly read DMAC command is 1, the PCIEC-DMAC reads the next command again from the memory and executes it after completion of that command. If the CCRE bit in the read DMAC command is 0, execution of the series of command chains ends upon completion of that command.

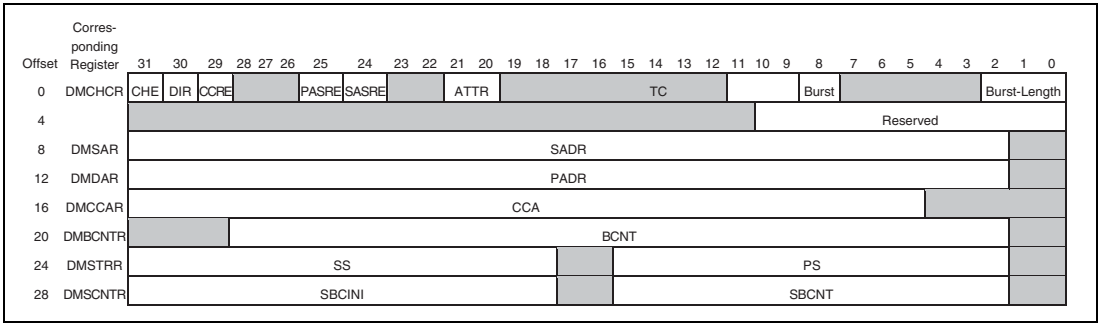


Figure 13.7 PCIEC-DMAC Command Format

To start a command chain on a channel, enable the channel while PCIEDMCHCRn[29].CCRE is set to 1. Before starting a command chain, store a chain of DMAC commands in a memory location accessible from the SuperHyway bus, and set the address of the first command to the PCIEDMCCARn.

PCIEC-DMAC read the DMAC command via SuperHyway by using 32-byte access. Put DMAC command on the memory, which is accessible from SuperHyway by 32-byte access.

Each DMAC command stored in memory must satisfy the conditions below.

- CHE field
  - — Must always be 1.
- ATTR field
  - — Set the ATTR field of the DMAC command in memory to the same value as the ATTR specified in the ATTR field of the PCIEC-DMAC control register. The value in the ATTR field cannot be modified by loading a command.
- TC field
  - — Set the TC field of the DMAC command in memory to the value that selects the same VC as that specified by the TC field of the PCIEC-DMAC control register. The VC to be used cannot be changed during command chain execution.
- RESERVED field
  - — Must be 8.
- CCA field
  - — The CCA field of last executed command must be 0.

(8) PCIEC-DMAC Interrupt Sources

The PCIEC-DMAC generates an interrupt for each channel that indicates the end of the transfer, as well as an interrupt that indicates an error termination on a common basis to all channels. For details, refer to section 13.5.11, Interrupt Request to the INTC.

13.5.8 Message Transmission and Reception

This section describes the PCIEC message transmission and reception by program control.

(1) Overview

In the PCI Express, messages are used to transmit information for standard-defined purposes, such as an interrupt notification, power management, error information, and an unlocking.

In this module, INTx interrupt reception and error information (for both transmission and reception) are processed by hardware according to the request; therefore, these messages need not be supported by software. When sending or receiving messages that are not automatically done by hardware, such as messages for INTx interrupt generation, unlocking, and power management, should be sent or received by software.

(2) Message Transmission

In the PCIEC, some messages are automatically sent by hardware, but to send other messages, transmission must be initiated by software or packets must be created by software.

(a) Automatic Message Transmission by Hardware

The PCIEC automatically sends a necessary message when an error occurs. The following messages are automatically sent.

- ERR_COR

  When transmission of the correctable error message is enabled, this message is automatically sent if a correctable error occurs.
- ERR_NONFATAL

  When transmission of the non-fatal error message is enabled, this message is automatically sent if a non-fatal error occurs.
- ERR_FATAL

  When transmission of the fatal error message is enabled, this message is automatically sent if a fatal error occurs.

(b) Message Transmission Initiated by Software

The PCIEC sends unlock or INTx interrupt messages under control by software. To send these messages, access the following registers.

- Unlocking (transmission of unlock message)

  Write 1 to the PCIEUNLOCKCR[0].ASTUNLOCK bit to send an unlock message and cancel the lock.

  The unlock message can be sent only in the Root Port.

  For details of he PCIEUNLOCKCR, refer to section 13.4.1 (10), Unlock Control Register (PCIEUNLOCKCR), and for details of lock transfer, refer to section 13.5.5 (6), Lock Requests.
- Slot power limit setting (transmission of Set_Slot_Power_Limit message)

  Write to the PCIEEXPCAP5 to send a Set_Slot_Power_Limit message. In the Set_Slot_Power_Limit message to be sent, bits [14:7] indicate the slot power limit value and bits [16:15] indicate the slot power limit scale.

  The Set_Slot_Power_Limit message can be sent only in the Root Port.

  For details of the slot power limit, refer to section 13.4.5 (30), PCIE Capability Register 5 (PCIEEXPCAP5).
- INTx interrupt generation and cancellation (transmission of Assert_INTx/Deassert_INTx message; x = A to D)

  Write 1 to the PCIEINTXR[16].ASTINTX bit to send the Assert_INTx message corresponding to the INTx interrupt specified in the PCICONF15[15:8].INTPIN bits and generate an INTx interrupt. Write 0 to the PCIEINTXR[16].ASTINTX bit to send the Deassert_INTx message corresponding to the INTx interrupt specified in the PCICONF15[15:8].INTPIN bits and cancel the INTx interrupt.

  The INTx interrupt can be generated or canceled only in the Endpoint.

  For details of PCIEINTXR, refer to section 13.4.1 (13), INTx Register (PCIEINTXR). For details of the INTx interrupt, refer to section 13.5.9, INTx Interrupts.

(c) Message Generation by Software

Messages other than the above can be sent by creating packets such as message codes by software. To transfer messages in this way, access the following registers.

- PCIEMSGALR
- PCIEMSGAHR
- PCIEMSGCTLR
- PCIEMSGD

In PCIEMSGALR/PCIEMSGAHR, specify the address at the time of the message issuance; in PCIEMSGCTLR, specify attributes at the time of the message issuance (selection of Msg/MsgD, MROUTE, MCODE). In PCIEMSGCTLR, enable the message issuance (MSGIE). Messages cannot be issued if MSGIE is cleared to 0.

By executing a write-access to the PCIEMSGD after setting these registers, a message is transmitted to the PCI Express. If PCIEMSGCTLR.MTYPE is cleared to 0 and Msg (no data) is selected, any value that is written is discarded. If PCIEMSGCTLR.MTYPE is set to 1 and MsgD (data attached) is selected, the value that is written is used as data. Only the MsgD having 1-DW data can be sent.

Depending on the above register settings, the PCIEC can also send messages that are not authorized in PCI Express standards, but correct operation cannot be guaranteed. Table 13.14 shows the messages that can be issued according to the PCI Express standards. The messages marked in the Remarks column can also be sent by the methods described in Notes, which differ from the usual method already described.

Table 13.14 Transmission-Authorized Messages

Transmit Mode	Message Name	Type	Message Code	Routing	Remarks
Root Port	PM_Active_State_Nak	Msg	0001 0100	100	
	PME_Turn_Off	Msg	0001 1001	011	
	Unlock	Msg	0000 0000	011	*1
	Set_Slot_Power_Limit	MsgD	0101 0000	100	*2
Endpoint	Assert_INTA	Msg	0010 0000	100	*3
	Assert_INTB	Msg	0010 0001	100	*3
	Assert_INTC	Msg	0010 0010	100	*3
	Assert_INTD	Msg	0010 0011	100	*3
	Deassert_INTA	Msg	0010 0100	100	*3
	Deassert_INTB	Msg	0010 0101	100	*3
	Deassert_INTC	Msg	0010 0110	100	*3
	Deassert_INTD	Msg	0010 0111	100	*3
	PM_PME	Msg	0001 1000	000	
	PME_TO_ACK	Msg	0001 1011	101	
	ERR_COR	Msg	0011 0000	000	*4
	ERR_NONFATAL	Msg	0011 0001	000	*4
	ERR_FATAL	Msg	0011 0011	000	*4

Notes: 1. This message can also be sent by writing 1 to PCIEUNLOCKCR[0].ASTUNLOCK.
2. This message can also be sent through write-access to PCIEEXPCAP5[14:7].
3. These messages can also be sent through write-access to PCIEINTXR.ASTINTX.
4. If an error is detected in the packets sent and received by the PCIEC module and a message transmission is specified in the associated configuration register, the message is automatically transmitted by hardware.

(3) Message Reception

In the PCIEC, reception of some messages (described later) is automatically handled by hardware. Reception of other messages that hardware does not support must be handled by software.

(a) Message Reception by Hardware

The PCIEC hardware handles reception of the following messages.

- Assert_INTA

  When receiving this message in the Root Port, the PCIEC sets the PCIEINTXR[0].INTA bit. If the PCICONF15[7:0].INTLINE bits are not set to FFh, setting the PCIEINTXR[0].INTA bit generates an INTA interrupt.

  When receiving this message in the Endpoint, the PCIEC handles it as a malformed TLP, and executes error processing. For details, refer to section 13.5.13, Error Processing.
- Assert_INTB

  When receiving this message in the Root Port, the PCIEC sets the PCIEINTXR[1].INTB bit. If the PCICONF15[7:0].INTLINE bits are not set to FFh, setting the PCIEINTXR[1].INTB bit generates an INTB interrupt (only in the Root Port).

  When receiving this message in the Endpoint, the PCIEC handles it as a malformed TLP, and executes error processing. For details, refer to section 13.5.13, Error Processing.
- Assert_INTC

  When receiving this message in the Root Port, the PCIEC sets the PCIEINTXR[2].INTC bit. If the PCICONF15[7:0].INTLINE bits are not set to FFh, setting the PCIEINTXR[2].INTC bit generates an INTC interrupt (only in the Root Port).

  When receiving this message in the Endpoint, the PCIEC handles it as a malformed TLP, and executes error processing. For details, refer to section 13.5.13, Error Processing.
- Assert_INTD

  When receiving this message in the Root Port, the PCIEC sets the PCIEINTXR[3].INTD bit. If the PCICONF15[7:0].INTLINE bits are not set to FFh, setting the PCIEINTXR[3].INTD bit generates an INTD interrupt (only in the Root Port).

  When receiving this message in the Endpoint, the PCIEC handles it as a malformed TLP, and executes error processing. For details, refer to section 13.5.13, Error Processing.
- Deassert_INTA

  When receiving this message in the Root Port, the PCIEC cleats the PCIEINTXR[0].INTA bit. If the PCICONF15[7:0].INTLINE bits are not set to FFh, clearing the PCIEINTXR[0].INTA bit cancels the INTA interrupt (only in the Root Port).

  When receiving this message in the Endpoint, the PCIEC handles it as a malformed TLP, and executes error processing. For details, refer to section 13.5.13, Error Processing.

- Deassert_INTB

  When receiving this message in the Root Port, the PCIEC cleats the PCIEINTXR[1].INTB bit. If the PCICONF15[7:0].INTLINE bits are not set to FFh, clearing the PCIEINTXR[1].INTB bit cancels the INTB interrupt (only in the Root Port).

  When receiving this message in the Endpoint, the PCIEC handles it as a malformed TLP, and executes error processing. For details, refer to section 13.5.13, Error Processing.

- Deassert_INTC

  When receiving this message in the Root Port, the PCIEC cleats the PCIEINTXR[2].INTC bit. If the PCICONF15[7:0].INTLINE bits are not set to FFh, clearing the PCIEINTXR[2].INTC bit cancels the INTC interrupt (only in the Root Port).

  When receiving this message in the Endpoint, the PCIEC handles it as a malformed TLP, and executes error processing. For details, refer to section 13.5.13, Error Processing.

- Deassert_INTD

  When receiving this message in the Root Port, the PCIEC cleats the PCIEINTXR[3].INTD bit. If the PCICONF15[7:0].INTLINE bits are not set to FFh, clearing the PCIEINTXR[3].INTD bit cancels the INTD interrupt (only in the Root Port).

  When receiving this message in the Endpoint, the PCIEC handles it as a malformed TLP, and executes error processing. For details, refer to section 13.5.13, Error Processing.

- ERR_COR

  The PCIEC sets the configuration register bits that indicate reception of a correctable error, and generates an INT_PCICERR interrupt. For details, refer to section 13.5.13, Error Processing.

- ERR_NONFATAL

  The PCIEC sets the configuration register bits that indicate reception of a non-fatal error, and generates an INT_PCINFERR interrupt. For details, refer to section 13.5.13, Error Processing.

- ERR_FATAL

  The PCIEC sets the configuration register bits that indicate reception of a fatal error, and generates an INT_PCIFERR interrupt. For details, refer to section 13.5.13, Error Processing.

- Unlock

  The PCI Express Endpoint and Root Port do not accept lock request; the PCIEC discards the Unlock message.

- Set_Slot_Power_Limit

  The PCIE sets the power information indicated by the received message in the configuration register (EXPCAP1[27:18]).

- Vendor_Define Type0

  This device does not support vendor-defined messages; the PCIEC handles this message as an unsupported request. For details, refer to section 13.5.13, Error Processing.

- Vendor_Define Type1
  This device does not support vendor-defined messages; the PCIEC discards this message.

The PCIEC standard specifies that the Assert_INTA/B/C/D, Deassert_INTA/B/C/D, ERR_COR, ERR_NONFATAL, ERR_FATAL, Unlock, and Set_Slot_Power_Limit messages must be sent and received in TC0. If such a message is received in a TC other than TC0, the PCI Express handles it as a malformed TLP (fatal error) and executes error processing.

(b) Reception of Messages to be Handled by Software

The PCIEC hardware detects reception of the following messages but does not automatically process them. Reception of these messages must be checked and processed by software as described below.

- PME_Active_State_Nak
- PM_PME
- PME_Turn_Off
- PME_To_Ack

A message reception can be checked via the following registers:

- PCIEMSGR
- PCIEMSGIER

In the PCIEMSGR, bits are defined for each specified message, and 1 is set to a bit corresponding to the received message when a message is received. In this situation, if the applicable bit in the PCIEMSGIER is set to 1 and if the PCIEINTER[5].INT_PCIMESE is set to 1, a PCIMES interrupt is generated.

Table 13.15 shows messages whose reception is detected by PCIEMSGR. The messages indicated in the Remarks column are hardware-processed, as will be described later.

Table 13.15 Messages Whose Reception can be Detected by the PCIEC

Bit Number in PCIEMSGR	Receive Message Name	Remarks
8	PM_Active_State_Nak	
9	PM_PME	
10	PME_Turn_OFF	
11	PME_TO_ACK	

13.5.9 INTx Interrupts

This section describes the INTx interrupts.

The INTx is a general term for the INTA/INTB/INTC/INTD interrupts that are used on legacy PCI buses. Whereas on a legacy PCI bus specific signal lines are used, in the PCI Express the INTx is emulated through device registers and through the sending and receiving of messages that represent the asserting/de-asserting of interrupts.

In the PCI Express, all devices are required to support INTx interrupts. This module also supports INTx interrupts. If MSI interrupts that have higher generality are used in this module, the use of INTx interrupts is prohibited. Whether MSI interrupts are used or not is determined during a configuration cycle, and the results of the determination are reflected in the MSICAP0[16].MSI Enable bit. If the MSICAP0[16].MSI Enable bit is set to 1, MSI interrupts are used, and INTx interrupts are disabled. For details, refer to section 13.5.10, MSI Interrupts.

INTx interrupts are used when an Endpoint requests an interrupt to a Root Port. Endpoints cannot receive an INTx interrupt. Similarly, Root Ports cannot generate an INTx interrupt.

(1) INTx Interrupt Setting (Root Port)

The setting of an INTx interrupt is performed by a Root Port during a configuration cycle. If a decision is made to use an INTx interrupt (and not to use an MSI interrupt) during a configuration cycle, the Root Port should perform the following settings:

- Set the PCICONF15[7:0].Interrupt Line of the Root Port to a value other than H'FF.
- Set the PCICONF15[15:8].Interrupt Pin of the Endpoint according to the interrupt to be assigned to the Endpoint.
- Clear the PCICONF1 [10].Interrupt Disable of the Endpoint to 0.

(2) INTx Interrupt Generation (Endpoint)

To generate an INTx interrupt from this module, set PCIEINTXR.ASTINTX to 1. To cancel the INTx interrupt, clear PCIEINTXR.ASTINTX to 0. By writing either 1 or 0 to this bit, an Assert_INTx/Deassert_INTx message is transmitted to the Root Port according to the interrupt type specified by the PCICONF15[15:8].Interrupt Pin.

When this module is used as a Root Port and MSI interrupts are used, an INTx interrupt cannot be generated if the PCICONF1 [10].Interrupt Disable is set to 1.

(3) INTx Interrupt Reception (Root Port)

Upon receiving an Assert_INTx message and detecting occurrence of an INTx interrupt, the PCIE sets the corresponding bit in the PCIEINTXR[3:0] to 1. Simultaneously, if a value other than H'FF is set to the PCICONF15[7:0].Interrupt Line, an interrupt to the INTC is generated. For details of interrupts to the INTC, refer to section 13.5.11, Interrupt Request to the INTC.

Upon receiving a Deassert_INTx message and detecting cancellation of an INTx interrupt, the PCIEC clears the corresponding bit in the PCIEINTXR[3:0] to 0.

When receiving an interrupt from the INTC by software, check the cause of the interrupt, and specify interrupt clearing for the Endpoint that generated the interrupt.

13.5.10 MSI Interrupts

This section describes the MSI interrupts.

The MSI interrupt is an interrupt method that is defined as a substitute for INTx. Compared with the INTx, the MSI can use a larger number of interrupt sources, and assign multiple sources to a single device. The MSI interrupt generates an interrupt when a Root Port performs a memory write to a specific internal area; as such, the MSI interrupt, unlike the INTx, does not require a dedicated line.

In the PCI Express, devices are required to support either MSI or MSI-X interrupts. This module supports MSI interrupts only. For PCI-SIG, the use of MSI interrupts rather than INTx interrupts is recommended, implying that in the future INTx interrupts may become an EOL (End of Life). However, if a legacy PCI bus that can potentially be connected via a PCI Express/PCI bridge does not support the MSI, or to enable the use of existing drivers that use INTx interrupts, the use of INTx interrupts is permitted.

Whether the PCI Express system should use INTx or MSI interrupts is determined by the Root Port during a configuration cycle. When this module is to be used as a Root Port, whether INTx or MSI interrupt is used should be determined during a configuration cycle, and the results of the determination must be set in the configuration register.

When this module is used as a Root Port, a maximum of 32 interrupt vectors can be used as MSI interrupts. In addition, this module supports both Per Vector Masking and the transmission of 64-bit address messages.

(1) MSI Interrupt Setting (Root Port)

When this module is used as a Root Port, whether INTx or MSI interrupts are used should be determined during a configuration cycle. If MSI interrupts are selected, the following settings should be performed.

(a) Enabling MSI Interrupts

Set MSICAP0[16].MSI Enable to 1 in all devices.

(b) Assigning Interrupt Vectors to Endpoints

Reference the MSICAP0[19:17].Multiple Message Capable of the Endpoint device, and determine the number of MSI interrupt vectors required by devices. The number of vectors to be assigned to the Endpoints is determined so that the number of vectors to be assigned to all devices is 32 or less, the number of vectors assigned to each Endpoint is 1 or greater and less than or equal to the requested number of assignments.

The number of vectors assigned to each Endpoint should be set in the MSICAP0[22:20].Multiple Message Enable.

(c) Setting Message Addresses

Set the transmission destination address of MSI messages in the MSICAP1[31:2].Lower Message Address and MSICAP2[31:0].Upper Message Address of Endpoint devices.

The transmission destination address of MSI messages should be the PCI address associated with the PCIEMSIFR register of the Module. When MSI interrupts are used, a part of the PCI space spanned by the base address register (BAR) of the Root Port must include the register space of the PCIE module.

Figure 13.8 shows a transfer destination address of a MSI message to be set in an Endpoint.

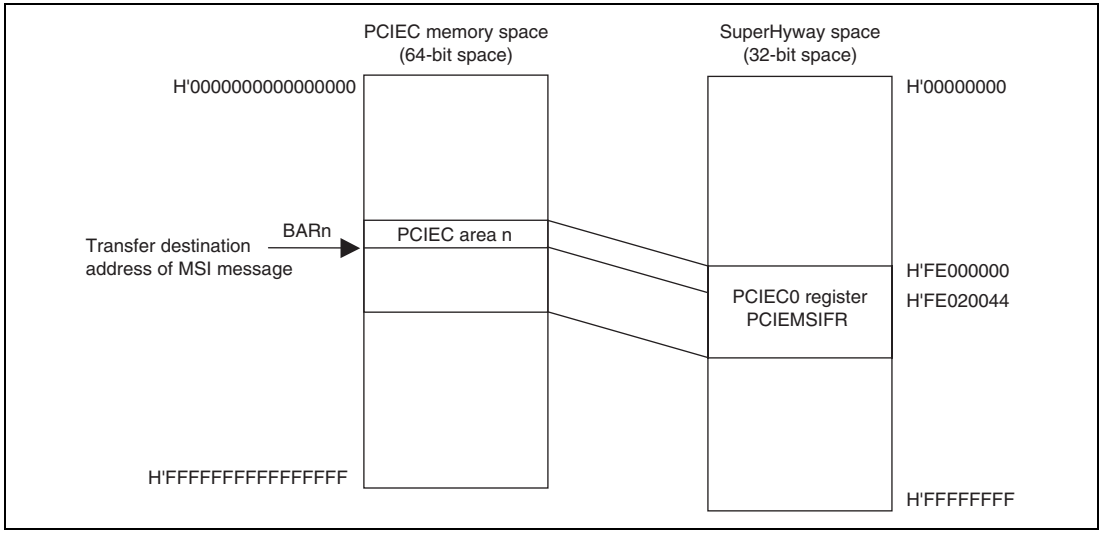


Figure 13.8 Transfer Destination Address of MSI Message to be Set in Endpoint

(d) Setting Message Data

Set the write data to be used during message transmission in the MSICAP3[15:0].Message Data of the Endpoint device.

In the 32-bit MSI vectors supported by this module, bits [12:8] in write data must specify the starting vector number of the MSI vectors assigned to the Endpoint, and all other bits must be cleared to 0. Figure 13.9 shows an example of an MSI message data to be set in an Endpoint.

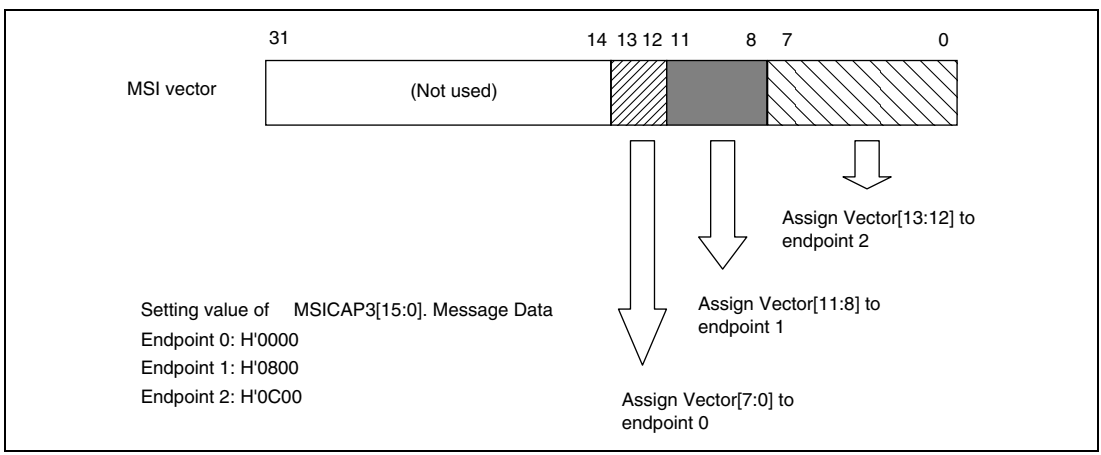


Figure 13.9 MSI Message Data to be Set in Endpoint
(Example When Vectors 8, 4, and 2 Are Assigned to Endpoints 0, 1, and 2, Respectively)

(2) MSI Interrupt Generation (Endpoint)

An MSI interrupt is generated when the vector number of an interrupt to be generated is written to the PCIEMSIR[4:0].MSIAST.

Writing to the PCIEMSIR[4:0].MSIAST sets a bit associated with the vector number in the MSICAP5[31:0].Message Pending. If 1 is set in the MSICAP5[31:0].Message Pending and the corresponding bit in the MSICAP4[31:0].Message Mask is 0, MSI interrupts are generated sequentially beginning with the lowest bit number.

An MSI interrupts is issued by executing a memory write that involves writing a value that is obtained by adding a vector number to MSICAP3[15:0].Message.Data to the address specified by MSICAP1[31:2].Lower Message Address and MSICAP2[31:0].Upper Message Address.

MSI interrupts can be generated only in Endpoints; Root Ports cannot generate MSI interrupts.

(3) MSI Interrupt Reception (Root Port)

When an MSI interrupt is received, a bit corresponding to the vector number set in PCIEMSIFR[31].MSIF is set to 1. Simultaneously, an interrupt is generated to the INTC.

13.5.11 Interrupt Request to the INTC

The PCIEC supports 68 types of interrupts. These interrupt signals are connected to the INTC in this LSI.

Table 13.16 lists the PCIEC interrupts.

Table 13.16 PCIEC Interrupts

Number	Interrupt Name	Description
0	INT_PCISERR	System error
1	INT_PCIFERR	PCIE Fatal error
2	INT_PCINFERR	PCIE Non Fatal error
3	INT_PCICERR	PCIE Correctable error
4	INT_PCIPOWER	Power-down interrupt
5	INT_PCIMES	PCIEC message receive interrupt
[8:6]	INT_TX_VC0[2:0]	PCIEC VC0 transmit interrupts
[11:9]	INT_RX_VC0[2:0]	PCIEC VC0 receive interrupts
12	INTPM	PCIEC power management interrupt
13	INTMAC	PCIEC MAC interrupt
14	INTDL	PCIEC data link interrupt
15	INTTL	PCIEC TLP interrupt
16	RX_VC0_ERR	PCIEC VC0 receive error interrupt
17	RX_VCX_ERR	PCIEC VCX receive error interrupt
[20:18]	INT_TX_VCX[2:0]	PCIEC VCX transmit interrupts
[23:21]	INT_RX_VCX[0]	PCIEC VCX receive interrupts
24	INT_TX_CTRL	PCIEC transmit control interrupt
25	INT_RX_CTRL	PCIEC receive control interrupt
26	INT_RX_ERP	PCIEC receive error interrupt
27	INT_DMACE	PCIEC-DMAC error interrupt
28	INT_DMAC0	PCIEC-DMAC0 interrupt
29	INT_DMAC1	PCIEC-DMAC1 interrupt
30	INT_DMAC2	PCIEC-DMAC2 interrupt
31	INT_DMAC3	PCIEC-DMAC3 interrupt
[63:32]	INT_MSI[31:0]	MSI interrupts
[67:64]	INT_PCIINTX[3:0]	INTx interrupts

Each of interrupts 0 to 3 (INT_PCISERR, INT_PCIFERR, INT_PCINFERR, and INT_PCICERR) indicates that the PCIEC has detected the corresponding error. For details, refer to section 13.5.13, Error Processing.

Interrupt 4 (INT_PCIPOWER) is used in power management. For details, refer to section 13.5.12, Power Management.

Interrupt 5 (INT_PCIMES) indicates that the PCIEC has received a message. For details, refer to section 13.5.8, Message Transmission and Reception.

Interrupts 6 to 26 are used when the PCIEC controller is controlled directly by software without using the PCIEC bridge. When the bridge is used, these interrupts are not used.

Interrupt 27 (INT_DMACE) indicates that an error has occurred in the DMAC. Each of interrupts 28 to 31 (INT_DMAC0 to INT_DMAC3) indicates that a transfer in the corresponding PCIEC-DMAC channel (PCIEC-DMAC0 to PCIEC-DMAC3) has been completed. For details, refer to section 13.5.7, DMA Transfer.

Each of interrupts 32 to 63 (INT_MSI[31:0]) indicates that an MSI interrupt has occurred. For details, refer to section 13.5.10, MSI Interrupts.

Each of interrupts 64 to 67 (INT_PCIINTX[3:0]) indicates that an INTx interrupt has occurred. For details, refer to section 13.5.9, INTx Interrupts.

13.5.12 Power Management

This section describes the power management function supported by the PCIEC.

(1) Overview

The PCIEC supports two types of power management, PCI-PM and ASPM. The PCI-PM (PCI Compatible Power Management) is a power management mode compatible with the PCI bus in which the PCIEC is shifted to a low-power state under software instructions. The ASPM (Active State Power Management) is a low-power mechanism specific to the PCI Express, in which the hardware automatically enters a low-power state when no transfer is done.

For the PCI-PM, the standard specifies the L1, L2, and L3 states in addition to the L0 state (normal state). The PCIEC supports the L1 state.

For the ASPM, the standard specifies state transitions from L0 (normal state) to L0s and L1. The PCIEC supports the transition to the L0s state.

(2) PCI-PM (PCI Compatible Power Management)

In the PCI-PM mode, the PCIEC can be shifted to low-power mode under software control. The following describes the procedures of transmission to the L1 state supported by the PCIEC and return from the L1 state to the normal state.

(a) When the PCIEC operates as a Root Port

- Transition from the L0 state to the L1 state

  When the PCIEC operates as a Root Port, the PCIEC (Root Port) issues an instruction to the external device (Endpoint) to enter the L1 state. The instruction to the external device (Endpoint) and the subsequent state transition processing must be done by software. To enter the L1 state, use the following procedure to control the transition by software.

  — Sending a request for transition to L1

  Request a transition to L1 through a configuration write access to write H'1 (indicating the D1 state) to the PCIEPMCAP1[1:0].Power State bits in the external device (Endpoint), which then requests a transition to L1.

  — Checking reception of the request for transition to L1

  Check that the Completion in response to the configuration write has been received from the external device (Endpoint).

  — Checking reception of PM_ENTER_L1_DLLP

  Check that the PCIEPMSR[23]. PMEnterL1RX bit is set to 1; that is, PM_ENTER_L1_DLLP from the external device (Endpoint) has been received.

  — Changing the power state

  Write H'1 (indicating the D1 state) to the PCIEPMCAP1[1:0].Power State bits in the PCIEC (Root Port).

  — Checking the L0 state

  Read the PCIEPMSR[18:16].PMSTATE bits in the PCIEC (Root Port) to check that they are set to the L0 state (H'1).

  — Specifying transition to the L1 state

  Write 1 to the PCIEPMCTLR[31].L1 Initiation bit in the PCIEC (Root Port) to instruct the PCIEC controller to enter the L1 state.

  — Checking transition to the L1 state

  Wait until the PCIEPMSR[31].L1FALLEdge bit in the PCIEC (Root Port) becomes 1 to check that the sequence of transition to L1 has been completed or suspended.

  After the PCIEPMSR[31].L1FALLEdge bit becomes 1, write 1 to the PCIEPMSR[31].L1FALLEdge bit to clear it.

Read the PCIEPMSR[18:16].PMSTATE bits in the PCIEC (Root Port) to check the current state after transition. If L1 (H'3) is read, the transition to L1 has been completed; if L0 (H'1) is read, the transition has been suspended.

- Return from the L1 state to the L0 state
  - — Requesting return from the PCIEC (Root Port)

    To issue a request from PCIEC (Root Port) to return to L0, the PCIEPMCAP1[1:0].Power State bits in the configuration register must be set to H'0 both in the PCIEC and external device. First, modify the PCIEPMCAP1[1:0].Power State bits in the configuration register of the PCIEC (Root Port) to H'0 (indicating the D0 state) through access via the SuperHyway bus. Next, write H'0 (indicating the D0 state) to the PCIEPMCAP1[1:0].Power State bits in the external device (Endpoint) through configuration write.

    After that, packet transmission and reception through PIO transfer or DMA transfer become available.
  - — Requesting return from the external device (Endpoint)

    To issue a request from the external device (Endpoint) to return to L0, the external device must write H'0 (indicating the D0 state) to the PCIEPMCAP1[1:0].Power State bits in the configuration register in the external device, and then transmit the PM_PME message to the PCIEC.

    Upon receiving the PM_PME message from the external device (Endpoint), the PCIEC must write H'0 (indicating the D0 state) to the PCIEPMCAP1[1:0].Power State bits in the configuration register in the PCIEC. Writing D0 to the Power State bits enables packets to be received from the Endpoint. When the Power State bits are not set to D0, the PCIEC handles any packet from the external device (Endpoint) except a message as an error (unsupported request).

    For reception of the PM_PME message, refer to section 13.5.8 (3), Message Reception.

(b) When the PCIEC operates as an Endpoint

- Transition from the L0 state to the L1 state

  When the PCIEC operates as an Endpoint, the PCIEC starts transition to the L1 state under a power-down instruction from the external device (Root Port). Transition from L0 to L1 requires control by software. The following describes the software processing required between a power-down instruction and a transition to L1.
  - — Receiving a request for transition to the L1 state

    When the external device (Root Port) writes H'1 (indicating the D1 state) to the PCIEPMCAP1[1:0].Power State bits in the PCIEC (Endpoint) through configuration write, the PCIEC receives a request for transition to L1. In the PCIEC, the change of the

PCIEPMCAP1[1:0].Power State value causes an INT_PCIPOWER interrupt. Detect this interrupt and start the processing of transition to L1.

— Waiting for completion of TLP transmission

Wait until the TLP in the transmit-wait state in the PCIEC controller has been transmitted; that is, wait until the PCIETXSR[63].TXEMP bit is read as 1.

After that, control the PCIEC by software so that no new TLP is transmitted until transition to L1 is completed.

— Returning the Completion in response to the configuration write

Write to the PCIEPWRCTLR[0].RCPL bit to send a Completion.

For most configuration requests, the PCIEC automatically returns a Completion. However, the PCIEC does not generate a Completion in response to the configuration write to the PCIEPMCAP1[1:0].Power State bits.

— Checking the L0 state

Read the PCIEPMSR[18:16].PMSTATE bits to check that they are set to the L0 state (H'1).

— Specifying transition to the L1 state

Write 1 to the PCIEPMCTLR[31].L1 Initiation bit to instruct the PCIEC controller to enter the L1 state.

— Checking transition to the L1 state

Wait until the PCIEPMSR[31].L1FALLEdge bit becomes 1 to check that the sequence of transition to L1 has been completed or suspended.

After the PCIEPMSR[31].L1FALLEdge bit becomes 1, write 1 to the PCIEPMSR[31].L1FALLEdge bit to clear it.

Read the PCIEPMSR[18:16].PMSTATE bits to check the current state after transition. If L1 (H'3) is read, the transition to L1 has been completed; if L0 (H'1) is read, it has been suspended.

- Return from the L1 state to the L0 state

— Requesting return from the PCIEC (Endpoint)

Return to the L0 state by writing H'0 (indicating the D0 state) to the PCIEPMCAP1[1:0].Power State bits in the PCIEC (Endpoint) to send the PM_PME message to the external device (Root Port).

For transmission of the PM_PME message, refer to section 13.5.8 (2) Message Transmission.

— Requesting return from the external device (Root Port)

When the external device (Root Port) writes H'0 (indicating the D0 state) to the PCIEPMCAP1[1:0].Power State bits in the configuration register through a configuration

write access, the PCIEC returns to the L0 state. When the PCIEPMCAP1[1:0].Power State bits are not set to D0, the PCIEC handles any received packet except a configuration access and a message as an error (unsupported request).

(3) ASPM (Active State Power Management)

When the ASPM mode is selected, the PCIEC is automatically shifted to the low-power mode by hardware. To use the ASPM mode, set the PCIEEXPCAP4[1:0].ASPM Control bits to H'1. Setting them to H'0 disables transition to the low-power mode through ASPM. In the initial state, these bits are set to H'0.

- Transition from the L0 state to the L0s state

  When the ASPM mode is selected, the PCIEC starts the sequence of transition to the L0s state when the idle state is detected for 4 μs. The idle state is detected when the following conditions are satisfied.

  (a) No TLP is being transmitted.

  (b) No TLP remains in the transmit-wait state in the PCIEC controller.

  (c) A TLP in the transmit-wait state cannot be sent because of insufficient credit in the flow control in the data link layer.

  (d) Completion has not been received in response to a transmitted Non-Posted request.

  (e) No DLLP is in the transmit-wait state.
- Return from the L0s state to the L0

  In the ASPM mode, the PCIEC automatically enters the L0 state when a packet to be transferred is generated.

13.5.13 Error Processing

This section describes the processing of errors detected by the PCIEC.

(1) Error Types

The PCIEC classifies detected errors into four types: three types defined in the PCI Express standard (correctable errors, non-fatal errors, and fatal errors) and other errors (system errors). It reports each type of error through the corresponding interrupt.

(a) Correctable Error

This type of error can be corrected through the PCI Express protocol. As the error is corrected by hardware, error processing by software is not needed in most cases.

When the conditions described later are satisfied, the PCIEC reports occurrence of a correctable error through the INT_PCICERR interrupt.

(b) Non-Fatal Error

This type of error cannot be corrected through the PCI Express protocol, but it is not a fatal error. After this error, a single packet is discarded but the subsequent data communication can be continued. In most cases, error processing by software such as retransmission of the discarded packet is needed.

When the conditions described later are satisfied, the PCIEC reports occurrence of a non-fatal error through the INT_PCINFERR interrupt.

(c) Fatal Error

This type of error cannot be corrected through the PCI Express protocol, and the communication channel must be reset to recover from the error. In most cases, the PCIEC must be initialized.

When the conditions described later are satisfied, the PCIEC reports occurrence of a fatal error through the INT_PCIFERR interrupt.

(d) Other Errors Detected by PCIEC (System Error)

The errors that do not fall into the above types are classified as system errors. The necessary processing depends on the cause of each error.

The PCIEC reports this type of error through the INT_PCISERR interrupt.

(2) Priority of Errors

Upon detecting multiple errors during reception of a single packet, the PCIEC reports only one error that the PCIEC determines as the most important according to the following priority among errors.

- Receiver Overflow
- Flow Control Protocol Error
- Malformed TLP
- Unsupported Request (UR), Completer Abort (CA), or Unexpected Completion
- Poisoned TLP Received

(3) Correctable Error

After a correctable error, communication can be recovered through hardware control, such as by an automatic retransmission request, without losing any transfer data. Therefore, recovery processing by software is usually not needed.

When a correctable error occurs, the PCIEEXPCAP2[16].Correctable Error Detected bit is set to 1. At this time, if the PCIEEXPCAP2[0].Correctable Error Reporting Enable bit is 1, the PCIEC operating as an Endpoint will send the ERR_COR message to report the error to the Root Port, and the PCIEC operating as a Root Port will generate an INT_PCICERR interrupt if the PCIEEXPCAP7[0].System Error on Correctable Error Enable bit is set to 1.

If the PCIEC operating as a Root Port receives the ERR_COR message from the Endpoint while the PCIEPCICONF15[17].SERR Enable bit is set to 1, the PCIEEXPCAP2[0].Correctable Error Reporting Enable bit is set to 1, and the PCIEEXPCAP7[0].System Error on Correctable Error Enable bit is set to 1, it will generate an INT_PCICERR interrupt.

The following errors are classified as correctable errors.

(a) Receiver error

8b/10b decode error

Disparity error

Elastic buffer overflow

Elastic buffer underflow

(b) BAD TLP

LCRC error

Sequence number error

(c) BAD DLLP

16-bit CRC error

(d) Replay timeout

Ack/Nak timeout

(e) Replay Num rollover

Replay Num rollover

(4) Non-Fatal Error

A non-fatal error cannot be corrected through hardware control and one packet of data will be lost. Therefore, recovery processing by software, such as retransmission processing, is needed.

When a non-fatal error occurs, the PCIEEXPCAP2[17].Non Fatal Error Detected bit is set to 1. In addition, the PCIEPCICONF1[30].Signaled System Error bit is set to 1 if the PCIEPCICONF1[8].SERR Enable bit is 1. When a non-fatal error occurs while the PCIEEXPCAP2[1].Non Fatal Error Reporting Enable bit is set to 1 or the PCIEPCICONF1[8].SERR Enable bit is set to 1, the PCIEC operating as an Endpoint will send the ERR_NONFATAL message to report the error to the Root Port, and the PCIEC operating as a Root Port will generate an INT_PCINFERR interrupt if the PCIEEXPCAP7[1].System Error on Non Fatal Error Enable bit is set to 1. However, if the error source is an unsupported request (described later), the PCIEC does not send the message or generate an interrupt when the conditions described later are satisfied.

When the PCIEC operating as a Root Port receives the ERR_NONFATAL message, it sets the PCIEPCICONF7[30].Received System Error bit to 1. If it receives the message while the PCIEPCICONF15[17].SERR Enable bit is set to 1 and the PCIEPCICONF1[8].SERR Enable bit is set to 1, it sets the PCIEPCICONF1[30].Signaled System Error bit to 1. If it receives the message while the PCIEPCICONF15[17].SERR Enable bit is set to 1, the PCIEEXPCAP2[1].Non Fatal Error Reporting Enable or PCIEPCICONF1[8].SERR Enable bit is set to 1, and the PCIEEXPCAP7[1].System Error on Non Fatal Error Enable bit is set to 1, it will generate an INT_PCINFERR interrupt. However, if the error source is an unsupported request (described later), the PCIEC does not generate an interrupt when the conditions described later are satisfied. In addition, in an advisory non-fatal error case (described later), the PCIEC performs different processing specified separately.

The following errors are classified as non-fatal errors.

(a) Poisoned TLP Reception

If a packet with the Poisoned attribute is written, it is handled as a non-fatal error.

When the PCIEC receives a write request with a packet having the Poisoned attribute from the PCI Express side, it does not transfer the request to the SuperHyway side. In this case, the PCIEC records the header of the packet in error header registers 0 to 3 (PCIEEH0R, PCIEEH1R, PCIEEH2R, and PCIEEH3R).

When the Completion received in response to a request issued by the PCIEC has the Poisoned attribute, the PCIEC does not transfer the data included in the Completion to the SuperHyway side.

(b) Unsupported Request (UR) Reception

The following requests are handled as unsupported requests.

- Access to the area outside the PCI address area specified by BAR
- Reception of a memory/IO request while the PCIE operates as a Root Port and the PCIEPCICONF1[2].Bus Master Enable bit is 0
- Access to the memory space while the PCICONF1[1].Memory Space Enable bit is 0
- Access to the IO space while the PCICONF1[0].IO Space Enable bit is 0
- Reception of a message with an undefined message code
- Reception of a Vendor Defined message of type 0
- Reception of a TLP that is not a configuration access or a message while not in the D0 state
- Access to an unimplemented configuration register
- Reception of a configuration request of type 1
- Reception of a packet without the No Snoop attribute while reception of the snoop transaction is prohibited through the VCCAP4[15].ReJeCT SNoop Transactions bit setting
- Reception of an MRdLk request

When the PCIEC receives an unsupported request, it sets the PCIEEXPCAP2[19].Unsupported Request Detected bit to 1.

At this time, if the PCIEEXPCAP2[3].Unsupported Request Reporting Enable bit is 0 and the PCIEPCICONF1[8].SERR Enable bit is 0, the PCIEC will not transmit a message or generate an interrupt.

Upon receiving an unsupported request, the PCIEC records the header of the received data in error header registers 0 to 3 (PCIEEH0R, PCIEEH1R, PCIEEH2R, and PCIEEH3R).

(c) Completion Timeout

When the PCIEC does not receive a Completion in response to a request that the PCIEC has issued to the PCI Express side, the PCIEC executes error processing for Completion timeout.

(d) Completer Abort (CA)

The PCIEC executes error processing for completer abort when the destination, in the internal bus space, of a packet sent from the PCI Express is either a PCI area of the internal bus space of the

PCIEC module or a PCIEC control register that is not allowed to be accessed by the PCI Express side, or when an error occurs on the SuperHyway bus during transfer of a packet from the PCI Express to the SuperHyway bus. When the received packet from the PCI Express is a Non-Posted request, the PCIEC returns a Completion having Completer Abort (CA)– Completion Status and sets the PCIECONF1[28].Signaled Target Abort bit to 1.

When the received packet from the PCI Express violates the PCI Express standard or is damaged, the PCIEC does not handle it as a completer abort but handles it as an unsupported request or a malformed TLP instead.

(e) Unexpected Completion Reception

When the received Completion is not valid as a response to the request that the PCIEC has issued, the PCIEC handles it as an unexpected Completion and executes error processing.

Reception of an unexpected Completion is one of the advisory non-fatal error cases described later.

(5) Fatal Error

A fatal error cannot be corrected through hardware control. A fatal error usually requires system-level recovery processing, such as initialization of the PCI Express. After a fatal error, correct data transfer is not possible without re-initialization.

When a fatal error occurs, the PCIEEXPCAP2[18].Fatal Error Detected bit is set to 1. In addition, the PCIEPCICONF1[30].Signaled System Error bit is set to 1 if the PCIEPCICONF1[8].SERR Enable bit is 1. When a fatal error occurs while the PCIEEXPCAP2[2].Fatal Error Reporting Enable bit is set to 1 or the PCIEPCICONF1[8].SERR Enable bit is set to 1, the PCIEC operating as an Endpoint will send the ERR_FATAL message to report the error to the Root Port, and the PCIEC operating as a Root Port will generate an INT_PCINFERR interrupt if the PCIEEXPCAP7[2].System Error on Fatal Error Enable bit is set to 1.

When the PCIEC operating as a Root Port receives the ERR_FATAL message, it sets the PCIEPCICONF7[30].Received System Error bit to 1. If it receives the message while the PCIEPCICONF15[17].SERR Enable bit is set to 1 and the PCIEPCICONF1[8].SERR Enable bit is set to 1, it sets the PCIEPCICONF1[30].Signaled System Error bit to 1. If it receives the message while the PCIEPCICONF15[17].SERR Enable bit is set to 1, the PCIEEXPCAP2[2].Fatal Error Reporting Enable or PCIEPCICONF1[8].SERR Enable bit is set to 1, and the PCIEEXPCAP7[2].System Error on Fatal Error Enable bit is set to 1, it will generate an INT_PCIFERR interrupt.

The following errors are classified as fatal errors.

(a) Data Link-Layer Protocol Error

(b) Flow Control Protocol Error

(c) Malformed TLP Reception

In the following cases, packets are determined as malformed TLPs. When such a packet is received, a fatal error occurs and the PCIE records the header of the packet in error header registers 0 to 3 (PCIEEH0R, PCIEEH1R, PCIEEH2R, and PCIEEH3R).

- Receiving a packet with data equal to or larger than Max Payload Size
- Receiving a packet with data length different from the size field value
- Receiving a TLP whose TD field of the header is 0 and with a digest field
- Receiving a TLP whose TD field of the header is 1 and without a digest field
- Receiving an Assert_INTx/Deassert message that does not use TC0
- Receiving a power management message that does not use TC0
- Receiving an error signal message that does not use TC0
- Receiving an unlock message that does not use TC0
- Receiving a Set_Slot_Power_Limit message that does not use TC0
- Receiving a Completion with data size different from the Length field value
- Receiving a Completion with data size larger than Max_Payload_Size
- Receiving a TLP of undefined TLP type or format
- Receiving a TLP using a TC that is not assigned
- Receiving a TLP for a disabled VC
- Receiving an IO request with TC != 0, Attr[1:0] != 0, Length[9:0] != 1, or Last DW[3:0] != 0
- Receiving a configuration request with TC != 0, Attr[1:0] != 0, Length[9:0] != 1, or Last DW[3:0] != 0
- Receiving an Assert_INTx/Deassert_INTx message during operation as an Endpoint

(6) System Error

This type includes errors that are not classified as any of the correctable, non-fatal, and fatal errors.

When a system error occurs, the PCIEC will generate an INT_PCISERR interrupt.

(a) DL_Down

DL_Down is the case when communication in the data link layer fails after the layer becomes active. DL_Down occurs when the data link layer is intentionally made inactive, but also occurs when the communication state of the channel seriously deteriorates during normal data transfer.

After a DL_Down, the PCIEC module must be initialized and a connection with the target device must be established again.

When a DL_Down occurs, all packets being transferred are discarded. The data transferred through the packets discarded due to DL_Down is not guaranteed; initialize the PCIEC after a DL_Down and transfer the data again. When the transfer is a DMA transfer, make settings in the DMAC control registers again and re-execute the DMA transfer.

When a DL_Down occurs, the PCIETCTLR[3].DLDOWN bit is set to 1. Read this bit to check whether a DL_DOWN has occurred if an INT_PCISERR interrupt occurs.

(b) CRS (Configuration request Retry Status) Reception

If the destination device has not been activated, a Completion with CRS Completion Status will be returned in response to a configuration request issued by the PCIEC. When a CRS is received while the PCIEEXPCAP7[4].CRS Software Visibility Enable bit is set to 1, the PCIEC generates an INT_PCISERR interrupt.

Upon receiving the CRS Completion Status, the PCIEC sets the PCIEPCTLR[16].CRS bit. Whether a CRS has been received can be checked by reading the PCIEPCTLR[16].CRS bit after issuing the first configuration request, without using the INT_PCISERR interrupt.

(The PCI Express standard prohibits the PCI Express device sending a CRS Completion once it returns a Completion other than CRS. Therefore, after a Completion other than CRS is received, it is not necessary to check CRS reception.)

(7) Advisory Non-Fatal Error Case

The PCI Express standard specifies that processing for some non-fatal errors must not be done for packet reception because the cause or importance of some non-fatal errors should be examined by the packet transmitter, software, or the system. Such non-fatal errors are the advisory non-fatal error cases described below. When the PCIEC detects an error included in these cases, it does not execute the usual non-fatal error processing, but performs the separate processing specified for each case.

(a) Return of Completion with UR/CA Status

When a Completion with UR/CA status is returned, the device receiving the Completion (the device that has issued a request) should determine the importance of the error; therefore, this case is specified as an advisory non-fatal error case.

When the PCIEC returns a Completion with UR/CA status, it does not execute the non-fatal error processing specified in the PCI Express standard; it records occurrence of the error in the

PCIEERRFR[0].SENDURCPL or PCIEERRFR[1].SENDCACPL bit and generates an INT_PCISERR interrupt if the corresponding bit in PCIEERRFER is set to 1.

(b) Completion Timeout

When no response to a request is received within a specified period and a Completion timeout occurs, the device or module issuing the request can determine whether to transmit the request again. The Completion timeout is specified as an advisory non-fatal error case to prohibit error processing by hardware because the number of re-transmissions can be specified by the software that controls request issuing.

When a Completion timeout occurs, the PCIEC does not execute the non-fatal error processing specified in the PCI Express standard; it records occurrence of the error in the PCIEERRFR[8].CPLTIMEOUT bit and generates an INT_PCISERR interrupt if the corresponding bit in PCIEERRFER is set to 1. When software determines that no more retransmission should be done, the software should start error processing such as transmission of ERR_COR.

(c) Unexpected Completion Reception

Reception of an unexpected Completion (Completion that is not valid in response to the transmitted response) is an advisory non-fatal error case. An unexpected Completion is usually caused by misrouting when a switching device is used. In this situation, the device that should have received this Completion must perform error processing according to the Completion timeout information, and therefore, reception of an unexpected Completion is specified as an advisory non-fatal error case.

When the PCIEC receives an unexpected Completion, it does not execute the non-fatal error processing specified in the PCI Express standard; it records occurrence of the error in the PCIEERRFR[12].UNEXPECTED COMPLETION bit and generates an INT_PCISERR interrupt if the corresponding bit in PCIEERRFER is set to 1.

(d) UR/CA Completion Reception

When a Completion with UR/CA status is received, an upper layer such as software should determine the importance of the error; therefore, this case is specified as an advisory non-fatal error case.

When the PCIEC receives a Completion with UR/CA status, it does not execute the non-fatal error processing specified in the PCI Express standard; it records occurrence of the error in the PCIEERRFR[4].RECEIVEURCPL or PCIEERRFR[5].RECEIVECACPL bit and generates an INT_PCISERR if the corresponding bit in PCIEERRFER is set to 1.

The PCI Express standard also specifies another advisory non-fatal error case: continued reception processing after reception of a poisoned TLP. In the PCIEC, continuing Poisoned data processing is not allowed, and therefore, this type of advisory non-fatal error never occurs. The PCI Express standard specifies still another case related to an intermediate receiver, but the PCIEC does not operate as an intermediate receiver and therefore does not execute the processing related to this case.

13.5.14 Access to Physical Layer Control Registers

This section describes the way to access physical layer control registers.

Do not change the value of physical layer control registers, except for bits allowed to do so. If the values of reserved register or bits are changed, the behavior of PCIEC modules including physical layer is not guaranteed.

(1) Abstract

The PCIEC module has registers for controlling the physical layer. These registers are mapped to the address space for physical layer registers, and they are accessed via the PHY control bus. The physical layer register address space is independent from the chip's physical and logical address spaces, and it is accessed by means of the PHY control bus via the registers of the PCIEC module.

This section describes the physical layer control registers and the PHY control bus used to access these registers.

(2) Physical Layer Control registers

The physical layer control registers are defined in the address space, which is spanned by 8-bit address and 4-bit lane number. The defined registers are shown in the Table 13.4. Do not access to the registers and bits, except for the bits defined in this table, or specially instructed to access.

(3) PHY control bus

The physical layer control register is accessed via the PHY control bus by using commands and ACK signals. The PHY control bus accesses the register by issuing read or write commands, and access is confirmed to be complete when an ACK is received. Control, including issuing of commands to the PHY control bus and ACK checking, is performed by software via the physical layer control register.

Table 13.17 lists the PHY control bus signals. Those signals are mapped on the PCIEC registers (PCIEPHYCTLR, PCIEPHYADRR, PCIEPHYDINR, PCIEPHYDOUTR), and should be

controlled by the software. To access to the physical layer control registers, perform initialize, read or write according to the sequence specified on figures 13.10 to 13.12.

Table 13.17 Signals for Physical Layer Control Register Access Bus

Signal	Bit Width	Corresponding register and bits	Meaning
CLKEN	1	PCIEPHYCTLR[0].PHYCKE	Clock Enable. When set to 1, a clock is supplied to the PHY control bus.
Reset	1	PCIEPHYCTLR[31].PHYRST	Reset. When set to 1, initialization takes place.
Address	8	PCIEPHYADRR[7:0].PHYADDR	Address. Specify the address of registers, in the physical layer address space.
Command	2	PCIEPHYADRR[17:16].PHYCMD	Specify the command. 2'b00: Idle 2'b01: Write 2'b10: Read 2'b11: Reserved (do not use)
DataIn	32	PCIEPHYDINR[31:0].PHYDIN	Write data. Specify the write data, when write command is issued.
LaneEn	4	PCIEPHYADRR[11:8].PHYLANE	Lane. Specify the lane to be accessed. More than one lane can be specified for write access. Only one lane can be specify for read access.
Ack	1	PCIEPHYADRR[24].PHYACK	Acknowledge signal. Indicate the acknowledge signal, which means the completion of bus access.
DataOut	32	PCIEPHYDOUTR[31:0].PHYDOUT	Read Data. Indicates the read data, when read command has completed.

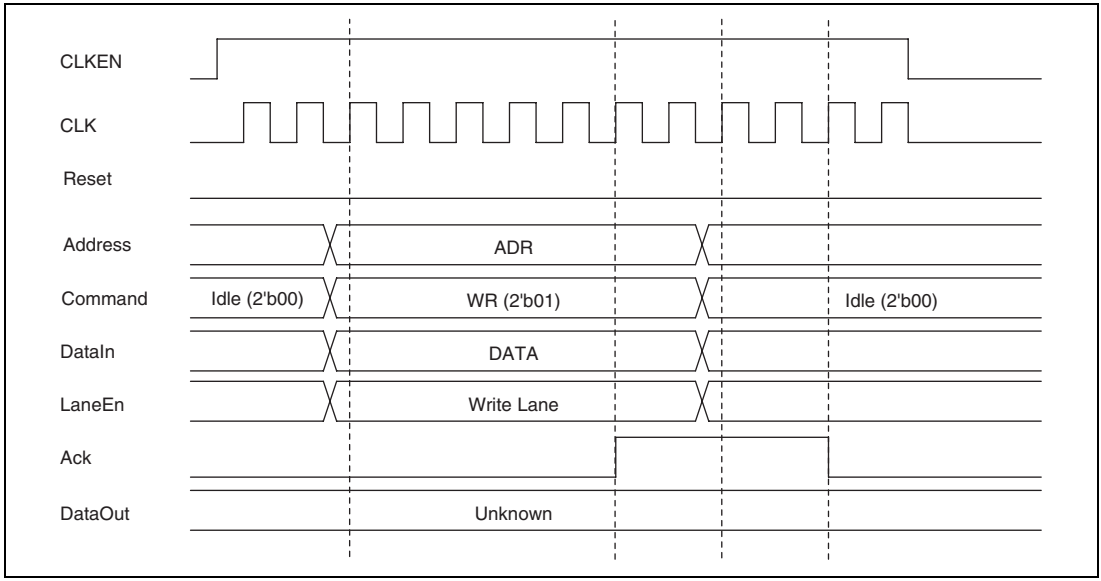


Figure 13.10 Write Sequence for Physical Layer Control Register Access Bus

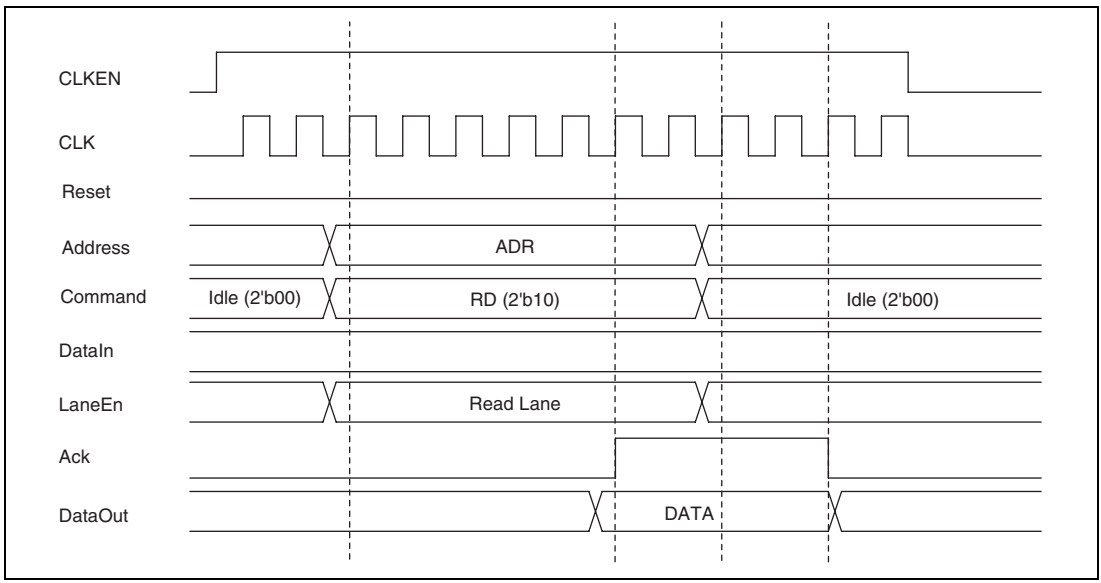


Figure 13.11 Read Sequence for Physical Layer Control Register Access Bus

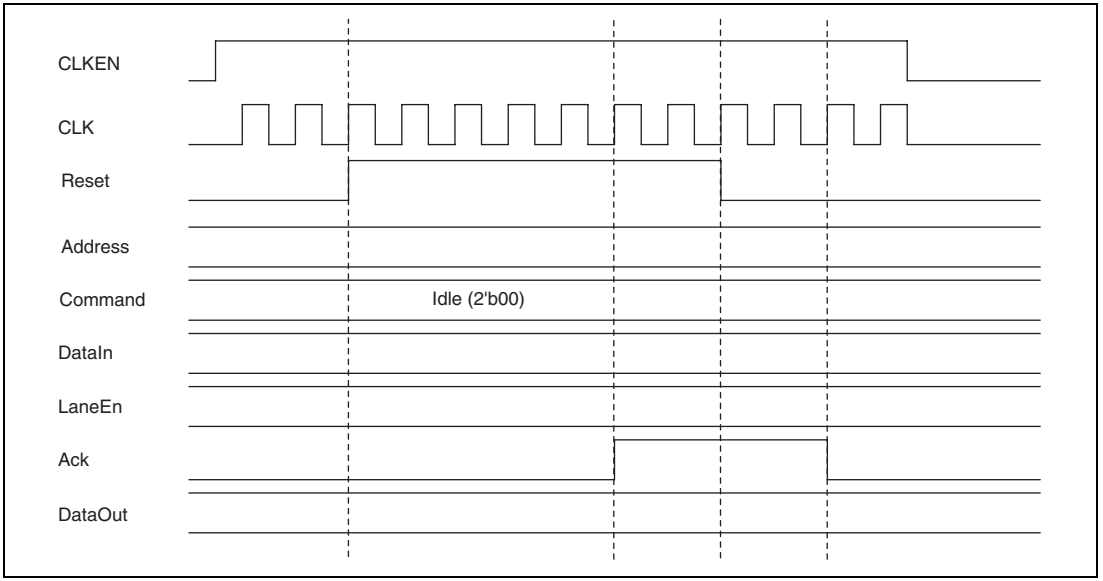


Figure 13.12 Initialization Sequence for Physical Layer Control Register Access Bus

(4) Start up process

At the start-up of PCIEC module, set the registers of physical layer control registers below as an initialization of PCIEC physical layer, before using PCIEC module.

In the case of x4(PCIEC0) + x1(PCIEC1) mode, perform the initialization to PCIEC0 if PCIEC0 is used, and perform the initialization to PCIEC1 if PCIEC1 is used.

In the case of x2(PCIEC0) + x1(PCIEC1) + x1(PCIEC2) mode, perform the initialization to PCIEC0 if PCIEC0 or PCIEC2 is used, and perform the initialization to PCIEC1 if PCIEC1 is used.

Table 13.17 Initialization of Physical Layer Control Register

Order	Address	Lane(PCIEC0)	Lane(PCIEC1)	Write Data
1	H'60	0-3	0	H'004B_008B
2	H'61	0-3	0	H'0000_7B41
3	H'64	0-3	0	H'00FF_4F00
4	H'65	0-3	0	H'0907_0907
5	H'66	0-3	0	H'0000_0010
6	H'74	0-3	0	H'0007_001C
7	H'79	0-3	0	H'01FC_000D
8	H'B0	0-3	0	H'0000_0610
9	H'67	0	0	H'0000_0400

13.5.15 Software Reset

The following describes how to initialize the bridge and controller in the PCIEC module by software.

(1) Purpose

A software reset should be used to recover this module from a disordered state caused by an abnormality such as a hardware failure. When a software reset is applied, the packets being transferred are discarded.

(2) Software Reset Procedure

- Set CFINIT to 0.
- Reset the Phy and then cancel the reset.
- Assert a software reset for PCIEC --> Deassert the reset
- Assert a software reset for SPW --> Deassert the reset

Write 1 to PCIESRSTR[0].SRST.

Write 0 to CFINIT.

Write 0 to PCIESRSTR[0].SRST.

Write 1 to PCIETXVC0SR[63].TXBUSCLR to initialize the PCIEC controller.

After the above steps, initialize the PCIEC.

13.5.16 Related Documents

- "PCI Express Base Specification Revision 1.1", March 28 2005, PCI-SIG
- "PCI Bus Power Management Interface Specification Revision 1.2", March 3 2004, PCI-SIG
- "PCI Local Bus Specification Revision 3.0", February 3 2004, PCI-SIG

Section 14 USB

14.1 USB

14.1.1 Features

As shown figure 14.1, USB Circuit Diagram. The USB module is constructed of USB-LINK (EHCI/OHCI/FUNCTION)/PHY, respective interfaces and common registers (REGS). This Chapter describes about the common registers (REGS) and USB-PHY external circuits. For USB-LNK (EHCI/OHCI/FUNCTION) registers, refer to register descriptions in section 14.2 to 14.4.

This module supports the following functions:

- Initialization of each USB module
- Port: Switching host and function (On-The-Go function not supported)
- Polarity: Switching OVC and VBUS

14.1.2 Circuit Diagram

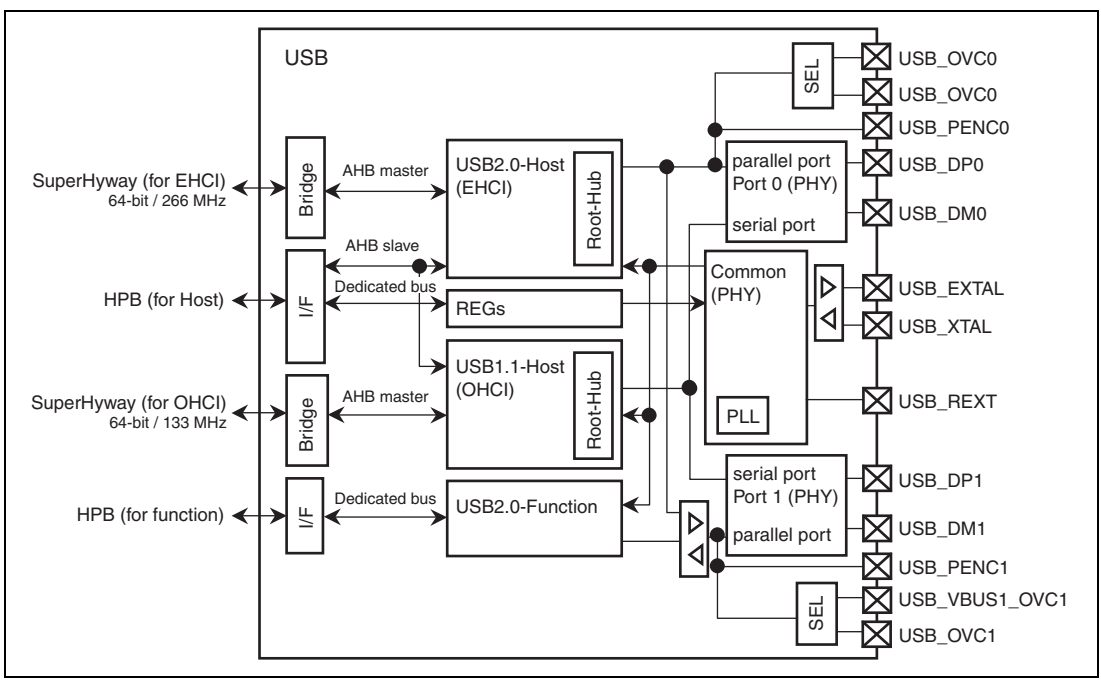


Figure 14.1 USB Circuit Diagram

14.1.3 External Pins

Table 14.1 Pin Configuration

Pin Name	I/O	Description
USB_EXTAL	Input	PLL oscillation clock, the input pin of external clock.
USB_XTAL	Input	PLL oscillation clock
USB_DP0	I/O	USB port 0 D+
USB_DM0	I/O	USB port 0 D-
USB_PENC0	Output	Port Enable 0: The initial value is 0. Host: Power supply control 1: Power supply IC is turned on. 0: Power supply IC is turned off.
USB_OVC0	Input	Over Current 0 Host: Power supply IC overcurrent detection Active-low
USB_OVC0*	Input	Over Current 0 Host: Power supply IC overcurrent detection Active-low, and active mode can be changed by register.
USB_DP1	I/O	USB port 1 D+
USB_DM1	I/O	USB port 1 D-
USB_PENC1	Output	Port Enable 1: The initial value is 0. Host: Power supply control 1: Power supply IC is turned on. 0: Power supply IC is turned off. Function: Pull-up enable 1: Enable 0: Disable
USB_VBUS1_OVC1	Input	VBUS/Over Current 1 Host: Power supply IC overcurrent detection, Function: Cable connection/disconnection detection Active-low
USB_OVC1*	Input	VBUS/Over Current 1 Host: Power supply IC overcurrent detection, Function: Cable connection/disconnection detection Active-low, and active mode can be changed by register.
USB_REXT	I/O	A pin for connecting the external resistor.

Note: * These pins are multiplexed with the DMAC0 pins.

14.1.4 Register Descriptions

The registers can be accessed only by the CPU. When they are accessed by other modules, operation is not guaranteed. Bit widths of these resisters are all 32 bits, so should be accessed in longword (32 bits) units. Access in other units is not supported.

Table 14.2 List of Registers

Register Name	Abbreviation	R/W	Address	Initial Value
Port Control 0	USBPCTL0	R/W	H'FFE7 0800	H'xxxxxxxx*[1]
Port Control 1	USBPCTL1	R/W	H'FFE7 0804	H'xxxxxxxx*[1]
Port Status	USBST	R	H'FFE7 0808	H'xxxxxxxx*[1]
EHCI Control 0	USBEH0	R/W	H'FFE7 080C	H'00000000
OHCI Control 0	USBOH0	R/W	H'FFE7 081C	H'00000000
USB Control 0	USBCTL0	R/W	H'FFE7 0858	H'00000224

Note: Addresses other than the above must not be write-accessed to. If written to, operation is not guaranteed. If read, an undefined value is read.

*1 For details on the initial value, see the descriptions of the individual registers.

[Legend for symbols used in register descriptions]

Initial value: The value in the register after a reset

—: Undefined value

R/W: The bit is readable and writable. The written value can be read.

R: The bit is read-only. The write value should always be 0.

W: The bit is write-only. The read value is undefined.

(1) Port Control 0 (USBPCTL0)

USBPCTL0 is a register that sets the port function.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	VBUS/OVC1	OVC0	—	—	—	PENC	OVC0_ACT	—	OVC1_ACT	POTR1
Initial value:	—	—	—	—	—	—	0	0	—	—	—	0	0	—	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	—	R	Reserved Setting other than 0 is prohibited.
9	VBUS/OVC1	0	R/W	Selects the input pin of VBUS/OVC1. 0: USB_VBUS1_OVC1 1: USB_OVC1*
8	OVC0	0	R/W	Selects the input pin of OVC0. 0: USB_OVC0 1: USB_OVC0*
7 to 5	—	—	R	Reserved Setting other than 0 is prohibited.
4	PENC	0	R/W	The bit for PENC1 at Function mode. 0: Output Low 1: Output High
3	OVC0_ACT	0	R/W	When the value of the bit 9 is 1: Host mode: OVC0 polarity inversion 0: Active Low 1: Active High
2	—	—	R	Reserved Setting other than 0 is prohibited.

Bit	Bit Name	Initial Value	R/W	Description
1	OVC1_ACT	0	R/W	When the value of the bit 8 is 1: Host mode: OVC1 polarity inversion 0: Active Low 1: Active High Function mode: VBUS1 polarity inversion 0: zero/one corresponds to connected/disconnected 1: zero/one corresponds to disconnected/connected
0	PORT1	0	R/W	Selects host or function for port 1. 0: Host (initial value) 1: Function

Note: * These pins are multiplexed with the DMAC0 pins.

(2) Port Control 1 (USBPCTL1)

USBPCTL1 is a register that is used for setting the port function or applying a software-reset. The target registers that are initialized when the RST bit is used are all USB registers except USBCTL, USBSTT.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RST	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PHY RST	PLL ENB	PHY ENB
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	RST	0	R/W	Resets the USB module when this bit is set to 1. To restore the USB module from the reset state, clear this bit to 0.
30 to 3	—	—	R	Reserved
2	PHY RST	0	R/W	Resets the USB-PHY. To use the USB module, set this bit to 1 after setting the PHY ENB to 1 and the PLL ENB to 1 so the PLL frequency is stabilized.
1	PLL ENB	0	R/W	Enables the PLL in the USB-PHY. To use the USB module, set this bit to 1 after setting the PHY ENB to 1.
0	PHY ENB	0	R/W	Enables the USB-PHY. To use the USB module, set this bit to 1.

Procedure of using USB module:

1. Set the USBPCTL1.PHY ENB (bit[0]) to 1.
2. Set the USBPCTL1.PLL ENB (bit[1]) to 1.
3. Confirm the USBST.PLL (bit[30]) and USBST.ACT (bit[31]) are set to 1.
4. Set the USBPCTL1.PHY RST (bit[2]) to 1.

(3) Port Status (USBST)

USBST is a register that indicates the USB module state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ACT	PLL	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	ACT	—*[1]	R	Indicates the USB module state. Confirm both ACT (bit[31]) and PLL (bit[30]) are set to 1 before setting registers of the USB module. 1: The USB module is active. 0: The USB module is initializing.
30	PLL	—*[2]	R	Indicates the USB PLL state. Confirm both ACT (bit[31]) and PLL (bit[30]) are set to 1 before setting registers of the USB module. 1: PLL oscillation is stable. 0: PLL oscillation is not stable.
29 to 0	—	—	R	Reserved Setting other than 0 is prohibited.

Notes: 1. Indicates 0 for approximately 1ms after the LSI starts up.
2. Clock input from USB_EXTAL/USB_XTAL is necessary for stabilization of PLL oscillation.

(4) EHCI Control 0 (USBEH0)

USBEH0 is a register that controls data alignment in the bridge section on the EHCI side.

For the set values, see section 14.1.5, Initial Settings.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0		H'00000000	R/W	Bus alignment control register For details, refer to section 14.1.5 Initial Settings.

(5) OHCI Control 0 (USBOH0)

USBOH0 is a register that controls data alignment in the bridge section on the OHCI side.

For the set values, see section 14.1.5, Initial Settings.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0		H'00000000	R/W	Bus alignment control register For details, refer to section 14.1.5 Initial Settings.

(6) USB Control 0 (USBCTL0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLK SEL	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved
9	—	1	R	Reserved
8	—	0	R	Reserved
7	CLKSEL	0	R/W	USB clock mode select 0: Crystal resonator (default) 1: External clock input
6	—	0	R	Reserved
5	—	1	R	Reserved
4, 3	—	All 0	R	Reserved
2	—	1	R	Reserved
1, 0	—	All 0	R	Reserved

14.1.5 Initial Settings

The following settings are necessary for using the USB modules.

- Setting 1
  Address: H'FFE70094
  Initial value: H'00400040
  Write value: H'00FF0040
- Setting 2
  Address: H'FFE7009C
  Initial value: H'00000000
  Write value: H'00000001

The following settings should be ensured before accessing the EHCI/OHCI/FUNCTION registers.

- Setting 1
  Address: H'FFE70804 (USBPCTL1)
  Write value: H'00000001 (Set the PHY ENB bit to 1 to release its standby mode.)
- Setting 2
  Address: H'FFE70804 (USBPCTL1)
  Write value: H'00000003 (Set the PLL ENB bit to 1 to operate internal PLL circuit.)
- Confirmation
  Address: H'FFE70808 (USBST)
  Bit 31=B'1 (USB is active state.)
  Bit 30=B'1 (USB PLL oscillation is stable.)
- Setting 3
  Address: H'FFE70804 (USBPCTL1)
  Write value: H'00000007 (Set the PHY RST bit to 1 to release the USB-PHY internal logic from reset state.)

The following settings are necessary for bus alignment processing.

Bus alignment on the EHCI side

- Settings
  Address: H'FFE7080C (USBEH0)
  Initial value: H'00000000
  Write value: See the table below.

Endian	Swap Setting	Register Value	Remarks
Big endian	No swap	H'0000_0003	Usually set this value.
	Byte swap	H'0000_0002	
	Word swap	H'0000_0001	
	Word-byte swap	H'0000_0000	
Little endian	No swap	H'0000_0000	Usually set this value.
	Byte swap	H'0000_0001	
	Word swap	H'0000_0002	
	Word-byte swap	H'0000_0003	

Bus alignment on the OHCI side

- Setting
  Address: H'FFE7081C (USBOH0)
  Initial value: H'00000000
  Write value: See the table below.

Endian	Swap Setting	Register Value	Remarks
Big endian	No swap	H'8800_0003	Usually set this value.
	Byte swap	H'8800_0002	
	Word swap	H'8800_0001	
	Word byte swap	H'8800_0000	
Little endian	No swap	H'0000_0000	Usually set this value.
	Byte swap	H'0000_0001	
	Word swap	H'0000_0002	
	Word byte swap	H'0000_0003	

14.1.6 Examples of Handling Unused Pins

- USB_DP0/1, USB_DM0/1 handling example
  open when unused these ports.
- USB_OVC0/1, USB_PENC0/1 handling example
  USB_OVC0/1: Not active when host mode, disconnect when function mode.
  (Example: Pulled-up when low-active setting in host mode.)
  USB_PENC0/1 should be open.
  Note that, USB_OVC0/1 and USB_PENC0/1 can be used as GPIO. When using as GPIO, refer to the GPIO handling.

14.2 USB2.0-HOST Controller

14.2.1 Overview

This document describes the EHCI standard. Refer to this document when developing USB host systems. For details on the EHCI standard, see the “Enhanced Host Controller Interface Specification for Universal Serial Bus Revision 1.0.”

14.2.2 Register Descriptions

The registers can be accessed only by the CPU. When they are accessed by other modules, operation is not guaranteed. Bit widths of these resisters are all 32 bits, so should be accessed in longword (32 bits) units. Access in other units is not supported.

The address is obtained by H'FFE70000 + offset.

Table 14.3 Host Controller Capability Registers

Offset	31 16	15 8	7 0
0	HCIVERSION	Reserved	CAPLENGTH
4	HCSPARAMS		
8	HCCPARAMS		
C	HCSP-PORTROUTE		

The address is obtained by H'FFE70010 + offset.

Table 14.4 Host Controller Operational Registers

Offset	31 0
0	USBCMD
4	USBSTS
8	USBINTR
C	FRINDEX
10	CTRLDSSEGMENT
14	PERIODICLISTBASE
18	ASYNCLISTADDR
1C to 3F	Reserved
40	CONFIGFLAG
44	PORTSC (1 – N_PORT)

Note: Addresses other than the above must not be write-accessed to. If written to, operation is not guaranteed. If read, an undefined value is read.

[Legend for symbols used in register descriptions]

Meanings of symbols in the HCD R/W column are shown below.

Attribute	Meaning
R	Read-only This is a read-only bit and cannot be modified.
W	Write-only This is a write-only bit and is always read as 0.
R/W	Read/write This bit is readable/writable.
R/WC	Read/write-clear Though this bit is readable/writable, only 1 should be written to clear the bit. Writing 0 is invalid.

(1) HCIVERSION Register

HCIVERSION is a register that indicates the version of the EHCI standard supported by the host controller in the BCD format.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HCIVERSION															
Reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Reset	HCD R/W	Description
31 to 16	HCIVERSION	H'0100	R	Host Controller Interface Version Number These bits indicate the EHCI standard version supported by the host controller (BCD format).

(2) CAPLENGTH Register

CAPLENGTH is a register that indicates the size of the entire capability register area.

Bit:	7	6	5	4	3	2	1	0
	CAPLENGTH							
Reset:	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Reset	HCD R/W	Description
7 to 0	CAPLENGTH	H'0010	R	Capability Register Length These bits indicate the size of the capability register area. Since the operational register area is located immediately after the capability register area, the location of the operational register area can be known by using this register value as an offset.

(3) HCSPARAMS Register

HCSPARAMS is a register that indicates host controller structure parameters, such as the number of ports.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DPN				—	—	—	P_INDI
Reset:	—	—	—	—	—	—	—	—	0	0	0	0	—	—	—	0
R/W:	—	—	—	—	—	—	—	—	R	R	R	R	—	—	—	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	N_CC				N_PCC				PRR	—	—	PCC	N_PORTS			
Reset:	0	0	0	1	0	0	1	0	0	—	—	1	0	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	—	—	R	R	R	R	R

Bit	Bit Name	Reset	HCD R/W	Description
31 to 24	—	—	—	Reserved
23 to 20	DPN	0000	R	Debug Port Number (Optional) These bits indicate the DebugPort number in N_PORTS ports. 0: There is no DebugPort. Other than 0: There is at least one DebugPort. The field value indicates the port number. A number greater than N_PORTS cannot be set in this field.
19 to 17	—	—	—	Reserved
16	P_INDI	0	R	Port Indicator (P_INDICATOR) Indicates whether or not the port supports the port indicator control function. 0: The port indicator control function is not supported. 1: The port indicator control function is supported. The port indicator can be controlled by the Port Indicator Control field in the register PORTSC.

Bit	Bit Name	Reset	HCD R/W	Description
15 to 12	N_CC	0001	R	Number of Companion Controller (N_CC) These bits indicate the number of the mounted USB1.1 companion host controllers. 0: There is no companion host controller. Only the EHCI host controller is mounted. Other than 0: Companion host controllers are mounted. The field value indicates the number of companion host controllers.
11 to 8	N_PCC	0010	R	Number of Ports per Companion Controller (N_PCC) These bits indicate the number of ports used for one companion host controller.
7	PRR	0	R	Port Routing Rule Specifies the routing rules that define to which companion host controller a given port is to be assigned. 0: Assigns N_PCC ports each in ascending order of cHC numbers. If N_PORT = 8, N_CC = 2, and N_PCC = 4, assigns ports No.1 to No.4 to the first companion host controller, and ports No.5 to No.8 to the second companion host controller. 1: Defines assignment rules in HCSP-PORTROUTE, and assigns the ports accordingly.
6, 5	—	—	—	Reserved
4	PPC	1	R	Port Power Control (PPC) Indicates whether the port power supply can be switched on and off. 1: Port power supply can be switched on and off. 0: Port power supply cannot be switched on and off. When this bit is 1, the Port Power bit in the register PORTSC can be set.
3 to 0	N_PORTS	0010	R	N_PORTS These bits indicate the number of downstream ports of the host controller. The number of port registers (PORTSC) in the operational register area is determined by this value. The value can be set in the range from H'1 to H'F.

(4) HCCPARAMS Register

HCCPARAMS is a register that indicates parameters relating to the host controller capabilities.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EECP								IST				—	ASPC	PFLF	64AC
Reset:	1	0	1	0	0	0	0	0	0	0	0	1	—	1	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	—	R	R	R

Bit	Bit Name	Reset	HCD R/W	Description
31 to 16	—	—	—	Reserved
15 to 8	EECP	H'A0	R	EHCI Extended Capabilities Pointer (EECP) To add capability information, an additional capability area can be provided in the PCI configuration area. This field indicates an offset for the added capability area.
7 to 4	IST	0001	R	Isochronous Scheduling Threshold These bits indicate cache mode for isochronous schedule data. When 7th bit (bit 7) is 0: Microframe cache mode. Field[6:4] specifies mframe to be cached. When 7th bit (bit 7) is 1: Frame cache mode. The entire frame is cached. When IST = 1000, the entire frame is cached. Therefore, the data structure corresponding to the current frame cannot be modified by software. Data structures of the next and the following microframes can be modified. When IST = 0010, two microframes are cached. Therefore, data structures of the 3rd and the following microframes can be modified. When IST = 0000, data structures are not cached; data structures are acquired at each microframe. Therefore, data structures of the next and the following microframes can be modified.
3	—	—	—	Reserved

Bit	Bit Name	Reset	HCD R/W	Description
2	ASPC	1	R	Asynchronous Schedule Park Capability Specifies whether or not the park function is supported for queue heads for HS in asynchronous schedules. 0: Park function is not supported. 1: Park function is supported. Park function can be enabled using Asynchronous Schedule Park Mode Enable and Asynchronous Schedule Park Mode Count in the register USBCMD.
1	PFLF	1	R	Programmable Frame List Flag Specifies whether or not the size of the frame list can be changed. 0: The number of frame list elements is fixed at 1024. Frame List Size in USBCMD is read-only and is set to 0. 1: The number of frame list elements can be changed between 512 and 256 using Frame List Size in USBCMD.
0	64AC	0	R	64-Bit Addressing Capability Specifies addressing mode for accessing data structures. 0: 32-bit addressing 1: 64-bit addressing

(5) HCSP-PORTROUTE Register

HCSP-PORTROUTE is a register that specifies to which companion host controller a DownStream port is to be assigned.

This register is valid only when the Port Routing Rules field in HCSPARAMS is set to 1.

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	HCSP-PORTROUTE											
Reset:	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	HCSP-PORTROUTE															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HCSP-PORTROUTE															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HCSP-PORTROUTE															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Reset	HCD R/W	Description
63 to 60	—	—	—	Reserved
59 to 0	HCSP-PORTROUTE	All 0	R	Specifies the correspondence between N_PORTS ports and companion host controllers. The number for a companion host controller is specified in 4 bits. There is a maximum of 15 N_PORTS, and the size of this register is 15 × 4 = 60 bits. For example, if ports are numbered 0h 1h 0h 1h ..., the corresponding companion host controllers are numbered 1, 2, 1, 2 ...

(6) USBCMD Register

USBCMD controls the ON/OFF of the host controller, host controller resets, and ON/OFF of schedule processing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ITC							
Reset:	—	—	—	—	—	—	—	—	0	0	0	0	1	0	0	0
R/W:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ASPME	—	ASPMC		LHCR	IAAD	ASE	PSE	FLS		HCR	RS
Reset:	—	—	—	—	1	—	1	1	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Reset	HCD R/W	Description
31 to 24	—	—	—	Reserved
23 to 16	ITC	H'08	R/W	Interrupt Threshold Control These bits specify the frequency (maximum interval) of hardware interrupts generated by the host controller. H'01: 1 microframe H'02: 2 microframes H'04: 4 microframes H'08: 8 microframes (initial value, 1 ms) H'10: 16 microframes (2 ms) H'20: 32 microframes (4 ms) H'40: 64 microframes (8 ms) This field must not be set when the HC Halted bit in USBSTS is 0.
15 to 12	—	—	—	Reserved

Bit	Bit Name	Reset	HCD R/W	Description
11	ASPME	1	R/W	Asynchronous Schedule Park Mode Enable (Optional) Enables park mode for asynchronous schedules. 0: Park mode is disabled. 1: Park mode is enabled. This field is fixed at 0 and can only be read unless the Asynchronous Schedule Park Capability bit in the register HCCPARAMS is set to 1. Conversely, if Asynchronous Schedule Park Capability is set to 1, the initial value of this bit is 1, and the bit can be read and written.
10	—	—	—	Reserved
9, 8	ASPMC	11	R/W	Asynchronous Schedule Park Mode Count (Optional) These bits specify the number of transactions that can be issued in a single operation from one QH in the asynchronous schedule. For example, if a transaction count of 3 is specified, bus transactions are executed 3 times from the fetched QH, and then the next QH is fetched. If Asynchronous Schedule Park Mode is set to 1, the value 0 must not be assigned to this field. This field is fixed at 0 and can only be read unless the Asynchronous Schedule Park Capability bit in the register HCCPARAMS is set to 1. Conversely, if Asynchronous Schedule Park Capability is set to 1, the initial value of this bit is 3, and the bit can be read and written.
7	LHCR	0	R/W	Light Host Controller Reset (Optional) Resets the EHCI host controller without affecting the status of the port or port ownership relations. In other words, this bit initializes all the registers except the PORTSC or CF register. By assigning the value 1 to this bit, the Light Host Controller Reset command can be executed. If this bit is read and it is 0, it can be concluded that the resetting is complete. If the bit is 1, the resetting operation is still in progress.

Bit	Bit Name	Reset	HCD R/W	Description
6	IAAD	0	R/W	Interrupt on Async Advance Doorbell Enables System Software to request the host controller to generate an interrupt when the asynchronous schedule processing is executed. When this bit is set, the host controller clears any cached asynchronous schedule data, and sets the Interrupt on Async Advance bit in the register USBSTS. If the Interrupt on Async Advance Enable bit in USBINTR is set, the host controller generates an interrupt on the next interrupt threshold.
5	ASE	0	R/W	Asynchronous Schedule Enable Specifies that the host controller executes or skips asynchronous schedules. 0: Asynchronous schedules are not executed. 1: Asynchronous schedules are executed using the register ASYNCLISTADDR.
4	PSE	0	R/W	Periodic Schedule Enable Specifies that the host controller executes or skips periodic schedules. 0: Periodic schedules are not executed. 1: Periodic schedules are executed using the register PERIODICLISTBASE.
3, 2	FLS	00	R/W	Frame List Size These bits specify the frame list size. Depending on the frame list size specified with this field, which bit in the register Frame List Index is used for the Frame List Current Index. These bits can be written only when Programmable Frame List Flag in HCCPARAMS is set. 00: 1024 elements (4096 bytes) 01: 512 elements (2048 bytes) 10: 256 elements (1024 bytes)

Bit	Bit Name	Reset	HCD R/W	Description
1	HCR	0	R/W	Host Controller Reset (HCRESET) Resets the host controller. If the host controller is reset by this bit, the RootHub register operates in the same manner as when the hardware reset is applied to the chip. When this bit is set, the host controller resets the pipeline, timer, counter, and state machine in the host controller, and sets the initial value. In addition, the host controller immediately terminates any transfer operation that is being conducted at the time. This resetting is not driven to any of the downstream ports. Although the resetting does not initialize the registers in the PCI configuration register area, all registers in the operational register area, including the Port register and the State Machine of the Port, are reset to their initial value. The ownership of the port is returned to the companion host controller. For this reason, after resetting the software must re-initialize the host controller in order to reset the host controller to the operating condition. The host controller clears this bit (i.e., if the bit is set to 1, resets it to 0) upon completion of the resetting operation. During the resetting process, software cannot clear this bit and interrupt the resetting process. Further, software cannot set the HCRESET when the HC Halted bit in the register USBSTS is 0. In other words, USBRESET must not be performed when the host controller is in the execution status.

Bit	Bit Name	Reset	HCD R/W	Description
0	RS	0	R/W	Run/Stop Controls the ON/OFF action of the entire host controller. 1: While the bit is 1, the host controller continues to execute the schedule. 0: When the bit is set to 0, the host controller terminates any communication currently in progress, and stops the operation. After the software has cleared this bit, the host controller must halt its operation within 16 microframes. The Halted bit in USBSTS can be used to verify that the host controller has terminated any transfer being executed by the host controller and transitioned to the Stop state. When the host controller is in the Halt state (USBSTS/Halted = 1), the software must not set the Run/Stop bit to 1.

(7) USBSTS Register

USBSTS indicates status information such as interrupt status and schedule status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ASS	PSS	R	HCH	—	—	—	—	—	—	IAA	HSE	FLR	PCD	UEI	UI
Reset:	0	0	0	1	—	—	—	—	—	—	0	0	0	0	0	0
R/W:	R	R	R	R	—	—	—	—	—	—	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC

Bit	Bit Name	Reset	HCD R/W	Description
31 to 16	—	—	—	Reserved
15	ASS	0	R	Asynchronous Schedule Status Indicates the current (actual) status of asynchronous schedules. 1: Asynchronous schedules are enabled. 0: Asynchronous schedules are disabled. It is not necessary for the host controller to enable/disable asynchronous schedules immediately after the software sets/clears the Asynchronous Schedule Enable bit in USBCMD to enable/disable asynchronous schedules (it is not necessary to reflect the software request immediately). When this bit matches the Asynchronous Schedule Enable bit in USBCMD, asynchronous schedules are recognized to be enabled or disabled.

Bit	Bit Name	Reset	HCD R/W	Description
14	PSS	0	R	Periodic Schedule Status Indicates the current (actual) status of periodic schedules. 1: Periodic schedules are enabled. 0: Periodic schedules are disabled. It is not necessary for the host controller to enable/disable periodic schedules immediately after the software sets/clears the Periodic Schedule Enable bit in USBCMD to enable/disable periodic schedules (it is not necessary to reflect the software request immediately). When this bit matches the Periodic Schedule Enable bit in USBCMD, periodic schedules are recognized to be enabled or disabled.
13	R	0	R	Reclamation This bit is set when asynchronous schedule empty state is detected.
12	HCH	1	R	HC Halted Indicates the host controller status. 0: The Run/Stop bit in USBCMD is set to 1. 1: The Run/Stop bit is cleared to 0 and the host controller is stopped. The Run/Stop bit is cleared by software or hardware (for example, by Internal Error).
11 to 6	—	—	—	Reserved
5	IAA	0	R/WC	Interrupt Async Advance By setting the Interrupt On Async Advance Doorbell bit in USBCMD, the software can force the host controller to generate an interrupt when executing the asynchronous schedule. Thus, this bit indicates that the host controller has executed the asynchronous schedule. This bit, being a status bit, indicates the occurrence of an interrupt due to this interrupt source. This interrupt is used to delete queue heads from asynchronous schedules.

Bit	Bit Name	Reset	HCD R/W	Description
4	HSE	0	R/WC	Host System Error The host controller sets this bit if a severe error occurs during access by the host system involving the use of the host controller module. In the PCI system, this bit is set in conditions including "PCI parity error", "PCI master abort", and "PCI target abort". If this error occurs, the host controller sets the Run/Stop bit to 0 to prevent any scheduled TD from being executed.
3	FLR	0	R/WC	Frame List Rollover The host controller sets this bit if the Frame List Index (FRINDEX) register rolls over from its maximum value to 0. At what value a RollOver occurs depends on the Frame List Size of USBCMD. If the Frame List Size is set to 1024, a RollOver occurs each time FRINDEX[13] toggles. Similarly, if the Frame List Size is 512, a RollOver occurs each time FRINDEX[12] toggles.
2	PCD	0	R/WC	Port Change Detect This bit is set by the host controller in the following cases: • When the Port Owner bit is changed from 0 to 1 in a port. • When a transition from J to K is detected in a suspended port and the Force Resume Transition bit is changed from 0 to 1. • When the Port Owner bit is written to 1 by software and the port ownership is released.
1	UEI	0	R/WC	USB Error interrupt This bit is set by the host controller when a USB transfer ends with an error (when the error counter underflows). When a transfer ends with an error at the TD with ioc = 1, both the UEI and USBINT bits are set.

Bit	Bit Name	Reset	HCD R/W	Description
0	UI	0	R/WC	USB Interrupt This bit is set by the host controller if the USB transfer ends when ioc = 1 and the TD retires. This bit is also set when a short packet is received (when the size of received data is smaller than expected.)

(8) USBINTR Register

USBINTR specifies the on/off of hardware interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	IAAE	HSEE	FLRE	PCDE	UEIE	UIE
Reset:	—	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Reset	HCD R/W	Description
31 to 6	—	—	—	Reserved
5	IAAE	0	R/W	Interrupt on Async Advance Enable The host controller generates hardware interrupts when the Interrupt on Async Advance bit in USBSTS is set while this bit is 1.
4	HSEE	0	R/W	Host System Error Enable The host controller generates hardware interrupts when the Host System Error Status bit in USBSTS is set while this bit is 1.
3	FLRE	0	R/W	Frame List Rollover Enable The host controller generates hardware interrupts when the Frame List Rollover bit in USBSTS is set while this bit is 1.
2	PCDE	0	R/W	Port Change Detect Enable The host controller generates hardware interrupts when the Port Change Detect bit in USBSTS is set while this bit is 1.
1	UEIE	0	R/W	USB Error Interrupt Enable The host controller generates hardware interrupts when the USBERRINT bit in USBSTS is set while this bit is 1.
0	UIE	0	R/W	USB Interrupt Enable The host controller generates hardware interrupts when the USBINT bit in USBSTS is set while this bit is 1.

(9) FRINDEX Register

FRINDEX indicates the current frame number which is used when the host controller references the Periodic Frame List. The register value is updated every 125 μs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	FI													
Reset:	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Reset	HCD R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	FI	All 0	R/W	Frame Index This value is incremented at every end of microframe. Bits[N:3] are used for the Current Index of the framelist. In other words, accesses to framelist are performed every 8 microframes (1 frame).

Frame List Size (USBCMD)	Number of Elements	N	Actual Size of Framelist
00b	1024	12	2^{12} = 4096
01b	512	11	2^{11} = 2048
10b	256	10	2^{10} = 1024

(10) CTRLDSSEGMENT Register

CTRLDSSEGMENT indicates the upper 32 bits [63:32] when the host controller accesses the 64-bit data structures.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CTRLDSSEGMENT															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTRLDSSEGMENT															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Reset	HCD R/W	Description
31 to 0	CTRLDSSEGMENT	All 0	R/W	Control Data Structure Data Segment These bits specify the upper 32 bits (bits 63 to 32) when the data structures of EHCI are to be accessed in 64 bits. This field is disabled if the 64-bit Addressing Capability field in HCCPARAMS is 0. If it is 1, access can be made to the data structures of EHCI. The data structures must reside within the same 4-Gbyte boundary.

(11) PERIODICLISTBASE Register

PERIODICLISTBASE specifies the base address for a periodic framelist.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA				—	—	—	—	—	—	—	—	—	—	—	—
Reset:	0	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Reset	HCD R/W	Description
31 to 12	BA	All 0	R/W	Base Address Specifies the start address of the periodic framelist placed on memory. This register value is loaded by software before the periodic schedule is executed by the host controller. The host controller can execute the periodic framelists in sequence using this register and FRINDEX. Setting for accessing the areas 0 and 1 is prohibited.
11 to 0	—	—	—	Reserved

(12) ASYNCLISTADDR Register

ASYNCLISTADDR specifies the pointer to queue heads in asynchronous schedules.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LPL															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LPL											—	—	—	—	—
Reset:	0	0	0	0	0	0	0	0	0	0	0	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—	—	—	—	—

Bit	Bit Name	Reset	HCD R/W	Description
31 to 5	LPL	All 0	R/W	Link Pointer Low Specifies the queue head for the next execution in asynchronous schedules. Since a queue head is allocated on the 32-bit boundary, only the upper 27 bits are specified. Setting for accessing the areas 0 and 1 is prohibited.
4 to 0	—	—	—	Reserved

(13) CONFIGFLAG Register

CONFIGFLAG specifies the ownership for all ports.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF
Reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	R/W

Bit	Bit Name	Reset	HCD R/W	Description
31 to 1	—	—	—	Reserved
0	CF	0	R/W	Config Flag Specifies the routing rule for all ports. 0: Each port is routed to the corresponding cHC. 1: All ports are routed to eHC.

(14) PORTSC (1 - N_PORT) Register

PORTSC controls ports and monitors port status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	WOE	WDE	WCE	PTC			
Reset:	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC		PO	PP	LS		—	PR	S	FPR	OC	OA	PEDC	PED	CSC	CCS
Reset:	0	0	1	0	0	0	—	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	—	R/W	R/W	R/W	R/WC	R	R/WC	R/WC	R/WC	R

Bit	Bit Name	Reset	HCD R/W	Description
31 to 23	—	—	—	Reserved
22	WOE	0	R/W	Wake on Over-Current Enable (WKOC_E) Enables detecting overcurrent for ports as a wakeup event to make transition to the resume from the suspend state.
21	WDE	0	R/W	Wake on Disconnect Enable (WKDSCNNT_E) Enables detecting device disconnection as a wakeup event to make transition to the resume from the suspend state.
20	WCE	0	R/W	Wake on Connect Enable (WKCNNT_E) Enables detecting device connection as a wakeup event to make transition to the resume from the suspend state.
19 to 16	PTC	0000	R/W	Port Test Control These bits control port test mode. 0000: Test mode is invalid. 0001: Test J_state 0010: Test K_state 0011: Test SE0_NAK 0100: Test packet 0101: Test Force_Enable

Bit	Bit Name	Reset	HCD R/W	Description
15, 14	PIC	00	R/W	Port Indicator Control These bits control the port indicator. 00: The port indicator function is turned off. 01: Amber 10: Greed This field is invalid when HCSPARAMS/F_INDICATOR is 0.
13	PO	1	R/W	Port Owner Controls the port ownership. 1: The companion host controller has the ownership of this port. 0: The EHCI host controller has the ownership of this port. When the connected device is not a high-speed device, this bit is set to 1 and the ownership is released by software. When the Configured bit in the register Configuration Flag is changed from 0 to 1, this bit is unconditionally cleared to 0. Conversely, when the Configured bit is changed from 1 to 0, this field is unconditionally set to 1.

Bit	Bit Name	Reset	HCD R/W	Description
12	PP	0	R/W	Port Power Controls the port power. The meaning of this bit differs depending on the PPC bit in HCSPARAMS.

PPC	PP
0	1 (read-only) The port power is always turned on and this bit is always 1.
1	0 or 1 (readable and writable) The port power can be switched on (when PP = 0) and off (when PP = 1).

When this bit is 0 or the port power is turned off, the port does not function and connection/disconnection is not detected.

When overcurrent is detected and PPC is 1, the PP bit for the port is cleared to 0 by the host controller and the port power is turned off.

Bit	Bit Name	Reset	HCD R/W	Description
11, 10	LS	00	R	Line Status These bits indicate the current D+/D- logic level. D+ is indicated at the 11th bit (bit 11) and D- is indicated at the 10th bit (bit 10). This field is used for detecting a low-speed device connection before port-reset or port-enable processing is executed. This field is only valid when Port Enable = 0 and Current Connect Status = 1. Since this field is valid during time from detecting connection to enabling the port. Therefore, it is used for determining whether the connected device is low speed or not.

LS	State	Description
00	SE0	Low-speed device. Performs EHCI reset
01	J-state	Not a low-speed device. Performs EHCI reset
10	K-state	Low-speed device. Releases port ownership
11	SE1	Not a low-speed device. Performs EHCI reset

9	—	—	—	Reserved

Bit	Bit Name	Reset	HCD R/W	Description
8	PR	0	R/W	Port Reset Controls the port resetting processing. 1: The port being reset 0: The port not being reset The setting of this bit by software commences the bus resetting operation that is defined in the USB 2.0 specifications. To complete the bus resetting operation, the software must write a 0 to this bit. The software must keep this bit at 1 until such time as the reset time defined in the USB 2.0 specifications has elapsed. Notes: 1. To change this bit from 0 to 1, the Port Enable bit must be set to 0. 2. Even if the software sets the bit to 0, there is some delay before the bit becomes 0. Even if this bit is read, the 0 is not read until the reset operation is complete. If the port is in the high-speed mode and the resetting has completed, the host controller automatically enables the port (Port Enable = 1). The host controller must complete the resetting operation and stabilize the port within 2 ms from the time the software set the bit to 0. For example, if the connection of a high-speed device is detected during the resetting of the port, the port must be enabled within 2 ms from the time the software set the bit to 0. 3. If the software uses this bit, the HC Halted bit in USBSTS must be set to 0. As long as USBSTS/HC Halted = 1, the host controller must continue to assert the Port Reset bit to 1.

Bit	Bit Name	Reset	HCD R/W	Description
7	S	0	R/W	Suspend Controls the Suspend operation of the port. 1: The port is in the Suspend state. 0: The port is not in the Suspend state. Depending on the setting of the Port Enable and Port Suspend bits, the port assumes the following condition:

Bit[Port Enable, Port Suspend]	Port State
0X	Disable
10	Enable
11	Suspend

In the Suspend state, any propagation of data from this port to the Down Stream is locked, except that Port Reset is conveyed. If a transfer is being conducted, the data is blocked after the current transfer operation is completed.

In the Suspend state, the port can detect a Resume.

Notes:

1. The status of this bit does not change after a transition to the Suspend state and when the current transfer operation, being executed, is pending on Suspend transition.
2. Any clearing of this bit by the software is ignored by the host controller.

   The host controller unconditionally clears this bit if the following conditions are met:

   — The software setting the Force Port Resume bit from 0 to 1

   — The software setting the Port Reset bit from 0 to 1

Bit	Bit Name	Reset	HCD R/W	Description
6	FPR	0	R/W	Force Port Resume Controls the Resume operation of the port. 1: It is detected that the port has resumed. Or, it is detected that the port is driving the Resume state. In other words, the port is in the Resume state. 0: It is not detected that the port is in the Resume state (K state) or driving the Resume state. This bit depends on the status of the Suspend bit. For example, if the port is not in the Suspend state (if both the Suspend bit and Port Enable bit are 1, it is Suspend, but not both bits are 1), even if this bit is set, the resulting status is undefined. When setting this bit, the software drives Resume. In other words, if the software itself wants to drive Resume, it sets this bit. The host controller sets this bit when a transition from J to K is detected when the ports is in the Suspend state. In other words, upon detecting a Resume, the host controller itself sets this bit. Since the host controller sets this bit after detecting the transition from J to K, the host controller must also set the Port Change Detect bit in USBSTS. However, if the software has set this bit, the host controller must not set the Port Change Detect bit in USBSTS. Notes (Setting by software): If EHCI is the owner of the port, the Resume operation is performed according to the USB 2.0 specifications. As long as this bit is 1, the Resume signal (Full-Speed K) continues to be driven. The software must resume for an appropriate length of time, and after performing a Resume for a sufficient length of time, it must set this bit to 0. When this bit is set from 1 to 0, the port returns to the high-speed mode (forces the bus to the idle state of the high-speed mode). Even if a 0 is written to this bit, the bit remains 1 until such time as a switching can be made to idle state of high-speed. The host controller must complete the switching within 2 ms after the software wrote a 0 to this bit.

Bit	Bit Name	Reset	HCD R/W	Description
5	OC	0	R/WC	Over-Current Change 1: Indicates that the port overcurrent state is changed from non-active to active. 0: Indicates that the state is not changed. This bit can be cleared by writing 1 by software.
4	OA	0	R	Over-Current Active 1: Indicates that the port is in the overcurrent state. 0: Indicates that the port is not in the overcurrent state. This bit is automatically cleared to 0 when the overcurrent state is cancelled.
3	PEDC	0	R/WC	Port Enable/Disable Change 1: Indicates that the port state has changed between enable and disable. 0: Indicates that the port state has not changed. This bit is set to 1 only when a condition to disable the port exists at EOF2 and the port is actually disabled (see section 11, Port Error, in the USB 2.0 Specification).

Bit	Bit Name	Reset	HCD R/W	Description
2	PED	0	R/W	Port Enabled/Disabled 1: The port is enabled. 0: The port is disabled. The port is enabled only when it is reset; the software cannot enable the port by resetting this bit. Only the host controller can enable the port. The host controller can set this bit only when it is verified by means of a reset sequence that the connected device is a high-speed device. The port is disabled only when a fault condition arises (due to the occurrence of a disconnection or other fault events) or by software. Notes: 1. The status of this bit does not change until the port status actually changes. Due to other host controller processing or bus processing, there may be situations where the port is slow to become enabled/disabled. 2. If this bit is 0b and the port is disabled, data does not propagate to the DownStream; any resetting, however, is conveyed.
1	CSC	0	R/WC	Connect Status Change 1: The Current Connect Status bit has changed. 0: Status is not changed. The host controller sets this bit to 1 when the state of device connection to the port is changed even though software has not clear this bit to 0. This bit can be cleared by writing 1 by software.
0	CCS	0	R	Current Connect Status 1: A device detected in the port. 0: No device detected in the port. This bit indicates the port state. This bit is not directly affected by event generation which sets the Connect Status Change bit.

14.2.3 Usage Notes

(1) Notes on Port Resetting

When an Over-Current occurs while resetting the port by EHCI (writing a 1 to the Port Reset bit in the register PORSC), the Port Reset bit may not be cleared despite the Port Reset is released (writing 0 to the Port Reset bit) without turning off the Port Power (to write a 0 to the Port Power bit). So in the case of detected Over-Current, be sure to turn the Port Power off in advance of releasing the Port Reset.

(2) Notes on NPS bit and PRS bit

When the NPS(No Power Swiching) bit in the register HcRhDescriptorA is 1, if the Over-Current occurs during 10 ms after issuing bus-reset by writing a 1 to the PRS (Port Reset Status) bit in the register HcRhPortStatus, The PRS bit is kept 1. Additionally, when the NPS bit in the register HcRhDescriptroA is 1 and the DR bit in the register HcRhDescriptorB is 1, if the Over-Current occurs during suspending the bus by writing a 1 to the PSS (Port Suspend Status) bit, The PSS bit is kept 1. So the NPS bit in the register HcRhDescriptorA should be kept 0, or the Low signal should not be input to the Over-Current pin.

14.3 USB1.1-Host Controller

14.3.1 Overview

The USB host interface embedded in this LSI has a root hub and a two-port USB transceiver, and operates in Full speed mode. Open HCI interfaces and registers are also embedded in this LSI.

For the development of software, refer to the Open HCI specifications as well.

(1) Features

- Support the Open HCI interface.
- Support the USB host interface.
- Root Hub function
- Operate in Full speed mode (12 Mbps) and Low speed mode (1.5 Mbps)
- Support over-current detection and Power source enable management.

14.3.2 Register Descriptions

Table 14.5 is a list of the registers in this module.

The USB host's registers are allocated to the address space of the I/O bus. This description is based on the Open HCI Rev.1.0. For details, refer to the Open HCI Rev1.0. Bit widths of these resisters are all 32 bits, so should be accessed in longword (32 bits) units. Access in other units is not supported.

Table 14.5 List of Open HCI Registers

Register Address	Register Name	R/W	Initial Value	Access Size
H'FFE70400	HcRevision register	R	H'00000010	32
H'FFE70404	HcControl register	R/W	H'00000000	32
H'FFE70408	HcCommandStatus register	R/W	H'00000000	32
H'FFE7040C	HcInterruptStatus register	R/W	H'00000000	32
H'FFE70410	HcInterruptEnable register	R/W	H'00000000	32
H'FFE70414	HcInterruptDisable register	R/W	H'00000000	32
H'FFE70418	HcHCCA register	R/W	H'00000000	32
H'FFE7041C	HcPeriodCurrentED register	R/W	H'00000000	32
H'FFE70420	HcControlHeadED register	R/W	H'00000000	32
H'FFE70424	HcControlCurrentED register	R/W	H'00000000	32

Register Address	Register Name	R/W	Initial Value	Access Size
H'FFE70428	HcBulkHeadED register	R/W	H'00000000	32
H'FFE7042C	HcBulkCurrentED register	R/W	H'00000000	32
H'FFE70430	HcDoneHead register	R/W	H'00000000	32
H'FFE70434	HcFmInterval register	R/W	H'00002EDF	32
H'FFE70438	HcFmRemaining register	R	H'00000000	32
H'FFE7043C	HcFmNumber register	R	H'00000000	32
H'FFE70440	HcPeriodicStart register	R/W	H'00000000	32
H'FFE70444	HcLSThreshold register	R/W	H'00000628	32
H'FFE70448	HcRhDescriptorA register	R/W	H'FF000902	32
H'FFE7044C	HcRhDescriptorB register	R/W	H'00060000	32
H'FFE70450	HcRhStatus register	R/W	H'00000000	32
H'FFE70454	HcRhPortStatus1 register	R/W	H'00000100	32
H'FFE70458	HcRhPortStatus2 register	R/W	H'00000100	32

[Legend for the Register Description]

Initial value: Value of the register after a reset.

—: The read value is undefined. The write value should always be 0.

R/W: Readable/writable register

R: Read only. The write value should always be 0.

Note: The register can be set when a 48 MHz clock is input.

Other than control registers for setting, all registers conform to the Open HCIs specification.

These control registers are only for this LSI.

(1) HcRevision Register

Register Name: HcRevision			Offset: 00 to 03
Bit	**Initial Value**	**R/W**	**Description**
31 to 8	—	—	Reserved These bits are always read as undefined. The write value should always be 0.
7 to 0	H'10	R	Revision These bits indicate the Open HCI Specification revision number implemented by the Hardware. (X.Y = XYh) USB Host Controller supports the Open HCI1.0 specification.

(2) HcControl Register

Register Name: HcControl			Offset: 04 to 07
Bit	**Initial Value**	**R/W**	**Description**
31 to 11	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
10	0	R/W	Remote Wakeup Connected Enable If a remote wakeup signal is supported, this bit enables that operation. Since remote wakeup signal is not supported, this bit is ignored.
9	0	R/W	Remote Wakeup Connected This bit indicates whether the Host Controller (HC) supports a remote wakeup signal.
8	0	R/W	Interrupt Routing This bit specifies interrupt routing: 0: Interrupts routed to normal interrupt processing unit (INT). 1: Interrupts routed to SMI.

Register Name: HcControl **Offset: 04 to 07**

Bit	Initial Value	R/W	Description
7, 6	00	R/W	Host Controller Functional State These bits set the Host Controller state. The state encodings are: 00: UsbReset 01: UsbResume 10: UsbOperational 11: UsbSuspend The Host Controller may force a state change from UsbSuspend to UsbResume after detecting resume signaling from a downstream port.
5	0	R/W	Bulk List Enable Setting this bit enables processing of the Bulk list.
4	0	R/W	Control List Enable Setting this bit enables processing of the Control list.
3	0	R/W	Isochronous Enable Clearing this bit disables the Isochronous List when the Periodic List is enabled (so Interrupt EDs may be serviced). While processing the Periodic List, the Host Controller will check this bit when it finds an isochronous ED.
2	0	R/W	Periodic List Enable Setting this bit enables processing of the Periodic (interrupt and isochronous) list. The Host Controller checks this bit prior to attempting any periodic transfers in a frame.
1, 0	00	R/W	Control Bulk Service Ratio These bits specify the number of Control Endpoints serviced for every Bulk Endpoint. Encoding is N-1 where N is the number of Control Endpoints (i.e. 00 = 1 Control Endpoint; 11 = 4 Control Endpoints).

(3) HcCommandStatus Register

Register Name: HcCommandStatus Offset: 08 to 0B

Bit	Initial Value	R/W	Description
31 to 18	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	00	R	Schedule Overrun Count These bits increment every time the SchedulingOverrun bit in HcInterruptStatus is set. The count wraps from 11 to 00.
15 to 4	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
3	0	R/W	Ownership Change Request Setting this bit by software sets the Ownership Change bit in HcInterruptStatus register. This bit is cleared by software.
2	0	R/W	Bulk List Filled Setting this bit indicates there is an active ED on the Bulk List. The bit can be set by either software or the Host Controller. The bit is cleared by the Host Controller each time it begins processing the head of the Bulk List.
1	0	R/W	Control List Filled Setting this bit indicates there is an active ED on the Control List. The bit can be set by either software or the Host Controller. The bit is cleared by the Host Controller each time it begins processing the head of the Control List.
0	0	R/W	Host Controller Reset This bit is set to initiate a software reset. This bit is cleared by the Host Controller upon completion of the reset operation.

(4) HcInterruptStatus Register

All bits are set by hardware and cleared by software.

These bits in this register can be cleared by writing 1 to bit positions to be cleared.

Register Name: HcInterruptStatus			Offset: 0C to 0F
Bit	**Initial Value**	**R/W**	**Description**
31	0	—	Reserved These bits are always read as 0. The write value should always be 0.
30	0	R/W	Ownership Change This bit is set when the OwnershipChangeRequest bit in HcCommandStatus register is set.
29 to 7	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
6	0	R/W	Root Hub Status Change This bit is set when the content of HcRhStatus register or the content of any HcRhPortStatus register has changed.
5	0	R/W	Frame Number Overflow This bit is set when bit 15 of FrameNumber changes value from 0 to 1 or from 1 to 0.
4	0	R/W	Unrecoverable Error This bit is set when HC detects a system error that is not USB related.
3	0	R/W	Resume Detected This bit is set when the Host Controller detects resume signaling on a downstream port.
2	0	R/W	Start of Frame This bit is set when the Frame manager signals a Start of Frame's event.
1	0	R/W	Writeback Done Head This bit is set after the Host Controller has written the value of HcDoneHead register to HccaDoneHead.
0	0	R/W	Scheduling Overrun This bit is set when the List Processor determines a Schedule Overrun has occurred.

(5) HcInterruptEnable Register

Writing 1 to a bit in this register sets the corresponding bit, while writing a 0 to a bit leaves the bit unchanged.

Register Name: HcInterruptEnable Offset: 10 to 13

Bit	Initial Value	R/W	Description
31	0	R/W	Master Interrupt Enable This bit is a global interrupt enable. Writing 1 allows interrupts to be enabled via the specific enable bits listed below.
30	0	R/W	Ownership Change Enable 0: Ignored 1: Interrupt due to Ownership Change is enabled.
29 to 7	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
6	0	R/W	Root Hub Status Change Enable 0: Ignored 1: Interrupt due to Root Hub Status Change is enabled.
5	0	R/W	Frame Number Overflow Enable 0: Ignored 1: Interrupt due to Frame Number Overflow is enabled.
4	0	R/W	Unrecoverable Error Enable This function is not supported. Writing is ignored.
3	0	R/W	Resume Detected Enable 0: Ignored 1: Interrupt due to Resume Detected is enabled.
2	0	R/W	Start of Frame Enable 0: Ignored 1: Interrupt due to Start of Frame is enabled.
1	0	R/W	Writeback Done Head Enable 0: Ignored 1: Interrupt due to Writeback Done Head is enabled.
0	0	R/W	Scheduling Overrun Enable 0: Ignored 1: Interrupt due to Scheduling Overrun is enabled.

(6) HcInterruptDisable Register

Writing 1 to a bit in this register clears the corresponding bit in the HcInterrupt Enable register, while writing 0 to a bit leaves the bit unchanged.

Register Name: HcInterruptDisable Offset: 14 to 17

Bit	Initial Value	R/W	Description
31	0	R/W	Master Interrupt Disable This bit is a global interrupt disable. Writing 1 disables all interrupts.
30	0	R/W	Ownership Change Disable 0: Ignored 1: Interrupt due to Ownership Change is disabled.
29 to 7	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
6	0	R/W	Root Hub Status Change Disable 0: Ignored 1: Interrupt due to Root Hub Status Change is disabled.
5	0	R/W	Frame Number Overflow Disable 0: Ignored 1: Interrupt due to Frame Number Overflow is disabled.
4	0	R/W	Unrecoverable Error Disable This function is not supported. Writing is ignored.
3	0	R/W	Resume Detected Disable 0: Ignored 1: Interrupt due to Resume Detected is disabled.
2	0	R/W	Start of Frame Disable 0: Ignored 1: Interrupt due to Start of Frame is disabled.
1	0	R/W	Writeback Done Head Disable 0: Ignored 1: Interrupt due to Writeback Done Head is disabled.
0	0	R/W	Scheduling Overrun Disable 0: Ignored 1: Interrupt generation due to Scheduling Overrun is disabled.

(7) HcHCCA Register

Register Name: HcHCCA			Offset: 18 to 1B
Bit	**Initial Value**	**R/W**	**Description**
31 to 8	All 0	R/W	HCCA Pointer to HCCA base address. (Within DDR Memory space) Setting for accessing the areas 0 and 1 is prohibited.
7 to 0	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

(8) HcPeriodCurrntED Register

Register Name: HcPeriodCurrentED			Offset: 1C to 1F
Bit	**Initial Value**	**R/W**	**Description**
31 to 4	All 0	R	Period Current ED Pointer to the current Periodic List ED. (Within DDR Memory space) Setting for accessing the areas 0 and 1 is prohibited.
3 to 0	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

(9) HcControlHeadED Register

Register Name: HcControlHeadED			Offset: 20 to 23
Bit	**Initial Value**	**R/W**	**Description**
31 to 4	All 0	R/W	Control Head ED Pointer to the Control List Head ED (within DDR Memory space). Setting for accessing the areas 0 and 1 is prohibited.
3 to 0	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

(10) HcControlCurrentED Register

Register Name: HcControlCurrentED Offset: 24 to 27

Bit	Initial Value	R/W	Description
31 to 4	All 0	R/W	Control Current ED Pointer to the current Control List ED (within DDR Memory space). Setting for accessing the areas 0 and 1 is prohibited.
3 to 0	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

(11) HcBulkHeadED Register

Register Name: HcBulkHeadED Offset: 28 to 2B

Bit	Initial Value	R/W	Description
31 to 4	All 0	R/W	Bulk Head ED Pointer to the Bulk List Head ED (within DDR Memory space). Setting for accessing the areas 0 and 1 is prohibited.
3 to 0	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

(12) HcBulkCurrentED Register

Register Name: HcBulkCurrentED Offset: 2C to 2F

Bit	Initial Value	R/W	Description
31 to 4	All 0	R/W	Bulk Current ED Pointer to the current Bulk List ED (within DDR Memory space). Setting for accessing the areas 0 and 1 is prohibited.
3 to 0	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

(13) HcDoneHead Register

Register Name: HcDoneHead			Offset: 30 to 33
Bit	**Initial Value**	**R/W**	**Description**
31 to 4	All 0	R	DoneHead Pointer to the current Done List Head ED (within DDR Memory space). Setting for accessing the areas 0 and 1 is prohibited.
3 to 0	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

(14) HcFmInterval Register

Register Name: HcFmInterval			Offset: 34 to 37
Bit	**Initial Value**	**R/W**	**Description**
31	0	R/W	Frame Interval Toggle This bit is toggled by Host Control Driver (HCD) whenever it loads a new value into FrameInterval bit.
30 to 16	All 0	R/W	FS Largest Data Packet These bits specify a value which is loaded into the Largest Data Packet Counter at the beginning of each frame.
15, 14	All 0	—	Reserved These bits are always read as 0. The write value should always be 0
13 to 0	H'2EDF	R/W	Frame Interval These bits specify the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is specified.

(15) HcFrameRemaining Register

Register Name: HcFrameRemaining Offset: 38 to 3B

Bits	Initial Value	R/W	Description
31	0	R	Frame Remaining Toggle This bit is loaded with FrameIntervalToggle when FrameRemaining is loaded.
30 to 14	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	All 0	R	Frame Remaining These bits are the 14-bit down counter used to time a frame. When the Host Controller is in the USB OPERATIONAL state, the counter decrements each 12 MHz clock period. When the count reaches 0, the end of a frame has been reached. The counter reloads with FrameInterval at that time. In addition, the counter reloads when the Host Controller transitions into USB OPERATIONAL.

(16) HcFmNumber Register

Register Name: HcFmNumber Offset: 3C to 3F

Bit	Initial Value	R/W	Description
31 to 16	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	H'0000	R	Frame Number These bits are the 16-bit up counter. The count is incremented coincident with the loading of FrameRemaining bit. The count will roll over from H'FFFF to H'0000.

(17) HcPeriodicStart Register

Register Name: HcPeriodicStart			Offset: 40 to 43
Bit	**Initial Value**	**R/W**	**Description**
31 to 14	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	All 0	R/W	Periodic Start These bits set a value used by the List Processor to determine where in a frame the Periodic List processing must begin.

(18) HcLSThreshold Register

Register Name: HcLSThreshold			Offset: 44 to 47
Bit	**Initial Value**	**R/W**	**Description**
31 to 12	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	H'628	R/W	LS Threshold These bits are a value used by the Frame manager to determine whether or not a low speed transaction can be started in the current frame.

(19) HcRhDescriptorA Register

This register is only reset by a power-on reset. It is written during system initialization to configure the Root Hub. This register should not be written during normal operation.

Register Name: HcRhDescriptorA Offset: 48 to 4B

Bit	Initial Value	R/W	Description
31 to 24	H'FF	R/W	Power-on to Power Good Time USB Host Controller power switching is effective within 2 ms. The bit value is represented as the number of 2 ms intervals. Only bits 25 and 24 can be written to. The remaining bits are read only as 0. It is not expected that these bits be written to anything other than 1h, but limited adjustment is allowed. These bits should be written to support the system implementation. These bits should always be written to a non-zero value.
23 to 13	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
12	0	R/W	No Over Current Protection USB Host Controller implements global over-current reporting 0: Over-current status is reported 1: Over-current status is not reported This bit should be written to support the external system port over-current implementation.
11	1	R/W	Over-current Protection Mode USB Host Controller implements global over-current reporting 0: Global Over-Current is reported. 1: Individual Over-Current is reported. This bit is only valid when NoOverCurrentProtection bit is cleared.
10	0	R	Device Type USB Host Controller is not a composite device.

Register Name: HcRhDescriptorA Offset: 48 to 4B

Bit	Initial Value	R/W	Description
9	0	R/W	No Power Switching USB Host Controller implements global power switching. 0: Ports are power switched. 1: Ports are always powered on. This bit should be written to support the external system port power switching implementation.
8	1	R/W	Power Switching Mode USB Host Controller implements a global power switching mode. 0: Global Switching 1: Individual Switching This bit is only valid when NoPowerSwitching is cleared.
7 to 0	H'02	R	Number Downstream Ports USB Host Controller supports one downstream port.

(20) HcRhDescriptorB Register

This register is only reset by a power-on reset. It is written during system initialization to configure the Root Hub. This register should not be written during normal operation.

Register Name: HcRhDescriptorB **Offset: 4C to 4F**

Bit	Initial Value	R/W	Description
31 to 16	H'0006	R/W	Port Power Control Mask USB Host Controller implements global-power switching. These bits are only valid if NoPowerSwitching is cleared and PowerSwitchingMode bit is set (individual port switching). When set, the port only responds to individual port power switching commands (set/ClearPortPower). When cleared, the port only responds to global power switching commands (set/ClearGlobalPower). 0: Device not removable 1: Global-power switching is masked Port Bit relationship Bit 16: Reserved Bit 17: Port 1 Bit 18: Port 2 : Bit 31: Port 15 Unimplemented ports are reserved. These bits are always read as 0. The write value should always be 0.
15 to 0	H'0000	R/W	Device Removable USB Host Controller ports default to removable devices. 0: Device removable 1: Device not removable Port Bit relationship Bit 0: Reserved Bit 1: Port 1 Bit 2: Port 2 : Bit 15: Port 15 Unimplemented ports are reserved. These bits are always read as 0. The write value should always be 0.

(21) HcRhStatus Register

This register is reset by the UsbReset state.

Register Name: HcRhStatus **Offset: 50 to 53**

Bit	Initial Value	R/W	Description
31	—	W	(write) Clear Remote Wakeup Enable Writing 1 to this bit clears DeviceRemoteWakeupEnable bit. Writing 0 has no effect
30 to 18	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
17	0	R/W	Over Current Indicator Change This bit is set when OverCurrentIndicator changes. Writing 1 clears this bit. Writing 0 has no effect.
16	0	R/W	(read) Local Power Status Change Not supported by this LSI. The read value should always be 0. (write) SetGlobalPower Write 1 issues a SetGlobalPower command to the ports. Writing 0 has no effect.
15	0	R/W	(read) Device Remote Wakeup Enable This bit enables ports' ConnectStatusChange as a remote wakeup event. 0: Disabled 1: Enabled (write) SetRemoteWakeupEnable Writing 1 sets the DeviceRemoteWakeupEnable bit. Writing 0 has no effect.
14 to 2	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
1	0	R	Over Current Indicator This bit reflects the state of the OVRCUR pin. This bit is only valid if NoOverCurrentProtection and OverCurrentProtectionMode bits are cleared. 0: No over-current condition 1: Over-current condition

Register Name: HcRhStatus **Offset: 50 to 53**

Bit	Initial Value	R/W	Description
0	0	R/W	(read) Local Power Status Not Supported by this LSI. The read value should always be 0. (write) ClearGlobalPower Writing 1 issues a ClearGlobalPower command to the ports. Writing 0 has no effect.

(22) HcRhPortStatus1, 2 Register

This register is reset by the UsbReset state.

Register Name: HcRhPortStatus1, 2 Offset: 54 to 57, 58 to 5B

Bit	Initial Value	R/W	Description
31 to 21	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
20	0	R/W	Port Reset Status Change This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.
19	0	R/W	Port Over Current Indicator Change This bit is set when OverCurrentIndicator changes. Writing 1 clears this bit. Writing 0 has no effect.
18	0	R/W	Port Suspend Status Change This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.
17	0	R/W	Port Enable Status Change This bit indicates that the port has been disabled due to a hardware event (PortEnableStatus bit is cleared). 0: Port has not been disabled. 1: PortEnableStatus bit has been cleared.
16	0	R/W	Connect Status Change This bit indicates a connection or disconnection event has been detected. Writing 1 clears this bit. Writing 0 has no effect. 0: No connect/disconnect event. 1: Hardware detection of connect/disconnect event. Note: If DeviceRemoveable is set, this bit resets to 1.
15 to 10	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

Register Name: HcRhPortStatus1, 2 Offset: 54 to 57, 58 to 5B

Bit	Initial Value	R/W	Description
9	×	R/W	(read) Low Speed Device Attached This bit defines the speed (and bus idle) of the attached device. It is only valid when CurrentConnectStatus is set. 0: Full Speed device 1: Low Speed device (write) ClearPortPower Writing 1 clears the PortPowerStatus bit. Writing 0 has no effect.
8	0	R/W	(read) Port Power Status This bit reflects the power state of the port regardless of power switching mode. 0: Port power is off. 1: Port power is on. Note: If NoPowerSwitching bit is set, the read value should always be 0. (write) SetPortPower Writing 1 sets the PortPowerStatus bit. Writing 0 has no effect.
7 to 5	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
4	0	R/W	(read) Port Reset Status 0: Port reset signal is not active. 1: Port reset signal is active. (write) SetPortReset Writing 1 sets the PortResetStatus bit. Writing 0 has no effect.

Register Name: HcRhPortStatus1, 2 Offset: 54 to 57, 58 to 5B

Bit	Initial Value	R/W	Description
3	0	R/W	(read) Port Over Current Indicator USB Host Controller supports global over-current reporting. This bit reflects the state of the OVRCUR pin dedicated to this port. This bit is only valid if NoOverCurrentProtection bit is cleared and OverCurrentProtectionMode is set. 0: No over-current condition 1: Over-current condition (write) ClearSuspendStatus Writing 1 initiates the selective resume sequence for the port. Writing 0 has no effect.
2	0	R/W	(read) Port Suspend Status 0: Port is not suspended 1: Port is selectively suspended (write) SetPortSuspend Writing 1 sets the PortSuspendStatus bit. Writing 0 has no effect.
1	0	R/W	(read) Port Enable Status 0: Port disabled. 1: Port enabled. (write) SetPortEnable Writing 1 sets the PortEnableStatus bit. Writing 0 has no effect.
0	0	R/W	(read) Current Connect Status 0: No device connected. 1: Device connected. Note: If DeviceRemoveable bit is set (not removable) this bit is always read as 1. (write) ClearPortEnable Writing 1 clears the PortEnableStatus bit. Writing 0 has no effect.

Note: ×: Will have an effect on the status of the transceiver.

14.4 USB2.0-FUNCTION Controller

This module is a USB controller that provides USB function operations and supports high-speed and full-speed transfers defined by USB specification 2.0.

This module supports all of the transfer types defined by the USB specification. It has up to 10 Kbytes of buffer memory for data transfer and provides a maximum of ten pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE9, based on the function devices or user system for communication. A local bus interface (dedicated to a DMA interface) is provided separately from the CPU bus interface for systems involving high-speed large-size data transfers.

14.4.1 Features

(1) Function Controller Supporting USB High-Speed Operation

- Supporting the interface conforming to UTMI+ specification 1.0 (UTMI+ level 3; the On-the-Go function not supported)

(2) All Types of USB Transfers Supported

Support of all types of USB transfers including isochronous transfer

- Control transfer
- Bulk transfer
- Interrupt transfer (high bandwidth transfer not supported)
- Isochronous transfer (high bandwidth transfer not supported)

(3) Bus Interfaces

- 32-bit-width HPB interface
- Two channels of DMA interface available
  (DMAC interface can be selected separately from the CPU bus interface)
- High-speed data transfer for access to the internal FIFOs at 60 Mbytes/second (in UTMI+ 8-bit mode)

(4) Pipe Configuration

- On-chip 10-Kbyte (max) buffer memory for USB communications
- Up to ten pipes can be selected (including the default control pipe)
- Programmable pipe configuration
- Endpoint numbers can be assigned flexibly to PIPE1 to PIPE9.

- Transfer conditions that can be set for each pipe:

PIPE0:	Control transfer, 64-byte fixed single buffer
PIPE1 and PIPE2:	Bulk transfers/isochronous transfer, continuous transfer mode, programmable buffer size (up to 2 Kbytes: double buffer can be specified)
PIPE3 to PIPE5:	Bulk transfer, continuous transfer mode, programmable buffer size (up to 2 Kbytes: double buffer can be specified)
PIPE6 to PIPE9:	Interrupt transfer, 64-byte fixed single buffer

(5) Features of USB Function Operation

- Support of high-speed transfer (480 Mbps) and full-speed transfer (12 Mbps)
- Automatic recognition of high-speed operation or full-speed operation based on automatic response to the reset handshake
- Control transfer stage control function
- Device state control function
- Auto response function for SET_ADDRESS request
- NAK response interrupt function (NRDY)
- SOF interpolation function

(6) Other Features

- Byte endian swap function for supporting both big and little endian data formats
- Transfer ending function using transaction count
- DMA transfer ending function using external triggers (TEND or WREND signal)
- SOF pulse output function
- BRDY interrupt event notification timing change function (BFRE)
- Function that automatically clears the buffer memory after the data for the pipe specified at the DxFIFO port has been read (DCLRM)
- NAK setting function for response PID generated by end of transfer (SHTNAK)

(7) Applications

Navigation systems, DVD recorders, set-top boxes, audio equipment, printers, external storage equipment, and other systems incorporating USB functions

14.4.2 Block Diagram

Figure 14.2 shows a block diagram of this module.

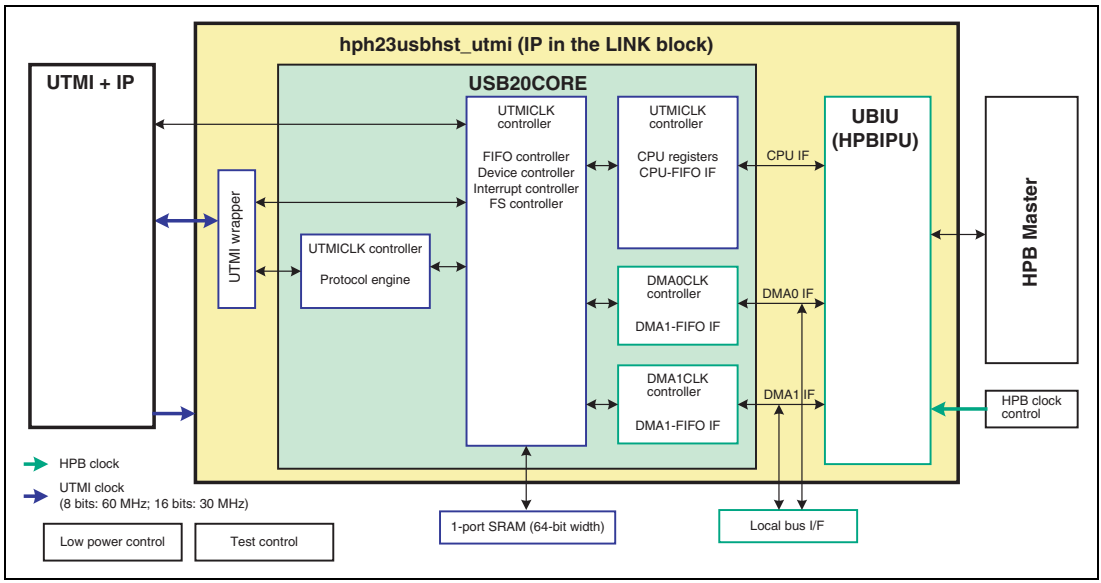


Figure 14.2 Block Diagram

Data transfer between this module and the host controller connected to the USB bus uses buffer memory assigned to each pipe. For communication between the function and host controllers, this module converts the data stored in buffer memory into USB data packets and outputs them to the USB bus serially, and also inputs data packets from the USB bus and stores them in buffer memory.

14.4.3 Functional Overview

(1) Controller Function

This module automatically detects the USB transfer speed.

(2) Bus Interface

This module supports the interface of a 32-bit-width HPB.

(a) FIFO Buffer Memory Access

The following two types of access are available for the FIFO buffer memory for USB data transfers. To read from or write to the FIFO buffer memory, access (read or write) the FIFO ports through the CPU or DMAC.

1. CPU access
   Specify a FIFO port address and write or read data to or from the FIFO buffer memory.
2. DMA access
   Specify a FIFO port address through the DMAC in the CPU or the dedicated DMAC and write or read data to or from the FIFO buffer memory.

USB data transfer is done in little endian. The byte endian swap function is available for FIFO port access; for 16-bit or 32-bit access, the endian can be switched through register settings.

(b) FIFO Buffer Memory Access from Local Bus

A local bus interface (dedicated to the DMA interface) is available separately from the CPU bus interface, enabling FIFO buffer access for high-speed large-size data transfer.

(3) USB Events

This module sends an interrupt to the user system to notify an event in USB operation. This module asserts the UCL_Dx_DREQ signal to notify that the pipe selected for the DMA interface has become ready for access.

Sending interrupts can be enabled or disabled separately for each interrupt type and source through software settings.

(4) USB Data Transfers

This module performs all types of USB data transfer: control transfer, bulk transfer, interrupt transfer, and isochronous transfer. The following numbers of pipes are available for each transfer type.

1. One pipe dedicated to control transfer
2. Four pipes dedicated to interrupt transfer
3. Three pipes dedicated to bulk transfer
4. Two pipes selectively used for bulk transfer or isochronous transfer

Each pipe should be set up as necessary for the desired USB transfer in accordance with the user system; for example, transfer type, endpoint number, or maximum packet size.

This module provides up to 10 Kbytes of buffer memory. For pipes dedicated for bulk transfer or selective pipes for bulk transfer or isochronous transfer, buffer memory assignment, buffer operating mode setting, or other necessary settings should be made in accordance with the user system. By setting the buffer operating mode, such as double buffer structure or continuous data packet transfer mode, fast data transfer is achieved with fewer interrupts.

Use three FIFO port registers to access the buffer memory from the user system controller CPU or DMA controller.

(5) Functions Available for Access from DMAC

This module provides two channels of the DMA interface having the following functions.

1. Transfer end notification function using a transfer end signal (only when the local bus is selected)
2. Automatic FIFO buffer clear function when a zero-length packet is received
3. Transfer ending function using the transaction counter

(6) SOF Pulse Output Function

This module provides a function for outputting an SOF pulse to indicate the SOF packet transfer timing. An SOF pulse is output when an SOF packet is received. Even if an SOF packet is damaged, pulses are output periodically using the SOF interpolation timer.

14.4.4 Register Specifications

Format of Register Description Tables

(1) Bit number — Each register is connected to the 16-bit internal bus. Bits 15 to 8 are allocated to an even address and bits 7 to 0 are allocated to an odd address.

(2) Register state after reset — Indicates the initial state of the register immediately after a reset. The H/W reset row shows the state after an external reset signal is input through the EXL_SYSRST pin. The USB reset row shows the state after a USB bus reset is detected. Notes on a reset operation may be added when necessary. "-" indicates that the bit is not manipulated by this module and retains the user setting. "?" indicates an undefined value.

(3) S/W access condition — Access type allowed for software.

(4) H/W access condition — Access type to be done by this module except for a reset operation.
- R: Read-only
- W: Write-only
- R/W: Readable/writable
- R(0): Only 0 can be read
- W(1): Only 1 can be written

(5) Remarks — Shows a remark number or a description number to be referred to.

(6) Name — Bit symbol and bit name.

(7) Function — Functional description. When a bit is read, the last value written by software or hardware will be read unless otherwise specified.

<Example>

The shaded bit has no function. The write value should always be 0.

(1) Bit number →	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit symbol →		A bit	B bit	C bit												
(2) H/W reset →	?	0	0	0												
USB reset →	?	0	-	-												

Bit	Name	Function	S/W	H/W	Remarks
15	This bit has no function. The write value should always be 0.				
14	A bit AAA enable	0: Operation is disabled 1: Operation is enabled	R/W	R	
13	B bit BBB operation	0: Low output 1: High output	R	W	
12	C bit CCC control	0: ... 1: ...	R(0)/ W(1)	R	
	(6)	(7)	(3)	(4)	(5)

<<Remarks>>

- **List of Registers**

Table 14.6 is a list of the registers in this module. Append H'FFE6_0 as the upper bits of the register addresses shown in the list.

Table 14.6 List of Registers

Address	Symbol	Name	Index
H'000	SYSCFG0	System configuration control register	
H'002	BUSWAIT	CPU bus wait register	
H'004	SYSSTS0	System configuration status register	
H'006			
H'008	DVSTCTR0	Device state control register	
H'00A			
H'00C	TESTMODE	Test mode register	
H'00E			
H'010	D0FBCFG	DMA0-FIFO bus configuration register	
H'012	D1FBCFG	DMA1-FIFO bus configuration register	
H'014	CFIFO	CFIFO port register	
H'016			
H'018	D0FIFO	D0FIFO port register	
H'01A			
H'01C	D1FIFO	D1FIFO port register	
H'01E			
H'020	CFIFOSEL	CFIFO port select register	
H'022	CFIFOCTR	CFIFO port control register	
H'024			
H'026			
H'028	D0FIFOSEL	D0FIFO port select register	
H'02A	D0FIFOCTR	D0FIFO port control register	
H'02C	D1FIFOSEL	D1FIFO port select register	
H'02E	D1FIFOCTR	D1FIFO port control register	
H'030	INTENB0	Interrupt enable register 0	
H'032	INTENB1	Interrupt enable register 1	
H'034			

Address	Symbol	Name	Index
H'036	BRDYENB	BRDY Interrupt enable register	
H'038	NRDYENB	NRDY Interrupt enable register	
H'03A	BEMPENB	BEMP Interrupt enable register	
H'03C	SOFCFG	SOF output configuration register	
H'03E			
H'040	INTSTS0	Interrupt status register 0	
H'042	INTSTS1	Interrupt status register 1	
H'044			
H'046	BRDYSTS	BRDY Interrupt status register	
H'048	NRDYSTS	NRDY Interrupt status register	
H'04A	BEMPSTS	BEMP Interrupt status register	
H'04C	FRMNUM	Frame number register	
H'04E	UFRMNUM	μFrame number register	
H'050	USBADDR	USB address register	
H'054	USBREQ	USB request type register	
H'056	USBVAL	USB request value register	
H'058	USBINDX	USB request index register	
H'05A	USBLENG	USB request length register	
H'05C	DCPCFG	DCP configuration register	
H'05E	DCPMAXP	DCP maximum packet size register	
H'060	DCPCTR	DCP control register	
H'062			
H'064	PIPESEL	Pipe window select register	
H'066			
H'068	PIPECFG	Pipe configuration register	
H'06A	PIPEBUF	Pipe buffer setting register	
H'06C	PIPEMAXP	Pipe maximum packet size register	
H'06E	PIPEPERI	Pipe cycle control register	
H'070	PIPE1CTR	PIPE1 control register	
H'072	PIPE2CTR	PIPE2 control register	
H'074	PIPE3CTR	PIPE3 control register	
H'076	PIPE4CTR	PIPE4 control register	

Address	Symbol	Name	Index
H'078	PIPE5CTR	PIPE5 control register	
H'07A	PIPE6CTR	PIPE6 control register	
H'07C	PIPE7CTR	PIPE7 control register	
H'07E	PIPE8CTR	PIPE8 control register	
H'080	PIPE9CTR	PIPE9 control register	
H'082-08E			
H'090	PIPE1TRE	PIPE1 transaction counter enable register	
H'092	PIPE1TRN	PIPE1 transaction counter register	
H'094	PIPE2TRE	PIPE2 transaction counter enable register	
H'096	PIPE2TRN	PIPE2 transaction counter register	
H'098	PIPE3TRE	PIPE3 transaction counter enable register	
H'09A	PIPE3TRN	PIPE3 transaction counter register	
H'09C	PIPE4TRE	PIPE4 transaction counter enable register	
H'09E	PIPE4TRN	PIPE4 transaction counter register	
H'0A0	PIPE5TRE	PIPE5 transaction counter enable register	
H'0A2	PIPE5TRN	PIPE5 transaction counter register	
H'0A4-0CE			
H'0D0	DEVADD0	Device address 0 configuration register	
H'0D2	DEVADD1	Device address 1 configuration register	
H'0D4	DEVADD2	Device address 2 configuration register	
H'0D6	DEVADD3	Device address 3 configuration register	
H'0D8	DEVADD4	Device address 4 configuration register	
H'0DA	DEVADD7	Device address 5 configuration register	
H'0DC	DEVADD6	Device address 6 configuration register	
H'0DE	DEVADD7	Device address 7 configuration register	
H'0E0	DEVADD8	Device address 8 configuration register	
H'0E2	DEVADD9	Device address 9 configuration register	
H'0E4	DEVADDA	Device address A configuration register	
H'0E6-100			
H'102	SUSPMODE	UTMI suspend mode register	

No register is allocated to the shaded addresses. Do not access these addresses.

- **List of Bit Symbols**

Table 14.7 is a list of the bit symbols in this module. Append H'FFE6_0 as the upper bits of the register addresses shown in the list.

Table 14.7 List of Bit Symbols

Addr	Register Name	Even Address								Odd Address							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H'000	SYSCFG0									HSE	DCFM	DRPD	DPRPU				USBE
H'002	BUSWAIT													BWAIT			
H'004	SYSSTS0	OVCMON											HDDM		IDMON	LNST	
H'006																	
H'008	DVSTCTR0						EXTLP0	VBOUT	WKUP	RWUPE	USBRST	RESUME	UACT		RHST		
H'00A																	
H'00C	TESTMODE													UTST			
H'00E																	
H'010	D0FBCFG			DFACC									TENDE				
H'012	D1FBCFG			DFACC									TENDE				
H'014	CFIFO0	CFPORT[31:16]															
H'016	CFIFO1	CFPORT[15:0]															
H'018	D0FIFO0	D0FPORT[31:16]															
H'01A	D0FIFO1	D0FPORT[15:0]															
H'01C	D1FIFO0	D1FIPORT[31:16]															
H'01E	D1FIFO1	D1FIPORT[15:0]															
H'020	CFIFOSEL	RCNT	REW			MBW			BIGEND			ISEL		CURPIPE			
H'022	CFIFOCTR	BVAL	BCLR	FRDY		DTLN											
H'024																	
H'026																	
H'028	D0FIFOSEL	RCNT	REW	DCLRM	DREQE	MBW			BIGEND					CURPIPE			

Addr	Register Name	Even Address 15	14	13	12	11	10	9	8	Odd Address 7	6	5	4	3	2	1	0
H'02A	D0FIFOCTR	BVAL	BCLR	FRDY		DTLN											
H'02C	D1FIFOSEL	RCNT	REW	DCLRM	DREQE	MBW			BIGEND					CURPIPE			
H'02E	D1FIFOCTR	BVAL	BCLR	FRDY		DTLN											
H'030	INTENB0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE								
H'032	INTENB1	OVRCRE	BCHGE		DTCHE	ATTCHE					EOFERRE	SIGNE	SACKE				
H'034																	
H'036	BRDYENB							PIPEBRDYE									
H'038	NRDYENB							PIPENRDYE									
H'03A	BEMPENB							PIPEBEMPE									
H'03C	SOFCFG								TRNENSEL		BRDYM	INTL	EDGESTS	SOFM			
H'03E																	
H'040	INTSTS0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ			VALID	CTSQ		
H'042	INTSTS1	OVRCR	BCHG		DTCH	ATTCH					EOFERR	SIGN	SACK				
H'044																	
H'046	BRDYSTS							PIPEBRDY									
H'048	NRDYSTS							PIPENRDY									
H'04A	BEMPSTS							PIPEBEMP									
H'04C	FRMNUM	OVRN	CRCE				FRNM										
H'04E	UFRMNUM														UFRNM		
H'050	USBADDR										USBADDR						
H'052																	
H'054	USBREQ	bRequest								bmRequestType							
H'056	USBVAL	wValue															
H'058	USBINDX	wIndex															
H'05A	USBLENG	wLength															
H'05C	DCPCFG												DIR				
H'05E	DCPMAXP	DEVSEL									MXPS						

Addr	Register Name	Even Address 15	14	13	12	11	10	9	8	Odd Address 7	6	5	4	3	2	1	0
H'060	DCPCTR	BSTS	SUREQ	CSCLR	CSSTS	SUREQCLR			SQCLR	SQSET	SQMON	PBUSY	PINGE		CCPL	PID	
H'062																	
H'064	PIPESEL													PIPESEL			
H'066																	
H'068	PIPECFG	TYPE					BFRE	DBLB	CNTMD	SHTNAK			DIR	EPNUM			
H'06A	PIPEBUF		BUFSIZE								BUFNMB						
H'06C	PIPEMAXP	DEVSEL					MXPS										
H'06E	PIPEPERI				IFIS										IITV		
H'070	PIPE1CTR	BSTS	INBUFM	CSCLR	CSSTS		ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY				PID	
H'072	PIPE2CTR	BSTS	INBUFM	CSCLR	CSSTS		ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY				PID	
H'074	PIPE3CTR	BSTS	INBUFM	CSCLR	CSSTS		ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY				PID	
H'076	PIPE4CTR	BSTS	INBUFM	CSCLR	CSSTS		ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY				PID	
H'078	PIPE5CTR	BSTS	INBUFM	CSCLR	CSSTS		ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY				PID	
H'07A	PIPE6CTR	BSTS		CSCLR	CSSTS			ACLRM	SQCLR	SQSET	SQMON	PBUSY				PID	
H'07C	PIPE7CTR	BSTS		CSCLR	CSSTS			ACLRM	SQCLR	SQSET	SQMON	PBUSY				PID	
H'07E	PIPE8CTR	BSTS		CSCLR	CSSTS			ACLRM	SQCLR	SQSET	SQMON	PBUSY				PID	
H'080	PIPE9CTR	BSTS		CSCLR	CSSTS			ACLRM	SQCLR	SQSET	SQMON	PBUSY				PID	
H'082-08E																	
H'090	PIPE1TRE							TRENB	TRCLR								
H'092	PIPE1TRN	TRNCNT															
H'094	PIPE2TRE							TRENB	TRCLR								

Addr	Register Name	Even Address								Odd Address							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H'096	PIPE2TRN	TRNCNT															
H'098	PIPE3TRE							TRENB	TRCLR								
H'09A	PIPE3TRN	TRNCNT															
H'09C	PIPE4TRE							TRENB	TRCLR								
H'09E	PIPE4TRN	TRNCNT															
H'0A0	PIPE5TRN							TRENB	TRCLR								
H'0A2	PIPE6TRN	TRNCNT															
H'0A4-0CE																	
H'0D0	DEVADD0		HPPHUB				HUBPORT			USBSPD							
H'0D2	DEVADD1		HPPHUB				HUBPORT			USBSPD							
H'0D4	DEVADD2		HPPHUB				HUBPORT			USBSPD							
H'0D6	DEVADD3		HPPHUB				HUBPORT			USBSPD							
H'0D8	DEVADD4		HPPHUB				HUBPORT			USBSPD							
H'0DA	DEVADD5		HPPHUB				HUBPORT			USBSPD							
H'0DC	DEVADD6		HPPHUB				HUBPORT			USBSPD							
H'0DE	DEVADD7		HPPHUB				HUBPORT			USBSPD							
H'0E0	DEVADD8		HPPHUB				HUBPORT			USBSPD							
H'0E2	DEVADD9		HPPHUB				HUBPORT			USBSPD							
H'0E4	DEVADDA		HPPHUB				HUBPORT			USBSPD							
H'0E6-100																	
H'102	SUSPMODE		SUSP														

System Configuration Control

- **System Configuration Control Register (SYSCFG0)** **<Address: H'000>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSE			DPRPU				USBE
?	?	?	?	?	?	?	?	0	?	?	0	?	?	?	0
?	?	?	?	?	?	?	?	-	?	?	-	?	?	?	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 8	No function assigned; the write value should always be 0.				
7	HSE High-speed operation enable	Enables or disables high-speed operation. 0: High-speed operation is disabled (full speed) 1: High-speed operation is enabled (the speed is detected by this module)	R/W	R	
6, 5	No function assigned; the write value should always be 0.				
4	DPRPU D+ line resistor control	Selects whether to pull-up the D+ line. 0: Not pulled up 1: Pulled up	R/W	R	
3 to 1	No function assigned; the write value should always be 0.				
0	USBE USB block operation enable	Enables or disables the USB block operation. 0: USB block operation is disabled 1: USB block operation is enabled	R/W	R	

<<Remarks>>

This register can be written to even while the UTMI clock is stopped. In this case, note that the written value takes effect only after the UTMI clock restarts oscillation.

(a) High-Speed Operation Enable Bit (HSE)

Setting this bit to 1 enables high-speed operation. When HSE = 1, this module selects high-speed or full-speed operation according to the reset handshake result.

When HSE = 0, this module selects full-speed operation.
When HSE = 1, this module executes the reset handshake protocol and automatically selects high-speed or full-speed operation according to the handshake result.

Write to this bit only while DPRPU = 0.

(b) D+ and D- Line Resistor Control Bits (DPRPU)

Table 14.8 shows the resistor settings for the USB data bus. Make resistor settings through the DPRPU bit.

Table 14.8 USB Data Bus Resistor Control

Settings	USB Data Bus Resistor Control		
DPRPU	**D- Line**	**D+ Line**	**Remarks**
0	Open	Open	
1	Open	Pulled up	Choose this setting to operate the data bus.

When this bit is set to 1, this module pulls up the D+ line to 3.3 V to signal an attach to the USB host.

When this bit is changed from 1 to 0, this module stops pulling up the D+ line to signal a detach to the USB host.

(c) USB Block Operation Enable Bit (USBE)

Enable or disable operation of the USB block in this module through this bit setting.
When this bit is changed from 1 to 0, this module initializes the bits shown in table 14.9.

Table 14.9 Register Bits Initialized When USBE is Cleared to 0

Register	Bit	Remark
SYSSTS0	LNST	
DVSTCTR0	RHST	
INTSTS0	DVSQ	
USBADDR	USBADDR	
USBREQ	bRequest	
	bmRequestType	
USBVAL	wValue	
USBINDX	wIndex	
USBLENG	wLength	

Write to this bit only while SuspendM = 1 and the UTMI clock is oscillating.

- **CPU Bus Wait Register (BUSWAIT)** **<Address: H'002>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												BWAIT			
?	?	?	?	?	?	?	?	?	?	?	?	1	1	1	1
?	?	?	?	?	?	?	?	?	?	?	?	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 4	No function assigned; the write value should always be 0.				
3 to 0	BWAIT CPU bus access wait setting	These bits specify the number of wait cycles for access to this module 0000: No wait cycle (access cycles: 2) : : 0010: Two wait cycles (access cycles: 4) : : 0100: Four wait cycles (access cycles: 6) : : 1111: Fifteen wait cycles (access cycles: 17) (initial value)	R/W	R	

<<Remarks>>

None

CPU Bus Access Wait Setting Bits (BWAIT)

This setting depends on the width of the UTMI interface bus.

This module has the following restriction on access to the registers allocated to address H'04 and larger addresses:

Access cycle restriction: When continuously accessing registers in this module, each access cycle must be 66 ns or longer.

To meet the above restriction, wait cycles should be inserted depending on the CPU clock frequency. The initial value specifies the maximum wait cycle (17 clock cycles). Select an appropriate setting according to the frequency.

Note that this setting is also applied to the wait cycle for access to the FIFO port registers. The maximum speed for access to the FIFO ports is determined as follows.

- MBW = B'10 (32-bit width): 60 Mbytes/second max.
- MBW = B'01 (16-bit width): 30 Mbytes/second max.
- MBW = B'00 (8-bit width): 15 Mbytes/second max.

(2) System Configuration Status

- **System Configuration Status Register (SYSSTS0)** **<Address: H'004>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVCMON													IDMON	LNST	
?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bit	Name	Function	S/W	H/W	Remarks
15, 14	OVCMON OVRCR0A and OVCUR0B pin monitor	These bits indicate the status of the signals input to the OVCUR0A and OVCUR0B pins. 00: OVCUR0A = low; OVCUR0B = low 01: OVCUR0A = low; OVCUR0B = high 10: OVCUR0A = high; OVCUR0B = low 11: OVCUR0A = high; OVCUR0B = high	R	W	
13 to 3	No function assigned; the write value should always be 0.				
2	IDMON ID0 pin monitor	Indicates the status of the signal input to the ID0 pin. 0: ID0 = low 1: ID0 = high	R	W	
1, 0	LNST USB data line status monitor	These bits indicate the status of the USB lines. For details, refer to the descriptions below.	R	W	

<<Remarks>>

None

(a) OVCUR0A and OVCUR0B Pin Monitor Bits (OVCMON)

Bit 15 of SYSSTS0 indicates the status of the OVCUR0A pin and bit 14 indicates the input status of the OVCUR0B pin.

(b) ID0 Pin Monitor Bit (IDMON)

This bit indicates the input status of the ID0 pin.

(c) USB Data Line Status Monitor Bits (LNST)

Table 14.10 shows the line status of the USB data bus of this module. The LNST bit of SYSSTS0 indicates the status of the USB data bus lines (D+ and D- lines).

Read the LNST bit after setting USBE = 1 and before the attach processing (setting DPRPU = 1).

Table 14.10 USB Data Bus Line Status

LNST [1]	LNST [0]	Full-Speed Operation	High-speed Operation	Chirp Operation
0	0	SE0	Squelch	Squelch
0	1	J state	Not squelch	Chirp J
1	0	K state	Invalid	Chirp K
1	1	SE1	Invalid	Invalid

Chirp: The reset handshake protocol is being executed in high-speed operation enabled state (HSE = 1).

Squelch: SE0 or idle state

Not squelch: High-speed J state or high-speed K state

Chirp J: Chirp J state

Chirp K: Chirp K state

(3) USB Signal Control

- **Device State Control Register (DVSTCTR0)** **<Address: H'008>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					EXTLP0	VBOUT	WKUP						RHST		
?	?	?	?	?	0	0	0	?	?	?	?	?	0	0	0
?	?	?	?	?	-	-	0	?	?	?	?	?	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 11	No function assigned; the write value should always be 0.				
10	EXTLP0 EXTLP0 pin output control	Controls the output to the EXTLP0 pin. 0: EXTLP0 outputs a low level (initial value) 1: EXTLP0 outputs a high level	R/W	R	
9	VBOUT VBOUT0 pin output control	Controls the output to the VBOUT0 pin. 0: VBOUT0 outputs a low level (initial value) 1: VBOUT0 outputs a high level	R/W	R	
8	WKUP Wakeup output	Enables or disables the remote wakeup (resume signal output). 0: Does not output the remote wakeup signal 1: Outputs the remote wakeup signal	R/W (1)	R/W (0)	
7 to 3	No function assigned; the write value should always be 0.				
2 to 0	RHST Reset handshake	These bits indicate the reset handshake state. For details, refer to the descriptions below.	R	W	

<<Remarks>>

None

(a) Remote Wakeup (Resume Signal Output) Enable/Disable Bit (WKUP)

This module outputs a remote wakeup signal to the USB bus.

The output period of the remote wakeup signal is controlled by this module. When the WKUP bit is set to 1 by software, this module outputs K-State for 10 ms and then clears this bit to 0.

According to the USB specification, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is output. If 1 is written to this bit immediately after detection of the suspended state, this module outputs K-State after waiting for 2 ms.

Write 1 to the WKUP bit only while the device is in the suspended state (DVSQ = 1xx) and the USB host enables the remote wakeup.
When setting this bit to 1, do not stop the internal clock even in the suspended state (set WKUP = 1 while SCKE = 1).

(b) Reset Handshake Status Bits (RHST)

These bits indicate the reset handshake results. Table 14.11 shows the reset handshake results.

Table 14.11 Reset Handshake Results

Bus State	RHST Bit Value
Powered or disconnected state	B'000
Reset handshake in progress	B'100
Connected at low speed	—
Connected at full speed	B'010
Connected at high speed	B'011

When this module detects a USB bus reset while HSE = 1, these bits are set to B'100. After that, when this module outputs ChirpK and detects ChirpJK from the USB host three times, these bits are set to B'011. If the high-speed operation is not established within 2.5 ms after the ChirpK output, these bits are set to B'010.

When this module detects a USB bus reset while HSE = 0, these bits are set to B'010.

When these bits are set to B'010 or B'011 after this module detects a USB bus reset, a DVST interrupt occurs.

(4) Test Mode

- **Test Mode Register (TESTMODE)** **<Address: H'00C>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												UTST			
?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 4	No function assigned; the write value should always be 0.				
3 to 0	UTST Test mode	For details, refer to the description below.	R/W	R	

<<Remarks>>

None

(a) Test Mode Bits (UTST)

This module outputs the USB test signal in high-speed mode according to the value set to these bits. Table 14.12 shows the test mode operation of this module.

Table 14.12 Test Mode Operation

Test Mode	UTST Bit Setting
Normal operation	B'0000
Test_J	B'0001
Test_K	B'0010
Test_SE0_NAK	B'0011
Test_Packet	B'0100
Test_Force_Enable	—
Reserved	B'0101 to B'0111

These bits should be set according to the SetFeature request from the USB host during high-speed operation.

This module will not shift to the suspended state while the setting of these bits is any value from 0001 to 0100.

(5) DMA-FIFO Bus Access Control

D0FBCFG controls the DMA0-FIFO bus access, and D1FBCFG controls the DMA1-FIFO bus access. When the DMA0-FIFO or DMA1-FIFO bus is connected to the local bus, the corresponding DMA-FIFO bus configuration register setting is ignored.

- **DMA0-FIFO Bus Configuration Register (D0FBCFG)** **<Address: H'010>**
- **DMA1-FIFO Bus Configuration Register (D1FBCFG)** **<Address: H'012>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DFACC									TENDE				
?	?	0	0	?	?	?	?	?	?	?	0	?	?	?	?
?	?	-	-	?	?	?	?	?	?	?	-	?	?	?	?

Bit	Name	Function	S/W	H/W	Remarks
15, 14	No function assigned; the write value should always be 0.				
13, 12	DFACC DMAx-FIFO access mode	These bits specify the mode for access to the corresponding FIFO port 00: Cycle steal mode (initial value) 01: 16-byte continuous access mode 10: 32-byte continuous access mode 11: Setting ignored	R/W	R	
11 to 5	No function assigned; the write value should always be 0.				
4	TENDE TENDx_N signal enable	Enables the TENDx_N signal input. 0: TENDx_N signal is disabled 1: TENDx_N signal is enabled	R/W	R	
3 to 0	No function assigned; the write value should always be 0.				

<<Remarks>>

None

(a) DMA Transfer FIFO Access Mode Select Bits (DFACC)

These bits specify the transfer mode when the UCL_Dx_DREQ output is used for DMA transfer. When the DMA FIFO bus is connected to the local bus, the setting of these bits is ignored.

(b) TENDx_N Input Signal Enable Bit (TENDE)

This bit specifies whether to enable the TEND input for FIFO write access by DMA.
When the DMA FIFO bus is connected to the local bus, the setting of this bit is ignored.

(6) FIFO Ports

- **CFIFO Port Register (CFIFO)** **<Address: H'014>**
- **D0FIFO Port Register (D0FIFO)** **<Address: H'018>**
- **D1FIFO Port Register (D1FIFO)** **<Address: H'01C>**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFOPORT (Low)															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

- **CFIFO Port Register (CFIFO)** **<Address: H'016>**
- **D0FIFO Port Register (D0FIFO)** **<Address: H'01A>**
- **D1FIFO Port Register (D1FIFO)** **<Address: H'01E>**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFOPORT(High)															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
31 to 0	FIFOPORT FIFO port	These bits should be accessed to read receive data from the FIFO buffer or write transmit data to the FIFO buffer.	R/W	R/W	

<<Remarks>>

None

(a) FIFO Port Control

The transfer buffer memory in this module is configured as FIFO (FIFO buffers). To access a FIFO buffer, use the corresponding FIFO port register. There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO ports. Each FIFO port is configured of a port register (CFIFO, D0FIFO, or D1FIFO) that handles the reading of data from and the writing of data to the buffer memory, a select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).

Each FIFO port has the following characteristics.

1. To access the FIFO buffer for the DCP, use the CFIFO port.
2. To access the FIFO buffer using DMA transfer, use the DxFIFO ports.
3. The DxFIFO ports can also be accessed by the CPU.
4. When using the functions specific to the FIFO ports, the pipe number set to the CURPIPE bits (selected pipe) cannot be changed (signal input/output of DMA-related pins, etc.).
5. The registers configuring a FIFO port do not affect other FIFO ports.
6. The same pipe should not be assigned to two or more FIFO ports.
7. There are two types of buffer state: the access right on the CPU side or on the SIE side. When the buffer memory access right is on the SIE side, the buffer cannot be accessed from the CPU.

When the DMA0-FIFO bus is connected to the local bus, the buffer memory cannot be accessed through the D0FIFO port. Similarly, when the DMA1-FIFO bus is connected to the local bus, the D1FIFO port cannot be used.

(b) FIFO Port Bits (CFIFO, D0FIFO, and D1FIFO)

When these bits are accessed by software, this module accesses the FIFO buffer assigned to the pipe number set in the CURPIPE bits in the select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL).

These bits can be accessed only while the FRDY bit in the control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1 (or while this module asserts the UCL_Dx_DREQ output signal).

The valid bits depend on the settings of the MBW bits and the BIGEND bit. Tables 14.13 to 14.15 show the valid bits.

Table 14.13 Endian Operation in 32-Bit Access (MBW = 10)

BIGEND	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	Address N + 3	Address N + 2	Address N + 1	Address N + 0
1	Address N + 0	Address N + 1	Address N + 2	Address N + 3

Table 14.14 Endian Operation in 16-Bit Access (MBW = 01)

BIGEND	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	Write: Disabled Read: Prohibited*		Odd address	Even address
1	Even address	Odd address	Write: Disabled Read: Prohibited*	

Table 14.15 Endian Operation in 8-Bit Access (MBW = 00)

BIGEND	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	Write: Disabled Read: Prohibited*			Write: Enabled Read: Enabled
1	Write: Enabled Read: Enabled	Write: Disabled Read: Prohibited*		

<<Remarks>>

* Word or byte read from an invalid register is prohibited.

- **CFIFO Port Select Register (CFIFOSEL)** **<Address: H'020>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCNT	REW			MBW			BIGEND			ISEL			CURPIPE		
0	0	?	?	0	0	?	0	?	?	0	?	?	0	0	0
-	-	?	?	-	-	?	-	?	?	-	?	?	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15	RCNT Read count mode	Specifies the read mode for the DTLN bits in CFIFOCTR. 0: The DTLN bits are cleared when all of the receive data has been read 1: The DTLN bits are decremented every time the receive data is read	R/W	R	
14	REW Buffer pointer rewind	Specify 1 to rewind the buffer pointer. 0: The buffer pointer is not rewound 1: The buffer pointer is rewound	R(0)/W	R	
13, 12	No function assigned; the write value should always be 0.				
11, 10	MBW CFIFO port access bit width	These bits specify the access width (bits) for the CFIFO port. 00: 8-bit width 01: 16-bit width 10: 32-bit width 11: Setting prohibited	R/W	R	
9	No function assigned; the write value should always be 0.				
8	BIGEND FIFO port endian control	Specifies the byte endian for the CFIFO port. 0: Little endian 1: Big endian	R/W	R	
7, 6	No function assigned; the write value should always be 0.				
5	ISEL FIFO port access direction when DCP is selected	Specifies the direction of FIFO port access when the DCP is selected through the CURPIPE bits. 0: Reading from the buffer memory is selected 1: Writing to the buffer memory is selected	R/W	R	
4 to 3	No function assigned; the write value should always be 0.				

Bit	Name	Function	S/W	H/W	Remarks
2 to 0	CURPIPE FIFO port access pipe specification	Specifies the pipe number through which the CFIFO port is accessed. 0000: DCP 0001: PIPE1 0010: PIPE2 : : 1000: PIPE8 1001: PIPE9	R/W	R	

<<Remarks>>

None

Read Count Mode Bit (RCNT)

When this bit is cleared to 0, this module clears the DTLN bits in CFIFOCTR to 0 when all of the receive data has been read from the FIFO buffer that is assigned to the pipe specified in the CURPIPE bits (called the specified pipe).

When this bit is set to 1, this module decrements the DTLN bits in CFIFOCTR every time receive data is read from the FIFO buffer that is assigned to the specified pipe.

(c) Buffer Pointer Rewind Bit (REW)

When this bit is set to 1 during FIFO buffer reading while the specified pipe is set to the data receive direction, reading is restarted from the first data in the FIFO buffer (for the double-buffer structure, reading is restarted from the first data in the current read buffer).

Do not set REW = 1 simultaneously with modification of the CURPIPE bits.
Be sure to check that FRDY = 1 before setting REW = 1.

Use the BCLR bit to restart writing to the FIFO buffer from the first data for the pipe set to the transmit direction.

(d) CFIFO Port Access Bit Width Bits (MBW)

Set the access width (bits) for the CFIFO port.

When this bit is set and reading is started while the pipe specified through the CURPIPE bits is set to the receive direction, do not modify this bit until all data has been read.

When the specified pipe is in the receive direction, set the CURPIPE and MBW bits simultaneously.

When the specified pipe is in the transmit direction, the width cannot be changed from the eight bits to 16 or 32 bits or from 16 bits to 32 bits while data is being written to the buffer memory. Even when the 8-bit or 16-bit width is selected, an odd number of bytes can be written through byte access control.

(e) CFIFO Port Endian Control Bit (BIGEND)

Specify the byte endian for the CFIFO port through this bit. For details, refer to (b) FIFO Port Bits in (2) CFIFO Port Register.

(f) FIFO Port Access Direction Bits When DCP is Selected (ISEL)

To modify this bit when the specified pipe is DCP, write to this bit and then read it to check that the read value matches the written value before moving to the next processing.

When this bit is modified during access to the FIFO buffer, the access results up to that point can be retained, and after the bit is restored to the previous value, access can be continued.

Set this bit and the CURPIPE bits simultaneously.

(g) FIFO Port Access Pipe Specification Bits (CURPIPE)

Specify a desired pipe number for which data is read or written through the CFIFO port.

To modify these bits, write to this bit and then read it to check that the read value matches the written value before moving to the next processing.

Do not specify the same pipe in the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

When these bits are modified during access to the FIFO buffer, the access results up to that point can be retained, and after the bits are restored to the previous value, access can be continued.

- **D0FIFO Port Select Register (D0FIFOSEL)** **<Address: H'028>**
- **D1FIFO Port Select Register (D1FIFOSEL)** **<Address: H'02C>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCNT	REW	DCLRM	DREQE	MBW			BIGEND					CURPIPE			
0	0	0	0	0	0	?	0	?	?	?	?	0	0	0	0
-	-	-	-	-	-	?	-	?	?	?	?	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15	RCNT Read count mode	Specifies the read mode for the DTLN bits in Dx_FIFOCTR. 0: The DTLN bits are cleared when all of the receive data has been read 1: The DTLN bits are decremented every time the receive data is read	R/W	R	
14	REW Buffer pointer rewind	Specify 1 to rewind the buffer pointer. 0: The buffer pointer is not rewound 1: The buffer pointer is rewound	R(0)/W	R	
13	DCLRM Auto buffer memory clear mode enable after specified pipe data is read	Enables or disables the auto buffer memory clear mode after the specified pipe data is read. 0: Auto buffer clear mode is disabled 1: Auto buffer clear mode is enabled	R/W	R	
12	DREQE UCL_Dx_DREQ signal output enable	Enables or disables output of the UCL_Dx_DREQ signal. 0: Output is disabled 1: Output is enabled	R/W	R	
11, 10	MBW FIFO port access bit width	These bits specify the access width (bits) for the FIFO port. 00: 8-bit width 01: 16-bit width 10: 32-bit width 11: Setting prohibited	R/W	R	
9	No function assigned; the write value should always be 0.				

Bit	Name	Function	S/W	H/W	Remarks
8	BIGEND FIFO port endian control	Specifies the byte endian for each FIFO port. 0: Little endian 1: Big endian	R/W	R	
7 to 4	No function assigned; the write value should always be 0.				
3 to 0	CURPIPE FIFO port access pipe specification	0000: None 0001: PIPE1 0010: PIPE2 : : 1000: PIPE8 1001: PIPE9	R/W	R	

<<Remarks>>

None

Read Count Mode Bit (RCNT)

When this bit is cleared to 0, this module clears the DTLN bits in DxFIFOCTR to 0 when all of the receive data has been read from the FIFO buffer (for the double-buffer structure, all receive data has been read from the current buffer) that is assigned to the pipe specified in the CURPIPE bits (called the specified pipe).

When this bit is set to 1, this module decrements the DTLN bits in DxFIFOCTR every time receive data is read from the FIFO buffer that is assigned to the specified pipe.

Be sure to clear this bit to 0 when DxFIFO is accessed with the BFRE bit set to 1.

(h) Buffer Pointer Rewind (REW)

When this bit is set to 1 during FIFO buffer reading while the specified pipe is set to the data receive direction, reading is restarted from the first data in the FIFO buffer (for the double-buffer structure, reading is restarted from the first data in the current read buffer). After this bit is set to 1 by software, this module automatically restores it to 0.

Do not set REW = 1 simultaneously with modification of the CURPIPE bits.
Be sure to check that FRDY = 1 before setting REW = 1.

When DxFIFO is accessed with the BFRE bit set to 1, do not set this bit to 1 in the state where short-packet data has been read.

Use the BCLR bit to restart writing to the FIFO buffer from the first data for the pipe in the transmit direction.

(i) Auto Buffer Memory Clear Mode Enable/Disable Bit (DCLRM)

Specify whether to automatically clear the FIFO buffer after the specified pipe data has been read. When this bit is set to 1, this module clears the buffer (processing specified by BCLR = 1) after a zero-length packet is received while the FIFO buffer assigned to the specified pipe is empty or after short-packet data is received and read while BFRE = 1.

Be sure to clear this bit to 0 when using this module with the BRDYM bit set to 1.

(j) UCL_Dx_DREQ Output Enable/Disable Bit (DREQE)

Enable or disable output of the UCL_Dx_DREQ signal through this bit.

To enable the UCL_Dx_DREQ signal output, set the CURPIPE bits and then set this bit to 1. Before modifying the CURPIPE bits, be sure to clear this bit to 0.

(k) DxFIFO Port Access Bit Width Bits (MBW)

Set the access width (bits) for the DxFIFO port through these bits. For details, refer to (c) CFIFO Port Access Bit Width Bits in (3), CFIFO Port Select Register (CFIFOSEL).

(l) DxFIFO Port Byte Endian Control Bit (BIGEND)

Specify the byte endian for the DxFIFO port through this bit. For details, refer to (b) FIFO Port Bits in (6), FIFO Ports.

(m) FIFO Port Access Pipe Specification Bits (CURPIPE)

Specify a desired pipe number for which data is read or written through the DxFIFO port.

To modify these bits, write to this bit and then read it to check that the read value matches the written value before moving to the next processing.

Do not specify the same pipe in the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

When these bits are modified during access to the FIFO buffer, the access results up to that point can be retained, and after the bits are restored to the previous value, access can be continued.

- **CFIFO Port Control Register (CFIFOCTR) <Address: H'022>**
- **D0FIFO Port Control Register (D0FIFOCTR) <Address: H'02A>**
- **D1FIFO Port Control Register (D1FIFOCTR) <Address: H'02E>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BVAL	BCLR	FRDY		DTLN											
0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0
-	-	-	?	-	-	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15	BVAL Buffer memory valid flag	Set this bit to 1 when writing has completed in the CPU-side FIFO buffer for the pipe specified in CURPIPE (called the specified pipe). 0: Invalid 1: Writing ended	R/ W(1)	R/W	
14	BCLR CPU buffer clear	Set this bit to 1 to clear the CPU-side buffer of the specified pipe. 0: Invalid 1: Buffer memory on the CPU side is cleared	R(0)/ W(1)	R	
13	FRDY FIFO port ready	Indicates whether the FIFO port is ready for access. 0: FIFO port access is not ready 1: FIFO port access is ready	R	W	
12	No function assigned; the write value should always be 0.				
11 to 0	DTLN Receive data length	These bits indicate the length of received data.	R	W	

<<Remarks>>

None

Buffer Memory Valid Flag (BVAL)

When the pipe specified through the CURPIPE bits (called the specified pipe) is set to the transmit direction, set this bit to 1 under any of the following conditions. After this bit is set to 1, this module switches the CPU-side FIFO buffer to the SIE side and makes it ready for transmission.

1. To transmit a short packet, set this bit to 1 after data has been written.
2. To transmit a zero-length packet, set this bit to 1 before writing data to the FIFO buffer.
3. For a pipe in the continuous transfer mode, set this bit to 1 after data of an integer multiple of MaxPacketSize bytes and less than BufferSize has been written.

After data of the MaxPacketSize bytes has been written to the pipe in the continuous transfer mode, this module automatically sets this bit to 1 to switch the CPU-side FIFO buffer to the SIE and make it ready for transmission.

Be sure to write 1 to this bit only while FRDY = 1.

Do not write 1 to this bit when the specified pipe is in the receive direction.

(n) CPU Buffer Clear Bit (BCLR)

When this bit is set to 1, this module clears the CPU side of the FIFO buffers assigned to the specified pipe.

When the FIFO buffer memory assigned to the specified pipe is in the double-buffer structure, this module clears only one of the FIFO buffers even while both buffers are ready to be read.

When the specified pipe is DCP, this module clears the FIFO buffer when BCLR is set to 1 regardless of whether the FIFO buffer is on the CPU or SIE side. To clear the SIE-side buffer, be sure to set the PID bits for DCP to NAK before setting BCLR = 1.

When the specified pipe is in the transmit direction, if 1 is simultaneously written to both the BVAL and BCLR bits, this module clears the data written up to that point and makes the pipe ready for transmission of a zero-length packet.

When the specified pipe is not DCP, be sure to write 1 to this bit only while FRDY = 1.

(o) FIFO Port Ready Bit (FRDY)

This bit indicates whether the FIFO port is ready to be accessed by the CPU (DMAC).

In the following cases, this module sets FRDY = 1 but data cannot be read from the FIFO port because it has no data. In these cases, set BCLR = 1 to clear the FIFO buffer and make it ready for the next data transfer.

1. When a zero-length packet is received while the FIFO buffer assigned for the specified pipe is empty
2. When short-packet data has been received and read while BFRE = 1

(p) Receive Data Length Bits (DTLN)

These bits indicate the length of received data. During FIFO buffer reading, the value of these bits depends on the RCNT bit setting as follows.

1. When RCNT = 0:

   These bits indicate the receive data length until the CPU (DMAC) has completed reading all receive data from one FIFO buffer.

   While BFRE = 1, this module retains the receive data length until BCLR = 1 even after reading has been completed.
2. When RCNT = 1:

   This module decrements the value in the DTLN bits every time data is read.

   (Decrements by 1 when MBW = 0 or by 2 when MBW = 1)

When all data has been read from one FIFO buffer, this module clears the DTLN bits to 0. In the double-buffer structure, if data reading has been completed in one FIFO buffer before completion in another FIFO buffer, the DTLN bits indicate the receive data length for the latter FIFO buffer when reading has been completed in the former FIFO buffer.

When reading these bits during FIFO buffer reading while RCNT = 1, note that these bits are updated within 150 ns after a read cycle for the FIFO port.

(7) Interrupt Enable

- **Interrupt Enable Register 0 (INTENB0)** **<Address: H'030>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE								
0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?
-	-	-	-	-	-	-	-	?	?	?	?	?	?	?	?

Bit	Name	Function	S/W	H/W	Remarks
15	VBSE VBUS interrupt enable	Enables or disables assertion of INT_N when a VBINT interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	
14	RSME Resume interrupt enable	Enables or disables assertion of INT_N when an RESM interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	
13	SOFE Frame number update interrupt enable	Enables or disables assertion of INT_N when an SOF interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	
12	DVSE Device state transition interrupt enable	Enables or disables assertion of INT_N when a DVST interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	
11	CTRE Control transfer stage transition interrupt enable	Enables or disables assertion of INT_N when a CTRT interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	
10	BEMPE Buffer empty interrupt enable	Enables or disables assertion of INT_N when a BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	

Bit	Name	Function	S/W	H/W	Remarks
9	NRDYE Buffer not ready response interrupt enable	Enables or disables assertion of INT_N when an NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	
8	BRDYE Buffer ready interrupt enable	Enables or disables assertion of INT_N when a BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	
7 to 0	No function assigned; the write value should always be 0.				

- **BRDY Interrupt Enable Register (BRDYENB)** **<Address: H'036>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PIPEBRDYE									
?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0
?	?	?	?	?	?	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 10	No function assigned; the write value should always be 0.				
9 to 0	PIPEBRDYE BRDY interrupt enable for each pipe	These bits specify whether to set the BRDY bit to 1 when a BRDY interrupt is detected in each pipe. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	

<<Remarks>>

* Each bit number corresponds to a pipe number.

(a) BRDY Interrupt Enable Bit for Each Pipe (PIPEBRDYE)

When this module detects a BRDY interrupt for the pipe corresponding to the PIPEBRDYE bit set to 1 by software, this module sets the corresponding PIPEBRDY bit in BRDYSTS to 1, sets the BRDY bit in INTSTS0 to 1, and asserts an interrupt signal through the INT_N pin.

While at least one of the PIPEBRDY bits in BRDYSTS is set to 1, if the corresponding PIPEBRDYE bit is changed from 0 to 1 by software, this module asserts an interrupt signal through the INT_N pin.

- **NRDY Interrupt Enable Register (NRDYENB)** **<Address: H'038>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PIPENRDYE									
?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0
?	?	?	?	?	?	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 10	No function assigned; the write value should always be 0.				
9 to 0	PIPENRDYE NRDY interrupt enable for each pipe	These bits specify whether to set the NRDY bit to 1 when an NRDY interrupt is detected in each pipe. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	

<<Remarks>>

* Each bit number corresponds to a pipe number.

NRDY Interrupt Enable Bit for Each Pipe (PIPENRDYE)

When this module detects an NRDY interrupt for the pipe corresponding to the PIPENRDYE bit set to 1 by software, this module sets the corresponding PIPENRDY bit in NRDYSTS to 1, sets the NRDY bit in INTSTS0 to 1, and asserts an interrupt signal through the INT_N pin.

While at least one of the PIPENRDY bits in NRDYSTS is set to 1, if the corresponding PIPENRDYE bit is changed from 0 to 1 by software, this module asserts an interrupt signal through the INT_N pin.

- **BEMP Interrupt Enable Register (BEMPENB)** **<Address: H'03A>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PIPEBEMPE									
?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0
?	?	?	?	?	?	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 10	No function assigned; the write value should always be 0.				
9 to 0	PIPEBEMPE BEMP interrupt enable for each pipe	These bits specify whether to set the BEMP bit to 1 when a BEMP interrupt is detected in each pipe. 0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	

<<Remarks>>

* Each bit number corresponds to a pipe number.

BEMP Interrupt Enable Bit for Each Pipe (PIPEBEMPE)

When this module detects a BEMP interrupt for the pipe corresponding to the PIPEBEMPE bit set to 1 by software, this module sets the corresponding PIPEBEMP bit in BEMPSTS to 1, sets the BEMP bit in INTSTS0 to 1, and asserts an interrupt signal through the INT_N pin.

While at least one of the PIPEBEMP bits in BEMPSTS is set to 1, if the corresponding PIPEBEMPE bit is changed from 0 to 1 by software, this module asserts an interrupt signal through the INT_N pin.

(8) SOF Control Register

- **SOF Pin Configuration Register (SOFCFG)** **<Address: H'03C>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									BRDYMM	INTL	EDGESTSTS	SOFM			
?	?	?	?	?	?	?	?	?	0	0	0	0	0	?	?
?	?	?	?	?	?	?	?	?	-	-	-	-	-	?	?

Bit	Name	Function	S/W	H/W	Remarks
15 to 7	No function assigned; the write value should always be 0.				
6	BRDYM PIPEBRDY interrupt status clear timing setting	Specifies the timing for clearing the PIPEBRDY interrupt status. 0: Software clears the status. 1: This module clears the status when the FIFO buffer is read or written to.	R/W	R	
5	INTL Interrupt output trigger type setting	Specifies the trigger type for the signal output through the INT_N pin. 0: Edge-trigger 1: Level-trigger	R/W	R	
4	EDGESTS Interrupt edge processing status	Indicates the status of the interrupt edge processing. 0: No interrupt edge is being processed 1: Interrupt edge processing is in progress	R/W	R	
3, 2	SOFM SOF pin function setting	These bits select the SOF pulse output mode. 00: SOF output disabled 01: SOF is output in 1-ms intervals 10: μSOF is output in 125-μs intervals 11: Reserved	R/W	R	
1, 0	No function assigned; the write value should always be 0.				

<<Remarks>>

* When setting BRDYM = 1, be sure to set INTL = 1 (level-trigger).

* When clearing the interrupt status and then stopping the system clock (SCKE = 0) while INTL = 0, be sure to check that EDGESTS = 0 before writing SCKE = 0.

(9) Interrupt Status

- **Interrupt Status Register 0 (INTSTS0)** **<Address: H'040>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ			VALID	CTSQ		
0	0	0	0	0	0	0	0	?	0	0	0	0	0	0	0
-	-	-	1	-	-	-	-	-	0	0	1	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15	VBINT VBUS change detection interrupt status	Indicates the VBUS change detection interrupt status. 0: VBUS interrupts not generated 1: VBUS interrupts generated	R/W(0)	W	
14	RESM Resume interrupt status	Indicates the resume detection interrupt status. 0: Resume interrupts not generated 1: Resume interrupts generated	R/W(0)	W	
13	SOFR Frame number refresh interrupt status	Indicates the frame number refresh interrupt status. 0: SOF interrupts not generated 1: SOF interrupts generated	R/W(0)	W	
12	DVST Device state transition interrupt status	Indicates the device state transition interrupt status. 0: Device state transition interrupts not generated 1: Device state transition interrupts generated	R/W(0)	W	
11	CTRT Control transfer stage transition interrupt status	Indicates the control transfer state transition interrupt status. 0: Control transfer state transition interrupts not generated 1: Control transfer state transition interrupts generated	R/W(0)	W	
10	BEMP BEMP interrupt status	Indicates the BEMP interrupt status. 0: BEMP interrupts not generated 1: BEMP interrupts generated	R	W	

Bit	Name	Function	S/W	H/W	Remarks
9	NRDY NRDY interrupt status	Indicates the NRDY interrupt status. 0: NRDY interrupts not generated 1: NRDY interrupts generated	R	W	
8	BRDY BRDY interrupt status	Indicates the BRDY interrupt status. 0: BRDY interrupts not generated 1: BRDY interrupts generated	R	W	
7	VBSTS VBUS input status	Indicates the VBUS pin input status. 0: The VBUS pin is at a low level 1: The VBUS pin is at a high level	R	W	
6 to 4	DVSQ Device state	These bits indicate the device state. 000: Powered state 001: Default state 010: Address state 011: Configured state 1xx: Suspended state	R	W	
3	VALID USB request receive	Indicates whether USB request reception has been detected. 0: Not detected 1: Setup packet received	R/W(0)	W	
2 to 0	CTSQ Control transfer stage	These bits indicate the control transfer stage. 000: Idle or setup stage 001: Control read data stage 010: Control read status stage 011: Control write data stage 100: Control write status stage 101: Control write (no data) status stage 110: Control transfer sequence error 111: Reserved	R	W	

<<Remarks>>

* To clear the status indicated by the VBINT, RESM, SOFR, DVST, or CTRT bit, write 0 only to the bit to be cleared and write 1 to the other bits. Do not write 0 to a status bit that indicates 0.

* This module detects any status change indicated by the VBINT and RESM bits of this register even while the clock is stopped (SCKE = 0) and notifies an interrupt if the interrupt is enabled. Clear the status by software only after the clock is enabled.

(a) VBUS Change Detection Interrupt Status Bit (VBINT)

This module sets this bit to 1 when a change in the value input to the VBUS pin (high to low or low to high) is detected. The module indicates the VBUS pin input value in the VBSTS bit. When a VBINT interrupt is generated, read the VBSTS bits several times by software to remove the chattering effect until the same value is read repeatedly from the bit.

(b) Resume Interrupt Status Bit (RESM)

This module sets this bit to 1 when a falling edge is detected on the DP pin while this module is in the suspended state (DVSQ = B'1XX).

(c) Frame Number Refresh Interrupt Status Bit (SOFR)

This module sets this bit to 1 under the following condition.

When the UACT bit is set to 1 by software, this module sets the SOFR bit to 1 with the frame number refresh timing. (This interrupt is detected in 1-ms intervals.)

Even if the SOF packet sent from the USB host is damaged, this module detects an SOFR interrupt by interpolating it internally.

(d) Device State Transition Interrupt Status Bit (DVST)

Upon detecting a change in the device state, this module updates the DVSQ value and sets the DVST bit to 1.

When this interrupt is generated, clear the status before this module detects the next device state transition.

(e) Control Transfer Stage Transition Interrupt Status Bit (CTRT)

Upon detecting a transition of the control transfer stage, this module updates the CTSQ value and sets the CTRT bit to 1.

When this interrupt is generated, clear the status before this module detects the next control transfer stage transition.

(f) Buffer Empty Interrupt Status Bit (BEMP)

This module sets this bit to 1 when at least one of the PIPEBEMP bits of BEMPSTS changes to 1 for the pipes that correspond to the PIPEBEMPE bits of BEMPENB that are set to 1 (when this module detects the BEMP interrupt state in at least one of the pipes for which the BEMP interrupt is enabled by software).

For the PIPEBEMP status set condition, refer to the description of the PIPEBEMP bits.

If software writes 0 to all PIPEBEMP bits corresponding to the pipes for which the BEMP interrupt is enabled through the PIPEBEMPE bits, this module clears the BEMP bit to 0.
This bit cannot be cleared to 0 by writing 0 by software.

(g) Buffer Not Ready Interrupt Status Bit (NRDY)

This module sets this bit to 1 when at least one of the PIPENRDY bits of BNRDYSTS changes to 1 for the pipes that correspond to the PIPENRDYE bits of NRDYENB that are set to 1 (when this module detects the NRDY interrupt state in at least one of the pipes for which the NRDY interrupt is enabled by software).

For the PIPENRDY status set condition, refer to the description of the PIPENRDY bits.

If software writes 0 to all PIPENRDY bits corresponding to the pipes for which the NRDY interrupt is enabled through the PIPENRDYE bits, this module clears the NRDY bit to 0.
This bit cannot be cleared to 0 by writing 0 by software.

(h) Buffer Ready Interrupt Status Bit (BRDY)

This module sets this bit to 1 when at least one of the PIPEBRDY bits of BRDYSTS changes to 1 for the pipes that correspond to the PIPEBRDYE bits of BRDYENB that are set to 1 (when this module detects the BRDY interrupt state in at least one of the pipes for which the BRDY interrupt is enabled by software).

For the PIPEBRDY status set condition, refer to the description of the PIPEBRDY bits.

If software writes 0 to all PIPEBRDY bits corresponding to the pipes for which the BRDY interrupt is enabled through the PIPEBRDYE bits, this module clears the BRDY bit to 0.
This bit cannot be cleared to 0 by writing 0 by software.

- **BRDY Interrupt Status Register (BRDYSTS)** **<Address: H'046>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PIPEBRDY									
?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0
?	?	?	?	?	?	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 10	No function assigned; the write value should always be 0.				
9 to 0	PIPEBRDY BRDY interrupt status for each pipe	These bits indicate the BRDY interrupt status in each pipe. 0: Interrupts not generated 1: Interrupts generated	R/W(0)	W	

<<Remarks>>

* Each bit number corresponds to a pipe number.

* To clear the status indicated by each bit of this register when BRDYM = 0, write 0 only to the bit to be cleared and write 1 to the other bits.

* When BRDYM = 0, clear each status bit of this register before accessing the corresponding FIFO.

BRDY Interrupt Status Bits for Each Pipe (PIPEBRDY)

Upon detecting a BRDY interrupt for a pipe, this module sets the corresponding PIPEBRDY bit in BRDYSTS to 1. At this time, if the corresponding bit in BRDYENB has been set to 1 by software, this module sets the BRDY bit in INTSTS0 to 1 and asserts an interrupt signal through the INT_N pin.

The set conditions and clear method of the BRDY interrupt depend on the settings of the BRDYM bit and BFRE bit for each pipe.

When BRDYM = 0 and BFRE=0:

With this setting, the BRDY interrupt shows that the FIFO port has become ready for access. This module generates an internal BRDY interrupt request trigger under any of the following conditions and sets the PIPEBRDY bit corresponding to the trigger-generated pipe to 1.

1. For a pipe set to transmission
   a. When the DIR bit is changed from 0 to 1 by software
   b. When this module completes packet transmission in the target pipe while the FIFO assigned to the pipe is not ready for write access by the CPU (the BSTS bit is read as 0)

      In the continuous transfer mode, a request trigger is generated when data of a single FIFO buffer has been transmitted.
   c. In the double-buffer structure, when one FIFO buffer is empty when writing to the other FIFO buffer is completed

      Even if transmission from a buffer is completed during writing to the other buffer, no request trigger is generated until the current writing is completed.
   d. In a pipe whose transfer type is set to isochronous transfer, when the buffer is flushed by hardware
   e. When the FIFO buffer is shifted from the write-not-ready state to write-ready state by writing 1 to the ACLRM bit

   For the DCP (that is, in the data transmission in the control transfer), no request trigger is generated.

2. For a pipe set to reception
   a. When packet reception is completed successfully and the FIFO buffer becomes ready to be read while the FIFO buffer assigned to the pipe is not ready for read access by the CPU (the BSTS bit is read as 0)

      No request trigger is generated for a transaction in which a data PID mismatch is detected.

      In the continuous transfer mode, no request is generated when the data size is MaxPacketSize and the buffer still has empty space.

      When a short packet is received, a request trigger is generated even if the FIFO buffer has empty space.

      When the transaction counter is used, a request trigger is generated when the specified number of packets have been received. In this case, a request trigger is generated even if the FIFO buffer has empty space.
   b. In the double-buffer structure, when one FIFO buffer is ready to be read when reading from the other FIFO buffer is completed

      Even if reception in a buffer is completed during reading from the other buffer, no request trigger is generated until the current reading is completed.

This interrupt does not occur in communication in the status stage of the control transfer.

The PIPEBRDY interrupt state for a pipe can be cleared to 0 by writing 0 to the PIPEBRDY bit corresponding to the pipe. At this time, write 1 to the bits corresponding to the other pipes.
Be sure to clear this interrupt state before accessing the corresponding FIFO buffer.

When BRDYM = 0 and BFRE = 1:

With this setting, when all data for a transfer has been read from a pipe, this module detects a BRDY interrupt and sets the corresponding PIPEBRDY bit to 1.

This module detects reception of the last data for a transfer under either of the following conditions.

1. When a short packet (including a zero-length packet) has been received
2. When the transaction counter (TRNCNT bits) is used and packets have been received for the number specified in the TRNCNT bits

When either of the above conditions is satisfied and reading of the data is completed, this module detects that all data for a transfer has been read.

If a zero-length packet is received while the FIFO buffer is empty, this module detects that all data for a transfer has been read when the zero-length packet data is toggled to the CPU side. In this case, before starting the next transfer, write 1 to the BCLR bit in FIFOCTR by software.

With this setting, this module does not detect BRDY interrupts in the transmit pipe.

The PIPEBRDY interrupt state for a pipe can be cleared to 0 by writing 0 to the PIPEBRDY bit corresponding to the pipe. At this time, write 1 to the bits corresponding to the other pipes.

When using this mode, do not change the BFRE bit setting until the processing for the transfer is completed. To change the BFRE bit setting during processing, clear all FIFO buffers for the corresponding pipe through the ACLRM bit setting.

When BRDYM = 1 and BFRE = 0:

With this setting, each bit value changes according to the BSTS bit status for each pipe; that is, this module sets the BRDY interrupt status to 1 or 0 according to the FIFO buffer state.

1. For a pipe set to transmission

   The PIPEBRDY bit is set to 1 when the FIFO port is ready to be written to and is cleared to 0 when the port becomes not ready to be written to.

   Note that the BRDY interrupt signal is not asserted when the transmit pipe for the DCP is ready to be written to.

2. For a pipe set to reception

   The PIPEBRDY bit is set to 1 when the FIFO port is ready to be read and is cleared to 0 when reading of all data is completed (the port becomes not ready to be read).

   If a zero-length packet is received while the FIFO buffer is empty, the corresponding PIPEBRDY bit indicates 1 and the BRDY interrupt signal continues to be asserted until BCLR = 1 is set by software.

With this setting, the PIPEBRDY bits cannot be cleared to 0 by software.

When setting BRDYM = 1, be sure to clear all BFRE bits (all pipes) to 0.
When setting BRDYM = 1, be sure to set the INTL bit to 1 (level-trigger).

- **NRDY Interrupt Status Register (NRDYSTS)** **<Address: H'048>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PIPENRDY									
?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0
?	?	?	?	?	?	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 10	No function assigned; the write value should always be 0.				
9 to 0	PIPENRDY NRDY interrupt status for each pipe	These bits indicate the NRDY interrupt status in each pipe. 0: Interrupts not generated 1: Interrupts generated	R/W(0)	W	

<<Remarks>>

* Each bit number corresponds to a pipe number.

* To clear the status indicated by each bit of this register, write 0 only to the bit to be cleared and write 1 to the other bits.

NRDY Interrupt Status Bits for Each Pipe (PIPENRDY)

Upon generating an NRDY interrupt for the pipe set to PID = BUF by software, this module sets the corresponding PIPENRDY bit in NRDYSTS to 1. At this time, if the corresponding bit in NRDYENB has been set to 1 by software, this module sets the NRDY bit in INTSTS0 to 1 and asserts an interrupt signal through the INT_N pin.

This module generates an internal NRDY interrupt request for a pipe under any of the following conditions.

No interrupt request is generated in the status data stage of the control transfer.

1. For a pipe set to transmission
   a. When an IN token is received while the FIFO buffer has no transmit data

      Upon receiving an IN token, this module generates an NRDY interrupt request and sets the PIPENRDY bit to 1.

      In a pipe whose transfer type is set to isochronous transfer, this module transmits a zero-length packet and sets the OVRN bit to 1.

2. For a pipe set to reception
   a. When an OUT token is received while the FIFO buffer has no empty space

      In a pipe whose transfer type is set to isochronous transfer, when an OUT token is received, this module generates an NRDY interrupt request, sets the PIPENRDY bit to 1, and sets the OVRN bit to 1.

      In a pipe whose transfer type is not the isochronous transfer, this module generates an NRDY interrupt request and sets the PIPENRDY bit to 1 upon transmitting the NAK Handshake after receiving the data subsequent to the OUT token.

      Note that no NRDY interrupt request is generated for retransmission (when a DATA-PID mismatch is detected). No NRDY interrupt request is generated when an error is found in the data packet.

   b. When a PING token is received while the FIFO buffer has no empty space

      Upon receiving a PING token, this module generates an NRDY interrupt request and sets the PIPENRDY bit to 1.

   c. In a pipe whose transfer type is set to isochronous transfer, when reception is not completed successfully in the interval frame

      This module generates an NRDY interrupt request and sets the PIPENRDY bit to 1 with the SOF reception timing.

- **BEMP Interrupt Status Register (BEMPSTS)** **<Address: H'04A>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PIPEBEMP									
?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0
?	?	?	?	?	?	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 10	No function assigned; the write value should always be 0.				
9 to 0	PIPEBEMP BEMP interrupt status for each pipe	These bits indicate the BEMP interrupt status in each pipe. 0: Interrupts not generated 1: Interrupts generated	R/W(0)	W	

<<Remarks>>

* Each bit number corresponds to a pipe number.

* To clear the status indicated by each bit of this register, write 0 only to the bit to be cleared and write 1 to the other bits.

BEMP Interrupt Status Bits for Each Pipe (PIPEBEMP)

Upon detecting a BEMP interrupt in the pipe set to PID = BUF by software, this module sets the corresponding PIPEBEMP bit in BEMPSTS to 1. At this time, if the corresponding bit in BEMPENB has been set to 1 by software, this module sets the BEMP bit in INTSTS0 to 1 and asserts an interrupt signal through the INT_N pin.

This module generates an internal BEMP interrupt request under any of the following conditions.

1. For a pipe set to transmission, when the FIFO buffer for the pipe is empty when transmission is completed (including transmission of a zero-length packet)

   In the single-buffer structure, this module generates an internal BEMP interrupt request for a pipe that is not DCP at the same time with the BRDY interrupt.

   Note that no internal BEMP interrupt is generated in the following cases.

   a. In the double-buffer structure, when software (DMAC) has started writing to the FIFO buffer on the CPU side before data transmission for one FIFO buffer is completed
   b. When the buffer is cleared (emptied) by writing 1 to the ACLRM or BCLR bit
   c. In the IN transfer (zero-length packet transmission) in the status stage of the control transfer

2. For a pipe set to reception

   When an amount of data larger than the MaxPacketSize setting is received successfully

   In this case, this module generates a BEMP interrupt request, sets the corresponding PIPEBEMP bit to 1, discards the received data, and sets the PID bits for the corresponding pipe to STALL (11).

   At this time, this module sends a STALL response.

   Note that no internal BEMP interrupt is generated in the following cases.

   a. When a CRC error or bit staffing error is found in the received data

   b. In SETUP transaction execution

Writing 0 to a PIPEBEMP bit clears the corresponding status.
Writing 1 to a PIPEBEMP bit is ignored.

(10) Frame Number Registers

- **Frame Number Register (FRMNUM)** **<Address: H'04C>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVRN	CRCE				FRNM										
0	0	?	?	?	0	0	0	0	0	0	0	0	0	0	0
-	-	?	?	?	-	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15	OVRN Overrun/underrun detection status	Indicates whether an overrun/underrun error was detected in the pipe set to isochronous transfer. 0: No error 1: An error occurred	R/W(0)	W	
14	CRCE CRC error detection status	Indicates whether a CRC error was detected in the pipe set to isochronous transfer. 0: No error 1: An error occurred	R/W(0)	W	
13 to 11	No function assigned; the write value should always be 0.				
10 to 0	FRNM Frame number	These bits indicate the latest frame number.	R	W	

<<Remarks>>

* The OVRN bit should be used for debugging. When designing a system, control the timing so that neither overrun nor underrun occurs.

(a) Overrun/Underrun Detection Status Bit (OVRN)

This bit is set to 1 when this module detects an overrun or an underrun in a pipe whose transfer type is set to isochronous transfer.
Upon detecting an overrun or an underrun, this module generates an internal NRDY interrupt request. For details, refer to NRDY Interrupt Status Registers (NRDYSTS), (9) Interrupt Status.

This bit can be cleared to 0 by writing 0 by software. At this time, write 1 to the other bits of this register.

This module sets this bit to 1 in either of the following two cases.

1. In a pipe whose transfer type is set to isochronous transmission, when an IN token is received before transmit data has been written to the FIFO buffer
2. In a pipe whose transfer type is set to isochronous reception, when an OUT token is received while at least one FIFO buffer is not empty

(b) CRC Error Detection Status Bit (CRCE)

This bit is set to 1 when this module detects a CRC error or a bit staffing error in a pipe whose transfer type is set to isochronous transfer.

This bit can be cleared to 0 by writing 0 by software. At this time, write 1 to the other bits of this register.

When a CRC error is detected, this module does not generate an internal NRDY interrupt request.

(c) Frame Number Bits (FRNM)

This module updates these bits to the latest frame number with the SOF output timing at 1-ms intervals or when an SOF is received.

When reading these bits by software, repeat reading until the same value is read at least twice.

- **µFrame Number Register (UFRMNUM)** **<Address: H'04E>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													UFRNM		
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 3	No function assigned; the write value should always be 0.				
2 to 0	UFRNM µframe number	These bits indicate the µframe number.	R	W	

<<Remarks>>

None

µFrame Number Bits (UFRMNUM)

In high-speed communication, these bits indicate the µframe number. When communication is not in the high-speed mode, these bits are set to 0x00.

When reading these bits by software, repeat reading until the same value is read at least twice.

(11) USB Address

- **USB Address Register (USBADDR)** **<Address: H'050>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									USBADDR						
?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Remarks
15 to 7	No function assigned; the write value should always be 0.				
6 to 0	USBADDR USB address	These bits indicate the USB address assigned by the host.	R	R/W	

(a) USB Address Bits (USBADDR)

Upon completing the SetAddress request processing successfully, this module indicates the received USB address in these bits.

This module indicates 0x00 in these bits upon detecting a USB reset.

(12) USB Request Registers

The USB request registers store setup requests for control transfers; that is, the values of the USB requests that have been received.

- **USB Request Type Register (USBREQ)** **<Address: H'054>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bRequest								bmRequestType							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Remarks
15 to 8	bRequest Request	Value of USB request bRequest	For P: R For H: R/W	For P: W For H: R	
7 to 0	bmRequestType Request type	Value of USB request bmRequestType	For P: R For H: R/W	For P: W For H: R	

<<Remarks>>

None

(a) USB Request Bits (bRequest)

These bits indicate the value of the USB request data that this module has received in the SETUP transaction. Writing to these bits by software is ignored.

(b) USB Request Type Bits (bRmRequestType)

These bits indicate the value of the USB request data that this module has received in the SETUP transaction. Writing to these bits by software is ignored.

- **USB Request Value Register (USBVAL)** **<Address: H'056>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
wValue															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Remarks
15 to 0	wValue Value	Value of USB request wValue	For P: R For H: R/W	For P: W For H: R	

<<Remarks>>

None

Value Bits (wValue)

These bits are used to write or read the value of USB request wValue. Bits 7 to 0 configure a lower byte.

These bits indicate the value of USB request wValue that this module has received in the SETUP transaction. Writing to these bits by software is ignored.

- **USB Request Index Register (USBINDX)** **<Address: H'058>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
wIndex															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Remarks
15 to 0	wIndex Index	Value of USB request wIndex	For P: R For H: R/W	For P: W For H: R	

<<Remarks>>

None

Index Bits (wIndex)

These bits are used to write or read the value of USB request wIndex. Bits 7 to 0 configure a lower byte.

These bits indicate the value of USB request wIndex that this module has received in the SETUP transaction. Writing to these bits by software is ignored.

- **USB Request Length Register (USBLENG)** **<Address: H'05A>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
wLength															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Remarks
15 to 0	wLength Length	Value of USB request wLength	For P: R For H: R/W	For P: W For H: R	

<<Remarks>>

None

Length Bits (wLength)

These bits are used to write or read the value of USB request wLength. Bits 7 to 0 configure a lower byte.

These bits indicate the value of USB request wLength that this module has received in the SETUP transaction. Writing to these bits by software is ignored.

(13) DCP Configuration Registers

- **DCP Maximum Packet Size Register (DCPMAXP)** **<Address: H'05E>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									MXPS						
?	?	?	?	?	?	?	?	?	1	0	0	0	0	0	0
?	?	?	?	?	?	?	?	?	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 7	No function assigned; the write value should always be 0.				
6 to 0	MXPS Maximum packet size	These bits specify the maximum data payload (maximum packet size) for the DCP.	R/W	R	

(a) Maximum Packet Size Bits (MXPS)

The maximum data payload (maximum packet size) for the DCP should be set in these bits. The initial value is 0x40 (64 bytes).

A value conforming to the USB specification must be set in the MXPS bits. The MXPS bits should be set only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When changing these bits after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing these bits. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

When MXPS = 0, do not write to the FIFO buffer memory or do not make the setting of PID = BUF.

- **DCP Control Register (DCPCTR)** **<Address: H'060>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSTS							SQCLR	SQSET	SQMON	PBUSY			CCPL	PID	
0	?	?	?	?	?	?	0	0	1	0	?	?	0	0	0
-	?	?	?	?	?	?	-	-	-	-	?	?	0	0	0

Bit	Name	Function	S/W	H/W	Remarks
15	BSTS Buffer status	Whether access to the DCP's FIFO buffer is enabled or disabled can be read. 0: Buffer access is disabled 1: Buffer access is enabled	R	W	
14 to 9	No function assigned; the write value should always be 0.				
8	SQCLR Sequence toggle bit clear	Enables the expectation of the sequence toggle bit in the next transaction in DCP transfer to be set in DATA0. 0: Invalid 1: Specifies DATA0	R (0)/ W (1)	R	
7	SQSET Sequence toggle bit set	Enables the expectation of the sequence toggle bit in the next transaction in DCP transfer to be set in DATA1. 0: Invalid 1: Specifies DATA1	R (0)/ W (1)	R	
6	SQMON Sequence toggle bit monitor	The expectation of the sequence toggle bit in the next transaction in DCP transfer can be read. 0: DATA0 1: DATA1	R	W	
5	SPBUSY Pipe busy	Whether the DCP communication state has actually transited to NAK when the PID bit setting in DCPCTR was changed from BUF to NAK can be read. 0: Has not finished the transition to NAK 1: Has finished the transition to NAK	R	W	
4, 3	No function assigned; the write value should always be 0.				

Bit	Name	Function	S/W	H/W	Remarks
2	CCPL Control transfer end enable	Setting this bit to 1 when the function controller function is selected enables the control transfer status stage to end. 0: Invalid 1: The control transfer is ended	R (0)/ W (1)	R/W (0)	
1, 0	PID Response PID	These bits specify the response type of this module in control transfer. 00: NAK response 01: BUF response (depending on the buffer state) 10: STALL response 11: STALL response	R/W	R/W	

<<Remarks>>

None

Buffer Status Bit (BSTS)

Whether the FIFO buffer allocated for the DCP can be accessed by the CPU can be read from this bit. The meaning of this bit differs depending on the ISEL bit setting, as shown below.

(1) When ISEL = 0: Indicates whether reading of the receive data is possible

(2) When ISEL = 1: Indicates whether writing of the transmit data is possible

(b) Bit for Sequence Toggle Bit Clear (SQCLR)

When this bit is set to 1 by software, this module sets the expectation of the sequence toggle bit for the corresponding pipe to DATA0. This bit is always read as 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

This bit should be set to 1 only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When setting this bit to 1 after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(c) Bit for Sequence Toggle Bit Set (SQSET)

When this bit is set to 1 by software, this module sets the expectation of the sequence toggle bit for the corresponding pipe to DATA1. This bit is always read as 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

This bit should be set to 1 only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When setting this bit to 1 after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(d) Bit for Sequence Toggle Bit Monitor (SQMON)

The expectation of the sequence toggle bit for the corresponding pipe can be read from this bit.

When the transaction has been processed normally, this module toggles this bit. Note however that this bit is not toggled when a DATA-PID mismatch occurs during transfer in the receiving direction.

When the SETUP packet is received normally, this module sets this bit to 1 (the expectation is set in DATA1). This module does not reference this bit during IN/OUT transactions of the status stage. Also, this bit is not toggled when processing ends normally.

(e) Pipe Busy Bit (PBUSY)

This module changes this bit from 0 to 1 when the USB transaction of the corresponding pipe has started. This bit is changed from 1 to 0 upon the completion of a single transaction.

Reading this bit after PID = NAK has been set by software is performed to confirm whether the pipe settings can be changed.

(f) Control Transfer End Enable Bit (CCPL)

When this bit is set to 1 by software when the corresponding PID bit setting is BUF, this module completes the control transfer stage. In other words, this module transmits an ACK handshake in response to an OUT transaction from the USB host in control read transfers, and transmits a zero-length packet in response to an IN transaction from the USB host in control write transfers and no-data control transfers. However, when a SET_ADDRESS request is detected, this module performs auto response from the SETUP stage up to completion of the status stage, regardless of this bit setting.

This module changes this bit from 1 to 0 on receiving a new SETUP packet.

When VALID = 1, 1 cannot be written to this bit by software.

(g) Response PID Bits (PID)

Change the setting of these bits from NAK to BUF by software while executing the data stage or status stage of control transfers.

This module changes the value of these bits in the following cases:

- When this module receives a SETUP packet, this module changes these bits to NAK (00). At this time, VALID = 1 can be read from these bits, and these bits cannot be changed by software until VALID = 0 has been set by software.
- If this module receives data whose size exceeds MaxPacketSize while BUF is being set to these bits by software, PID = STALL (11) can be read from these bits.
- When a control transfer sequence error has been detected by this module, PID = STALL (1x) can be read from these bits.
- When a USB bus reset has been detected by this module, PID = NAK can be read from these bits.

This module does not reference these bits during the SET_ADDRESS request processing (automatic processing).

(14) Pipe Configuration Registers

The settings for PIPE1 to PIPE9 should be made through register settings of PIPESEL, PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI, PIPExCTR, PIPExTRE, and PIPExTRN.

After the pipe to be used has been selected by PIPESEL, the functions of the pipe should be set using PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI. Note that PIPExCTR, PIPExTRE, and PIPExTRN can be set regardless of the pipe selection by PIPESEL.

- **Pipe Window Select Register (PIPESEL)** **<Address: H'064>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												PIPESEL			
?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 4	No function assigned; the write value should always be 0.				
3 to 0	PIPESEL Pipe window select	These bits specify a pipe for registers at addresses H'68 to H'6E. 0000: Not selected 0001: PIPE1 0010: PIPE2 0011: PIPE3 0100: PIPE4 0101: PIPE5 0110: PIPE6 0111: PIPE7 1000: PIPE8 1001: PIPE9	R/W	R	

<<Remarks>>

* When PIPESEL = B'0000 is set, 0 is read from all bits in the above related registers. Also, when PIPESEL = B'0000 is set, writing to registers located at addresses H'68 to H'6E is invalid.

(a) Pipe Window Select Bits (PIPESEL)

When a value between B'0001 and B'1001 is set in these bits by software, the information and settings for the corresponding pipe can be read from the registers at addresses H'68 to H'6C. After

the pipe is specified by these bits, the values that are set to addresses H'68 to H'6C by software will be reflected in the transfer mode of the corresponding pipe.

When these bits are set to B'0000 by software, all of the bits in registers at addresses H'68 to H'6C are read as 0. At this time, writing to addresses H'68 to H'6C by software is invalid.

- **Pipe Configuration Register (PIPECFG)** **<Address: H'068>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE					BFRE	DBLB	CNTMD	SHTNAK			DIR	EPNUM			
0	0	?	?	?	0	0	0	0	?	?	0	0	0	0	0
-	-	?	?	?	-	-	-	-	?	?	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15, 14	TYPE Transfer type	These bits specify the transfer type for the corresponding pipe specified by the PIPESEL bits. 00: Pipe use disabled 01: Bulk transfer 10: Interrupt transfer 11: Isochronous transfer	R/W	R	
13 to 11	No function assigned; the write value should always be 0.				
10	BFRE BRDY interrupt operation specification	Specifies the timing for this module to notify generation of a BRDY interrupt for the corresponding pipe. 0: BRDY interrupt notification upon transmitting or receiving data 1: BRDY interrupt notification upon completion of reading data	R/W	R	
9	DBLB Double buffer mode	Specifies a single buffer or double buffer as the FIFO buffer used by the corresponding pipe. 0: Single buffer 1: Double buffer	R/W	R	
8	CNTMD Continuous transfer mode	Specifies whether continuous transfer mode is selected for communication with the corresponding pipe. 0: Non-continuous transfer mode 1: Continuous transfer mode	R/W	R	
7	SHTNAK Pipe disabled at end of transfer	Specifies whether to change the PID bit setting to NAK at the end of transfer when the corresponding pipe is set to the receiving direction. 0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer	R/W	R	

Bit	Name	Function	S/W	H/W	Remarks
6, 5	No function assigned; the write value should always be 0.				
4	DIR Transfer direction	Specifies the transfer direction for the corresponding pipe. 0: Receiving 1: Transmitting	R/W	R	
3 to 0	EPNUM Endpoint number	These bits specify the endpoint number for the corresponding pipe.	R/W	R	

<<Remarks>>

None

Transfer Type Bits (TYPE)

The USB transfer type for the pipe set in the PIPESEL bits (selected pipe) should be set in these bits. The selected pipe and the transfer types that can be set by these bits are shown in table 14.16.

Table 14.16 Selected Pipe and Transfer Types Settable by TYPE Bits

Selected Pipe	TYPE Bits	USB Transfer Type
PIPE1 or PIPE2	B'01 or B'11	Bulk transfer or isochronous transfer
PIPE3 to PIPE5	B'01	Bulk transfer
PIPE6 to PIPE9	B'10	Interrupt transfer

Before setting the PID bits to BUF (this will start USB communication using the selected pipe), set the TYPE bits to a value other than B'00.

These bits should be changed only when the PID bits of the selected pipe are set to NAK. When changing these bits after the PID bit setting of the selected pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing these bits. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(b) BRDY Interrupt Operation Specification Bit (BFRE)

This bit is valid when the selected pipe is between PIPE1 and PIPE5.

If this bit is set to 1 by software and the selected pipe is used in the receiving direction (this means that the DIR bit is 0), this module detects the end of transfer and issues a BRDY interrupt when reading of that packet has completed. When a BRDY interrupt is generated under these conditions,

1 must be written to the BCLR bit by software. The FIFO buffer allocated for the selected pipe cannot receive data until 1 is written to the BCLR bit.

If this bit is set to 1 by software and the selected pipe is used in the transmitting direction (this means that the DIR bit is 1), this module does not generate BRDY interrupts. For details, refer to the PIPEBRDY interrupt status.

This bit should be changed only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When changing this bit setting after USB communication using the selected pipe has been performed, in addition to confirming the states of the above three bits, write 1 to the ACLRM bit and then write 0 to it in succession by software to clear the FIFO buffer allocated for the selected pipe.

When changing this bit after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(c) Double Buffer Mode Bit (DBLB)

This bit is valid when the selected pipe is between PIPE1 and PIPE5.

If this bit is set to 1 by software, this module allocates two buffers with the FIFO buffer size specified by the BUFSIZE bits in PIPEBUF for the selected pipe. This means that the FIFO buffer capacity allocated by this module for the selected pipe is as follows:

(BUFSIZE + 1)*64*(DBLB + 1) [byte]

If this bit is set to 1 by software and the selected pipe is used in the transmitting direction (this means that the DIR bit is 1), this module does not generate BRDY interrupts. For details, refer to the PIPEBRDY interrupt status.

This bit should be changed only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When changing this bit setting after USB communication using the selected pipe has been performed, in addition to confirming the states of the above three bits, write 1 to the ACLRM bit and then write 0 to it in succession by software to clear the FIFO buffer assigned to the selected pipe.

When changing this bit after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(d) Continuous Transfer Mode Bit (CNTMD)

This bit is valid when the selected pipe is between PIPE1 and PIPE5, and bulk transfer is set as the transfer type for the selected pipe.

This module judges whether transmitting to/receiving from the FIFO buffer allocated for the selected pipe has completed or not using this bit setting, as shown in table 14.17.

Table 14.17 Relationship between CNTMD Bit Setting and Method for Determining Completion of FIFO Buffer Transmission/Reception

CNTMD Bit Setting	Method for Determining Reading Enabled State and Transmission Enabled State
0	[Condition for enabling FIFO buffer reading when receiving direction is set (DIR = 0)] When this module has received a single packet
	[Condition for enabling FIFO buffer transmission when transmitting direction is set (DIR = 1)] When either (1) or (2) is satisfied (1) When software (or the DMAC) has written the maximum packet size of data to the FIFO buffer (2) When software (or the DMAC) has written a short packet size of data (including the case of zero bytes) to the FIFO buffer, and has also written 1 to the BVAL bit
1	[Condition for enabling FIFO buffer reading when receiving direction is set (DIR = 0)] (1) When the number of data bytes received in the FIFO buffer allocated for the selected pipe has become equal to the number of allocated bytes ((BUFSIZE+1)*64) (2) When this module has received a short packet other than a zero-length packet (3) When this module has received a zero-length packet while the FIFO buffer allocated for the selected pipe already contains data (4) When the number of received packets becomes equal to the transaction counter number set for the selected pipe by software
	[Condition for enabling FIFO buffer transmission when transmitting direction is set (DIR = 1)] When (1), (2), or (3) is satisfied (1) When the number of data bytes written by software (or the DMAC) has become equal to one buffer of the FIFO buffer size allocated for the selected pipe (2) When software (or the DMAC) has written data bytes (including the case of zero bytes) that are less than one buffer of the FIFO buffer size allocated for the selected pipe to the FIFO buffer, and has also written 1 to the BVAL bit (3) When software (or the DMAC) has written data bytes (including the case of zero bytes) that are less than one buffer of the FIFO buffer size allocated for the selected pipe to the FIFO buffer, and has also asserted the DENDx_N signal simultaneously with the last writing

This bit should be changed only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When changing this bit setting after USB communication using the selected pipe has been performed, in addition to confirming the states of the above three bits, write 1 to the ACLRM bit and then write 0 to it in succession by software to clear the FIFO buffer assigned to the selected pipe.

When changing this bit after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(e) Bit for Pipe Disabled at End of DCP Transfer (SHTNAK)

This bit is valid when the selected pipe is between PIPE1 and PIPE5, and the receiving direction is selected.

When this bit is set to 1 by software for a pipe set to the receiving direction, this module changes the PID bits corresponding to the selected pipe to NAK when the end of transfer has been determined for the selected pipe. This module recognizes that transfer has ended when either one of the following conditions 1 or 2 is satisfied.

1. When a short packet size of data (including a zero-length packet) is received normally
2. When the transaction counter is used, and packets for the number of transactions set in the transaction counter are received normally

This bit should be changed only when CSSTS = 0 and PID = NAK. When changing this bit after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

This bit should be cleared to 0 for pipes set to the transmitting direction.

(f) Transfer Direction Bit (DIR)

When this bit is cleared to 0 by software, this module uses the selected pipe in the receiving direction. When this bit is set to 1, this module uses the selected pipe in the transmitting direction.

This bit should be changed only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When changing this bit setting after USB communication using the selected pipe has been performed, in addition to confirming the states of the above three bits, write 1 to the ACLRM bit and then write 0 to it in succession by software to clear the FIFO buffer assigned to the selected pipe.

When changing this bit after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(g) Endpoint Number Bits (EPNUM)

The endpoint number corresponding to the selected pipe should be set in these bits by software. A setting of 0000 means that the pipe is unused.

These bits should be changed only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When changing these bits after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing these bits. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

The endpoint number should be set so that the combination of the DIR bit and EPNUM bit settings does not overlap with settings of other pipes. (The setting of EPNUM = 0000 (selected pipe is not used) can be set for more than one pipe.)

- **Pipe Buffer Setting Register (PIPEBUF)** **<Address: H'06A>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BUFSIZE							BUFNMB							
?	0	0	0	0	0	?	?	0	0	0	0	0	0	0	0
?	-	-	-	-	-	?	?	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15	No function assigned; the write value should always be 0.				
14 to 10	BUFSIZE Buffer size	These bits specify the FIFO buffer size for the pipe selected with the PIPESEL bits (corresponding pipe). H'00: 64 bytes H'01: 128 bytes : H'1F: 2 Kbytes	R/W	R	
9, 8	No function assigned; the write value should always be 0.				
7 to 0	BUFNMB Buffer number	These bits specify the FIFO buffer number for the corresponding pipe (from H'4 to H'87).	R/W	R	

<<Remarks>>

* The bits in this register should be set only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits by software.

* When changing the bits in this register after the PID bit setting of the selected pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing the bits. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

Buffer Size Bits (BUFSIZE)

The FIFO buffer size to be allocated for the corresponding pipe should be set in these bits. The buffer memory is allocated in blocks, and 64 bytes comprise one block.

When the DBLB bit is set to 1 by software, this module allocates two buffers with the FIFO buffer size specified by these bits for the selected pipe. The FIFO buffer capacity allocated by this module for the selected pipe is as follows:

(BUFSIZE + 1)*64*(DBLB + 1) [byte]

The value that can be set in these bits differs according to the pipe, as shown below.

- When PIPE1 to PIPE5 is the selected pipe: Any value from H'0 to H'1F can be set.
- When PIPE6 to PIPE9 is the selected pipe: Only H'0 can be set.

When the CNTMD bit is set to 1, an integer multiple of MaxPacketSize should be set in the BUFSIZE bits.

(h) Buffer Number Bits (BUFNMB)

The first block number in the FIFO buffer to be allocated for the corresponding pipe should be set in these bits.

The FIFO buffer blocks allocated for the selected pipe by this module are determined as follows:

Block number: BUFNMB to block number of (BUFNMB + (BUFSIZE + 1)*(DBLB + 1) – 1)

These bits should be set so that the memory size range is not exceeded (0 [H'00] to 8640 [H'87] for 8.5 Kbytes). The following conditions also need to be satisfied.

- H'00 is for DCP only.
- H'04 is for PIPE6 only. However, if PIPE6 is not used, H'04 can be used for other pipes. When the selected pipe is PIPE6, writing to these bits is invalid; this module automatically assigns BUFNMB = H'04 for PIPE6.
- H'05 is for PIPE7 only. However, if PIPE7 is not used, H'05 can be used for other pipes. When the selected pipe is PIPE7, writing to these bits is invalid; this module automatically assigns BUFNMB = H'05 for PIPE7.
- H'06 is for PIPE8 only. However, if PIPE8 is not used, H'06 can be used for other pipes. When the selected pipe is PIPE8, writing to these bits is invalid; this module automatically assigns BUFNMB = H'06 for PIPE8.
- H'07 is for PIPE9 only. However, if PIPE9 is not used, H'07 can be used for other pipes. When the selected pipe is PIPE9, writing to these bits is invalid; this module automatically assigns BUFNMB = H'07 for PIPE9.

- **Pipe Maximum Packet Size Register (PIPEMAXP)** **<Address: H'06C>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					MXPS										
?	?	?	?	?	0	0	0	0	0(1)	0	0	0	0	0	0
?	?	?	?	?	-	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 11	No function assigned; the write value should always be 0.				
10 to 0	MXPS Maximum packet size	These bits specify the maximum data payload (maximum packet size) for the corresponding pipe. H'1 to H'40 bytes can be set for PIPE6 to PIPE9.	R/W	R	

<<Remarks>>

* The initial value of the MXPS bits is H'00 when no pipe is selected with the PIPESEL bits in PIPESEL and H'40 when a pipe is selected with the PIPESEL bits in PIPESEL.

Device Select Bits (DEVSEL)

When the host controller function is selected, the USB device address of the communicating party should be set in these bits.

These bits should be set after setting the DEVADDx register corresponding to the setting of these bits. For example, if DEVSEL = 0010 is to be set, first set DEVADD2 at address H'D4.

These bits should be changed only when CSSTS = 0 and PID = NAK. When changing these bits after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing these bits. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

When the function controller function is selected, these bits should be set to 0000.

(i) Maximum Packet Size Bits (MXPS)

The maximum data payload (maximum packet size) for the selected pipe should be set in these bits. A value between 1 byte (H'1) and 1024 bytes (H'400) can be set for PIPE1 and PIPE2. A value of 8 bytes (H'8), 16 bytes (H'10), 32 bytes (H'20), 64 bytes (H'40), and 512 bytes (H'200)

can be set for PIPE3 to PIPE5 (bits MXPS[2:0] are not available). A value between 1 byte (H'1) and 64 bytes (H'40) can be set for PIPE6 to PIPE9.

The initial value of these bits is H'40 (64 bytes).

The MXPS bits should be set to a value defined by the USB specification for each transfer type. When a pipe for which isochronous transfer has been specified is used in split-transaction communication, a value of 188 bytes or less must be set in the MXPS bits.

The MXPS bits should be changed only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When changing these bits after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing these bits. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

When MXPS = 0 is set, the FIFO buffer must not be written to and the PID bits must not be set to BUF.

- **Pipe Cycle Control Register (PIPEPERI)** **<Address: H'06E>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			IFIS										IITV		
?	?	?	0	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	-	?	?	?	?	?	?	?	?	?	-	-	-

Bit	Name	Function	S/W	H/W	Remarks
15 to 13	No function assigned; the write value should always be 0.				
12	IFIS Isochronous IN buffer flush	Enables or disables buffer flush when isochronous-IN transfer is selected for the pipe set in the PIPESEL bits (corresponding pipe). 0: Buffer is not flushed 1: Buffer is flushed	R/W	R	
11 to 3	No function assigned; the write value should always be 0.				
2 to 0	IITV Interval error detection interval	These bits specify the transfer interval timing for the corresponding pipe in terms of the frame timing divided by an n-th power of 2.	R/W	R	

<<Remarks>>

None

Interval Error Detection Interval Bits (IITV)

The interval error detection interval should be set in these bits in terms of the frame timing divided by an n-th power of 2.

These bits should be changed only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When changing these bits after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing these bits. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

In the case of setting these bits and performing USB communication, and then changing these bits to a different value, first set the PID bits to NAK, set the ACLRM bit to 1, and initialize the interval timer.

These bits are not available for PIPE3 to PIPE5. 0 should be written to the bit locations that correspond to PIPE3 to PIPE5.

The IITV bits can be set when isochronous transfer is set as the transfer type for the selected pipe.

- When isochronous-OUT transfer is set for the selected pipe

  When no data packet is received during the (μ) frame for each interval set using the IITV bits, this module generates an NRDY interrupt.

  An NRDY interrupt is also generated when this module fails to receive data due to an error (CRC error, etc.) occurring in the data packet or the FIFO buffer being full (because software (or the DMAC) was slow in reading data from the FIFO buffer).

  The timing for generating an NRDY interrupt is when an SOF packet is received. Even when an SOF packet is corrupted, the internal interpolation function enables an NRDY interrupt to be generated at the timing to receive the SOF packet.

  However, when the IITV bits are set to a value other than 0, an NRDY interrupt is generated when an SOF packet is received at each interval after the interval counter has started.

  When the PID bits are set to NAK by software after the interval counter has started, this module does not generate an NRDY interrupt even though an SOF packet is received.

  The conditions for starting the interval counter differ according to the IITV bit setting.

  (a) When IITV is 0

  The interval counter starts from the (μ) frame following the (μ) frame in which the PID bits of the selected pipe were changed to BUF.

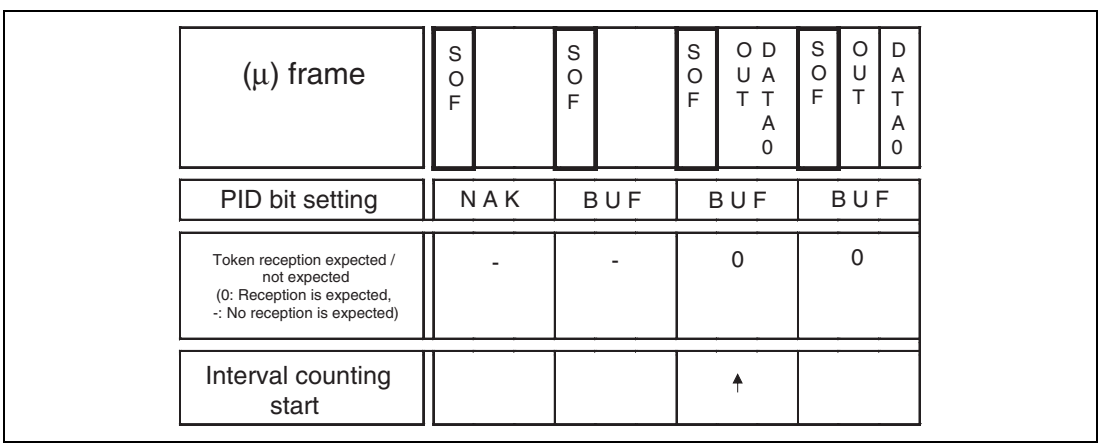


Figure 14.3 Relationship between (μ) Frame and Token Reception Expectation when IITV = 0

(b) When IITV is not 0

The interval counter starts from the completion of the first data packet normal reception following the (μ) frame in which the PID bits of the selected pipe were changed to BUF.

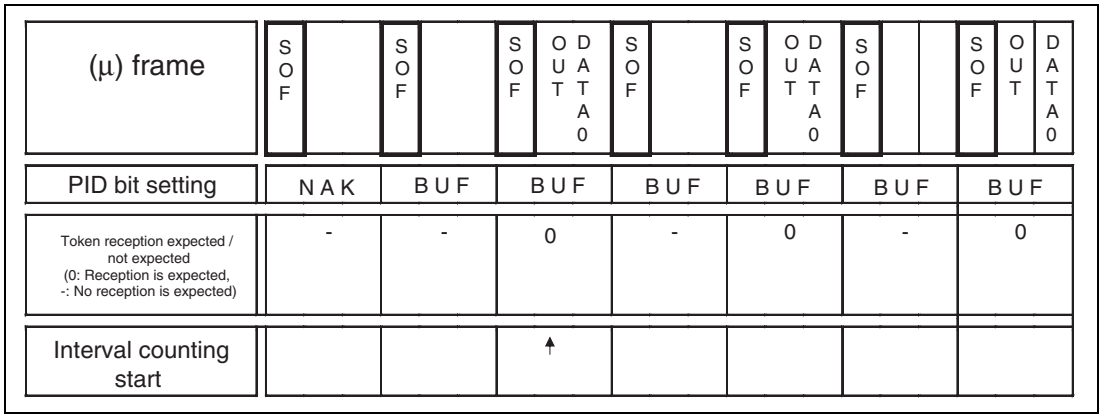


Figure 14.4 Relationship between (μ) Frame and Token Reception Expectation when IITV ≠ 0

- When isochronous-IN transfer is set for the selected pipe

  This is used in combination with the IFIS bit being set to 1. When IFIS = 0, a data packet is sent in response to the received token regardless of the IITV bit setting.

  When IFIS = 1 is set, this module clears the FIFO buffer if an IN-token is not received during the (μ) frame for each interval set using the IITV bits when there is data to be transmitted in the FIFO buffer.

  The FIFO buffer is also cleared when this module could not receive an IN-token normally because a bus error (CRC error, etc.) has occurred in the IN-token.

  The timing for clearing the FIFO buffer is when an SOF packet is received. Even when an SOF packet is corrupted, the internal interpolation function enables the FIFO buffer to be cleared at the timing to receive the SOF packet.

  The conditions for starting the interval counter differ according to the IITV bit setting (similar to isochronous-OUT transfer).

  The condition for interval counting is one of the following condition 1, 2, or 3:

  1. When a hardware reset is performed to this module (the IITV bit setting is also cleared to 0 in this case)
  2. When ACLRM = 1 is set by software
  3. When this module has detected a USB reset

(15) Pipe Control Registers

- **PIPE1 Control Register (PIPE1CTR)** **<Address: H'070>**
- **PIPE2 Control Register (PIPE2CTR)** **<Address: H'072>**
- **PIPE3 Control Register (PIPE3CTR)** **<Address: H'074>**
- **PIPE4 Control Register (PIPE4CTR)** **<Address: H'076>**
- **PIPE5 Control Register (PIPE5CTR)** **<Address: H'078>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSTS	INBUFM				ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBSY				PID	
0	0		?	?	0	0	0	0	0	0	?	?	?	0	0
-	-	?	?	?	-	-	-	-	-	-	?	?	?	0	0

Bit	Name	Function	S/W	H/W	Remarks
15	BSTS Buffer status	The FIFO buffer status of the corresponding pipe can be read. 0: Buffer access is disabled 1: Buffer access is enabled	R	W	
14	INBUFM Transmission buffer monitor	The FIFO buffer status of the corresponding pipe when the corresponding pipe is set to the transmitting direction can be read. 0: There is no data to be transmitted in the FIFO buffer 1: There is data to be transmitted in the FIFO buffer	R	W	
13 to 11	No function assigned; the write value should always be 0.				
10	ATREPM Auto response mode	Enables or disables auto response of the corresponding pipe. 0: Auto response is disabled 1: Auto response is enabled (Regardless of the FIFO buffer status of the corresponding pipe, a zero-length packet is always sent at transmission, and a NAK response is returned and an NRDY interrupt generated at reception.)	R/W	R	

Bit	Name	Function	S/W	H/W	Remarks
9	ACLRM Auto buffer clear mode	Enables or disables auto buffer clear mode of the corresponding pipe. 0: Auto buffer clear mode is disabled 1: Auto buffer clear mode is enabled (all buffers are initialized)	R/W	R	
8	SQCLR Sequence toggle bit clear	Set this bit to 1 when the expectation of the sequence toggle bit in the next transaction of the corresponding pipe is to be set in DATA0. 0: Invalid 1: Specifies DATA0	R (0)/ W (1)	R	
7	SQSET Sequence toggle bit set	Set this bit to 1 when the expectation of the sequence toggle bit in the next transaction of the corresponding pipe is to be set in DATA1. 0: Invalid 1: Specifies DATA1	R (0)/ W (1)	R	
6	SQMON Sequence toggle bit monitor	The expectation of the sequence toggle bit in the next transaction of the corresponding pipe can be read. 0: DATA0 1: DATA1	R	W	
5	PBUSY Pipe busy	Whether the corresponding pipe is currently being used by the USB bus can be read. 0: Corresponding pipe is not being used by the USB bus 1: Corresponding pipe is being used by the USB bus	R	W	
4 to 2	No function assigned; the write value should always be 0.				
1, 0	PID Response PID	These bits specify the response type for the next transaction of the corresponding pipe. 00: NAK response 01: BUF response (depending on the buffer state) 10: STALL response 11: STALL response	R/W	R/W	

<<Remarks>>

None

(a) Buffer Status Bit (BSTS)

Whether the FIFO buffer allocated for the corresponding pipe can be accessed by the CPU can be read from this bit. The meaning of this bit differs depending on the DIR, BFRE, and DCLRM bit settings.

Table 14.18 BSTS Bit Operation

DIR Bit Setting	BFRE Bit Setting	DCLRM Bit Setting	Meaning of BSTS Bit
0	0	0	This bit is read as 1 when reading the receive data from the FIFO buffer is possible. This bit is read as 0 when reading of data has completed.
		1	This setting is prohibited.
	1	0	This bit is read as 1 when reading the receive data from the FIFO buffer is possible. This bit is read as 0 when 1 is written to the BCLR bit by software after reading of data has completed.
		1	This bit is read as 1 when reading the receive data from the FIFO buffer is possible. This bit is read as 0 when reading of data has completed.
1	0	0	This bit is read as 1 when writing the transmit data to the FIFO buffer is possible. This bit is read as 0 when writing of data has completed.
		1	This setting is prohibited.
	1	0	This setting is prohibited.
		1	This setting is prohibited.

(b) IN Buffer Monitor Bit (INBUFM)

If the corresponding pipe is set to the transmitting direction (DIR = 1), this bit is read as 1 when software (or the DMAC) has finished writing data to the FIFO buffer for at least one buffer.

This bit is read as 0 by this module when this module has finished transmitting the data in the single FIFO buffer to which writing has been completed. When a double buffer is used (DBLB = 1), this bit is read as 0 when this module has finished transmitting data for two buffers and also software (or the DMAC) has not yet finished writing data for one buffer.

When the corresponding pipe is set to the receiving direction (DIR = 0), this bit has the same value as the BSTS bit.

(c) Auto Response Mode Bit (ATREPM)

When bulk transfer is set as the transfer type for the corresponding pipe, this bit can be set to 1. When this bit is set to 1, this module responds to the token from the USB host as shown below.

- When bulk-IN transfer (TYPE = 01 and DIR = 1) is set for the corresponding pipe
  When both ATREPM = 1 and PID = BUF are set, this module transmits a zero-length packet in response to an IN token.
  Each time this module receives an ACK response from the USB host (a single transaction consists of IN token reception → zero-length packet transmission → ACK response reception), this module updates (toggles) the sequence toggle bit (DATA-PID). BRDY and BEMP interrupts are not generated.
- When bulk-OUT transfer (TYPE = 01 and DIR = 0) is set for the corresponding pipe
  When both ATREPM = 1 and PID = BUF are set, this module sends a NAK response in response to an OUT token (or PING token) and generates an NRDY interrupt.

This bit should be changed only when CSSTS = 0 and PID = NAK. When changing this bit after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

When performing USB communication with this bit set to 1, this bit should be set to 1 only when the FIFO buffer is empty. The FIFO buffer must not be written to during USB communication with this bit set to 1.

When isochronous transfer is set as the transfer type for the corresponding pipe, this bit must be cleared to 0.

(d) Auto Buffer Clear Mode Bit (ACLRM)

To delete all contents of the FIFO buffer allocated for the corresponding pipe, write 1 to the ACLRM bit and then write 0 to it in succession.

Table 14.19 shows the contents cleared by this module when the ACLRM bit is set to 1 and then to 0, and the cases in which the contents need to be cleared.

Table 14.19 Contents Cleared by This Module when ACLRM = 1 is Set

No.	Contents Cleared by ACLRM Bit Operation	Case Requiring Clearing of Contents
1	All contents in the FIFO buffer allocated for the corresponding pipe (both buffers when a double buffer is selected)	
2	Interval counter value when isochronous transfer is set as the transfer type for the corresponding pipe	When the interval counter value needs to be reset
3	Internal flag relating to BFRE bit	When the BFRE bit setting is changed
4	FIFO buffer toggle control	When the DBLB bit setting is changed
5	Internal flag relating to transaction count	When the transaction counter function is forcibly terminated

This bit should be changed only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When changing this bit after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(e) Bit for Sequence Toggle Bit Clear (SQCLR)

When this bit is set to 1 by software, this module sets the expectation of the sequence toggle bit for the corresponding pipe to DATA0. This bit is always read as 0.

This bit should be set to 1 only when CSSTS = 0 and PID = NAK. When setting this bit to 1 after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(f) Bit for Sequence Toggle Bit Set (SQSET)

When this bit is set to 1 by software, this module sets the expectation of the sequence toggle bit for the corresponding pipe to DATA0. This bit is always read as 0.

This bit should be set to 1 only when CSSTS = 0 and PID = NAK. When setting this bit to 1 after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(g) Bit for Sequence Toggle Bit Monitor (SQMON)

The expectation of the sequence toggle bit for the corresponding pipe can be read from this bit.

If the transfer type for the corresponding pipe is other than isochronous transfer, this module toggles this bit when the transaction has been processed normally. Note however that this bit is not toggled when a DATA-PID mismatch occurs during transfer in the receiving direction.

(h) Pipe Busy Bit (PBUSY)

This module changes this bit from 0 to 1 when the USB transaction of the corresponding pipe has started. This bit is changed from 1 to 0 upon the completion of a single transaction.

Reading this bit after PID = NAK has been set by software is performed to confirm whether the pipe settings can be changed.

(i) Response PID Bits (PID)

The response of this module for each pipe should be set in these bits by software.

The default value of these bits is NAK. When performing USB transfer at the corresponding pipe, change the setting of these bits to BUF. The basic operations (operations without errors in the communication packet) of this module for each PID bit setting are shown in table 14.20.

When changing these bits from BUF to NAK by software when the corresponding pipe is in the middle of USB communication, after these bits have been set to NAK, confirm PBUSY = 1 to make sure that the USB transfer status of the corresponding pipe has actually transited to the NAK state. However, if these bits were changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

Even though these bits are changed to NAK by software after S-Split of a Split transaction is issued (CSSTS = 1) for the corresponding pipe, the transaction continues to be executed until its end.

This module changes the value of these bits in the following cases:

- When the corresponding pipe is set to the receiving direction and software has set the SHTNAK bit of the corresponding pipe to 1, PID = NAK can be read from these bits as soon as this module recognizes the end of transfer.
- When this module receives a data packet whose payload exceeds MaxPacketSize for the corresponding pipe, PID = STALL (11) can be read from these bits.
- When a USB bus reset has been detected by this module, PID = NAK can be read from these bits.

Write 10 to these bits to change PID = NAK (00) to PID = STALL. Write 11 to these bits to change PID = BUF (01) to PID = STALL. Write 10 first and then 00 to these bits to change PID = STALL (11) to PID = NAK. To change the STALL state to the BUF state, change these bits to the NAK state first and then change them to the BUF state.

Table 14.20 Module Operations with Various PID Bit Settings

PID Bit Setting	Transfer Type (TYPE Bit Setting)	Transfer Direction (DIR Bit Setting)	Operation of This Module
00 (NAK)	Bulk (TYPE = 01) or interrupt (TYPE = 10)	Does not depend on the setting	The NAK response is returned to the token from the USB host. However, when ATREPM = 1 is set, this module carries out the operations listed in (c) Auto Response Mode Bit (ATREPM).
	Isochronous (TYPE = 11)	Does not depend on the setting	No response is returned to the token from the USB host.
01 (BUF)	Bulk (TYPE = 01)	Receiving direction (DIR = 0)	If the FIFO buffer of the corresponding pipe can receive data, the OUT token from the USB host is received and the ACK response is returned. If data cannot be received, the NAK response is returned. If the FIFO buffer of the corresponding pipe can receive data, the PING token from the USB host is received and the ACK response is returned. If data cannot be received, the NYET response is returned.
	Interrupt (TYPE = 10)	Receiving direction (DIR = 0)	If the FIFO buffer of the corresponding pipe can receive data, the OUT token from the USB host is received and the ACK response is returned. If data cannot be received, the NAK response is returned.
	Bulk (TYPE = 01) or interrupt (TYPE = 10)	Transmitting direction (DIR = 1)	If the corresponding FIFO buffer can transmit data, data is transmitted in response to the token from the USB host. If data cannot be transmitted, the NAK response is returned.

PID Bit Setting	Transfer Type (TYPE Bit Setting)	Transfer Direction (DIR Bit Setting)	Operation of This Module
01 (BUF)	Isochronous (TYPE = 11)	Receiving direction (DIR = 0)	If the FIFO buffer of the corresponding pipe can receive data, the OUT token from the USB host is received. If data cannot be received, the data is discarded.
		Transmitting direction (DIR = 1)	If the corresponding FIFO buffer can transmit data, data is transmitted in response to the token from the USB host. If data cannot be transmitted, a zero-length packet is transmitted.
10 (STALL) or 11 (STALL)	Bulk (TYPE = 01) or interrupt (TYPE = 10)	Does not depend on the setting	The STALL response is returned to the token from the USB host.
	Isochronous (TYPE = 11)	Does not depend on the setting	No response is returned to the token from the USB host.

- **PIPE6 Control Register (PIPE6CTR) <Address: H'07A>**
- **PIPE7 Control Register (PIPE7CTR) <Address: H'07C>**
- **PIPE8 Control Register (PIPE8CTR) <Address: H'07E>**
- **PIPE9 Control Register (PIPE9CTR) <Address: H'080>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSTS						ACLRM	SQCLR	SQSET	SQMON	PBSY				PID	
0	?	?	?	?	?	0	0	0	0	0	?	?	?	0	0
-	?	?	?	?	?	-	-	-	-	-	?	?	?	0	0

Bit	Name	Function	S/W	H/W	Remarks
15	BSTS Buffer status	The FIFO buffer status of the corresponding pipe can be read. 0: Buffer access is disabled 1: Buffer access is enabled	R	W	
14 to 10	No function assigned; the write value should always be 0.				
9	ACLRM Auto buffer clear mode	Enables or disables auto buffer clear mode of the corresponding pipe. 0: Auto buffer clear mode is disabled 1: Auto buffer clear mode is enabled (all buffers are initialized)	R/W	R	
8	SQCLR Sequence toggle bit clear	Set this bit to 1 when the expectation of the sequence toggle bit in the next transaction of the corresponding pipe is to be set in DATA0. 0: Invalid 1: Specifies DATA0	R (0)/ W (1)	R	
7	SQSET Sequence toggle bit set	Set this bit to 1 when the expectation of the sequence toggle bit in the next transaction of the corresponding pipe is to be set in DATA1. 0: Invalid 1: Specifies DATA1	R (0)/ W (1)	R	

Bit	Name	Function	S/W	H/W	Remarks
6	SQMON Sequence toggle bit monitor	The expectation of the sequence toggle bit in the next transaction of the corresponding pipe can be read. 0: DATA0 1: DATA1	R	W	
5	PBUSY Pipe busy	Whether the corresponding pipe is currently being used by the USB bus can be read. 0: Corresponding pipe is not being used by the USB bus 1: Corresponding pipe is being used by the USB bus	R	W	
4 to 2	No function assigned; the write value should always be 0.				
1 to 0	PID Response PID	These bits specify the response type for the next transaction of the corresponding pipe. 00: NAK response 01: BUF response (depending on the buffer state) 10: STALL response 11: STALL response	R/W	R/W	

<<Remarks>>

None

Buffer Status Bit (BSTS)

Refer to (a) Buffer Status Bit (BSTS) in (1) to (5).

(j) Auto Buffer Clear Mode Bit (ACLRM)

To delete all contents of the FIFO buffer allocated for the corresponding pipe, write 1 to the ACLRM bit and then write 0 to it in succession.

Table 14.21 shows the contents cleared by this module when the ACLRM bit is set to 1 and then to 0, and the cases in which the contents need to be cleared.

Table 14.21 Contents Cleared by This Module when ACLRM = 1 is Set

No.	Contents Cleared by ACLRM Bit Operation	Case Requiring Clearing of Contents
1	All contents in the FIFO buffer allocated for the corresponding pipe	
2	Interval counter value when isochronous transfer is set as the transfer type for the corresponding pipe	When the interval counter value needs to be reset
3	Internal flag relating to BFRE bit	When the BFRE bit setting is changed
4	Internal flag relating to transaction count	When the transaction counter function is forcibly terminated

This bit should be changed only when CSSTS = 0, PID = NAK, and no pipe is set in the CURPIPE bits. When changing this bit after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing this bit. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(k) Bit for Sequence Toggle Bit Clear (SQCLR)

Refer to (e) Bit for Sequence Toggle Bit Clear (SQCLR) in (1) to (5).

(l) Bit for Sequence Toggle Bit Set (SQSET)

Refer to (f) Bit for Sequence Toggle Bit Set (SQSET) in (1) to (5).

(m) Bit for Sequence Toggle Bit Monitor (SQMON)

Refer to (g) Bit for Sequence Toggle Bit Monitor (SQMON) in (1) to (5).

(n) Pipe Busy Bit (PBUSY)

Refer to (h) Pipe Busy Bit (PBUSY) in (1) to (5).

(o) Response PID Bits (PID)

Refer to (i) Response PID Bits (PID) in (1) to (5).

(16) Transaction Counters

- **PIPE1 Transaction Counter Enable Register (PIPE1TRE)** **<Address: H'090>**
- **PIPE2 Transaction Counter Enable Register (PIPE2TRE)** **<Address: H'094>**
- **PIPE3 Transaction Counter Enable Register (PIPE3TRE)** **<Address: H'098>**
- **PIPE4 Transaction Counter Enable Register (PIPE4TRE)** **<Address: H'09C>**
- **PIPE5 Transaction Counter Enable Register (PIPE5TRE)** **<Address: H'0A0>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TRENB	TRCLR								
?	?	?	?	?	?	0	0	?	?	?	?	?	?	?	?
?	?	?	?	?	?	-	-	?	?	?	?	?	?	?	?

Bit	Name	Function	S/W	H/W	Remarks
15 to 10	No function assigned; the write value should always be 0.				
9	TRENB Transaction counter enable	Enables or disables the transaction counter function. 0: Transaction counter function is disabled 1: Transaction counter function is enabled	R/W	R	
8	TRCLR Transaction counter clear	Setting this bit to 1 clears the transaction counter value to 0. 0: Invalid 1: Current count is cleared.	R (0)/ W (1)	R	
7 to 0	No function assigned; the write value should always be 0.				

<<Remarks>>

* The bits in this register should be set only when CSSTS = 0 and PID = NAK.
When changing the bits in this register after the PID bit setting of the selected pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing the bits. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

(a) Transaction Counter Enable Bit (TRENB)

For a pipe that has been set to the receiving direction, when this bit is set to 1 after the total number of packets has been set in the TRCNT bits by software, this module performs the

following operations when packets for the same number as the TRCNT bit setting have been received.

- When continuous transfer mode is selected (CNTMD = 1), this bit should be toggled by the CPU even though the FIFO buffer is not full when reception has finished.
- When SHTNAK = 1 is set, the PID bits of the corresponding pipe are changed to NAK when packets for the same number as the TRCNT bit setting have been received.
- When both DENDE = 1 and PKTMD = 0 are set, the DEND signal is asserted when the last data is to be read after receiving packets for the same number as the TRCNT bit setting.
- When BFRE = 1 is set, the BRDY interrupt signal is asserted when the last data has been read after receiving packets for the same number as the TRCNT bit setting.

For a pipe that has been set to the transmitting direction, this bit should be cleared to 0.

When the transaction counter function is not used, clear this bit to 0. When the transaction counter function is used, set the TRNCNT bits before setting this bit to 1. Note that this bit should be set to 1 before receiving the first packet that will be counted as a transaction.

(b) Transaction Counter Clear Bit (TRCLR)

When this bit is set to 1 by software, this module clears the current counter value of the transaction counter for the corresponding pipe, and then writes 0 to this bit.

- **PIPE1 Transaction Counter Register (PIPE1TRN)** **<Address: H'092>**
- **PIPE2 Transaction Counter Register (PIPE2TRN)** **<Address: H'096>**
- **PIPE3 Transaction Counter Register (PIPE3TRN)** **<Address: H'09A>**
- **PIPE4 Transaction Counter Register (PIPE4TRN)** **<Address: H'09E>**
- **PIPE5 Transaction Counter Register (PIPE5TRN)** **<Address: H'0A2>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRNCNT															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit	Name	Function		S/W	H/W	Remarks
15 to 0	TRNCNT Transaction counter	At write:	These bits set the number of transactions in DMA transfer.	R/W	R/W	
		At read:	When TRENB = 0, the set number of transactions can be read. When TRENB = 1, the number of transactions that is being counted can be read.			

<<Remarks>>

None

Transaction Counter Bits (TRNCNT)

For a pipe that has been set to the receiving direction, when the TRENB bit is set to 1 after the total number of packets that must be received has been set in these bits by software, this module performs the operations listed in (a) Transaction Counter Enable Bit (TRENB).

When the TRENB bit is 0, the number of transactions set by software can be read from these bits. When the TRENB bit is 1, the number of transactions that is being counted can be read from these bits.

The TRNCNT bit value is incremented by 1 when all of the following conditions (a) to (c) are satisfied in the receiving state.

(a) TRENB = 1

(b) (TRNCNT setting ≠ Current counter value + 1) at packet reception

(c) The payload of the received packet matches the MXPS bit setting

The TRNCNT bit value is cleared to 0 when any one of the following conditions (1) to (3) is satisfied.

(1) When all of the following conditions (a) to (c) are satisfied
  (a) TRENB = 1
  (b) (TRNCNT setting = Current counter value + 1) at packet reception
  (c) The payload of the received packet matches the MXPS bit setting
(2) When both of the following conditions (a) and (b) are satisfied
  (a) TRENB = 1
  (b) A short packet is received
(3) When both of the following conditions (a) and (b) are satisfied
  (a) TRENB = 1
  (b) The TRCLR bit is set to 1 by software

For a pipe that has been set to the transmitting direction, these bits should be cleared to 0.

When the transaction counter function is not used, clear these bits to 0.

These bits should be changed only when CSSTS = 0, PID = NAK, and TRENB = 0. When changing these bits after the PID bit setting of the corresponding pipe has been changed from BUF to NAK, confirm CSSTS = 0 and PBUSY = 0 before changing these bits. However, if the PID bit setting was changed to NAK by this module, the PBUSY bit value does not have to be confirmed by software.

When changing the value of these bits, set TRCLR = 1 prior to the setting of TRENB = 1.

(17) UTMI Suspend Mode Register

- **UTMI Suspend Mode Register (SUSPMODE)** **<Address: H'102>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SUSPM														
?	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	-	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bit	Name	Function	S/W	H/W	Remarks
15	No function assigned; the write value should always be 0.				
14	SUSPM UTMI SuspendM control	Controls the SuspendM signal sent to the UTMI. 0: UTMI suspend mode 1: UTMI normal mode	R/W	R/W	
13 to 0	No function assigned; the write value should always be 0.				

<<Remarks>>

None

(a) UTMI SuspendM Control Bit (SUSPM)

This bit is used for controlling the SuspendM signal sent to the UTMI. The initial value is 0, so the UTMI is in the suspend mode. This bit should be set to 1 when this module is operating.

In the UTMI specification, the SuspendM signal is normally used for clock output control. When SuspendM = 0, the clock supply to the LINK is stopped.

For details, refer to the respective UTMI Specifications.

When this bit is cleared to 0 (UTMI clock is stopped), this module cannot be written to but reading is possible. However, the registers listed in table 14.22 can be written to even when this bit is 0.

Table 14.22 Registers that Can be Written to by Software when SUSPM = 0

Address	Register Name
H'000	SYSCFG0
H'002	BUSWAIT
H'102	SUSPMODE

Note that the value written to SYSCFG0 while the UTMI clock is stopped (SUSPM = 0), will be reflected in the setting after the UTMI clock starts oscillating (SUSPM = 1).

14.4.5 Operation

(1) System Control and Oscillation Control

This section describes the register operations that are necessary for the initial settings of this module, and the registers necessary for power consumption control.

(a) Resets

Table 14.23 lists the types of resets for this module. For the initialized states of the registers following the reset operations, see section 14.4.4, Register Specifications.

Table 14.23 Types of Reset

Name	Operation
Hardware reset	Low level input from the EXL_SYSRST pin
USB bus reset	Automatically detected by this module from the D+ and D– lines

(b) USB Data Bus Resistor Control

This module performs switching of a pull-up resistor for the D+ signal and a pull-down resistor for the D+ and D- signals. These signals can be pulled up or down using the DPRPU and DRPD bits in SYSCFG0.

When the DPRPU bit in SYSCFG0 is cleared to 0 during communication with the host controller, the pull-up resistor (or the terminal resistor) of the USB data line is disabled, making it possible to notify the host controller of the device disconnection.

(c) Clock Supply

Table 14.24 shows the two clock inputs necessary for this module.

Table 14.24 Clock Inputs

Input Clock	Function
CPU clock	CPU clock input. Any clock frequency is possible.
UTMI clock	Clock input from the UTMI. The clock frequency differs according to the UTMI interface bus width. For 8-bit interface: 60 MHz For 16-bit interface: 30 MHz The UTMI clock oscillation can be controlled using the SuspendM bit. For details, refer to the UTMI Specifications.

When a local bus is connected, it is possible to input an FIFO access clock different from the CPU clock.

(d) Notes at Clock Stop

- The CPU clock and UTMI clock can be stopped during disconnection or when in the suspended state.
- If clock supply has been stopped when the USB function module is in the suspended state with the function controller function selected, the clock supply must be restarted on recovery from the suspended state (resume signal detection). The UTMI clock supply requires to be restarted within 2.5 ms after a resume interrupt has been generated.

(2) Interrupt Functions

(a) Overview of Interrupt Functions

Table 14.25 lists the interrupt functions of this module.

Table 14.25 List of Interrupt Functions

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
VBINT	VBUS interrupt	When a change in the state of the VBUS input pin has been detected (low to high or high to low)		VBSTS
RESM	Resume interrupt	When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)		—
SOFR	Frame number update interrupt	• SOFRM = 0: When an SOF packet with a different frame number is received • SOFRM = 1: When the SOF packet with the μframe number of 0 could not be received due to corruption of a packet		—
DVST	Device state transition interrupt	When a device state transition is detected • A USB bus reset is detected • The suspended state is detected • Set address request is received • Set configuration request is received		DVSQ

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
CTRT	Control transfer stage transition interrupt	When a stage transition is detected in control transfer • Setup stage is completed • Control write transfer status stage transition • Control read transfer status stage transition • Control transfer end • A control transfer sequence error occurred		CTSQ
BEMP	Buffer empty interrupt	• When transmission of all data items in the buffer memory has been completed and the buffer is empty • When a packet whose size exceeds the maximum packet size has been received		PIPEBEMP
NRDY	Buffer not ready interrupt	• When an IN token, OUT token, or PING token has been received and a NAK response is returned • When a CRC error or bit stuffing error occurred while receiving data in isochronous transfer • When an overrun/underrun occurred during isochronous transfer		PIPENRDY
BRDY	Buffer ready interrupt	When the buffer is ready (reading or writing is enabled)		PIPEBRDY
OVRCR	OVRCR interrupt	When a change in the state of the OVRCR input pin has been detected		OVCMON
BCHG	Bus change interrupt	When a change in the state of the USB bus has been detected		—

Table 14.26 shows the IFL_INT pin operation of this module. When more than one interrupt source is generated, the IFL_INT pin output mode can be selected by the INTL bit in INTENB1.

The IFL_INT pin operation should be specified in conjunction with the specifications of the user system.

Table 14.26 IFL_INT Pin Operation

IFL_INT Pin Operation / INTL Setting	One Interrupt Source is Generated	Two or More Interrupt Sources are Generated
Edge-trigger (INTL = 0)	Low-level output until the source is cleared	When one source is cleared, the pin is negated (high-level pulse output) for 32 clock cycles at 48 MHz
Level-trigger (INTL = 1)	Low-level output until the source is cleared	Low-level output until all sources are cleared

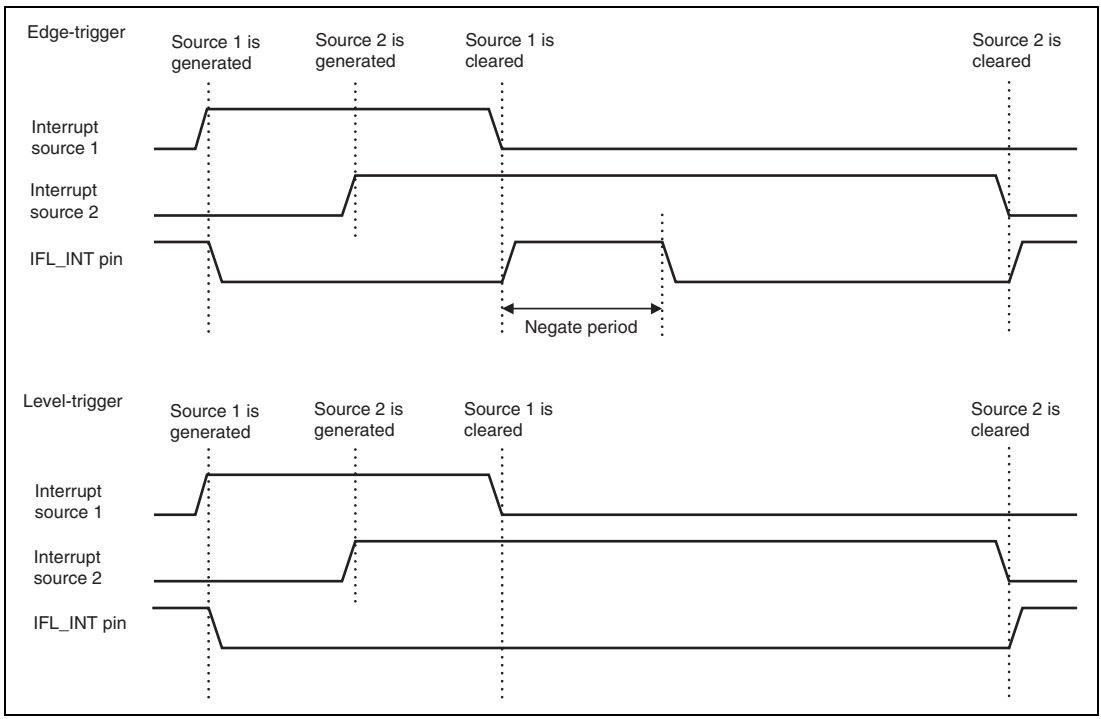


Figure 14.5 IFL_INT Pin Operation

Figure 14.6 shows a diagram relating to interrupts of this module.

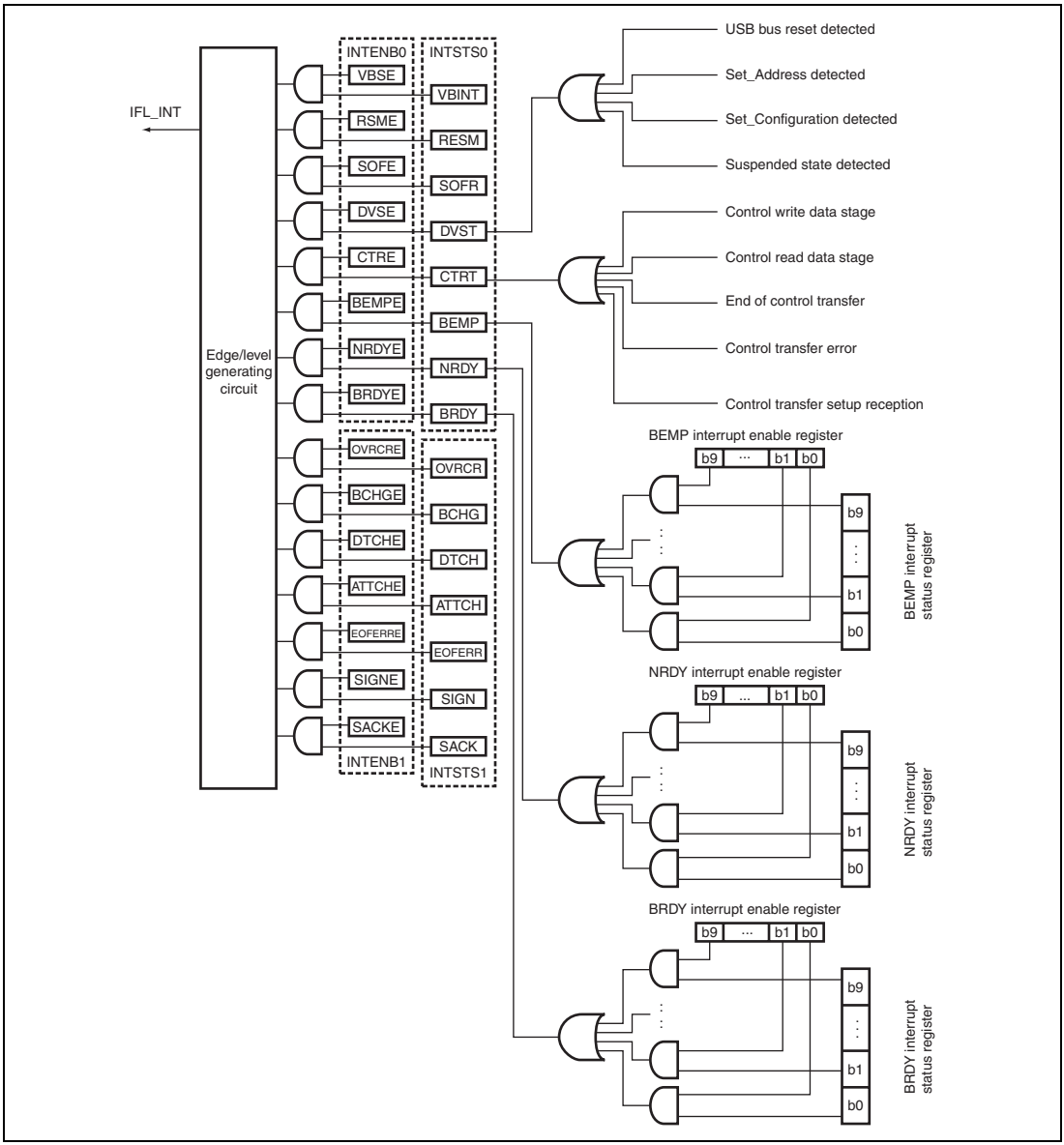
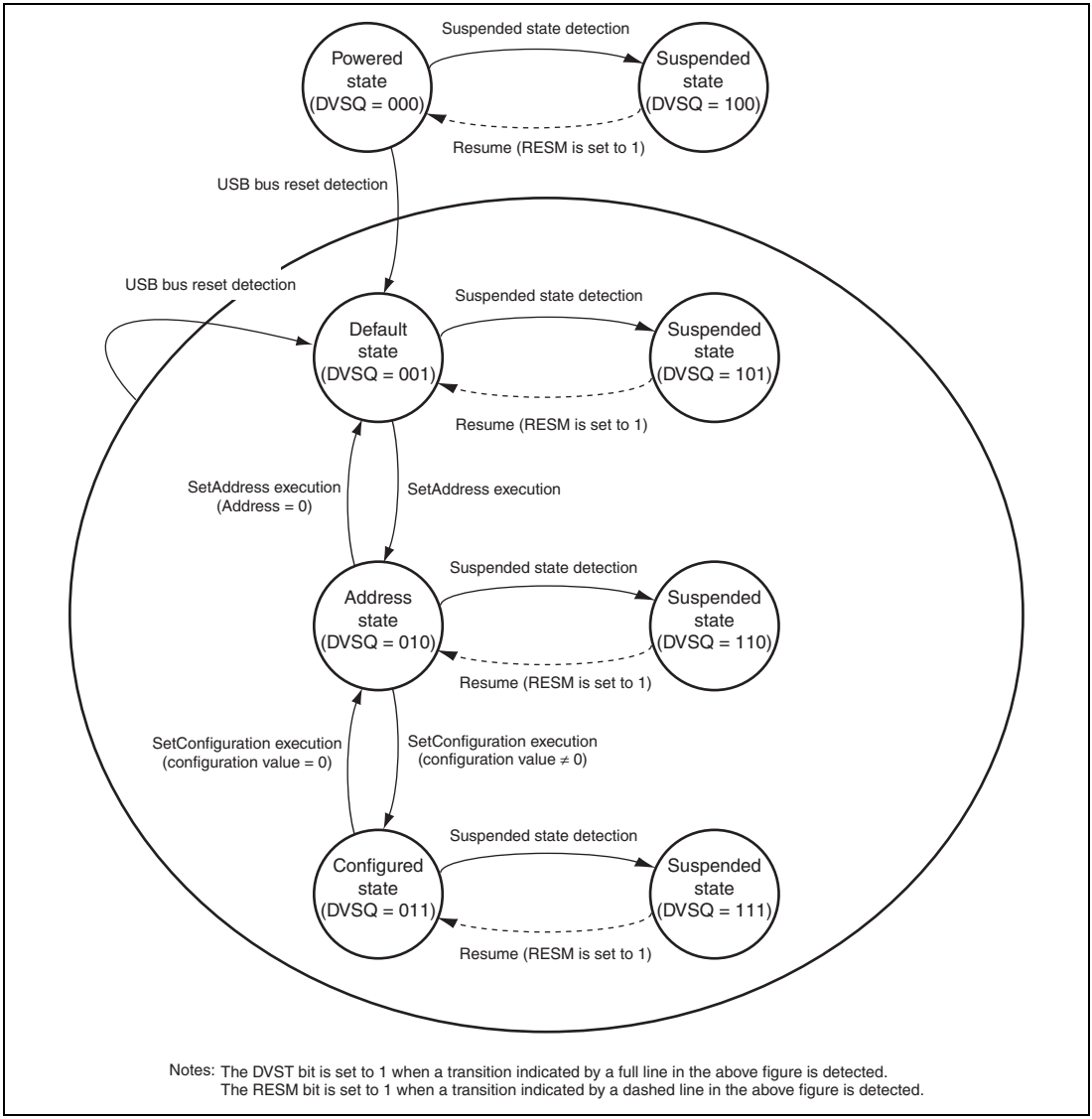


Figure 14.6 Items Relating to Interrupts

(b) Device State Transition Interrupt

Figure 14.7 shows a diagram of this module's device state transitions. This module controls device states and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually by INTENB0. The device state that made a transition can be confirmed using the DVSQ bit in INTSTS0.

To make a transition to the default state, the device state transition interrupt is generated after the reset handshake protocol has been completed.



Notes: The DVST bit is set to 1 when a transition indicated by a full line in the above figure is detected.
The RESM bit is set to 1 when a transition indicated by a dashed line in the above figure is detected.

Figure 14.7 Device State Transitions

(c) Control Transfer Stage Transition Interrupt

Figure 14.8 shows a diagram of how this module handles the control transfer stage transitions. This module controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled individually by INTENB0. The transfer stage that made a transition can be confirmed using the CTSQ bit in INTSTS0.

The control transfer sequence errors are described below. If an error occurs, the PID bits in DCPCTR are set to 1X (STALL).

1. During control read transfers
   - — At the IN token of the data stage, an OUT or PING token is received when there have been no data transfers at all
   - — An IN token is received at the status stage
   - — A packet for which the data packet is DATAPID = DATA0 is received at the status stage
2. During control write transfers
   - — At the OUT token of the data stage, an IN token is received when there have been no ACK responses at all
   - — A packet for which the first data packet is DATAPID = DATA0 is received at the data stage
   - — At the status stage, an OUT or PING token is received
3. During no-data control transfers
   - — At the status stage, an OUT or PING token is received

At the data stage of a control write transfer, if the number of received data items exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the status stage of a control read transfer, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (SERR = 1), the CTSQ = 110 value is retained until CTRT = 0 is written from the user system (the interrupt status is cleared). Therefore, while CTSQ = 110 is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (This module retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

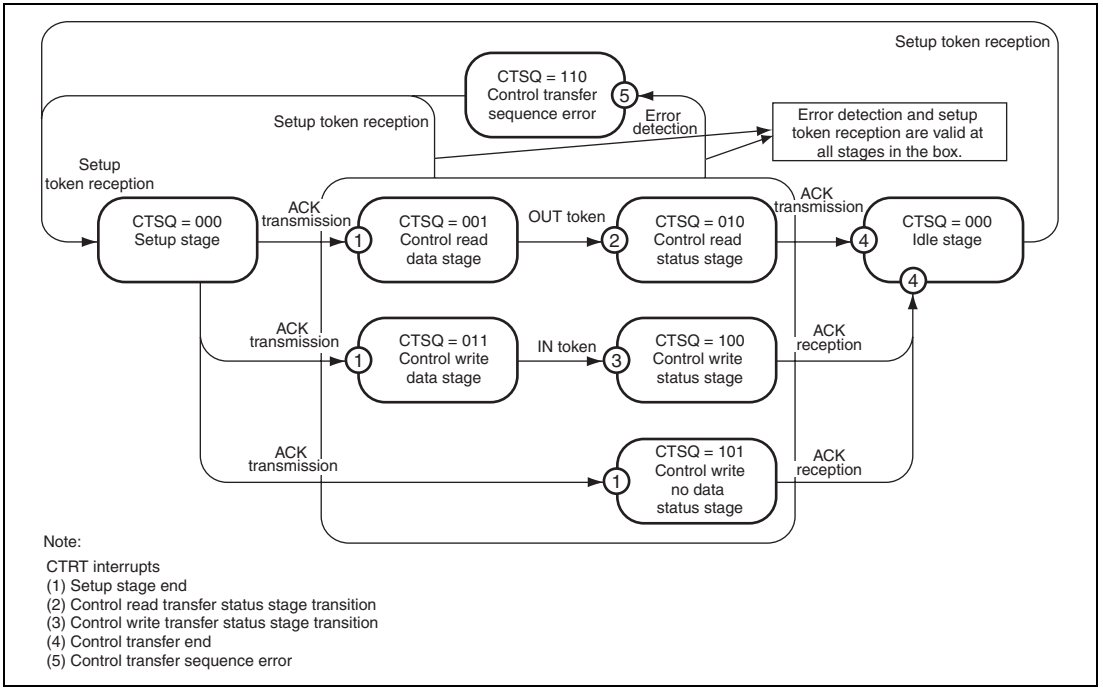


Figure 14.8 Control Transfer Stage Transitions

(3) Pipe Control

Table 14.27 lists the pipe setting items of this module. With USB data transfer, data transmission has to be carried out using the logic pipe called the endpoint. This module has ten pipes that are used for data transfer. Settings should be entered for each of the pipes in conjunction with the specifications of the user system.

Table 14.27 Pipe Setting Items

Register Name	Bit Name	Setting Contents	Remarks
DCPCFG PIPECFG	TYPE	Specifies the transfer type	PIPE1 to PIPE9: Can be set
	BFRE	Selects the BRDY interrupt mode	PIPE1 to PIPE5: Can be set
	DBLB	Selects a double buffer	PIPE1 to PIPE5: Can be set
	CNTMD	Selects continuous transfer or non-continuous transfer	PIPE1 and PIPE2: Can be set only when bulk transfer has been selected PIPE3 to PIPE5: Can be set
	DIR	Selects the transfer direction	IN or OUT can be set
	EPNUM	Endpoint number	PIPE1 to PIPE9: Can be set Set a value other than 0000 when the pipe is used.
	SHTNAK	Selects the disabled state for the pipe when transfer ends	PIPE1 and PIPE2: Can be set only when bulk transfer has been selected PIPE3 to PIPE5: Can be set
PIPEBUF	BUFSIZE	Buffer memory size	DCP: Cannot be set (fixed at 256 bytes) PIPE1 to PIPE5: Can be set (a maximum of 2 Kbytes can be specified) PIPE6 to PIPE9: Cannot be set (fixed at 64 bytes)
	BUFNMB	Buffer memory number	DCP: Cannot be set (areas fixed at H'0 to H'3) PIPE1 to PIPE5: Can be set (can be specified in areas H'8 to H'87) PIPE6 to PIPE9: Cannot be set (areas fixed at H'4 to H'7)
DCPMAXP	DEVSEL	Device select	Can be referenced only when the host controller function is selected

Register Name	Bit Name	Setting Contents	Remarks
PIPEMAXP	MXPS	Maximum packet size	The setting must conform to the USB specification
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set only when isochronous transfer has been selected PIPE3 to PIPE9: Cannot be set
	IITV	Interval counter	PIPE1 and PIPE2: Can be set only when isochronous transfer has been selected PIPE3 to PIPE9: Cannot be set
DCPCTR PIPExCTR	BSTS	Buffer status	The receive and transmit buffer states of the DCP can be switched by the ISEL bit
	INBUFM	IN buffer monitor	Incorporated only in PIPE3 to PIPE5
	SUREQ	Setup request	Can be set only in DCP
	SUREQCLR	SUREQ clear	Can be set only in DCP
	CSCLR	CSSTS clear	
	CSSTS	Split status confirmation	
	ATREPM	Auto response mode	PIPE1 to PIPE5: Can be set Can be set only when the function controller function is selected
	ACLRM	Auto buffer clear	PIPE1 to PIPE9: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence monitor	Confirms the data toggle bit
	PBUSY	Pipe busy confirmation	
	PID	Response PID	
PIPExTRE	TRENB	Transaction counter enable	PIPE1 to PIPE5: Can be set
	TRCLR	Current transaction counter clear	PIPE1 to PIPE5: Can be set
PIPExTRN	TRNCNT	Transaction counter	PIPE1 to PIPE5: Can be set

(a) Switching Procedure for Pipe Control Registers

The pipe control register bits shown below can be changed only when USB communication is disabled (PID = NAK). Figure 14.9 shows the procedure for switching the USB communication enabled state (PID = BUF) in order to change the pipe control register settings.

The registers which must not be set in the USB communication enabled state (PID = BUF) are as follows:

- Bits in DCPCFG and DCPMAXP
- Bits SQCLR and SQSET in DCPCTR
- Bits in PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- Bits ATREPM, ACLRM, SQCLR, and SQSET in PIPExCTR
- Bits in PIPExTRE and PIPExTRN

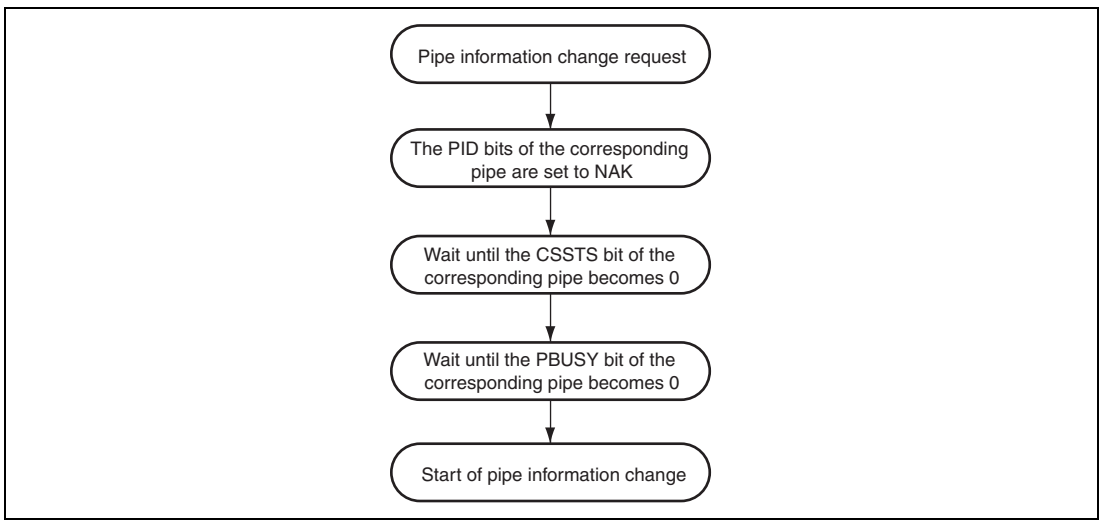


Figure 14.9 Procedure for Changing Pipe Information When in the USB Communication Enabled State (PID = BUF)

The pipe control register bits shown below can be changed when the corresponding pipe is not set in the CURPIPE bits for any of the FIFO ports (CPU-FIFO, DMA0-FIFO, and DMA1-FIFO).

The registers which must not be set when the corresponding pipe is set in the CURPIPE bits for the FIFO ports are as follows:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI

To change the pipe information, change the CURPIPE setting to a pipe that is not to be changed. Note that after the pipe information has been changed, the FIFO buffer for the DCP should be cleared using the BCLR bit.

(4) FIFO Buffer

Operations relating to the FIFO buffer in this module are described here. If not specified otherwise, the operations are the same regardless of whether the host controller function or function controller function is selected.

(a) FIFO Buffer Allocation

Figure 14.10 shows an example of a memory map for the FIFO buffer in this module. The FIFO buffer is an area shared by this module and the CPU used for controlling the user system. In the FIFO buffer status, there are times when the access right to the buffer memory is allocated to the user system (CPU side), and times when it is allocated to this module (SIE side).

The FIFO buffer sets independent areas for each pipe. In the memory areas, 64 bytes comprise one block, and the memory areas are set using the first block number and the number of blocks (specified using the BUFNMB and BUFSIZE bits in PIPEBUF). When continuous transfer mode is selected using the CNTMD bit in PIPExCFG, the BUFSIZE bit setting must be set to an integer multiple of the maximum packet size. Also, when a double buffer is selected using the DBLB bit in PIPExCFG, the memory area specified by the BUFSIZE bits in PIPEBUF is allocated for two buffers for the same pipe.

Moreover, three FIFO ports are used for access to the FIFO buffer (reading and writing data). A pipe is assigned to the FIFO port by specifying the pipe number using the CURPIPE bits in C/DxFIFOSEL.

The FIFO buffer statuses of the various pipes can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPExCTR. Also, the access right of the FIFO port can be confirmed using the FRDY bit in C/DxFIFOCTR.

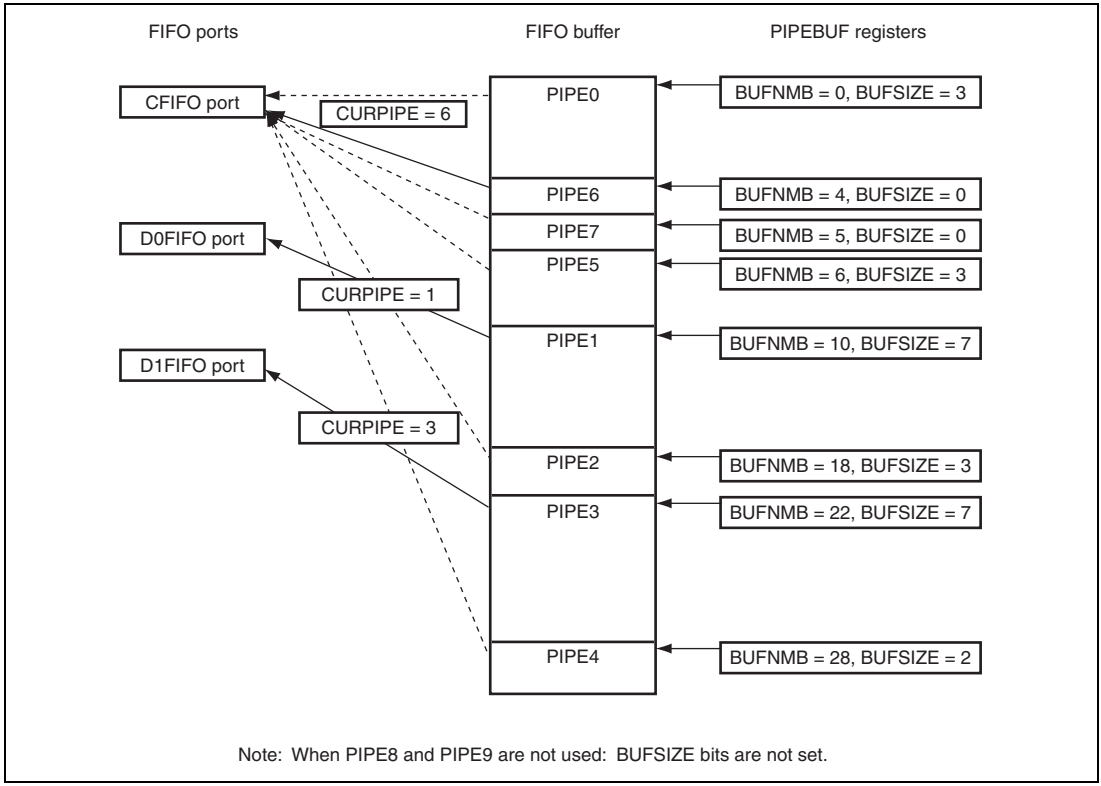


Figure 14.10 Example of a FIFO Buffer Memory Map

(b) FIFO Buffer Clearing

Table 14.28 shows the clearing of the FIFO buffer by this module. The FIFO buffer can be cleared using the three bits indicated below.

Table 14.28 List of FIFO Buffer Clearing Methods

Bit Name	BCLR	DCLRM	ACLRM
Register	CFIFOCTR DxFIFOCTR	DxFIFOSEL	PIPExCTR
Function	Clears the FIFO buffer on the CPU side.	In this mode, after the data of the specified pipe has been read, the FIFO buffer is cleared automatically.	This is the auto buffer clear mode, in which all of the received packets are discarded.
Clearing method	Cleared by writing 1.	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(5) FIFO Port Functions

The functions relating to the FIFO ports are described here. Table 14.29 shows the settings for the FIFO port functions of this module. In write access, writing data until the buffer is full (or the maximum packet size for non-continuous transfers) automatically enables sending data to the USB bus. To enable sending of data before the buffer is full (or less than the maximum packet size for non-continuous transfers), the BVAL bit in C/DxFIFOCTR must be set to 1 to end the writing (TEND signal for DMA transfers). Also, to send a zero-length packet, the BCLR bit in C/DxFIFOCTR must be used to clear the buffer and then the BVAL bit set to 1 in order to end the writing.

In read access, reception of new packets is automatically enabled if all of the data items have been read. Data cannot be read when a zero-length packet is being received (DTLN = 0), so the BCLR bit in C/DxFIFOCTR must be used to clear the buffer. The length of the data being received can be confirmed using the DTLN bit in C/DxFIFOCTR.

Table 14.29 FIFO Port Function Settings

Register Name	Bit Name	Function	Note
C/DxFIFOSEL	RCNT	Selects the DTLN read mode	
	REW	Buffer memory rewind (re-read, rewrite)	
	DCLRM	Automatically clears data received for a specified pipe after the data has been read	For DxFIFO only
	DREQE	Asserts the DREQ signal	For DxFIFO only
	MBW	FIFO port access bit width	
	BIGEND	Selects the FIFO port endian	
	ISEL	FIFO port access direction	For DCP only
	CURPIPE	Selects the current pipe	
C/DxFIFOCTR	BVAL	Ends writing to the buffer memory	
	BCLR	Clears the buffer memory on the CPU side	
	FRDY	FIFO port ready monitor	
	DTLN	Confirms the length of received data	

(a) FIFO Port Selection

Table 14.30 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed is selected using the CURPIPE bits in C/DxFIFOSEL. After the pipe has been selected, confirm that the written CURPIPE value can be read correctly. (If the previous pipe number is read, this module is still in the middle of changing the pipe.) After this confirmation has been made, FRDY = 1 should be confirmed before accessing the FIFO port.

Also, the bus width to be accessed should be selected using the MBW bit. The buffer memory access direction conforms to the ISEL bit for the DCP, and conforms to the DIR bit in PIPExCFG for the other pipes.

Table 14.30 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
PIPE1 to PIPE9	CPU access	CFIFO port register
		DxFIFO port register
	DMA access	DxFIFO port register

(b) DxFIFO Auto Clear Mode (DxFIFO Port Reading Direction)

If 1 is set for the DCLRM bit in DxFIFOSEL, this module automatically clears the buffer memory of the corresponding pipe when reading of the data from the buffer memory has been completed.

Table 14.31 shows the packet reception and buffer memory clearing processing for each of the various settings. As shown in table 14.31, the buffer clearing conditions depend on the BFRE bit setting. Using the DCLRM bit eliminates the need for the buffer to be cleared by software even if a situation occurs that necessitates clearing of the buffer. This makes it possible to carry out DMA transfers without involving software.

This function can be set only when reading from the buffer memory.

Table 14.31 Packet Reception and Buffer Memory Clearing Processing by Software

Register Setting / Buffer Status When Packet is Received	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Doesn't need to be cleared	Doesn't need to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Zero-length packet reception	Needs to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Normal short packet reception	Doesn't need to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Transaction count ended	Doesn't need to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared

(c) BRDY Interrupt Timing Selection Function

By setting the BFRE bit in PIPECFG, it is possible to keep the BRDY interrupt from being generated when a data packet consisting of the maximum packet size is received.

When using DMA transfers, this function can be used to generate an interrupt only when the last data item has been received. The last data item refers to the reception of a short packet, or the ending of the transaction counter. When the BFRE bit is set to 1, the BRDY interrupt is generated after the received data has been read. When the DTLN bit in DxFIFOCTR is read, the length of the data received in the last data packet to have been received can be confirmed.

Table 14.32 shows the timing at which the BRDY interrupts are generated by this module.

Table 14.32 Timing at which BRDY Interrupts are Generated

Register setting / Buffer State When Packet is Received	BFRE = 0	BFRE = 1
Buffer full (normal packet received)	When packet is received	Not generated
Zero-length packet received	When packet is received	When packet is received
Normal short packet received	When packet is received	When reading of the received data from the buffer memory has been completed
Transaction count ended	When packet is received	When reading of the received data from the buffer memory has been completed

Note: This function is valid only when reading from the buffer memory. When writing to the buffer memory, the BFRE bit should be fixed at 0.

(6) Control Transfers (DCP)

Data transfers of the data stage of control transfers are done using the default control pipe (DCP). The DCP buffer memory is a 256-byte single buffer, and is a fixed area that is shared for both control read transfers and control write transfers. The buffer memory can be accessed through only the CFIFO port.

This module always sends an ACK response in response to a normal setup packet for this module. The operations of this module in the setup stage are shown below.

1. When a new SETUP packet is received, this module sets the following bits:
   — Set the VALID bit in INTSTS0 to 1.
   — Set the PID bits in DCPCTR to NAK.
   — Clear the CCPL bit in DCPCTR to 0.
2. When a data packet is received right after the SETUP packet, this module stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should always be carried out after first setting VALID = 0. In the VALID = 1 state, PID = BUF cannot be set, and so the data stage cannot be terminated.

Using the function of the VALID bit, this module is able to interrupt the processing of a request currently being processed if a new USB request is received during a control transfer, and can send a response in response to the newest request.

Also, this module automatically judges the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the USB request that was received, and then distinguishes between control read transfers, control write transfers, and no-data control transfers, and controls the stage transitions. For an incorrect sequence, the sequence error of the control transfer stage transition interrupt is generated, and software is notified. For information on the stage control of this module, see figure 14.8.

Data transfers corresponding to USB requests that have been received should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the ISEL bit in CFIFOSEL.

If the data being transferred is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

With control write transfers during high-speed operation, the NYET handshake response is returned based on the state of the buffer memory.

Control transfers are ended by setting the CCPL bit to 1 with the PID bits in DCPCTR set to PID = BUF.

After the above settings have been entered, this module automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

1. For control read transfers

   This module sends a zero-length packet and receives an ACK response from the USB host.
2. For control write transfers and no-data control transfers

   The zero-length packet is received from the USB host, and this module sends an ACK response.

This module automatically responds to a normal SET_ADDRESS request. If any of the following errors occur in the SET_ADDRESS request, a response from software is necessary.

1. Any transfer other than a control read transfer: bmRequestType ≠ H'00
2. If a request error occurs: wIndex ≠ H'00
3. Any transfer other than a no-data control transfer: wLength ≠ H'00
4. If a request error occurs: wValue > H'7F
5. Control transfer of a device state error: DVSQ = B'011 (configured)

For all requests other than the SET_ADDRESS request, a response from the corresponding software is required.

(7) Bulk Transfers (PIPE1 to PIPE5)

The buffer memory specifications for bulk transfers (single/double buffer setting or continuous/non-continuous transfer mode setting) can be selected. The maximum size that can be set for the buffer memory is 2 Kbytes. The buffer memory state is controlled by this module, with a response sent automatically for a PING packet/NYET handshake.

(8) Interrupt Transfers (PIPE6 to PIPE9)

This module carries out interrupt transfers in accordance with the timing controlled by the host controller. For interrupt transfers, PING packets are ignored (no responses are sent), and the ACK, NAK, and STALL responses are returned without an NYET handshake response being made.

This module does not support high-bandwidth transfer for interrupt transfers.

(9) Isochronous Transfers (PIPE1 and PIPE2)

This module has the following functions pertaining to isochronous transfers.

1. Notification of isochronous transfer error information
2. Interval counter (specified by the IITV bits)
3. Isochronous-IN transfer data setup control (IDLY function)
4. Isochronous-IN transfer buffer flush function (specified by the IFIS bit)
5. SOF pulse output function

This module does not support high-bandwidth transfer for isochronous transfers.

(a) Interval Counter

The isochronous transfer interval can be set using the IITV bits in PIPEPERI. The interval counter enables the functions shown in table 14.33. When the host controller function is selected, this module generates the token issuance timing.

Table 14.33 Interval Counter Functions

Transfer Direction	Function	Conditions for Detection
IN	IN buffer flush function	When an IN token cannot be normally received in the interval frame during an isochronous-IN transfer
OUT	Notifies that a token is not being received	When an OUT token cannot be normally received in the interval frame during an isochronous-OUT transfer

Interval counting is carried out when an SOF packet is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF packet is corrupted. The frame interval that can be set is 2^{IITV} (μ) frames.

This module initializes the interval counter under the following conditions.

1. Hardware reset

   The IITV bits are initialized.
2. Buffer memory clearing using the ACLRM bit

   The IITV bits are not initialized but the counter value is. Setting the ACLRM bit to 0 starts counting from the value set in the IITV bits.

After the interval counter has been initialized, the interval counter is started under the following conditions 1 or 2 when a packet has been transferred normally.

1. An SOF packet is received following transmission of data in response to an IN token in the PID = BUF state.
2. An SOF packet is received after data following an OUT token is received in the PID = BUF state.

The interval counter is not initialized under the conditions noted below.

1. When the PID bits are set to NAK or STALL

   The interval counter does not stop. This module attempts the transactions at the subsequent interval.
2. At an USB bus reset or when the USB is suspended

   The IITV bits are not initialized. When an SOF packet is received, the interval counter is restarted from the value prior to the reception of the SOF packet.

(b) Setup of Data to be Transmitted using Isochronous Transfer

With isochronous data transmission using this module, after data has been written to the buffer memory, a data packet can be sent with the next frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

If a double buffer is used for the buffer memory, transmission will be enabled for only one of the two buffers even after the writing of data to both buffers has been completed. The enabled buffer memory is the one to which the data writing was completed first. For this reason, even if multiple IN tokens are received, the buffer memory that can be sent is only one packet's worth of data.

When an IN token is received, if the buffer memory is in the transmission enabled state, this module transmits the data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 14.11 shows an example of transmission using the isochronous transfer transmission data setup function with this module, when IITV = 0 (every frame) has been set. Sending of a zero-length packet is displayed in the figure as Null, in a shaded box.

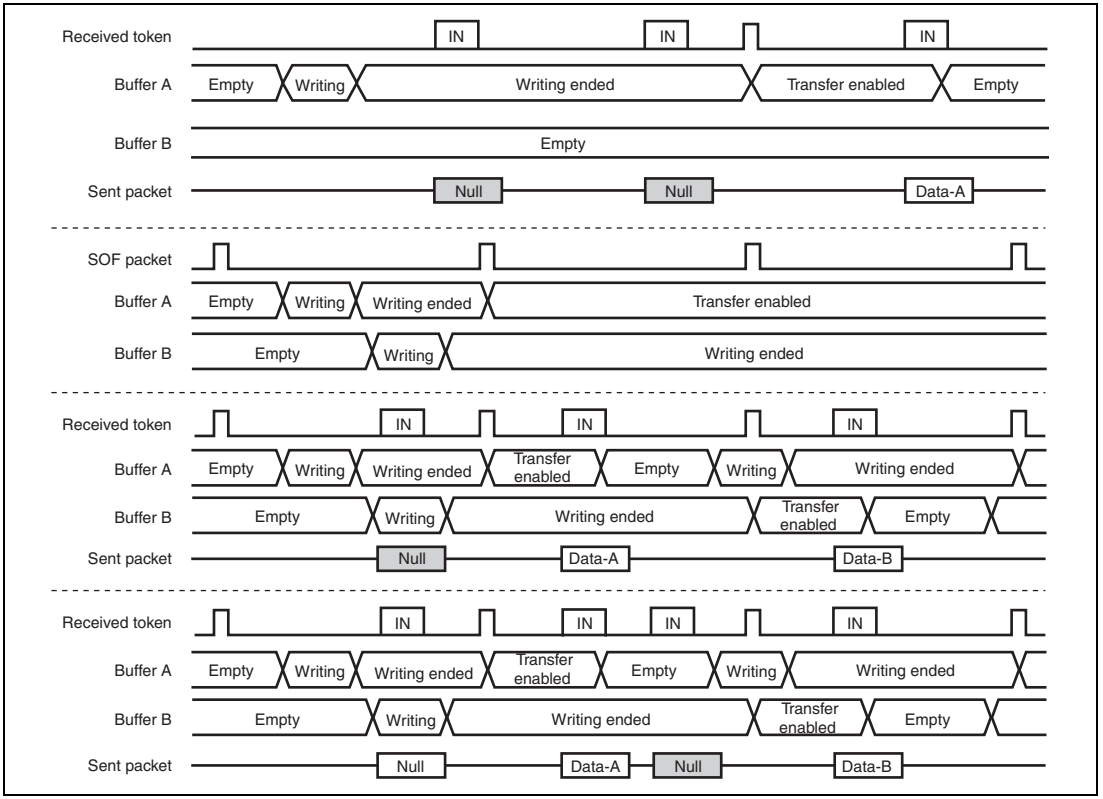


Figure 14.11 Example of Data Setup Function Operation

(c) Transmission Buffer Flush using Isochronous Transfer

If a (μ) SOF packet of the next frame is received without receiving an IN token in the interval frame during isochronous data transmission, this module operates as if the IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.

If a double buffer is being used and writing to both buffers has been completed, the buffer memory that was cleared is seen as the data having been sent at the same interval frame, and transmission is enabled for the buffer memory that is not discarded with (μ) SOF packet reception.

The timing for starting operation of the buffer flush function varies depending on the value set for the IITV bits.

- When IITV = 0

  The buffer flush operation starts from the next frame after the pipe becomes valid.

- In any cases other than IITV = 0
  The buffer flush operation is carried out subsequent to the first normal transaction.

Figure 14.12 shows an example of the buffer flush function of this module. When an unanticipated token is received prior to the interval frame, this module sends the written data or a zero-length packet as an underrun error, according to the data setup status.

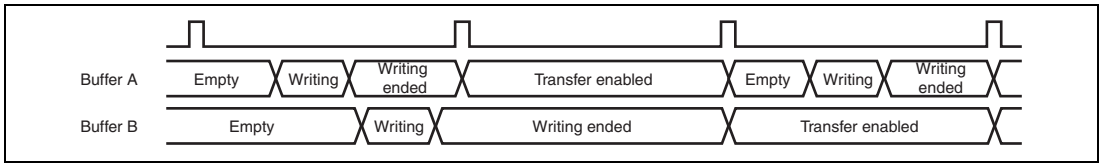


Figure 14.12 Example of Buffer Flush Function Operation

Figure 14.13 shows an example of this module generating an interval error. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by (1) in the figure, and the buffer flush function is activated.

If an interval error occurs during an IN transfer, the buffer flush function is activated; and if it occurs during an OUT transfer, an NRDY interrupt is generated.

The OVRN bit should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the figure, responses occur based on the buffer memory status.

- IN direction
  - If the buffer is in the transmission enabled state, the data is transferred as a normal response.
  - If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.
- OUT direction
  - If the buffer is in the reception enabled state, the data is received as a normal response.
  - If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

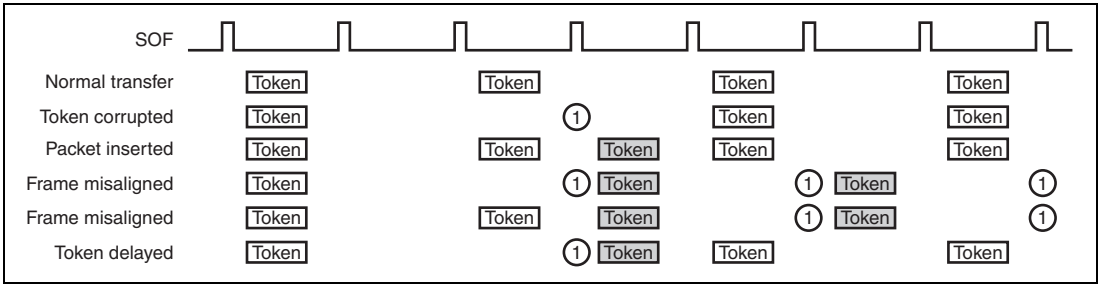


Figure 14.13 Example of an Interval Error Being Generated when IITV = 1

(10) SOF Interpolation Function

If data could not be received at intervals of 1 ms (when using full-speed operation) or 125 μs (when using high-speed operation) because an SOF packet was corrupted or missing, this module internally interpolates the SOF. The SOF interpolation operation begins when an SOF packet is received with USBE = 1 and SCKE = 1 set. The interpolation function is initialized under the following conditions.

- Hardware reset
- USB bus reset
- Suspended state detected

Also, the SOF interpolation operates under the following specifications.

- The frame interval (125 μs or 1 ms) conforms to the results of the reset handshake protocol.
- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, either 125 μs or 1 ms is counted with the internal clock of 48 MHz, and interpolation is carried out.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received. (With suspended transitions in high-speed operation, interpolation continues for 3 ms after the last packet is received.)

The SOF interpolation can also operate with the following functions.

- Refreshing of the frame number and the μ-frame number
- SOFR interrupt timing and μSOF lock
- SOF pulse output
- Isochronous transfer interval counter

If an SOF packet is missing during full-speed operation, the FRNM bits in FRMNUM0 are not refreshed. If a μSOF packet is missing during high-speed operation, the UFRNM bits in FRMNUM1 are refreshed. However, if a μSOF packet of μFRNM = B'000 is missing, the FRNM bits are not refreshed. In this case, the FRNM bits are not refreshed even if successive μSOF packets other than μFRNM = 000 are received normally.

(a) SOF Pulse Output

SOF pulses can be output from this module at the SOF timing when SOF output is enabled. When the SOFM bits in SOFCFG are set to B'01 (SOF output at every 1 ms) or B'10 (SOF output at every 125 μs), an active-low pulse, referred to as the SOF signal, is output from the SOF_N pin. For the pulse timing, see figure 14.14. SOF signals can be output at regular intervals due to reception of SOF packets and the SOF interpolation function.

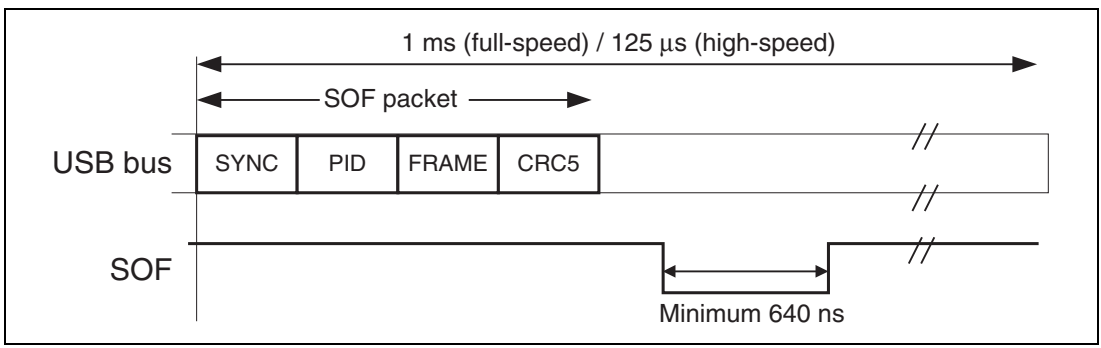


Figure 14.14 SOF Output Timing

Section 15 Direct Memory Access Controller 0 (DMAC0)

This LSI includes an on-chip direct memory access controller 0 (DMAC0). Instead of the CPU, the DMAC0 can be used to perform data transfers among external devices equipped with DACK (transfer request acceptance signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

15.1 Features

- Number of channels: Six channels (channels 0 to 3 can accept an external request)
- Address space: 4-Gbyte architecture
- Transfer data length: Byte, word (2 bytes), longword (4 bytes), 16 bytes, and 32 bytes
- Maximum transfer count: 16,777,216
- Transfer requests:

  Choice of external request (channels 0 to 3), on-chip peripheral module request, or auto-request

  The FLCTL can issue an on-chip peripheral module request
- Bus mode:

  Cycle steal mode (normal mode or intermittent mode)
- Data transfers:

  Normal mode

  Repeat mode

  Reload mode

  Multi-dimensional mode (multi-dimensional transfer, scatter/gather transfer, stride transfer)
- Priority: Selectable between fixed channel priority mode and round-robin mode
- Interrupt request: An interrupt request can be generated to the CPU after half of transfers have ended, all transfers have ended, or an address error has occurred
- External request detection: Choice of low/high level detection and rising/falling edge detection of DREQ input
- DMA transfer end notification signal: Active levels for DACK can be specified independently

Figure 15.1 shows a block diagram of the DMAC0.

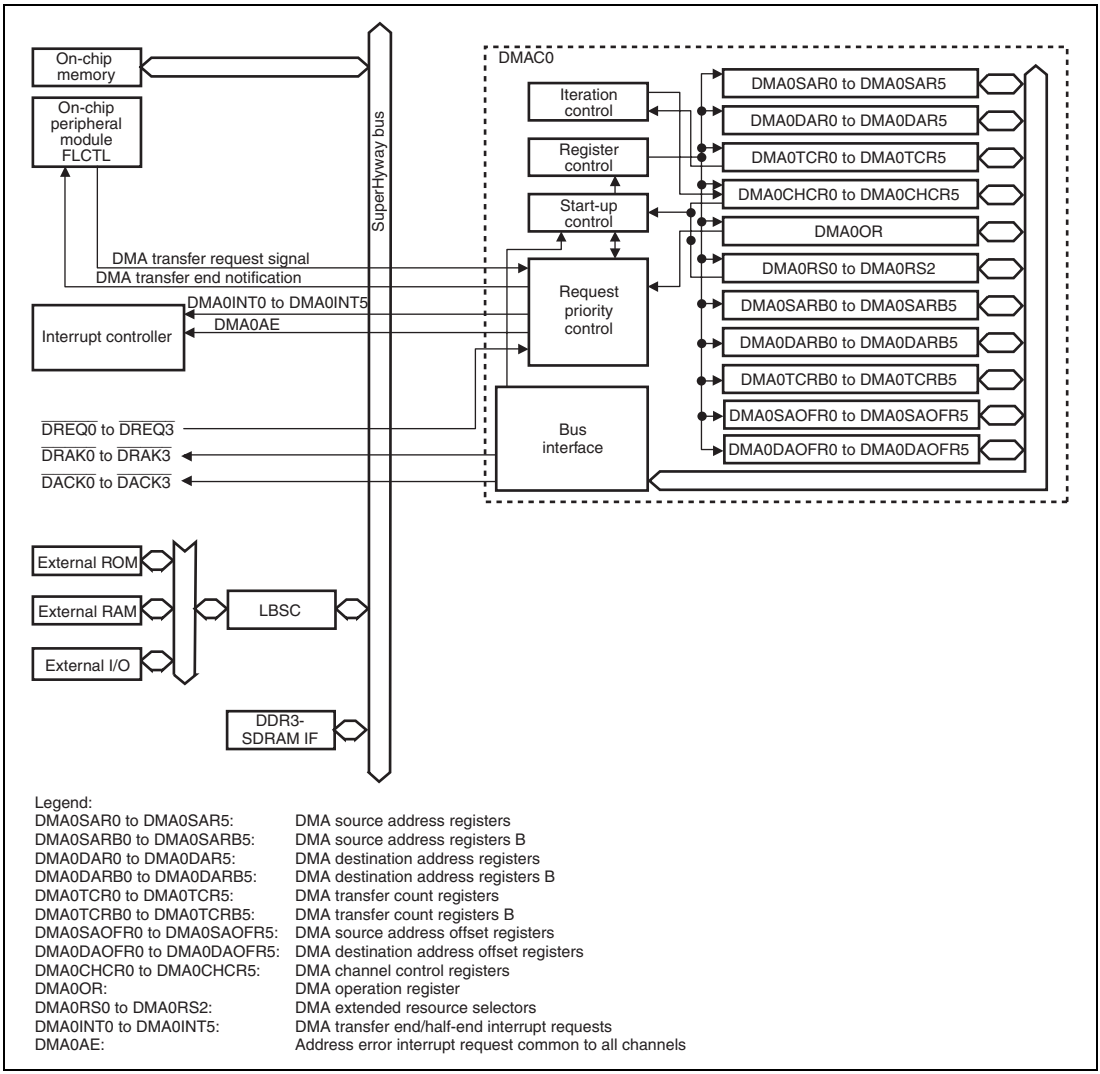


Figure 15.1 Block Diagram of DMAC0

15.2 Input/Output Pins

The DMAC0-related external pins are shown below.

Table 15.1 shows the configuration of the pins that are connected to external device. The DMAC0 has pins for four channels (channels 0 to 3) used in the external bus.

Table 15.1 Pin Configuration for the External Bus

Channel	Function	Pin Name	I/O	Description
0	DMA transfer request	$\overline{\text{DREQ0}}$*1	Input	DMA transfer request input from external device to channel 0
	DREQ0 acceptance confirmation	$\overline{\text{DRAK0}}$*2	Output	Notifies acceptance of DMA transfer request and start of execution from channel 0 to external device
	DMA transfer end notification	$\overline{\text{DACK0}}$*2	Output	Outputs strobe to DMA transfer request from channel 0 to external device
1	DMA transfer request	$\overline{\text{DREQ1}}$*1	Input	DMA transfer request input from external device to channel 1
	DREQ1 acceptance confirmation	$\overline{\text{DRAK1}}$*2	Output	Notifies acceptance of DMA transfer request and start of execution from channel 1 to external device
	DMA transfer end notification	$\overline{\text{DACK1}}$*2	Output	Outputs strobe to DMA transfer request from channel 1 to external device
2	DMA transfer request	$\overline{\text{DREQ2}}$*1	Input	DMA transfer request input from external device to channel 2
	DREQ2 acceptance confirmation	$\overline{\text{DRAK2}}$*2	Output	Notifies acceptance of DMA transfer request and start of execution from channel 2 to external device
	DMA transfer end notification	$\overline{\text{DACK2}}$*2	Output	Outputs strobe to DMA transfer request from channel 2 to external device
3	DMA transfer request	$\overline{\text{DREQ3}}$*1	Input	DMA transfer request input from external device to channel 3
	DREQ3 acceptance confirmation	$\overline{\text{DRAK3}}$*2	Output	Notifies acceptance of DMA transfer request and start of execution from channel 3 to external device.
	DMA transfer end notification	$\overline{\text{DACK3}}$*2	Output	Outputs strobe to DMA transfer request from channel 3 to external device

Notes: 1. The initial value is low-level detection.
2. The initial value is low-active.

15.3 Register Configuration

Table 15.2 shows the register configuration.

Table 15.2 Register Configuration of DMAC0 (1)

Channel	Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size*3	Sync Clock
0	DMA0 source address register 0	DMA0SAR0	R/W	H'FE808020	H'1E808020	32	Bck
	DMA0 destination address register 0	DMA0DAR0	R/W	H'FE808024	H'1E808024	32	Bck
	DMA0 transfer count register 0	DMA0TCR0	R/W	H'FE808028	H'1E808028	32	Bck
	DMA0 channel control register 0	DMA0CHCR0	R/W*1	H'FE80802C	H'1E80802C	32	Bck, Pck*4
1	DMA0 source address register 1	DMA0SAR1	R/W	H'FE808030	H'1E808030	32	Bck
	DMA0 destination address register 1	DMA0DAR1	R/W	H'FE808034	H'1E808034	32	Bck
	DMA0 transfer count register 1	DMA0TCR1	R/W	H'FE808038	H'1E808038	32	Bck
	DMA0 channel control register 1	DMA0CHCR1	R/W*1	H'FE80803C	H'1E80803C	32	Bck, Pck*4
2	DMA0 source address register 2	DMA0SAR2	R/W	H'FE808040	H'1E808040	32	Bck
	DMA0 destination address register 2	DMA0DAR2	R/W	H'FE808044	H'1E808044	32	Bck
	DMA0 transfer count register 2	DMA0TCR2	R/W	H'FE808048	H'1E808048	32	Bck
	DMA0 channel control register 2	DMA0CHCR2	R/W*1	H'FE80804C	H'1E80804C	32	Bck, Pck*4
3	DMA0 source address register 3	DMA0SAR3	R/W	H'FE808050	H'1E808050	32	Bck
	DMA0 destination address register 3	DMA0DAR3	R/W	H'FE808054	H'1E808054	32	Bck
	DMA0 transfer count register 3	DMA0TCR3	R/W	H'FE808058	H'1E808058	32	Bck
	DMA0 channel control register 3	DMA0CHCR3	R/W*1	H'FE80805C	H'1E80805C	32	Bck, Pck*4
0 to 5	DMA0 operation register	DMA0OR	R/W*2	H'FE808060	H'1E808060	16	Bck, Pck*5
4	DMA0 source address register 4	DMA0SAR4	R/W	H'FE808070	H'1E808070	32	Bck
	DMA0 destination address register 4	DMA0DAR4	R/W	H'FE808074	H'1E808074	32	Bck
	DMA0 transfer count register 4	DMA0TCR4	R/W	H'FE808078	H'1E808078	32	Bck
	DMA0 channel control register 4	DMA0CHCR4	R/W*1	H'FE80807C	H'1E80807C	32	Bck, Pck*4
5	DMA0 source address register 5	DMA0SAR5	R/W	H'FE808080	H'1E808080	32	Bck
	DMA0 destination address register 5	DMA0DAR5	R/W	H'FE808084	H'1E808084	32	Bck
	DMA0 transfer count register 5	DMA0TCR5	R/W	H'FE808088	H'1E808088	32	Bck
	DMA0 channel control register 5	DMA0CHCR5	R/W*1	H'FE80808C	H'1E80808C	32	Bck, Pck*4

Channel	Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size*[3]	Sync Clock
0	DMA0 source address register B0	DMA0SARB0	R/W	H'FE808120	H'1E808120	32	Bck
	DMA0 destination address register B0	DMA0DARB0	R/W	H'FE808124	H'1E808124	32	Bck
	DMA0 transfer count register B0	DMA0TCRB0	R/W	H'FE808128	H'1E808128	32	Bck
1	DMA0 source address register B1	DMA0SARB1	R/W	H'FE808130	H'1E808130	32	Bck
	DMA0 destination address register B1	DMA0DARB1	R/W	H'FE808134	H'1E808134	32	Bck
	DMA0 transfer count register B1	DMA0TCRB1	R/W	H'FE808138	H'1E808138	32	Bck
2	DMA0 source address register B2	DMA0SARB2	R/W	H'FE808140	H'1E808140	32	Bck
	DMA0 destination address register B2	DMA0DARB2	R/W	H'FE808144	H'1E808144	32	Bck
	DMA0 transfer count register B2	DMA0TCRB2	R/W	H'FE808148	H'1E808148	32	Bck
3	DMA0 source address register B3	DMA0SARB3	R/W	H'FE808150	H'1E808150	32	Bck
	DMA0 destination address register B3	DMA0DARB3	R/W	H'FE808154	H'1E808154	32	Bck
	DMA0 transfer count register B3	DMA0TCRB3	R/W	H'FE808158	H'1E808158	32	Bck
4	DMA0 source address register B4	DMA0SARB4	R/W	H'FE808170	H'1E808170	32	Bck
	DMA0 destination address register B4	DMA0DARB4	R/W	H'FE808174	H'1E808174	32	Bck
	DMA0 transfer count register B4	DMA0TCRB4	R/W	H'FE808178	H'1E808178	32	Bck
5	DMA0 source address register B5	DMA0SARB5	R/W	H'FE808180	H'1E808180	32	Bck
	DMA0 destination address register B5	DMA0DARB5	R/W	H'FE808184	H'1E808184	32	Bck
	DMA0 transfer count register B5	DMA0TCRB5	R/W	H'FE808188	H'1E808188	32	Bck
0	DMA0 source address offset register 0	DMA0SAOFR0	R/W	H'FE808220	H'1E808220	32	Bck
	DMA0 destination address offset register 0	DMA0DAOFR0	R/W	H'FE808224	H'1E808224	32	Bck
1	DMA0 source address offset register 1	DMA0SAOFR1	R/W	H'FE808230	H'1E808230	32	Bck
	DMA0 destination address offset register 1	DMA0DAOFR1	R/W	H'FE808234	H'1E808234	32	Bck
2	DMA0 source address offset register 2	DMA0SAOFR2	R/W	H'FE808240	H'1E808240	32	Bck
	DMA0 destination address offset register 2	DMA0DAOFR2	R/W	H'FE808244	H'1E808244	32	Bck
3	DMA0 source address offset register 3	DMA0SAOFR3	R/W	H'FE808250	H'1E808250	32	Bck
	DMA0 destination address offset register 3	DMA0DAOFR3	R/W	H'FE808254	H'1E808254	32	Bck

Channel	Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size*[3]	Sync Clock
4	DMA0 source address offset register 4	DMA0SAOFR4	R/W	H'FE808270	H'1E808270	32	Bck
	DMA0 destination address offset register 4	DMA0DAOFR4	R/W	H'FE808274	H'1E808274	32	Bck
5	DMA0 source address offset register 5	DMA0SAOFR5	R/W	H'FE808280	H'1E808280	32	Bck
	DMA0 destination address offset register 5	DMA0DAOFR5	R/W	H'FE808284	H'1E808284	32	Bck
0, 1	DMA0 extended resource selector 0	DMA0RS0	R/W	H'FE809000	H'1E809000	16	Pck
2, 3	DMA0 extended resource selector 1	DMA0RS1	R/W	H'FE809004	H'1E809004	16	Pck
4, 5	DMA0 extended resource selector 2	DMA0RS2	R/W	H'FE809008	H'1E809008	16	Pck

Notes: 1. To clear the flag, the HE and TE bits in DMA0CHCR can be read as 1, and then, 0 can be written to.
2. To clear the flag, the AE and NMIF bits in DMA0OR can be read as 1, and then, 0 can be written to.
3. Accessing with other access sizes is prohibited.
4. The synchronous clock for the HE and TE bits in DMA0CHCR is Bck, and the synchronous clock for the other bits in DMA0CHCR is Pck.
5. The synchronous clock for the AE, NMIF, and DME bits in DMA0OR is Bck, and the synchronous clock for the CMS and PR bits in DMA0OR is Pck.

Table 15.2 Register Configuration of DMAC0 (2)

Channel	Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiple Exception	Sleep/ Light Sleep by SLEEP instruction	Module Standby
0	DMA0 source address register 0	DMA0SAR0	Undefined	Retained	Retained	Retained
	DMA0 destination address register 0	DMA0DAR0	Undefined	Retained	Retained	Retained
	DMA0 transfer count register 0	DMA0TCR0	Undefined	Retained	Retained	Retained
	DMA0 channel control register 0	DMA0CHCR0	H'40000000	Retained	Retained	Retained
1	DMA0 source address register 1	DMA0SAR1	Undefined	Retained	Retained	Retained
	DMA0 destination address register 1	DMA0DAR1	Undefined	Retained	Retained	Retained
	DMA0 transfer count register 1	DMA0TCR1	Undefined	Retained	Retained	Retained
	DMA0 channel control register 1	DMA0CHCR1	H'40000000	Retained	Retained	Retained
2	DMA0 source address register 2	DMA0SAR2	Undefined	Retained	Retained	Retained
	DMA0 destination address register 2	DMA0DAR2	Undefined	Retained	Retained	Retained
	DMA0 transfer count register 2	DMA0TCR2	Undefined	Retained	Retained	Retained
	DMA0 channel control register 2	DMA0CHCR2	H'40000000	Retained	Retained	Retained
3	DMA0 source address register 3	DMA0SAR3	Undefined	Retained	Retained	Retained
	DMA0 destination address register 3	DMA0DAR3	Undefined	Retained	Retained	Retained
	DMA0 transfer count register 3	DMA0TCR3	Undefined	Retained	Retained	Retained
	DMA0 channel control register 3	DMA0CHCR3	H'40000000	Retained	Retained	Retained
0 to 5	DMA0 operation register	DMA0OR	H'0000	Retained	Retained	Retained
4	DMA0 source address register 4	DMA0SAR4	Undefined	Retained	Retained	Retained
	DMA0 destination address register 4	DMA0DAR4	Undefined	Retained	Retained	Retained
	DMA0 transfer count register 4	DMA0TCR4	Undefined	Retained	Retained	Retained
	DMA0 channel control register 4	DMA0CHCR4	H'40000000	Retained	Retained	Retained
5	DMA0 source address register 5	DMA0SAR5	Undefined	Retained	Retained	Retained
	DMA0 destination address register 5	DMA0DAR5	Undefined	Retained	Retained	Retained
	DMA0 transfer count register 5	DMA0TCR5	Undefined	Retained	Retained	Retained
	DMA0 channel control register 5	DMA0CHCR5	H'40000000	Retained	Retained	Retained
0	DMA0 source address register B0	DMA0SARB0	Undefined	Retained	Retained	Retained
	DMA0 destination address register B0	DMA0DARB0	Undefined	Retained	Retained	Retained
	DMA0 transfer count register B0	DMA0TCRB0	Undefined	Retained	Retained	Retained
1	DMA0 source address register B1	DMA0SARB1	Undefined	Retained	Retained	Retained
	DMA0 destination address register B1	DMA0DARB1	Undefined	Retained	Retained	Retained
	DMA0 transfer count register B1	DMA0TCRB1	Undefined	Retained	Retained	Retained

Channel	Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiple Exception	Sleep/ Light Sleep by SLEEP instruction	Module Standby
2	DMA0 source address register B2	DMA0SARB2	Undefined	Retained	Retained	Retained
	DMA0 destination address register B2	DMA0DARB2	Undefined	Retained	Retained	Retained
	DMA0 transfer count register B2	DMA0TCRB2	Undefined	Retained	Retained	Retained
3	DMA0 source address register B3	DMA0SARB3	Undefined	Retained	Retained	Retained
	DMA0 destination address register B3	DMA0DARB3	Undefined	Retained	Retained	Retained
	DMA0 transfer count register B3	DMA0TCRB3	Undefined	Retained	Retained	Retained
4	DMA0 source address register B4	DMA0SARB4	Undefined	Retained	Retained	Retained
	DMA0 destination address register B4	DMA0DARB4	Undefined	Retained	Retained	Retained
	DMA0 transfer count register B4	DMA0TCRB4	Undefined	Retained	Retained	Retained
5	DMA0 source address register B5	DMA0SARB5	Undefined	Retained	Retained	Retained
	DMA0 destination address register B5	DMA0DARB5	Undefined	Retained	Retained	Retained
	DMA0 transfer count register B5	DMA0TCRB5	Undefined	Retained	Retained	Retained
0	DMA0 source address offset register 0	DMA0SAOFR0	Undefined	Retained	Retained	Retained
	DMA0 destination address offset register 0	DMA0DAOFR0	Undefined	Retained	Retained	Retained
1	DMA0 source address offset register 1	DMA0SAOFR1	Undefined	Retained	Retained	Retained
	DMA0 destination address offset register 1	DMA0DAOFR1	Undefined	Retained	Retained	Retained
2	DMA0 source address offset register 2	DMA0SAOFR2	Undefined	Retained	Retained	Retained
	DMA0 destination address offset register 2	DMA0DAOFR2	Undefined	Retained	Retained	Retained
3	DMA0 source address offset register 3	DMA0SAOFR3	Undefined	Retained	Retained	Retained
	DMA0 destination address offset register 3	DMA0DAOFR3	Undefined	Retained	Retained	Retained
4	DMA0 source address offset register 4	DMA0SAOFR4	Undefined	Retained	Retained	Retained
	DMA0 destination address offset register 4	DMA0DAOFR4	Undefined	Retained	Retained	Retained
5	DMA0 source address offset register 5	DMA0SAOFR5	Undefined	Retained	Retained	Retained
	DMA0 destination address offset register 5	DMA0DAOFR5	Undefined	Retained	Retained	Retained
0, 1	DMA0 extended resource selector 0	DMA0RS0	H'0000	Retained	Retained	Retained
2, 3	DMA0 extended resource selector 1	DMA0RS1	H'0000	Retained	Retained	Retained
4, 5	DMA0 extended resource selector 2	DMA0RS2	H'0000	Retained	Retained	Retained

15.4 Register Descriptions

15.4.1 DMA0 Source Address Registers 0 to 5 (DMA0SAR0 to DMA0SAR5)

DMA0SAR are 32-bit readable/writable registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the source address of the next transfer.

A word or longword boundary address should be specified when a word or longword transfer is performed respectively. A 16-byte or 32-byte boundary value should be specified when a 16-byte or 32-byte transfer is performed respectively.

The initial values of DMA0SAR are undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.4.2 DMA0 Source Address Registers B0 to B5 (DMA0SARB0 to DMA0SARB5)

DMA0SARB are 32-bit readable/writable registers.

In repeat/reload mode, DMA0SARB specify the source address of a DMA transfer that is set in DMA0SAR again.

In multi-dimensional mode, DMA0SARB are used by the DMAC0 as next-page source address hold registers.

The data written to DMA0SAR by the CPU is also written to DMA0SARB. To set the address that is different from the DMA0SAR address, write data to DMA0SAR, then, to DMA0SARB.

A word or longword boundary address should be specified when a word or longword transfer is performed respectively. A 16-byte or 32-byte boundary value should be specified when a 16-byte or 32-byte transfer is performed respectively.

The initial values of DMA0SARB are undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.4.3 DMA0 Destination Address Registers 0 to 5 (DMA0DAR0 to DMA0DAR5)

DMA0DAR are 32-bit readable/writable registers that specify the destination address of a DMA transfer. During a DMA transfer, these registers indicate the destination address of the next transfer.

A word or longword boundary address should be specified when a word or longword transfer is performed respectively. A 16-byte or 32-byte boundary value should be specified when a 16-byte or 32-byte transfer is performed respectively.

The initial values of DMA0DAR are undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.4.4 DMA0 Destination Address Registers B0 to B5 (DMA0DARB0 to DMA0DARB5)

DMA0DARB are 32-bit readable/writable registers.

In repeat/reload mode, DMA0DARB specify the destination address of a DMA transfer that is set in DMA0SAR again.

In multi-dimensional mode, DMA0DARB are used by the DMAC0 as next-page destination address hold registers.

The data written to DMA0DAR by the CPU is also written to DMA0DARB. To set the address that is different from the DMA0DAR address, write data to DMA0DAR, then, to DMA0DARB.

A word or longword boundary address should be specified when a word or longword transfer is performed respectively. A 16-byte or 32-byte boundary value should be specified when a 16-byte or 32-byte transfer is performed respectively.

The initial values of DMA0DARB are undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.4.5 DMA0 Transfer Count Registers 0 to 5 (DMA0TCR0 to DMA0TCR5)

DMA0TCR are 32-bit readable/writable registers that specify the DMA transfer count. When the value is set to H'00000001, H'00FFFFFF, H'00000000, the transfer count is 1, 16,777,215, and 16,777,216 (the maximum) respectively. During a DMA transfer, these registers indicate the remaining transfer count.

The upper eight bits in DMA0TCR (bits 31 to 24) are always read as 0. The write value should always be 0.

The initial values of DMA0TCR are undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—								
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.4.6 DMA0 Transfer Count Registers B0 to B5 (DMA0TCRB0 to DMA0TCRB5)

DMA0TCRB are 32-bit readable/writable registers. The data written to DMA0TCR by the CPU is also written to DMA0TCRB.

When the half-end function is used, DMA0TCRB are used as initial value hold registers for half-end detection.

Also, DMA0TCRB specify the number of DMA transfers which are set in DMA0TCR again in repeat mode. In repeat mode, write 0 to bits DMA0TCRB[31:24].

DMA0TCRB specify the number of DMA transfers and are used as transfer count counters in reload mode and multi-dimensional mode. Bits DMA0TCRB[15:0] operate as transfer count counters. When the values are 0, values of DMA0SAR and DMA0DAR are updated, and the value of bits DMA0TCRB[31:16] is loaded to bits DMA0TCRB[15:0]. Set the number of transfers until reloading starts to bits DMA0TCRB[31:16]. In reload mode and multi-dimensional mode, a value from H'FFFF (65635 times) to H'0001 (1 time) can be specified in bits DMA0TCRB[31:16] and DMA0TCRB[15:0], and set the same number to bits DMA0TCRB[31:16] and DMA0TCRB[15:0]. Also, clear the HIE bit in DMA0CHCR to 0 and do not use the half-end function in reload mode or multi-dimensional mode.

The initial values of DMA0TCRB are undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.4.7 DMA0 Source Address Offset Registers 0 to 5 (DMA0SAOFR0 to DMA0SAOFR5)

DMA0SAOFR are 32-bit readable/writable registers used when multi-dimensional mode is set. The settings of DMA0SAOFR are valid only when multi-dimensional mode is selected (RPT[3:0] in DMA0CHCR = B'1101 or B'1111).

Bits DMA0SAOFR[15:0] set the amount by which the transfer source address is increased with each transfer.

Bits DMA0SAOFR[31:16] specify the address offset used when bits DMA0SAOFR[15:0] is 0 and DMA0SAR is reloaded.

Make settings to bits DMA0SAOFR[15:0] and DMA0SAOFR[31:16] such that the resulting addresses match the transfer size and address boundary.

For details on multi-dimensional mode, refer to section 15.5.7, Multi-Dimensional Mode Transfer.

The initial values of DMA0SAOFR are undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.4.8 DMA0 Destination Address Offset Registers 0 to 5 (DMA0DAOFR0 to DMA0DAOFR5)

DMA0DAOFR are 32-bit readable/writable registers used when multi-dimensional mode is set. The settings of DMA0DAOFR are valid only when multi-dimensional mode is selected (RPT[3:0] in DMA0CHCR = B'1101 or B'1110).

Bits DMA0DAOFR[15:0] set the amount by which the transfer destination address is increased with each transfer.

Bits DMA0DAOFR[31:16] specify the address offset used when bits DMA0DAOFR[15:0] is 0 and DMA0DAR is reloaded.

Make settings to bits DMA0DAOFR[15:0] and DMA0DAOFR[31:16] such that the resulting addresses match the transfer size and address boundary.

For details on multi-dimensional mode, refer to section 15.5.7, Multi-Dimensional Mode Transfer.

The initial values of DMA0DAOFR are undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.4.9 DMA0 Channel Control Registers 0 to 5 (DMA0CHCR0 to DMA0CHCR5)

DMA0CHCR are 32-bit readable/writable registers that control the DMA transfer mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	LCKN	—	RPT[3:0]				—	DO	RL	—	TS[2]	HE	HIE	AM	AL
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R/W	R/(W)*	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM[1:0]		SM[1:0]		RS[3:0]				DL	DS	TB	TS[1:0]		IE	TE	DE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

Note: * R/(W): To clear the flag, 0 can be written to. Writing 1 does not affect the value of the flag.

Bit	Bit Name	Initial Value	R/W	Descriptions
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30	LCKN	1	R/W	Bus Lock Signal Disable 0: Bus lock signal output enabled 1: Bus lock signal output disabled (initial value) Note: This bit can be cleared to 0 only when PCMCIA ATA complement mode with DACKBST enabled is selected as the external request mode. When setting this bit to 1, also set the TB bit in DMA0CHCR to 1. Channels should be used from the channel with the highest priority. For details on PCMCIA ATA complement mode, refer to section 11, Bus State Controller (LBSC). For channels 4 and 5, do not clear this bit to 0. The write value should always be 1.
29	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
28 to 25	RPT[3:0]	0000	R/W	DMA Setting Update Specification 0000: Normal mode 0001: Repeat mode DMA0SAR/DMA0DAR/DMA0TCR repeated 0010: Repeat mode DMA0DAR/DMA0TCR repeated 0011: Repeat mode DMA0SAR/DMA0TCR repeated 0101: Reload mode DMA0SAR/DMA0DAR reloaded according to the DMA0TCRB setting 0110: Reload mode DMA0DAR reloaded according to the DMA0TCRB setting 0111: Reload mode DMA0SAR reloaded according to the DMA0TCRB setting 1101: Multi-dimensional mode DMA0SAR address updated by the value specified in bits DMA0SAOFR[15:0] and DMA0SAOFR[31:16], DMA0DAR address updated by the value specified in bits DMA0DAOFR[15:0] and DMA0DAOFR[31:16], and the DM and SM settings ignored 1110: Multi-dimensional mode DMA0DAR address updated by the value specified in bits DMA0DAOFR[15:0] and DMA0DAOFR[31:16] and the DM setting ignored 1111: Multi-dimensional mode DMA0SAR address updated by the value specified in bits DMA0SAOFR[15:0] and DMA0SAOFR[31:16] and the SM setting ignored Other than above: Setting prohibited
24	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
23	DO	0	R/W	DMA Overrun Selects whether DREQ is detected by overrun 0 or by overrun 1. This bit is valid in only DMA0CHCR0 to DMA0CHCR3. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1
22	RL	0	R/W	Request Check Level Selects whether the DRAK signal output is an active-high or active-low. This bit is valid in only DMA0CHCR0 to DMA0CHCR3. If the DRAK active direction is changed, reflecting the change on the external pins requires one cycle of the external bus clock after writing to the register is completed. 0: DRAK is an active-low output 1: DRAK is an active-high output
21	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
20	TS[2]	0	R/W	DMA Transfer Size Specification Specifies the DMA transfer size with bits TS[1:0]. When the transfer source or transfer destination is a register in an on-chip peripheral module register that the access size is specified, the transfer data size for the register should be the same as the access size. For the address set to DMA0SAR or DMA0DAR as transfer source or transfer destination, the transfer data size should be the same as the address boundary. TS[2], TS[1], TS[0] 000: Byte units 001: Word (2-byte) units 010: Longword (4-byte) units 011: 16-byte units 100: 32-byte units Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Descriptions
19	HE	0	R/(W)*	Half-End Flag After HIE (bit 18) is set to 1 and the number of transfers is half of the value of DMA0TCR (the value after a 1-bit right shift), which is set before transfer, the HE bit becomes 1. • HE is set to 1 when the number of transfers is an even number ((DMA0TCR setting before transfer)/2) • HE is set to 1 when the number of transfers is an odd number ((DMA0TCR setting before transfer − 1)/2) • HE is set to 1 when the number of transfer is the maximum transfer count 8,388,608 (H'00800000) The HE bit is not set when transfers are ended by an NMI interrupt or address error, or by clearing the DE or DME bit in DMA0OR before the number of transfers is decreased to half of the DMA0TCR value set before the transfer. The HE bit is kept set when the transfer ends by an NMI interrupt or address error, or clearing the DE bit (bit 0) or the DME bit in DMA0OR after the HE bit is set to 1. To clear the HE bit, write 0 after reading 1 from the HE bit. 0: DMA transfer in progress or DMA transfer has been aborted DMA0TCR > (DMA0TCR setting before transfer)/2 To clear the HE bit, write 0 only after the HE bit is read as 1. When the HE bit is read as 0 or not to clear this bit, write 1 to the HE bit. Writing 1 to the HE bit does not affect the value of this bit. 1: DMA0TCR = (DMA0TCR setting before transfer)/2

Bit	Bit Name	Initial Value	R/W	Descriptions
18	HIE	0	R/W	Half-End Interrupt Enable Specifies whether an interrupt request is generated to the CPU when the read cycle of the transfer that the number of transfers is decreased to half of the DMA0TCR value set before the transfer has ended. If the HIE bit is set to 1, an interrupt request is generated to the CPU when the HE bit is set. To confirm that the half of the transfer has ended, execute a dummy read of the destination space after issuing the SYNCO instruction. Clear this bit to 0 while reload mode or multi-dimensional mode is set. 0: Half-end interrupt disabled 1: Half-end interrupt enabled
17	AM	0	R/W	Acknowledge Mode Selects whether DACK is output in a data read cycle or in a data write cycle. DACK is output only for LBSC space transfers. This bit is valid in only DMA0CHCR0 to DMA0CHCR3. 0: DACK output in a read cycle (DACK is output only when the DMA transfer source is LBSC space.) 1: DACK output in a write cycle (DACK is output only when the DMA transfer destination is LBSC space.)
16	AL	0	R/W	Acknowledge Level Specifies whether the DACK signal output is high-active or low-active. This bit is valid in only DMA0CHCR0 to DMA0CHCR3. If DACK active direction has been changed, reflecting the change on the external pins requires two cycles of the external bus clock after writing to register is completed. 0: DACK output low-active 1: DACK output high-active

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	DM[1:0]	00	R/W	Destination Address Mode 1, 0 Specify whether the DMA destination address is incremented or decremented. 00: Destination address is fixed 01: Destination address is incremented Byte unit transfer: +1 Word unit transfer: +2 Longword unit transfer: +4 16-byte unit transfer: +16 32-byte unit transfer: +32 10: Destination address is decremented Byte unit transfer: −1 Word unit transfer: −2 Longword unit transfer: −4 Setting prohibited in 16/32-byte unit transfer 11: Setting prohibited For any setting (00, 01, or 10), specifying a transfer size greater than the bus width divides bus cycles into two or more, and increases the number of addresses for the divided bus cycles. When RPT[3:0] = B'1101 or B'1110 is set, this setting is invalid and update is performed based on the DMA0DAOFR[15:0] bit setting.

Bit	Bit Name	Initial Value	R/W	Descriptions
13, 12	SM[1:0]	00	R/W	Source Address Mode 1, 0 Specify whether the DMA source address is incremented or decremented. 00: Source address is fixed 01: Source address is incremented Byte unit transfer: +1 Word unit transfer: +2 Longword units transfer: +4 16-byte unit transfer: +16 32-byte unit transfer: +32 10: Source address is decremented Byte unit transfer: −1 Word unit transfer: −2 Longword unit transfer: −4 Setting prohibited in 16/32-byte unit transfer 11: Setting prohibited For any setting (00, 01, or 10), specifying a transfer size greater than the bus width divides bus cycles into two or more, and increases the number of addresses for the divided bus cycles. When RPT[3:0] = B'1101 or B'1110 is set, this setting is invalid and update is performed based on the DMA0SAOFR[15:0] bit setting.
11 to 8	RS[3:0]	0000	R/W	Resource Select 3 to 0 Specify the transfer request source. To change the transfer request source, the DMA enable (DE) bit should be cleared to 0. 0000: External request or dual address mode 0100: Auto-request 1000: On-chip peripheral module request Selected by DMA0 extended resource selector (DMA0RS0 to DMA0RS2) Other than above: Setting prohibited Note: External request specification is valid in only DMA0CHCR0 to DMA0CHCR3. The external request cannot be specified in DMA0CHCR4 and DMA0CHCR5.

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 input and the detecting level. These bits are valid in only DMA0CHCR0 to DMA0CHCR3. Even in channels 0 to 3, 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 (DREQ) 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 for DMA transfers. 0: Cycle steal mode 1: Burst mode Note: This bit can be set to 1 only when PCMCIA ATA complement mode with DACKBST enabled is selected as the external request mode. When setting this bit to 1, also set the LCKN bit in DMA0CHCR to 1. Channels should be used from the channel with the highest priority. For details on PCMCIA ATA complement mode, refer to section 11, Local Bus State Controller (LBSC). For channels 4 and 5, do not set this bit to 1. The write value should always be 0.
4, 3	TS[1:0]	00	R/W	DMA Transfer Size Specification See the description of TS[2] (bit 20).
2	IE	0	R/W	Interrupt Enable Specifies whether an interrupt request is generated to the CPU at the end of the final DMA transfer. Setting this bit to 1 generates an interrupt request (DMA0INT) to the CPU when the TE bit is set to 1 and a read cycle of the final DMA transfer has ended. To confirm that the final transfer has ended, execute a dummy read of the destination space after issuing the SYNCO instruction. 0: Interrupt request disabled 1: Interrupt request enabled

Bit	Bit Name	Initial Value	R/W	Descriptions
1	TE	0	R/(W)*	Transfer End Flag The TE bit is set to 1 when the DMA0 transfer count register (DMA0TCR) is set to 0 (when the DMAC0 starts executing the final DMA transfer). The TE bit is not set, if DMA transfer ends due to an NMI interrupt or DMA address error before DMA0TCR is cleared to 0, or if DMA transfer is ended by clearing the DE bit and DME bit in the DMA0 operation register (DMA0OR). To clear the TE bit, the TE bit should be read as 1, and then, 0 is written to. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: DMA transfer in progress or DMA transfer has been aborted To clear the TE bit, write 0 only after the TE bit is read as 1. When the TE bit is read as 0 or not to clear the bit, write 1 to the TE bit. Writing 1 to the TE bit does not affect the value of this bit. 1: DMA0TCR = 0 (when the final DMA transfer is being performed or the DMA transfer ends)
0	DE	0	R/W	DMA Enable Enables or disables DMA transfer. In auto-request mode, DMA transfer starts by setting the DE bit and the DME bit in DMA0OR to 1. The TE, NMIF, and AE bits in DMA0OR should be 0. In an external request or on-chip peripheral module request, DMA transfer starts if a DMA transfer request is generated by the corresponding devices or corresponding peripheral modules after the DE and DME bits are set to 1. In this case, too, the TE, NMIF, and AE bits should be 0. Clearing the DE bit to 0 can abort DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

Note: * To clear the flag, 0 can be written to. Writing 1 does not affect the value of the flag.

15.4.10 DMA0 Operation Register (DMA0OR)

DMA0OR is a 16-bit readable/writable register that specifies the priority of channels in DMA transfer. Also, this register shows the DMA transfer status.

DMA0OR is a register common to all channels.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CMS[1:0]		—	—	PR[1:0]		—	—	—	—	—	AE	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R/(W)*	R/(W)*	R/W

Note: * To clear the flag, 0 can be written to. Writing 1 does not affect the value of the flag.

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	CMS[1:0]	00	R/W	Cycle Steal Mode Select 1, 0 Select normal mode or intermittent mode in cycle steal mode. To validate intermittent mode, the bus mode for all channels should be cycle steal mode. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Executes a DMA transfer after waiting 16 Bck clock of the external clock 11: Intermittent mode 64 Executes a DMA transfer after waiting 64 Bck clock of the external clock For details, see section 15.5.3 (1) (b), Intermittent Mode 16, Intermittent Mode 64.

Bit	Bit Name	Initial Value	R/W	Descriptions
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PR[1:0]	00	R/W	Priority Mode 1, 0 Determine the priority 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 for CH0 to CH5 When round-robin mode is specified, only cycle steal mode (DMA0CHCR.LCKN = 1, DMA0CHCR.TB = 0) can be set for all channels.
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	AE	0	R/(W)*	Address Error Flag Indicates that an address error occurred during DMA transfer. This bit is set under the following conditions. • Value set in DMA0SAR or DMA0DAR does not match the transfer size boundary • Transfer source or transfer destination is in an undefined space on the address map • Transfer source or transfer destination is in module stop mode When the AE bit in DMA0OR is set to 1, DMA transfers are disabled even if the DE bit in DMA0CHCR of all channels and the DME bit in DMA0OR are set to 1. 0: No DMAC0 address error To clear the AE bit, write 0 only after AE bit is read as 1. When the AE bit read as 0 or not to clear the bit, write 1 to the AE bit. Writing 1 to the AE bit does not affect the value of this bit. 1: Address error occurs during DMA transfer

Bit	Bit Name	Initial Value	R/W	Descriptions
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 DMA0CHCR and the DME bit in DMA0OR are set to 1. When an NMI is input, DMA transfer stops. After returning from the NMI interrupt processing, restart a transfer after setting again on all channels. When the DMAC0 does not operate, the NMIF bit is set to 1 even if the NMI interrupt is input. 0: No NMI interrupt To clear the NMIF bit, write 0 only after the NMIF bit is read as 1. When the NMIF bit is read as 0 or not to clear the bit, write 1 to the NMIF bit. Writing 1 to the NMIF bit does not affect the value of this bit. 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 DMA0CHCR are set to 1, transfer is enabled. In this case however the TE bit in DMA0CHCR of the channel that executes transfer, and the NMIF and AE bits in DMA0OR that corresponds to the channel should all be 0. If the DME bit is cleared to 0, transfers on all channels are aborted. If an on-chip peripheral module request is set in any channel among channels 0 to 5, clearing the DME bit to 0 to abort DMA transfer must be performed after the DMA transfer request from the relevant on-chip peripheral module has been cleared. 0: DMA transfers on all channels are disabled 1: DMA transfers on all channels are enabled

Note: * To clear the flag, 0 can be written to. Writing 1 does not affect the value of the flag.

15.4.11 DMA0 Extended Resource Selectors 0 to 2 (DMA0RS0 to DMA0RS2)

DMA0RS are 16-bit readable/writable registers. DMA0RS0, DMA0RS1, and DMA0RS2 specify DMA transfer request sources from peripheral modules for channels 0 and 1, channels 2 and 3, and channels 4 and 5, respectively. These registers can specify a transfer request from the FLCTL.

When MID/RID other than the values listed in table 15.3 is specified, the operation of this LSI is not guaranteed. The transfer request from DMA0RS is valid only when the resource select bits RS3 to RS0 in DMA0CHCR have been set to B'1000. When the bits are not set to B'1000, the transfer request source is not accepted even if DMA0RS has been set.

- DMA0RS0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C1MID[5:0]						C1RID[1:0]		C0MID[5:0]						C0RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMA0RS1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C3MID[5:0]						C3RID[1:0]		C2MID[5:0]						C2RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMA0RS2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C5MID[5:0]						C5RID[1:0]		C4MID[5:0]						C4RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMA0RS0

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C1MID[5:0]	000000	R/W	Transfer request source module ID5 to ID0 (MID) for channel 1. See table 15.3.
9, 8	C1RID[1:0]	00	R/W	Transfer request source register ID1 and ID0 (RID) for channel 1. See table 15.3.
7 to 2	C0MID[5:0]	000000	R/W	Transfer request source module ID5 to ID0 (MID) for channel 0. See table 15.3.
1, 0	C0RID[1:0]	00	R/W	Transfer request source register ID1 and ID0 (RID) for channel 0. See table 15.3.

- DMA0RS1

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C3MID[5:0]	000000	R/W	Transfer request source module ID5 to ID0 (MID) for channel 3. See table 15.3.
9, 8	C3RID[1:0]	00	R/W	Transfer request source register ID1 and ID0 (RID) for channel 3. See table 15.3.
7 to 2	C2MID[5:0]	000000	R/W	Transfer request source module ID5 to ID0 (MID) for channel 2. See table 15.3.
1, 0	C2RID[1:0]	00	R/W	Transfer request source register ID1 and ID0 (RID) for channel 2. See table 15.3.

- DMA0RS2

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C5MID[5:0]	000000	R/W	Transfer request source module ID5 to ID0 (MID) for channel 5. See table 15.3.
9, 8	C5RID[1:0]	00	R/W	Transfer request source register ID1 and ID0 (RID) for channel 5. See table 15.3.
7 to 2	C4MID[5:0]	000000	R/W	Transfer request source module ID5 to ID0 (MID) for channel 4. See table 15.3.
1, 0	C4RID[1:0]	00	R/W	Transfer request source register ID1 and ID0 (RID) for channel 4. See table 15.3.

Table 15.3 List of Transfer Request Sources

Peripheral Module	Setting for One Channel (MID and RID Fields)	MID	RID	Function
FLCTL	H'83	B'100000	B'11	Transmit/receive of data part
	H'87	B'100001	B'11	Transmit/receive of management code part

15.5 Operation

When DMA transfer is requested, the DMAC0 starts transfer according to the determined channel priority. When the transfer end conditions are satisfied, the DMAC0 ends transfer. Transfer requests have three modes: auto-request mode, external request mode, and on-chip peripheral module request mode.

15.5.1 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or data transfer destination, but they can also be generated in external devices or on-chip peripheral modules that are neither the transfer source nor the transfer destination.

Transfers can be requested in three modes: auto-request, external request, and on-chip peripheral module request. The transfer request is selected by bits RS3 to RS0 in DMA0CHCR and DMA0RS0 to DMA0RS2, according to DMA channels.

(1) Auto-Request Mode

Auto-request mode is a mode that automatically generates a transfer request signal in the DMAC0 when there is no transfer request signal from an external source, like memory-to-memory transfer or a transfer between memory and an on-chip peripheral module that cannot generate a transfer request. When the DE bit in DMA0CHCR for each channel and the DME bit in DMA0OR common to all channels are set to 1, transfers are started. Note however that the AE and NMIF bits in DMA0OR should both be 0.

(2) External Request Mode

External request mode is a mode that starts transfer by the transfer request signal ($\overline{\text{DREQ0}}$ to $\overline{\text{DREQ3}}$) from the external device of this LSI. This mode is valid in only channels 0 to 3. Table 15.4 shows the external request mode settings. While DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), DMA transfer starts when DREQ is input.

Table 15.4 External Request Mode Setting with RS Bits

DMA0CHCR						
RS3	**RS2**	**RS1**	**RS0**	**Address Mode**	**Transfer Source**	**Transfer Destination**
0	0	0	0	Dual address mode	Any	Any

Choose whether DREQ is detected by edge or level with the DREQ level (DL) bit and DREQ select (DS) bit in DMA0CHCR0 to DMA0CHCR3 shown in table 15.5. The source of the transfer request does not have to be the transfer source or transfer destination.

Table 15.5 External Request Detection Selection with DL and DS Bits

DMA0CHCR			
DL	**DS**	**Detection of External Request**	
0	0	Low level detection	(initial value: $\overline{\text{DREQ}}$)
	1	Falling edge detection	
1	0	High level detection	
	1	Rising edge detection	

When DREQ is accepted, the DREQ pin cannot accept requests. After acknowledge DACK is output to the accepted DREQ, the DREQ pin can accept requests again.

When DREQ is used for level detection, the timing to detect the next DREQ after outputting DACK depends on the DO bit in DMA0CHCR.

For details, see section 15.5.8, DREQ Pin Sampling Timing.

Table 15.6 Selecting External Request Detection with DO Bit

DMA0CHCR		
DO	**External Request**	
0	Overrun 0	(initial value)
1	Overrun 1	

DACK can be output to only LBSC space, and is output at the same timing as $\overline{\text{CSn}}$. The setting whether DACK is output during the reading or writing cycle is selected by the AM bit in DMA0CHCR shown in table 15.7.

Table 15.7 Acknowledge Mode Selection with AM Bit

DMA0CHCR		
AM	**External Request**	
0	DACK output during the reading cycle	(initial value)
1	DACK output during the writing cycle	

(3) On-Chip Peripheral Module Request Mode

On-chip peripheral module request mode is a mode that performs transfer by a DMA transfer request signal from an on-chip peripheral module. The DMA transfer request signal is a transfer request from the FLCTL which is set by DMA0RS0 to DMA0RS2.

If the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0) in on-chip peripheral module request mode, a transfer is performed by a transfer request signal.

When a transmit FIFO data empty request of the FLCTL is specified as the transfer request, the transfer destination must be the FLDTFIFO register of the FLCTL. Likewise, when a receive FIFO data full request of the FLCTL is specified as the transfer request, the transfer source must be the FLDTFIFO register of the FLCTL.

Table 15.8 shows the settings required to select the on-chip peripheral module request mode.

Table 15.8 List of On-Chip Peripheral Module Request Modes

DMA0CHCR	DMA0RS		DMA Transfer	DMA Transfer			Bus
RS[3:0]	**MID**	**RID**	**Request Source**	**Request Signal**	**Source**	**Destination**	**Mode**
1000	100000	11	FLCTL data part transmitter	Transmit FIFO data empty request	Any	FLDTFIFO	Cycle steal
			FLCTL data part receiver	Receive FIFO data full request	FLDTFIFO	Any	Cycle steal
	100001	11	FLCTL management code part transmitter	Transmit FIFO data empty request	Any	FLECFIFO	Cycle steal
			FLCTL management code part receiver	Receive FIFO data full request	FLECFIFO	Any	Cycle steal

15.5.2 Channel Priority

When the DMAC0 receives transfer requests on two or more channels simultaneously, it transfers data according to a determined priority. The mode is chosen between fixed mode and round-robin mode by the PR1 and PR0 bits in DMA0OR.

(1) Fixed Mode

In fixed mode, the channel priority does not change. There are two kinds of fixed modes as shown below. These are selected by bits PR1 and PR0 in DMA0OR.

- DMA0OR.PR = 00: CH0 > CH1 > CH2 > CH3 > CH4 > CH5
- DMA0OR.PR = 01: CH0 > CH2 > CH3 > CH1 > CH4 > CH5

(2) Round-Robin Mode

In round-robin mode, each time data of one transfer unit (byte, word, longword, 16-byte, or 32-byte unit) is transferred on one channel, the channel on which the transfer has just ended is the bottom of the priority. Figure 15.2 shows the round-robin mode operation. The priority of round-robin mode immediately after reset is CH0 > CH1 > CH2 > CH3 > CH4 > CH5.

When round-robin mode is specified, only cycle steal mode (DMA0CHCR.LCKN = 1, DMA0CHCR.TB = 0) can be set for all channels.

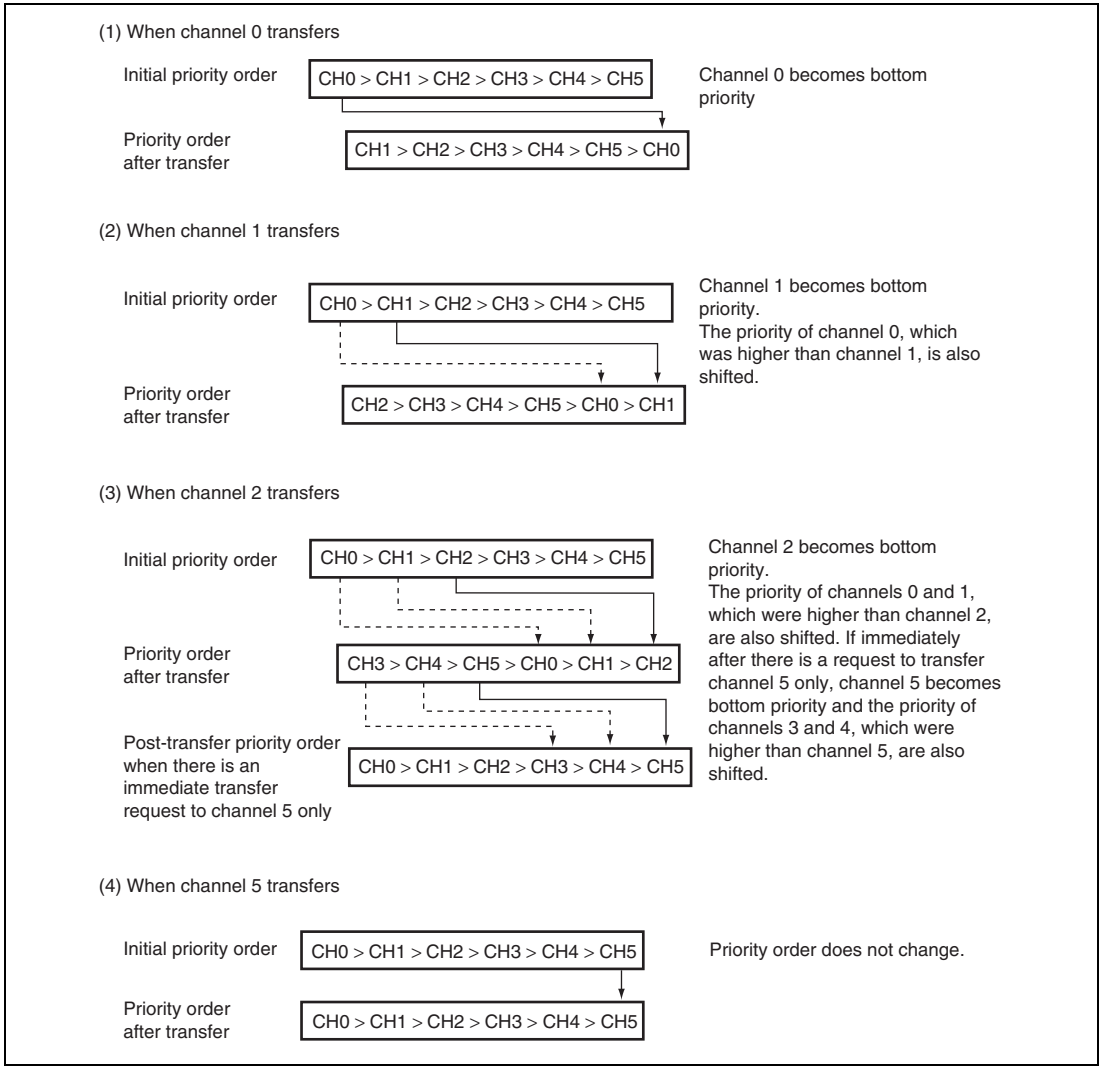


Figure 15.2 Round-Robin Mode

Figure 15.3 shows how the priority changes when channels 0 and 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC0 operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. As channel 0 has a higher priority, the channel 0 transfer starts (channel 3 is waiting for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are waiting for transfer).
4. When the channel 0 transfer ends, channel 0 has the lowest priority.
5. As channel 1 has a higher priority than channel 3 at this point, the channel 1 transfer starts (channel 3 is waiting for transfer).
6. When the channel 1 transfer ends, channel 1 has the lowest priority.
7. The channel 3 transfer starts.
8. When the channel 3 transfer ends, channels 3 and 2 have lower priority so that channel 3 has the lowest priority.

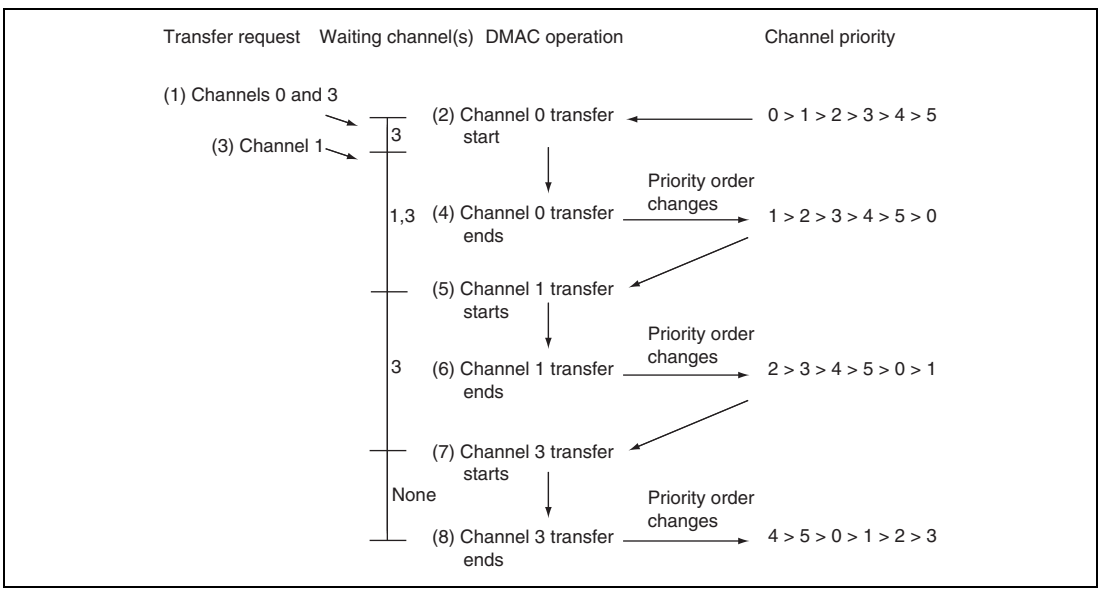


Figure 15.3 Channel Priority in Round-Robin Mode

15.5.3 DMA Transfer Types

Tables 15.9 and 15.10 show the transfer intervals that can be supported by the DMAC0.

Table 15.9 DMA Transfer Intervals for Auto-Request and External Request*

	Transfer Destination		
Transfer Source	**LBSC Space**	**DDR3-SDRAM Space**	**L Memory or L2C Memory**
LBSC space	Y	Y	Y
DDR3-SDRAM space	Y	Y	Y
L memory or L2C memory	Y	Y	Y

Legend:

Y: Transfer is enabled

Note: * External requests apply to only channels 0 to 3.

Table 15.10 DMA Transfer Intervals for On-Chip Peripheral Module Request*[2]

	Transfer Destination			
Transfer Source	**LBSC Space**	**DDR3-SDRAM Space**	**FLCTL*[1]**	**L Memory or L2C Memory**
LBSC space	N	N	Y	N
DDR3-SDRAM space	N	N	Y	N
FLCTL*[1]	Y	Y	N	Y
L memory or L2C memory	N	N	Y	N

Legend:

Y: Transfer is enabled

N: Transfer is disabled

Notes: 1. This is the access size that is permitted by the FLDTFIFO/FLECFIFO register of the FLCTL which is the transfer source or destination.

2. The transfer source or destination must be the request source register for an on-chip peripheral module request.

(1) Bus Modes

Bus modes include cycle steal mode and burst mode. The mode is selected by the LCKN and TB bits in the DMA0 channel control register (DMA0CHCR).

(a) Cycle Steal Mode (DMA0CHCR.LCKN = 1, DMA0CHCR.TB = 0)

In cycle steal mode, the DMAC0 does not keep the SuperHyway bus mastership, and obtains the bus mastership in every transfer unit of read or write cycle.

Figure 15.4 shows an example of DMA transfer timing in cycle steal mode.

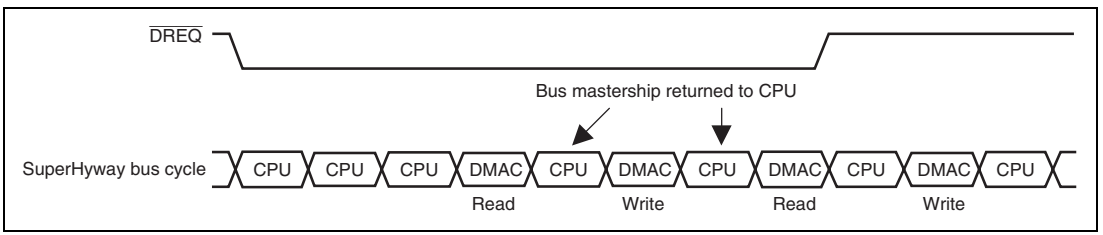


Figure 15.4 DMA Transfer Timing Example in Cycle Steal Mode (DREQ Low Level Detection)

(b) Intermittent Mode 16 (DMA0OR.CMS = 10, DMA0CHCR.LCKN = 1, DMA0CHCR.TB = 0), Intermittent Mode 64 (DMA0OR.CMS = 11, DMA0CHCR.LCKN = 1, DMA0CHCR.TB = 0)

In intermittent mode of cycle steal, the DMAC0 gives the SuperHyway bus mastership to other bus master whenever a one-transfer unit (byte, word, longword, or 16-byte or 32-byte unit) is completed. After that, if the next transfer request occurs, the DMAC0 issues the next transfer request after waiting for 16 or 64 clocks in Bck, transfers data of one-transfer unit again, and returns the SuperHyway bus mastership to other bus master. These operations are repeated until the transfer end condition is satisfied. It is possible to make lower the ratio of bus occupation by DMA transfer than cycle steal mode.

When the DMAC0 issues the next transfer request again, DMA transfer can be postponed in case of entry updating due to cache miss.

The intermittent modes must be cycle steal mode in all channels.

Figure 15.5 shows an example of DMA transfer timing in cycle steal intermittent mode.

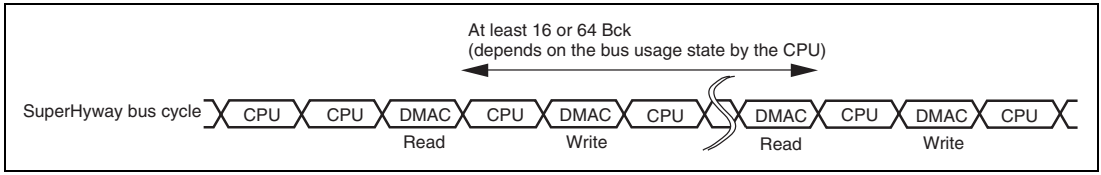


Figure 15.5 DMA Transfer Timing Example in Cycle Steal Intermittent Mode

(c) Burst Mode (DMA0CHCR.LCKN = 0, DMA0CHCR.TB = 1)

In burst mode, once the DMAC0 obtains the SuperHyway bus mastership, the transfer is performed continuously without releasing the bus mastership until the transfer end condition is satisfied. If the DREQ is detected at level in external request mode, when the DREQ pin is not active, the DMAC0 passes the SuperHyway bus mastership to the other bus master after the DMA transfer request that has already been accepted ends, even if the transfer end conditions have not been satisfied.

Burst mode can be set only when PCMCIA ATA complement mode with DACKBST enabled is selected as the external request mode (in channels 0 to 3). Burst mode cannot be set for channels 4 and 5. Figure 15.6 shows DMA transfer timing in burst mode.

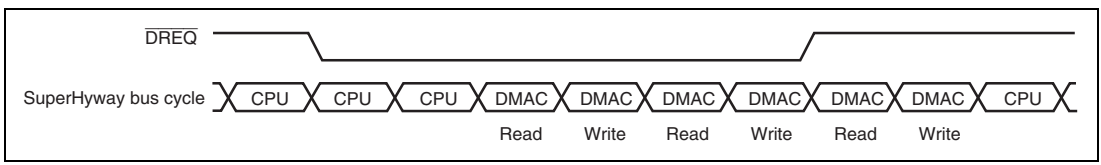


Figure 15.6 DMA Transfer Timing Example in Burst Mode (DREQ Low Level Detection)

(2) Bus Mode and Channel Priority

Figure 15.7 shows the bus modes and channel priority in fixed channel priority mode.

In fixed channel priority mode (CH0 > CH1), when channel 1 is transferring in burst mode, the transfer of channel 0 starts immediately if there is a transfer request to channel 0 with a higher priority.

At this time, if channel 0 is also in burst mode, the channel 1 transfer continues after the channel 0 transfer has completely ended. (Figure 15.7 (h))

When channel 0 is in cycle steal mode, channel 0 with a higher priority performs the transfer of one transfer unit and the channel 1 transfer is continuously performed without releasing the bus mastership. The bus mastership then switches between the two in the order of channel 0, channel 1, channel 0, and channel 1. (Figure 15.7 (d))

In other words, the bus status looks as if the CPU cycle reached after the transfer in cycle steal mode is replaced with transfer in burst mode.

When multiple channels are operating in burst modes, the channel with the highest priority is executed first.

When DMA transfer is executed in the multiple channels, the bus mastership is not given to the bus master until all competing burst transfers are completed.

In round-robin mode, the priority changes according to the specification shown in figure 15.3. However, the channel in cycle steal mode and the channel in burst mode cannot be mixed.

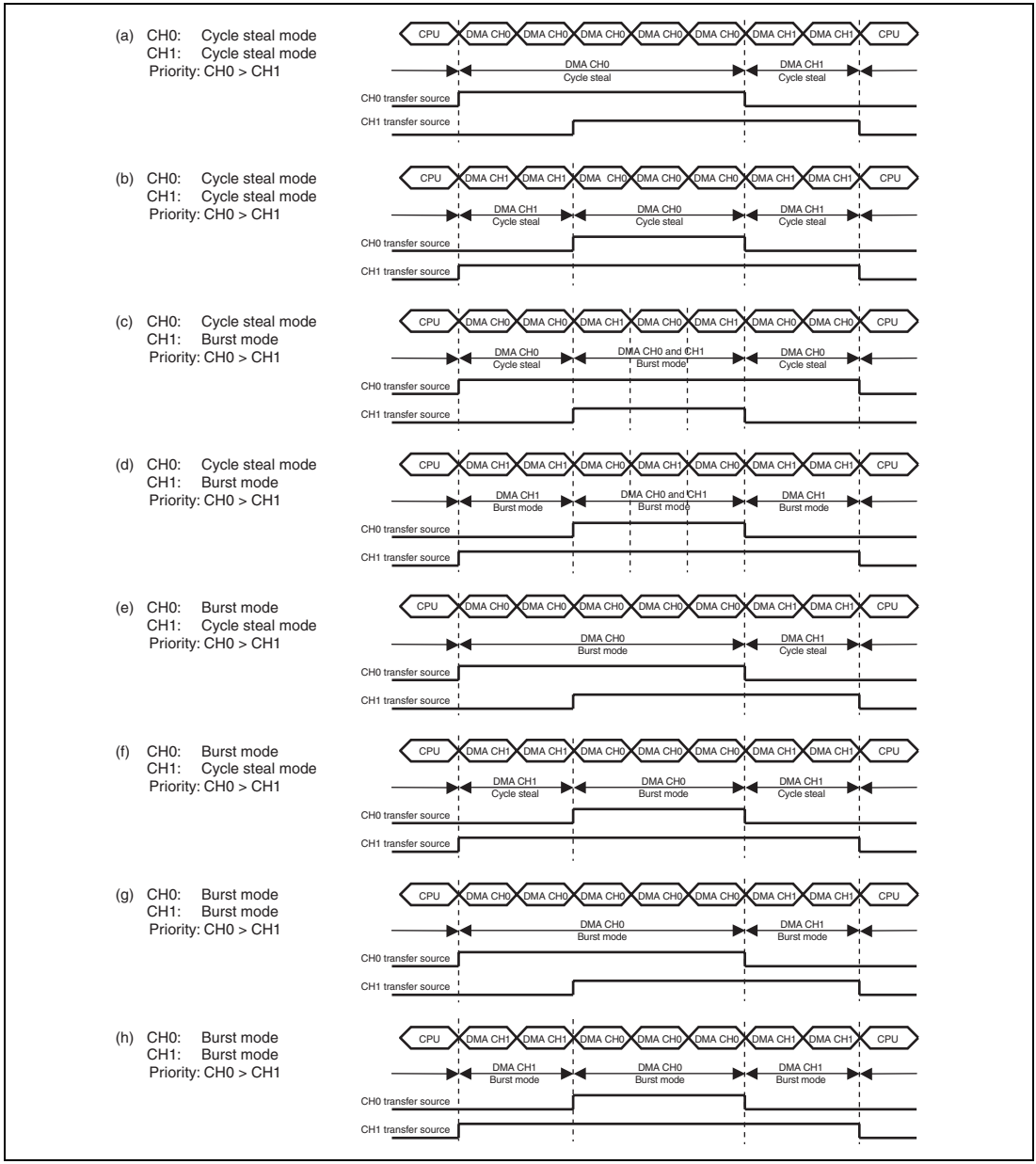


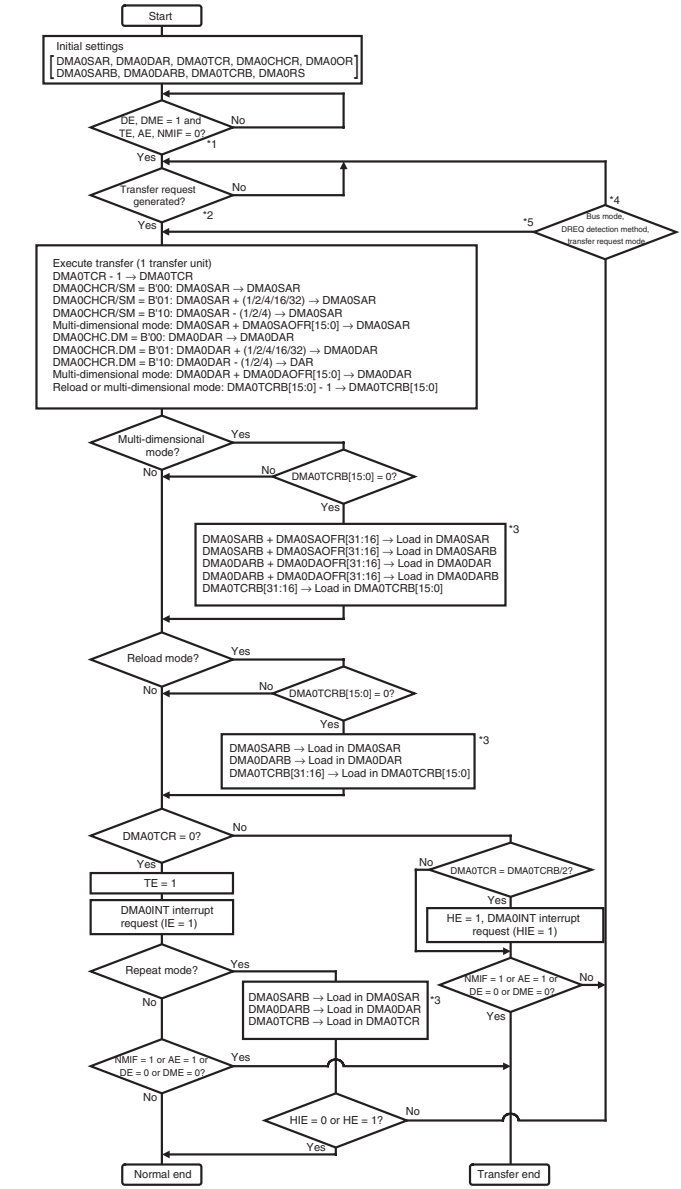
Figure 15.7 Bus Mode and Channel Priority in Fixed Channel Priority Mode

15.5.4 DMA Transfer Flow

After intended transfer conditions are set to the DMA0 source address register (DMA0SAR), DMA0 destination address register (DMA0DAR), DMA0 transfer count register (DMA0TCR), DMA0 channel control register (DMA0CHCR), DMA0 operation register (DMA0OR), and DMA0 extended resource selector (DMA0RS), the DMAC0 transfers data according to the following procedure.

1. Checks 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 DMAC0 transfers one transfer unit of data (depending on the settings of bits TS0, TS1, and TS2). In auto-request mode, the transfer starts automatically when the DE and DME bits are set to 1. The DMA0TCR value is decremented for each transfer. The actual transfer flows depend on the bus mode.
3. When the specified number of transfers has been completed (when the DMA0TCR value is 0), the transfer ends successfully. If the IE bit in DMA0CHCR is set to 1 at this time, a DMA0INT interrupt is sent to the CPU.
4. When an address error or an NMI interrupt is generated by the DMAC0, the transfer is aborted. Transfers are also aborted when the DE bit in DMA0CHCR or the DME bit in DMA0OR is cleared to 0.

Figure 15.8 shows a flowchart of DMA transfer.



Notes: 1. In repeat mode, a transfer request is accepted at TE =1 when HIE = 1 and HE = 0 (half-end interrupt enabled and HE cleared to 0 after being set to 1).
2. In auto-request mode, transfer starts when bits DE and DME are set to 1 while bits NMIF, AE, and TE are all 0 or bits TE and HIE are 1 and HE is 0 (in repeat mode).
3. Data loading to DMA0SAR, DMA0SARB, DMA0DAR, and DMA0DARB differs according to the operating conditions of each mode.
4. Cycle steal mode or level detection (external request) is used for DREQ in burst mode.
5. Edge detection (external request) is used for DREQ in burst mode or auto-request in burst mode.

Figure 15.8 Flowchart of DMA Transfer

15.5.5 Repeat Mode Transfer

A repeat mode transfer of the DMAC0 enables DMA transfers to be repeated without specifying the transfer settings every time.

Using the repeat mode function in combination with the half-end function enables virtual double buffer transfer execution. This allows efficient execution of the following procedures. As an example, an operation involving receiving audio data and compressing it is described below.

This example describes the processing for receiving audio data 40 words at a time and sequentially compressing it. In this example, audio data can be received from any memory.

1. DMAC0 settings
   - Set the address of the desired memory in DMA0SAR.
   - Set the address of an internal memory data storage area in DMA0DAR.
   - Set H'50 (80 times) to DMA0TCR.
   - Set the following values to DMA0CHCR.

     RPT (bits 28 to 25) = B'0010: Repeat mode (use DMA0DAR as a repeat area)

     HIE (bit 18) = B'1: DMA0TCR/2 interrupt generated

     DM (bits 15 and 14) = B'01: DMA0DAR incremented

     SM (bits 13 and 12) = B'01: DMA0SAR incremented

     IE (bit 2) = B'1: Interrupt enabled

     DE (bit 0) = B'1: DMA transfer enabled

     Set bits such as the TS bit according to the usage conditions.
   - Set bits CMS and PR in DMA0OR according to the usage conditions and set the DME bit to 1.
2. Start DMA transfer from the desired memory.
3. DMA0TCR is decreased to half of the initial value and an interrupt is generated.

   After reading DMA0CHCR and confirming that HE (bit 19) is set to 1 by an interrupt processing, clear HE (bit 19) to 0 and compress the 40 words of audio data from the address set in DMA0DAR.
4. DMA0TCR is cleared to 0 and an interrupt is generated.

   After reading DMA0CHCR and confirming that TE (bit 1) is set to 1 with an interrupt processing, clear TE (bit 1) to 0 and compress the 40 words of audio data from the address that is obtained by adding 40 to the address set in DMA0DAR. After this operation, the value of DMA0DARB is copied to DMA0DAR in DMAC0 and initialized, and the value of DMA0TCRB is copied to DMA0TCR and initialized to H'50 (80 times).

5. Steps 2 to 4 are repeated until the DME or DE bit is set to B'0, or an NMI interrupt is generated. (If the HE bit is not cleared to 0 in the procedure 2 or if the TE bit is not cleared to 0 in the procedure 4, the transfer is stopped when both the HE and TE bits are set to 1.)

This function enables sequential audio compression by switching a storing buffer for data received consequentially and a data buffer for processing signals alternately.

15.5.6 Reload Mode Transfer

In a reload mode transfer, according to the setting of the RPT bits in DMA0CHCR, the value set in DMA0SARB/DMA0DARB is reloaded to DMA0SAR/DMA0DAR at each transfer for the number which is set in bits DMA0TCRB[31:16] and DMA0TCRB[15:0], and the transfer is repeated until DMA0TCR is 0 without specifying the transfer again. This function is effective when data transfer with specific area is repeatedly executed.

In reload mode, DMA0TCRB is used as a reload counter. See section 15.4.6, DMA0 Transfer Count Registers B0 to B5, and set DMA0TCRB appropriately.

Figure 15.9 shows an example of reload mode settings.

The transfer source and destination addresses in reload mode transfer when using the following register settings are shown below.

- Register settings
  Set the transfer source address in DMA0SAR (the data written to DMA0SAR is also written to DMA0SARB).
  Set the transfer destination address in DMA0DAR.
  Set H'0000000C to TCR (12 transfers).
  Set H'00040004 to TCRB (reload every 4 transfers).
  Make settings in DMA0CHCR as follows.
    RPT (bits 28 to 25) = B'0111: Reload mode (DMA0SAR is reloaded)
    DM (bits 15 and 14) = B'01: DMA0DAR incremented
    SM (bits 13 and 12) = B'01: DMA0SAR incremented
    TS (bits 20, 4, and 3) = B'010: Transfers in longword (4-byte) units

- The DMA transfer source and destination addresses resulting from the above register settings are as follows:

1st transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR
2nd transfer: Transfer source address = DMA0SAR + H'04, transfer destination address = DMA0DAR + H'04
3rd transfer: Transfer source address = DMA0SAR + H'08, transfer destination address = DMA0DAR + H'08
4th transfer: Transfer source address = DMA0SAR + H'0C, transfer destination address = DMA0DAR + H'0C
5th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'10 (DMA0SARB value reloaded in DMA0SAR)
6th transfer: Transfer source address = DMA0SAR + H'04, transfer destination address = DMA0DAR + H'14
7th transfer: Transfer source address = DMA0SAR + H'08, transfer destination address = DMA0DAR + H'18
8th transfer: Transfer source address = DMA0SAR + H'0C, transfer destination address = DMA0DAR + H'1C
9th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'20 (DMA0SARB value reloaded in DMA0SAR)
10th transfer: Transfer source address = DMA0SAR + H'04, transfer destination address = DMA0DAR + H'24
11th transfer: Transfer source address = DMA0SAR + H'08, transfer destination address = DMA0DAR + H'28
12th transfer: Transfer source address = DMA0SAR + H'0C, transfer destination address = DMA0DAR + H'2C

Figure 15.9 Example of Operation Based on Reload Mode Settings

15.5.7 Multi-Dimensional Mode Transfer

The multi-dimensional mode can be used to perform multi-dimensional, scatter/gather, and stride transfers.

In multi-dimensional mode, the DMA0SAR address is updated with the values specified in bits DMA0SAOFR[15:0] and DMA0SAOFR[31:16], and the DMA0DAR address is updated with the values specified in bits DMA0DAOFR[15:0] and DMA0DAOFR[31:16]. DMA0TCRB is used to specify the transfer count for updating DMA0SAR and DMA0DAR and also as a transfer counter. Bits DMA0TCRB[15:0] function as a transfer counter. When their value reaches 0, DMA0SAR and DMA0DAR are updated, and the value of bits DMA0TCRB[31:16] is loaded into bits DMA0TCRB[15:0]. Bits DMA0SAOFR[15:0] and DMA0DAOFR[15:0] set the amount by which the source and destination addresses are incremented respectively after each transfer. Bits DMA0SAOFR[31:16] and DMA0DAOFR[31:16] set the address offsets used for reloading DMA0SAR and DMA0DAR when the value of bits DMA0TCRB[15:0] reaches 0.

- When bits RPT[3:0] in DMA0CHCR are set to B'1101

  The DMA0SAR and DMA0DAR addresses are updated according to the settings of DMA0SAOFR and DMA0DAOFR.

  The settings of bits SM[1:0] as well as bits DM[1:0] in DMA0CHCR are ignored.

  DMA0TCRB[15:0] ≠ 0: DMA0SAR + DMA0SAOFR[15:0] → DMA0SAR
  DMA0DAR + DMA0DAOFR[15:0] → DMA0DAR

  DMA0TCRB[15:0] = 0: DMA0SARB + DMA0SAOFR[31:16] → DMA0SAR
  DMA0SARB + DMA0SAOFR[31:16] → DMA0SARB
  DMA0DARB + DMA0DAOFR[31:16] → DMA0DAR
  DMA0DARB + DMA0DAOFR[31:16] → DMA0DARB

- When bits RPT[3:0] in DMA0CHCR are set to B'1110

  The DMA0DAR address is updated according to the setting of DMA0DAOFR.

  The setting of bits DM[1:0] in DMA0CHCR is ignored.

  DMA0TCRB[15:0] ≠ 0: DMA0DAR + DMA0DAOFR[15:0] → DMA0DAR

  DMA0TCRB[15:0] = 0: DMA0DARB + DMA0DAOFR[31:16] → DMA0DAR
  DMA0DARB + DMA0DAOFR[31:16] → DMA0DARB

- When bits RPT[3:0] in DMA0CHCR are set to B'1111

  The DMA0SAR address is updated according to the setting of DMA0SAOFR.

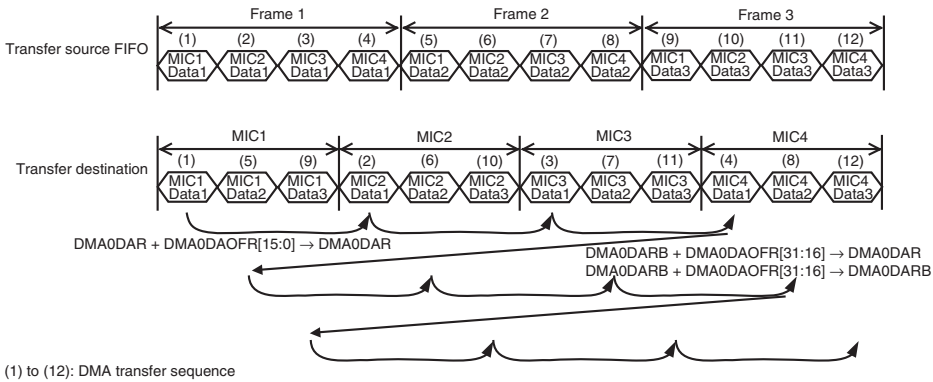
  The setting of bits SM[1:0] in DMA0CHCR are ignored.

  DMA0TCRB[15:0] ≠ 0: DMA0SAR + DMA0SAOFR[15:0] → DMA0SAR

  DMA0TCRB[15:0] = 0: DMA0SARB+ DMA0SAOFR[31:16] → DMA0SAR
  DMA0SARB+ DMA0SAOFR[31:16] → DMA0SARB

Figure 15.10 shows an example of multi-dimensional transfer operation, figure 15.11 shows an example of scatter transfer operation, figure 15.12 shows an example of gather transfer operation, and figure 15.13 shows an example of stride transfer operation.

The multi-dimensional transfer function can be used to reorder audio data from the transfer source FIFO memory and transfer it, as shown below.

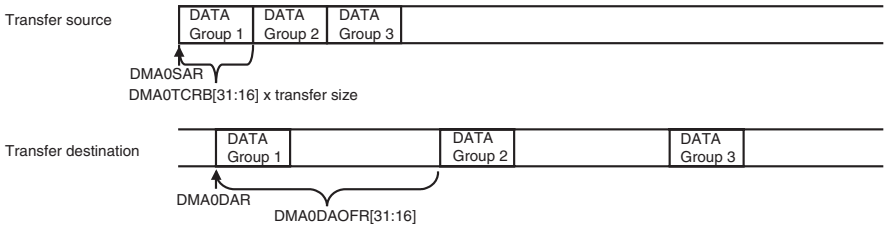


(1) to (12): DMA transfer sequence

- DMAC0 settings
  DMA0SAR: Any memory address (FIFO assumed in this example)
  DMA0DAR: Any memory address
  DMA0TCR = H'0000000C (12 transfers)
  DMA0TCRB = H'00040004 (DMA0DARB + DMA0DAOFR[31:16] → DMA0DAR and DMA0DARB + DMA0DAOFR[31:16] → DMA0DARB every 4 transfers)
  DMA0DAOFR = H'0002_0006
  DMA0CHCR: As follows.
  RPT[3:0] = B'1110: Multi-dimensional mode, DMA0DAR address updated by the value specified in bits DMA0DAOFR[15:0] and DMA0DAOFR[31:16]
  TS[2:0] = B'001: Transfers in word units
  SM[1:0] = B'00: DMA0SAR fixed
  DE = B'1: Transfer enabled
  Set other bits such as RS to match the usage conditions.
  Set the CMS and PR bits in DMA0OR to match the usage conditions. Set the DME bit to 1.

- The DMA0SAR and DMA0DAR addresses resulting from the above register settings are as follows.
  1st transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR
  2nd transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'0006
  3rd transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'000C
  4th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'0012
  5th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'0002
  6th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'0008
  7th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'000E
  8th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'0014
  9th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'0004
  10th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'000A
  11th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'0010
  12th transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR + H'0016

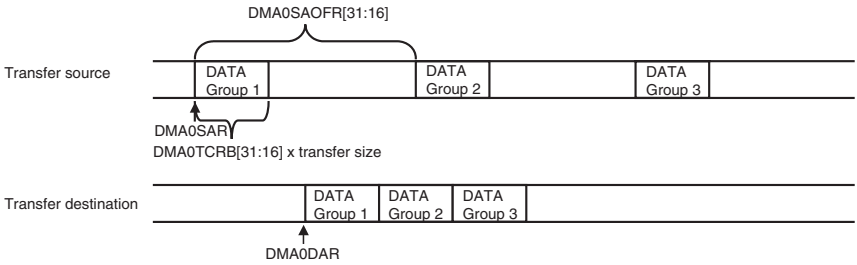
Figure 15.10 Multi-Dimensional Transfer Operation Example



• DMAC0 settings
DMA0SAR: Any memory address
DMA0DAR: Any memory address
DMA0TCR = H'0000000C (12 transfers)
DMA0TCRB = H'00040004 (DMA0DARB + DMA0DAOFR[31:16] → DMA0DAR and DMA0DARB + DMA0DAOFR[31:16] → DMA0DARB every 4 transfers)
DMA0DAOFR = H'0200_0004 (Set bits DMA0DAOFR[15:0] to produce an offset equal to the transfer size.)
DMA0CHCR: As follows.
RPT[3:0] = B'1110: Multi-dimensional mode, DMA0DAR address updated by the value specified in bits DMA0DAOFR[15:0] and DMA0DAOFR[31:16]
TS[2:0] = B'010: Transfers in longword units
SM[1:0] = B'01: DMA0SAR incremented
DE = B'1: Transfer enabled
Set other bits such as RS to match the usage conditions.
Set the CMS and PR bits in DMA0OR to match the usage conditions. Set the DME bit to 1.

• The DMA0SAR and DMA0DAR addresses resulting from the above register settings are as follows.
1st transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR
2nd transfer: Transfer source address = DMA0SAR + H'0004, transfer destination address = DMA0DAR + H'0004
3rd transfer: Transfer source address = DMA0SAR + H'0008, transfer destination address = DMA0DAR + H'0008
4th transfer: Transfer source address = DMA0SAR + H'000C, transfer destination address = DMA0DAR + H'000C
5th transfer: Transfer source address = DMA0SAR + H'0010, transfer destination address = DMA0DAR + H'0200
6th transfer: Transfer source address = DMA0SAR + H'0014, transfer destination address = DMA0DAR + H'0204
7th transfer: Transfer source address = DMA0SAR + H'0018, transfer destination address = DMA0DAR + H'0208
8th transfer: Transfer source address = DMA0SAR + H'001C, transfer destination address = DMA0DAR + H'020C
9th transfer: Transfer source address = DMA0SAR + H'0020, transfer destination address = DMA0DAR + H'0400
10th transfer: Transfer source address = DMA0SAR + H'0024, transfer destination address = DMA0DAR + H'0404
11th transfer: Transfer source address = DMA0SAR + H'0028, transfer destination address = DMA0DAR + H'0408
12th transfer: Transfer source address = DMA0SAR + H'002C, transfer destination address = DMA0DAR + H'040C

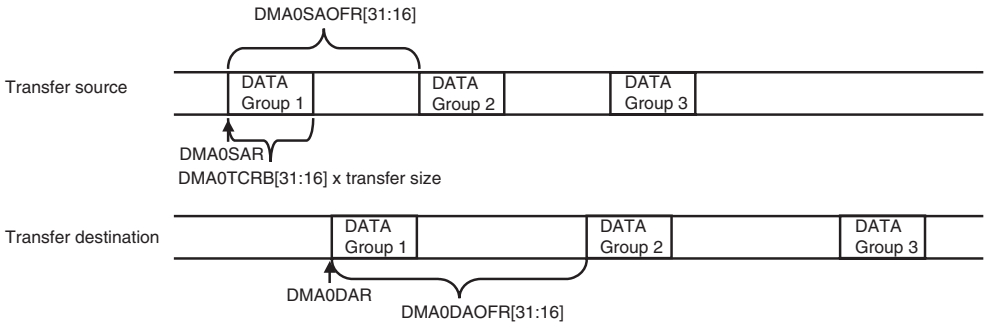
Figure 15.11 Scatter Transfer Operation Example



• DMAC0 settings
DMA0SAR: Any memory address
DMA0DAR: Any memory address
DMA0TCR = H'0000000C (12 transfers)
DMA0TCRB = H'00040004 (DMA0SARB + DMA0SAOFR[31:16] → DMA0SAR and DMA0SARB + DMA0SAOFR[31:16] → DMA0SARB every 4 transfers)
DMA0SAOFR = H'0100_0004 (Set bits DMA0SAOFR[15:0] to produce an offset equal to the transfer size.)
DMA0CHCR: As follows.
RPT[3:0] = B'1111: Multi-dimensional mode, DMA0SAR address updated by the value specified in bits DMA0SAOFR[15:0] and DMA0SAOFR[31:16]
TS[2:0] = B'010: Transfers in longword units
DM[1:0] = B'01: DMA0DAR incremented
DE = B'1: Transfer enabled
Set other bits such as RS to match the usage conditions.
Set the CMS and PR bits in DMA0OR to match the usage conditions. Set the DME bit to 1.

• The DMA0SAR and DMA0DAR addresses resulting from the above register settings are as follows.
1st transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR
2nd transfer: Transfer source address = DMA0SAR + H'0004, transfer destination address = DMA0DAR + H'0004
3rd transfer: Transfer source address = DMA0SAR + H'0008, transfer destination address = DMA0DAR + H'0008
4th transfer: Transfer source address = DMA0SAR + H'000C, transfer destination address = DMA0DAR + H'000C
5th transfer: Transfer source address = DMA0SAR + H'0100, transfer destination address = DMA0DAR + H'0010
6th transfer: Transfer source address = DMA0SAR + H'0104, transfer destination address = DMA0DAR + H'0014
7th transfer: Transfer source address = DMA0SAR + H'0108, transfer destination address = DMA0DAR + H'0018
8th transfer: Transfer source address = DMA0SAR + H'010C, transfer destination address = DMA0DAR + H'001C
9th transfer: Transfer source address = DMA0SAR + H'0200, transfer destination address = DMA0DAR + H'0020
10th transfer: Transfer source address = DMA0SAR + H'0204, transfer destination address = DMA0DAR + H'0024
11th transfer: Transfer source address = DMA0SAR + H'0208, transfer destination address = DMA0DAR + H'0028
12th transfer: Transfer source address = DMA0SAR + H'020C, transfer destination address = DMA0DAR + H'002C

Figure 15.12 Gather Transfer Operation Example



• DMAC0 settings
DMA0SAR: Any memory address
DMA0DAR: Any memory address
DMA0TCR = H'0000000C (12 transfers)
DMA0TCRB = H'00040004 (DMA0SARB + DMA0SAOFR[31:16] → DMA0SAR and DMA0SARB + DMA0SAOFR[31:16] → DMA0SARB, DARB + DMA0DAOFR[31:16] → DMA0DAR and DMA0DARB + DMA0DAOFR[31:16] → DMA0DARB, every 4 transfers)
DMA0SAOFR = H'0100_0004 (Set bits DMA0SAOFR[15:0] to produce an offset equal to the transfer size.)
DMA0DAOFR = H'0200_0004 (Set bits DMA0DAOFR[15:0] to produce an offset equal to the transfer size.)
DMA0CHCR: As follows.
RPT[3:0] = B'1111: Multi-dimensional mode, DMA0SAR address updated by the value specified in bits DMA0SAOFR[15:0] and DMA0SAOFR[31:16], and DMA0DAR address updated by the value specified in bits DMA0DAOFR[15:0] and DMA0DAOFR[31:16]
TS[2:0] = B'010: Transfers in longword units
DE = B'1: Transfer enabled
Set other bits such as RS to match the usage conditions.
Set the CMS and PR bits in DMA0OR to match the usage conditions. Set the DME bit to 1.

• The DMA0SAR and DMA0DAR addresses resulting from the above register settings are as follows.
1st transfer: Transfer source address = DMA0SAR, transfer destination address = DMA0DAR
2nd transfer: Transfer source address = DMA0SAR + H'0004, transfer destination address = DMA0DAR + H'0004
3rd transfer: Transfer source address = DMA0SAR + H'0008, transfer destination address = DMA0DAR + H'0008
4th transfer: Transfer source address = DMA0SAR + H'000C, transfer destination address = DMA0DAR + H'000C
5th transfer: Transfer source address = DMA0SAR + H'0100, transfer destination address = DMA0DAR + H'0200
6th transfer: Transfer source address = DMA0SAR + H'0104, transfer destination address = DMA0DAR + H'0204
7th transfer: Transfer source address = DMA0SAR + H'0108, transfer destination address = DMA0DAR + H'0208
8th transfer: Transfer source address = DMA0SAR + H'010C, transfer destination address = DMA0DAR + H'020C
9th transfer: Transfer source address = DMA0SAR + H'0200, transfer destination address = DMA0DAR + H'0400
10th transfer: Transfer source address = DMA0SAR + H'0204, transfer destination address = DMA0DAR + H'0404
11th transfer: Transfer source address = DMA0SAR + H'0208, transfer destination address = DMA0DAR + H'0408
12th transfer: Transfer source address = DMA0SAR + H'020C, transfer destination address = DMA0DAR + H'040C

Figure 15.13 Stride Transfer Operation Example

15.5.8 DREQ Pin Sampling Timing

Figures 15.14 to 15.23 show the timing that the DREQ input is sampled in each bus mode.

Figures 15.14, 15.17, and 15.21 show the timing that the DREQ input is sampled when byte transfer is performed in 8-, 16-, or 32-bit bus width, word transfer is performed in 16- or 32-bit bus width, or longword transfer is performed in 32-bit bus width. DACK is output once in a single DMA transfer.

Figures 15.15, 15.18, and 15.22 show the timing that the DREQ input is sampled when word transfer is performed in 8-bit bus width, longword transfer is performed in 8- or 16-bit bus width, or 16- or 32-byte transfer is performed in 8-, 16-, or 32-bit bus width. These figures suppose that DACK of a single DMA transfer is divided.

Figures 15.16, 15.19, and 15.23 show the timing that the DREQ input is sampled when word transfer is performed in 8-bit bus width, longword transfer is performed in 8- or 16-bit bus width, or 16- or 32-byte transfer is performed in 8-, 16-, or 32-bit bus width. These figures suppose that DACK of a single DMA transfer is connected.

When word transfer is performed in 8-bit bus width, longword transfer is performed in 8- or 16-bit bus width, or 16- or 32-byte transfer is performed in 8-, 16-, or 32-bit bus width, DMA transfer units are divided into multiple bus cycles. If DMA transfer size is divided into multiple bus cycles and CS is negated between bus cycles, DACK output is divided, like CS. For details, see section 11.5.13, Register Settings for Divided-Up $\overline{\text{DACKn}}$ Output.

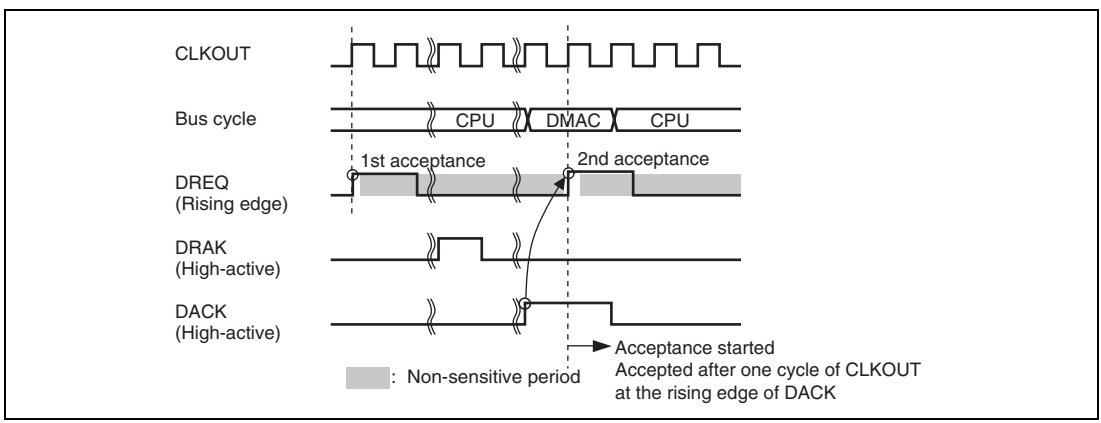


Figure 15.14 Example 1 of DREQ Input Detection in Cycle Steal Mode Edge Detection (Byte Transfer in 8/16/32-Bit Bus Width, Word Transfer in 16/32-Bit Bus Width, or Longword Transfer in 32-Bit Bus Width)

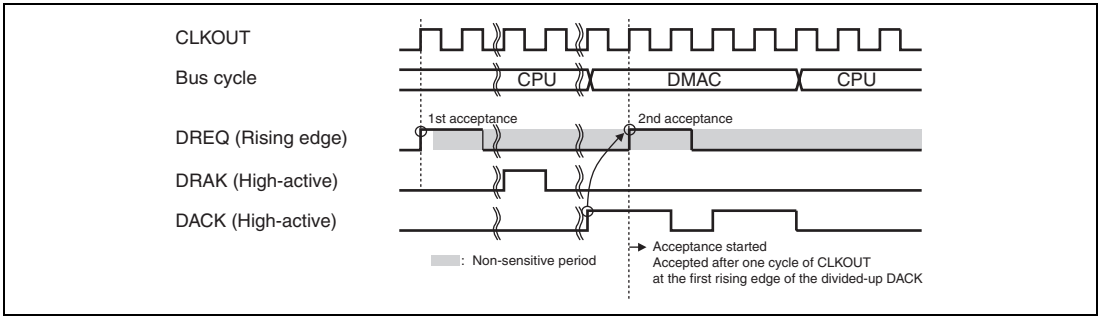


Figure 15.15 Example 2 of DREQ Input Detection in Cycle Steal Mode Edge Detection (Word Transfer in 8-Bit Bus Width, Longword Transfer in 8/16-Bit Bus Width, 16/32-Byte Transfer in 8/16/32-Bit Bus Width: DACK of Single DMA Transfer Divided)

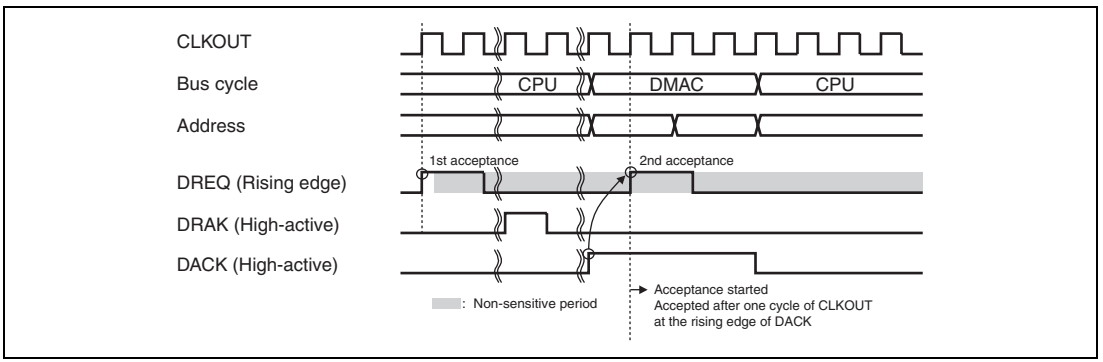


Figure 15.16 Example 3 of DREQ Input Detection in Cycle Steal Mode Edge Detection (Word Transfer in 8-Bit Bus Width, Longword Transfer in 8/16-Bit Bus Width, or 16/32-Byte Transfer in 8/16/32-Bit Bus Width: DACK of Single DMA Transfer is Connected)

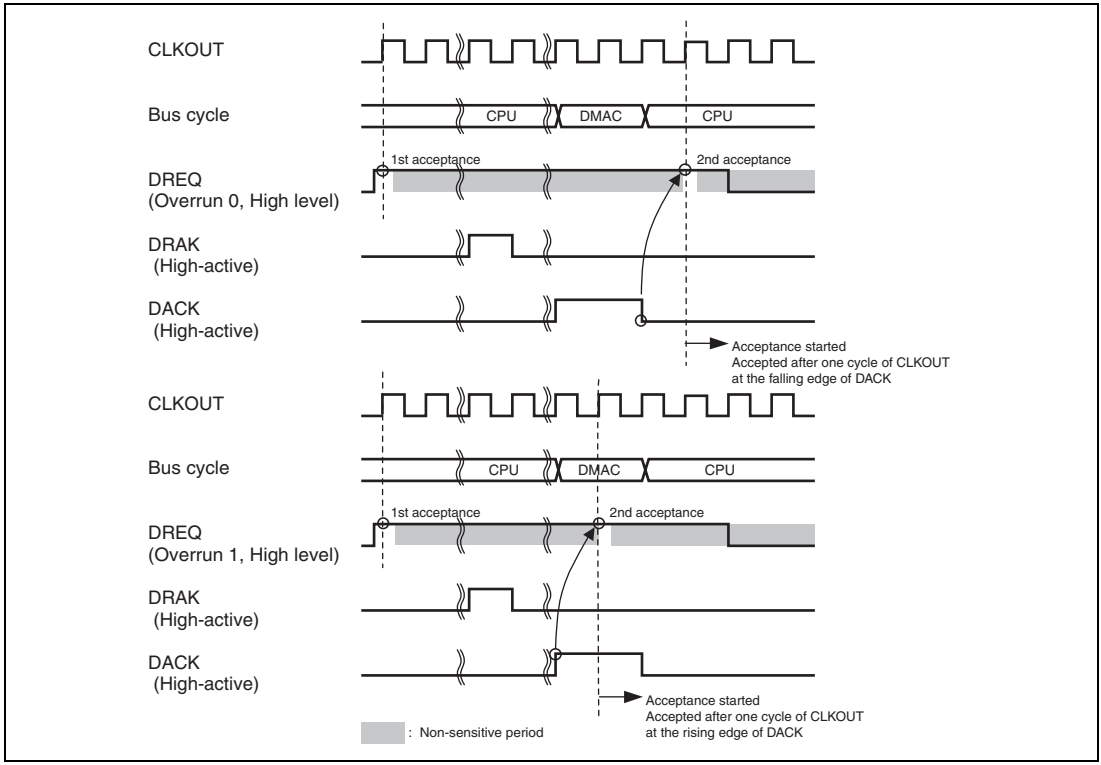


Figure 15.17 Example 1 of DREQ Input Detection in Cycle Steal Mode Level Detection (Byte Transfer in 8/16/32-Bit Bus Width, Word Transfer in 16/32-Bit Bus Width, or Longword Transfer in 32-Bit Bus Width)

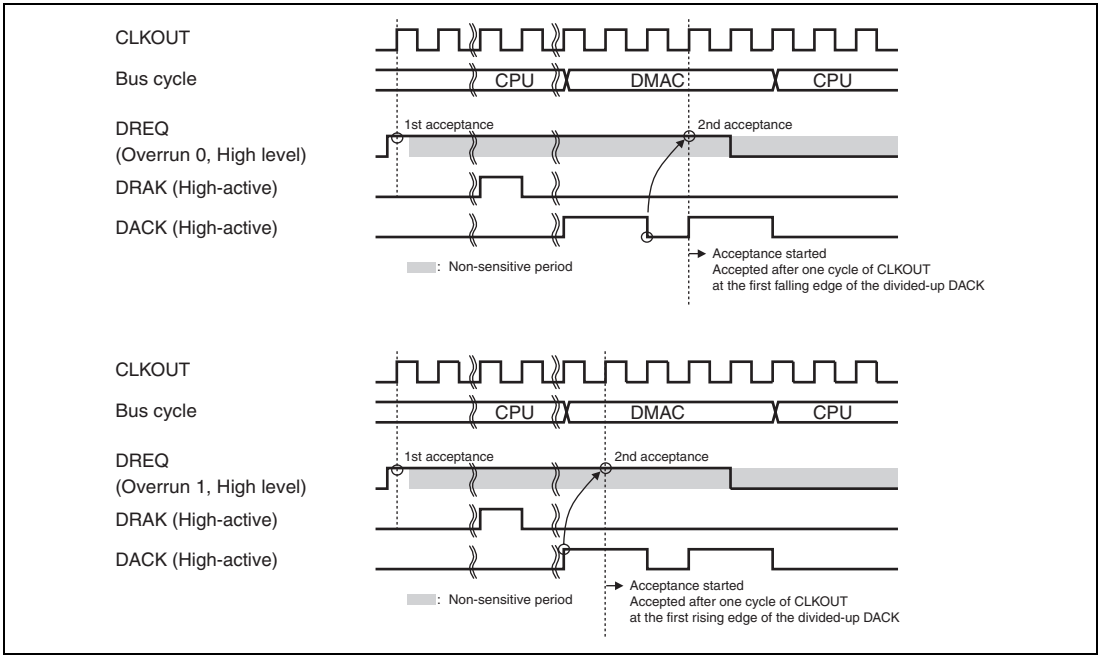


Figure 15.18 Example 2 of DREQ Input Detection in Cycle Steal Mode Level Detection (Word Transfer in 8-Bit Bus Width, Longword Transfer in 8/16-Bit Bus Width, 16/32-Byte Transfer in 8/16/32-Bit Bus Width: DACK of Single DMA Transfer Divided)

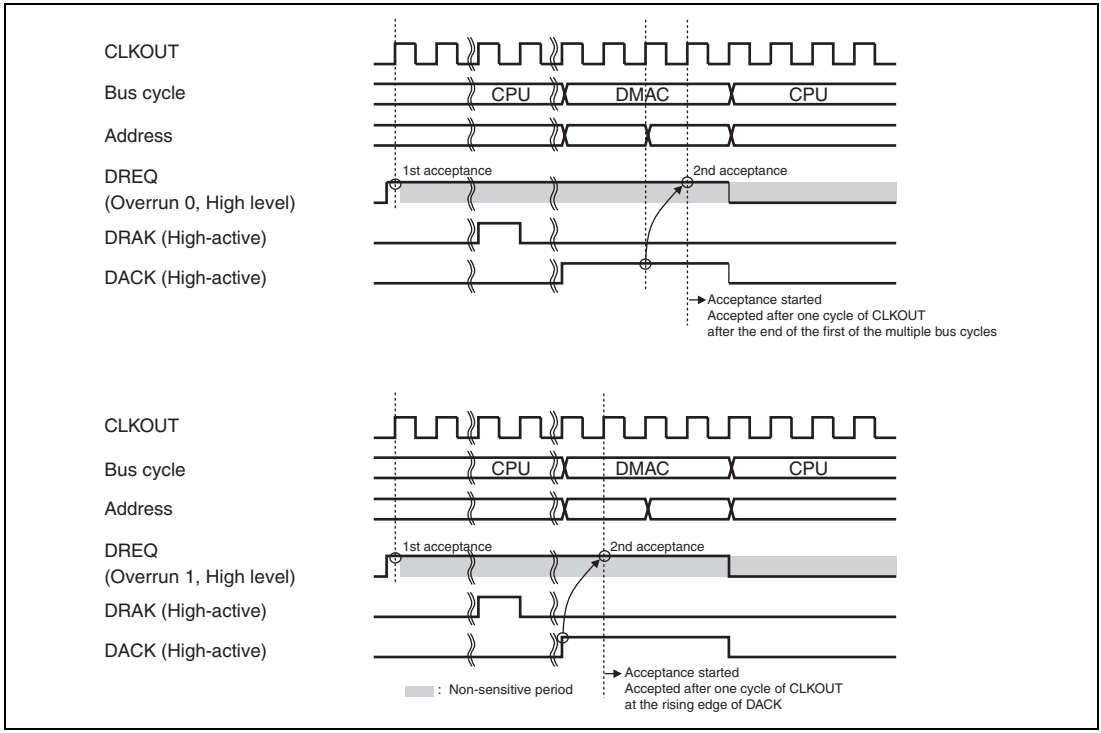


Figure 15.19 Example 3 of DREQ Input Detection in Cycle Steal Mode Level Detection (Word Transfer in 8-Bit Bus Width, Longword Transfer in 8/16-Bit Bus Width, or 16/32-Byte Transfer in 8/16/32-Bit Bus Width: DACK of Single DMA Transfer is Connected)

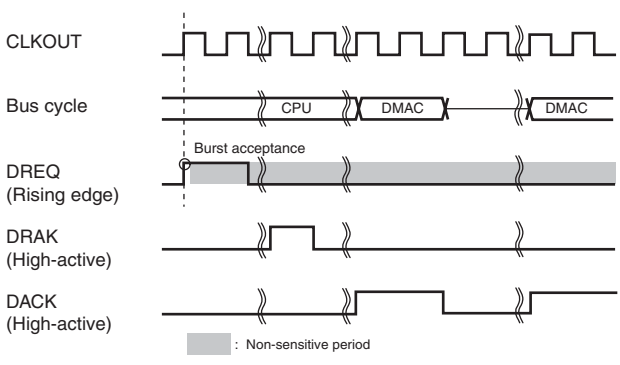


Figure 15.20 Example of DREQ Input Detection in Burst Mode Edge Detection

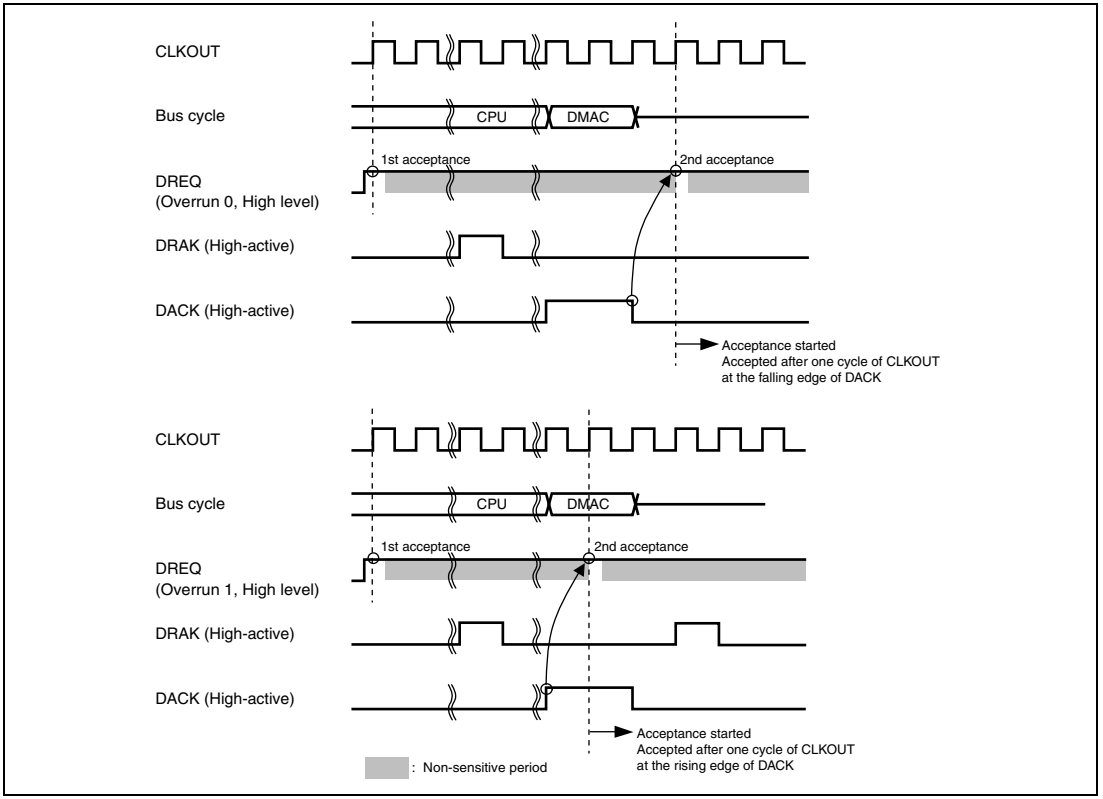


Figure 15.21 Example 1 of DREQ Input Detection in Burst Mode Level Detection (Byte Transfer in 8/16/32-Bit Bus Width, Word Transfer in 16/32-Bit Bus Width, or Longword Transfer in 32-Bit Bus Width)

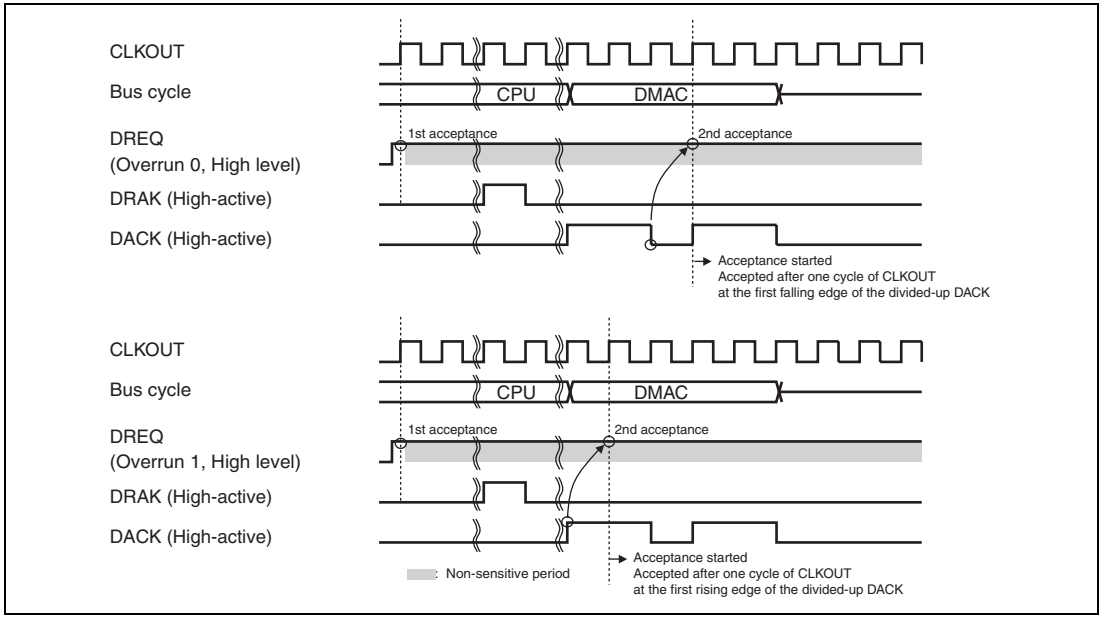


Figure 15.22 Example 2 of DREQ Input Detection in Burst Mode Level Detection (Word Transfer in 8-Bit Bus Width, Longword Transfer in 8/16-Bit Bus Width, 16/32-Byte Transfer in 8/16/32-Bit Bus Width: DACK of Single DMA Transfer Divided)

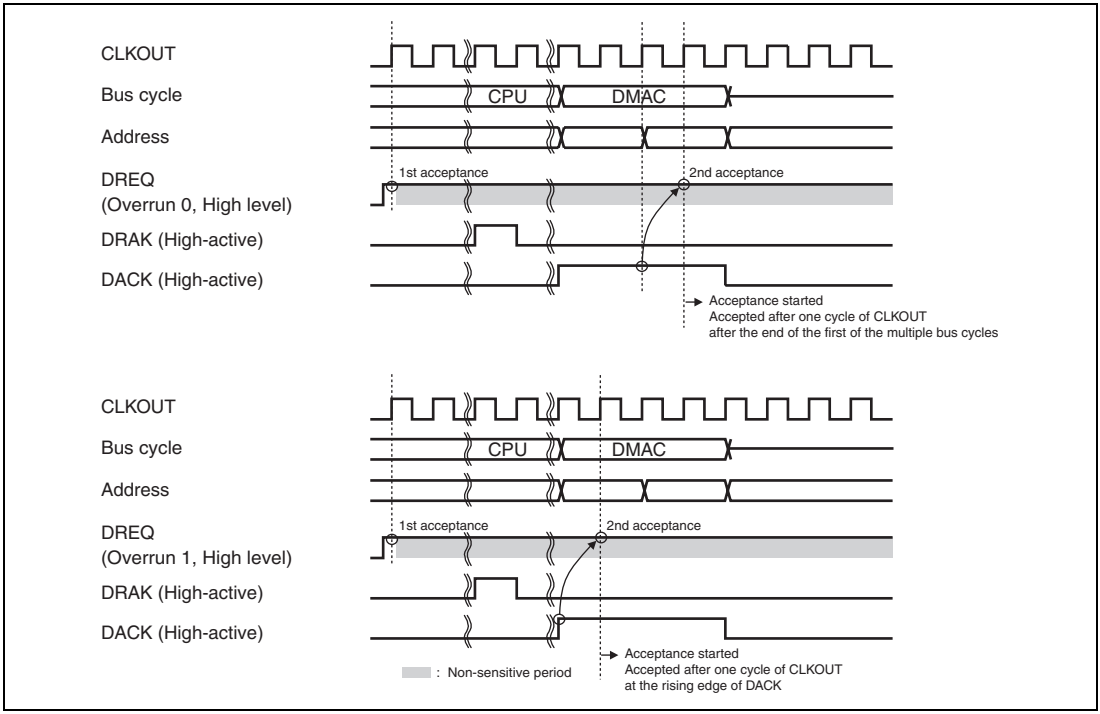


Figure 15.23 Example 3 of DREQ Input Detection in Burst Mode Level Detection (Word Transfer in 8-Bit Bus Width, Longword Transfer in 8/16-Bit Bus Width, or 16/32-Byte Transfer in 8/16/32-Bit Bus Width: DACK of Single DMA Transfer is Connected)

15.6 DMAC0 Interrupt Sources

The DMAC0 has interrupt sources for seven types of interrupts: DMA transfer end/half-end interrupt request (DMA0INT0 to DMA0INT5), one for each channel and DMA address error interrupt request (DMA0AE), which is common to all channels.

Table 15.11 lists the interrupt sources. Interrupt sources are sent independently to the interrupt controller.

Table 15.11 DMAC0 Interrupt Sources

Interrupt Factor	Description
DMA0INT0	Channel 0 DMA transfer end/half-end interrupt
DMA0INT1	Channel 1 DMA transfer end/half-end interrupt
DMA0INT2	Channel 2 DMA transfer end/half-end interrupt
DMA0INT3	Channel 3 DMA transfer end/half-end interrupt
DMA0INT4	Channel 4 DMA transfer end/half-end interrupt
DMA0INT5	Channel 5 DMA transfer end/half-end interrupt
DMA0AE	DMA address error interrupt common to all channels

15.7 Usage Notes

Note the following things when using this DMAC0.

15.7.1 Module Stop Function and Frequency Change

While the DMAC0 is operating, if the standby control register (MSTPCR1) is used to start operation (clear the bit to 0) or stop operation (set the bit to 1) or if the frequency is changed, the operation of this LSI is not guaranteed.

Check that the DME bit (bit 0) in DMA0OR is 0 or the DE bits (bit 0) in DMA0CHCR0 to DMA0CHCR5 are all 0 before setting the standby control register (MSTPCR1) or changing the frequency.

15.7.2 Address Error

When the AE bit in DMA0OR is set to 1 and a DMA address error occurs, reset the registers for all channels and then start transfer.

15.7.3 Divided-Up DACK Output

When 16- or 32-byte transfer is performed in 8-, 16-, or 32-bit bus width, longword transfer is performed in 8- or 16-bit bus width, or word transfer is performed in 8-bit bus width, DMA transfer units are divided into multiple bus cycles. Note that DACK output is divided-up, like CS, if DMA transfer size is divided into multiple bus cycles and CS is negated between bus cycles.

Section 16 Direct Memory Access Controller 1 (DMAC1)

16.1 Overview

The direct memory access controller 1 (DMAC1) connected to the internal bus (SuperHyway) can be used instead of the CPU to perform high-speed data transfers among external memory (DDR-SDRAM) and on-chip memory.

16.1.1 Features

- Four channels
- Address space: Supports up to 64-bit address spaces on the architecture and up to 32-bit address spaces in the actual product.
- Transfer data capacity
  - ch0, ch1: A value from 4 bytes (H'0000_0004) to 536,870,912 bytes (H'2000_0000) can be set in 4-byte units
  - ch2, ch3: A value from 1 byte (H'0000_0001) to 536,870,912 bytes (H'2000_0000) can be set in 1-byte units

  The transfer data capacity can be independently set for data transfer from the transfer source and data transfer to the transfer destination.
- Transfer data length
  - ch0, ch1: Automatically selected from 4, 8, 16, or 32 bytes according to the transfer source and destination addresses and remaining transfer data size.
  - ch2, ch3: Can be selected from 1, 2, 4, 8, or 32 bytes through register setting.

  Note however that if the transfer source address or transfer destination address is not set at the address boundary of the selected byte size, data is sent 1 byte each until the address boundary is reached. When the remaining transfer data has become smaller than the selected data size, data is sent 1 byte each until the transfer has completed.
- Dual address mode
- Priority: Fixed channel priority mode
- Interrupt request: DMA transfer end interrupt (DMA1TEn), transfer source transfer error interrupt (DMA1SEn), and transfer destination transfer error interrupt (DMA1DEn) can be generated in each channel (n = 0 to 3; corresponds to each channel).
- Data transfer
  - ch0, ch1: Contiguous region transfer, stride transfer, and gather/scatter transfer are possible among resources on the SuperHyway bus.
  - ch2, ch3: Contiguous region transfer is possible among resources on the SuperHyway bus.

- Command chain
  - ch0, ch1: Multiple data transfers can be executed continuously according to the data transfer instruction set in the specified address.
  - ch2, ch3: Command chain is not supported.
- SuperHyway transactions which can be issued
  - ch0, ch1
    - 4-, 8-, 16-, or 32-byte transfer
  - ch2, ch3
    - 1-, 2-, 4-, 8-, or 32-byte transfer

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the DMAC1.

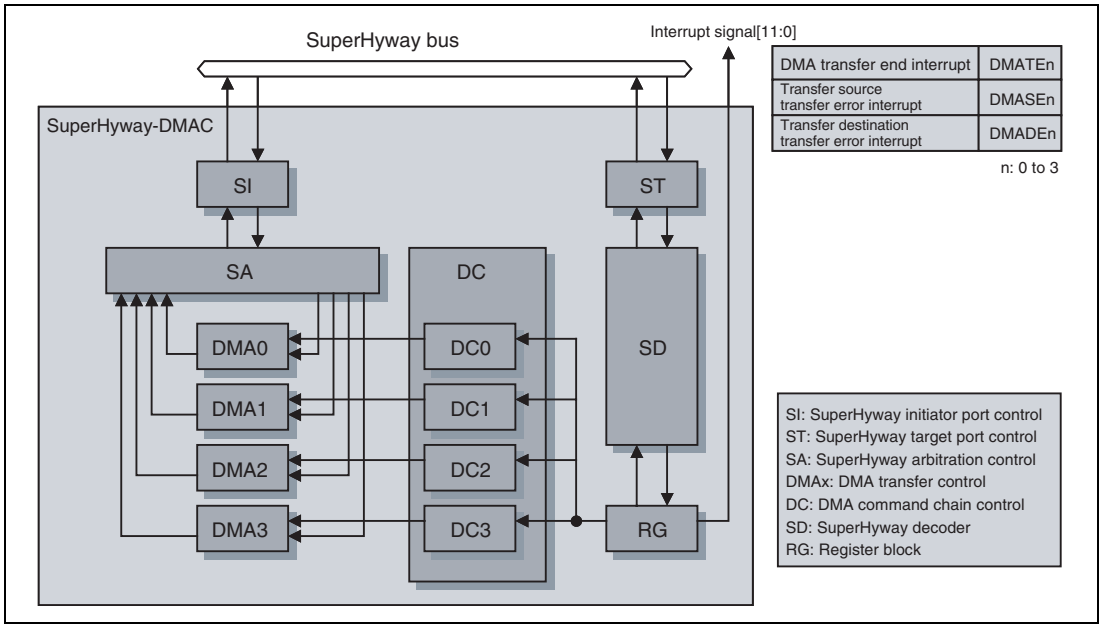


Figure 16.1 Block Diagram of DMAC1

16.1.3 External Pins

There are no external pins associated with the DMAC1.

16.1.4 Register Configuration

Table 16.1 shows the register configuration. For the relationship between a channel and its registers, DMA1SAR in ch0 is shown as DMA1SAR0. The registers should be accessed only by the access size shown in the table.

Table 16.1 Register Configuration

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
DMA1 operation register	DMA1OR	R/W	H'FEA0_0010	H'1EA0_00010	32
DMA1 source address registers 0 to 3	DMA1SAR0 to DMA1SAR3	R/W	ch0: H'FEA0_0020 ch1: H'FEA0_0120 ch2: H'FEA0_0220 ch3: H'FEA0_0320	ch0: H'1EA0_00020 ch1: H'1EA0_00120 ch2: H'1EA0_00220 ch3: H'1EA0_00320	32
DMA1 destination address registers 0 to 3	DMA1DAR0 to DMA1DAR3	R/W	ch0: H'FEA0_0028 ch1: H'FEA0_0128 ch2: H'FEA0_0228 ch3: H'FEA0_0328	ch0: H'1EA0_0028 ch1: H'1EA0_0128 ch2: H'1EA0_0228 ch3: H'1EA0_0328	32
DMA1 byte count registers 0 to 3	DMA1BCNTR0 to DMA1BCNTR3	R/W	ch0: H'FEA0_0030 ch1: H'FEA0_0130 ch2: H'FEA0_0230 ch3: H'FEA0_0330	ch0: H'1EA0_0030 ch1: H'1EA0_0130 ch2: H'1EA0_0230 ch3: H'1EA0_0330	32
DMA1 stride count registers 0 and 1	DMA1SBCNTR0 and DMA1SBCNTR1	R/W	ch0: H'FEA0_0034 ch1: H'FEA0_0134	ch0: H'1EA0_0034 ch1: H'1EA0_0134	32
DMA1 stride registers 0 and 1	DMA1STRR0 and DMA1STRR1	R/W	ch0: H'FEA0_0038 ch1: H'FEA0_0138	ch0: H'1EA0_0038 ch1: H'1EA0_0138	32
DMA1 command chain address registers 0 and 1	DMA1CCAR0 and DMA1CCAR1	R/W	ch0: H'FEA0_0040 ch1: H'FEA0_0140	ch0: H'1EA0_0040 ch1: H'1EA0_0140	32
DMA1 channel control registers 0 to 3	DMA1CHCR0 to DMA1CHCR3	R/W	ch0: H'FEA0_0048 ch1: H'FEA0_0148 ch2: H'FEA0_0248 ch3: H'FEA0_0348	ch0: H'1EA0_0048 ch1: H'1EA0_0148 ch2: H'1EA0_0248 ch3: H'1EA0_0348	32
DMA1 channel status registers 0 to 3	DMA1CHSR0 to DMA1CHSR3	R/(W)*	ch0: H'FEA0_004C ch1: H'FEA0_014C ch2: H'FEA0_024C ch3: H'FEA0_034C	ch0: H'1EA0_004C ch1: H'1EA0_014C ch2: H'1EA0_024C ch3: H'1EA0_034C	32
DMA1 source transfer size registers 2 and 3	DMA1STRS2 and DMA1STRS3	R/W	ch2: H'FEA0_0260 ch3: H'FEA0_0360	ch2: H'1EA0_0260 ch3: H'1EA0_0360	32
DMA1 destination transfer size registers 2 and 3	DMA1DTRS2 and DMA1DTRS3	R/W	ch2: H'FEA0_0270 ch3: H'FEA0_0370	ch2: H'1EA0_0270 ch3: H'1EA0_0370	32

Note: * To clear the SE, DE, and TE bits, only 1 can be written to.

Table 16.2 Register States in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep/ Light Sleep	Module Standby
DMA1 operation register	DMA1OR	H'00000000	Retained	Retained	Retained
DMA1 source address registers 0 to 3	DMA1SAR0 to DMA1SAR3	H'00000000	Retained	Retained	Retained
DMA1 destination address registers 0 to 3	DMA1DAR0 to DMA1DAR3	H'00000000	Retained	Retained	Retained
DMA1 byte count registers 0 to 3	DMA1BCNTR0 to DMA1BCNTR3	H'00000000	Retained	Retained	Retained
DMA1 stride count registers 0 and 1	DMA1SBCNTR0 and DMA1SBCNTR1	H'00000000	Retained	Retained	Retained
DMA1 stride registers 0 and 1	DMA1STRR0 and DMA1STRR1	H'00000000	Retained	Retained	Retained
DMA1 command chain address registers 0 and 1	DMA1CCAR0 and DMA1CCAR1	H'00000000	Retained	Retained	Retained
DMA1 channel control registers 0 to 3	DMA1CHCR0 to DMA1CHCR3	H'00000000	Retained	Retained	Retained
DMA1 channel status registers 0 to 3	DMA1CHSR0 to DMA1CHSR3	H'00000000	Retained	Retained	Retained
DMA1 source transfer size registers 2 and 3	DMA1STRS2 and DMA1STRS3	H'00000000	Retained	Retained	Retained
DMA1 destination transfer size registers 2 and 3	DMA1DTRS2 and DMA1DTRS3	H'00000000	Retained	Retained	Retained

16.2 Register Descriptions

The DMAC1 has the following registers.

(1) Common

- DMA1 operation register (DMA1OR)

(2) Channel 0

- DMA1 source address register 0 (DMA1SAR0)
- DMA1 destination address register 0 (DMA1DAR0)
- DMA1 byte count register 0 (DMA1BCNTR0)
- DMA1 stride count register 0 (DMA1SBCNTR0)
- DMA1 stride register 0 (DMA1STRR0)
- DMA1 command chain address register 0 (DMA1CCAR0)
- DMA1 channel control register 0 (DMA1CHCR0)
- DMA1 channel status register 0 (DMA1CHSR0)

(3) Channel 1

- DMA1 source address register 1 (DMA1SAR1)
- DMA1 destination address register 1 (DMA1DAR1)
- DMA1 byte count register 1 (DMA1BCNTR1)
- DMA1 stride count register 1 (DMA1SBCNTR1)
- DMA1 stride register 1 (DMA1STRR1)
- DMA1 command chain address register 1 (DMA1CCAR1)
- DMA1 channel control register 1 (DMA1CHCR1)
- DMA1 channel status register 1 (DMA1CHSR1)

(4) Channel 2

- DMA1 source address register 2 (DMA1SAR2)
- DMA1 destination address register 2 (DMA1DAR2)
- DMA1 byte count register 2 (DMA1BCNTR2)
- DMA1 channel control register 2 (DMA1CHCR2)
- DMA1 channel status register 2 (DMA1CHSR2)
- DMA1 source transfer size register 2 (DMA1STRS2)
- DMA1 destination transfer size register 2 (DMA1DTRS2)

(5) Channel 3

- DMA1 source address register 3 (DMA1SAR3)
- DMA1 destination address register 3 (DMA1DAR3)
- DMA1 byte count register 3 (DMA1BCNTR3)
- DMA1 channel control register 3 (DMA1CHCR3)
- DMA1 channel status register 3 (DMA1CHSR3)
- DMA1 source transfer size register 3 (DMA1STRS3)
- DMA1 destination transfer size register 3 (DMA1DTRS3)

Legends for register description:

Initial value:	Register value after a reset
—:	Undefined value
R/W:	Can be read from or written to; the written value can be read.
R/WC0:	Can be read from or written to; writing 0 initializes the bit but writing 1 is ignored.
R/WC1:	Can be read from or written to; writing 1 initializes the bit but writing 0 is ignored.
R:	Read-only; the write value should always be 0.

16.2.1 DMA1 Operation Register (DMA1OR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMA1E	Reserved														
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
31	DMA1E	0	R/W	DMA Master Enable Enables or disables DMA transfers. Set this bit to 1 to use the DMA transfer function. When this bit is cleared to 0, transfers on all channels are aborted. If transfer has been aborted, even if this bit is set to 1 again, the aborted DMA transfer will not be resumed. When transfer has been aborted, the transfers up to two addresses before the values indicated in DMA1DAR0 to DMA1DAR3 will be completed, but the data values of the previous and current addresses cannot be guaranteed.
30 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

16.2.2 DMA1 Source Address Registers 0 to 3 (DMA1SAR0 to DMA1SAR3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMA1SAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMA1SAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 0	DMA1SAR	All 0	R/W	These bits specify the transfer source address in DMA transfer. During DMA transfer, these bits indicate the transfer source address that is being issued currently. These bits correspond to a 32-bit physical address space. Only a 4-byte boundary address can be specified in DMA1SAR0 and DMA1SAR1, and bits 1 and 0 are reserved in these registers. In these registers, bits 1 and 0 are always read as 0 and the write value should always be 0.

16.2.3 DMA1 Destination Address Registers 0 to 3 (DMA1DAR0 to DMA1DAR3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMA1DAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMA1DAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 0	DMA1DAR	All 0	R/W	These bits specify the transfer destination address in DMA transfer. During DMA transfer, these bits indicate the transfer destination address that is being issued currently. These bits correspond to a 32-bit physical address space. Only a 4-byte boundary address can be specified in DMA1DAR0 and DMA1DAR1, and bits 1 and 0 are reserved in these registers. In these registers, bits 1 and 0 are always read as 0 and the write value should always be 0.

16.2.4 DMA1 Byte Count Registers 0 to 3 (DMA1BCNTR0 to DMA1BCNTR3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved			BCNT												
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BCNT															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 0	BCNT	All 0	R/W	These bits specify the transfer byte count. If these bits are specified as 0, 2^29 (= 536,870,912) bytes are transferred. Only a value in 4-byte units can be specified in DMA1BCNTR0 and DMA1BCNTR1, and bits 1 and 0 are reserved in these registers. In these registers, bits 1 and 0 are always read as 0 and the write value should always be 0. In ch0 and ch1, the number of bytes transferred from the transfer source is counted. In ch2 and ch3, the settings of the DMA1 source transfer size registers and DMA1 destination transfer size registers determine whether the number of bytes is counted at the transfer source or the transfer destination. (1) Source transfer size register ≥ Destination transfer size register The number of bytes transferred to the transfer destination is counted. (2) Source transfer size register < Destination transfer size register The number of bytes transferred from the transfer source is counted.

16.2.5 DMA1 Stride Count Registers 0 and 1 (DMA1SBCNTR0 and DMA1SBCNTR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SBCINI														Reserved	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SBCNT														Reserved	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 18	SBCINI	All 0	R/W	Initial Stride Counter These bits specify the initial value of the stride counter. The initial value of the number of data bytes to be transferred as a block in stride transfer or gather/scatter transfer is set. Note that only a multiple of four can be specified. A quarter of the target value (a value excluding the lower 2 bits) should be specified in this field.
17, 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 2	SBCNT	All 0	R/W	Stride Counter These bits specify the stride counter which indicates the number of data bytes to be transferred as a block in stride transfer or gather/scatter transfer. During data transfer, these bits indicate the number of remaining bytes to be transferred. When BCNT $\neq$ 0 and SBCNT = 0, data transfer is continued by loading the SBINI value. Note that only a multiple of four can be specified. A quarter of the target value (a value excluding the lower 2 bits) should be specified in this field.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

16.2.6 DMA1 Stride Registers 0 and 1 (DMA1STRR0 and DMA1STRR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SS														Reserved	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DS														Reserved	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 18	SS	All 0	R/W	Stride Width of Transfer Source Address These bits specify the stride width of the transfer source address. Note that only a multiple of four can be specified. A quarter of the target value (a value excluding the lower 2 bits) should be specified in this field.
17, 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 2	DS	All 0	R/W	Stride Width of Transfer Destination Address These bits specify the stride width of the transfer destination address. Note that only a multiple of four can be specified. A quarter of the target value (a value excluding the lower 2 bits) should be specified in this field.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

16.2.7 DMA1 Command Chain Address Registers 0 and 1 (DMA1CCAR0 and DMA1CCAR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCA															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCA											Reserved				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 5	CCA	All 0	R/W	Command Chain Address These bits specify the address of the command stream of the first command chain when command chains are executed. During command chain execution, these bits specify the address of the command chain to be executed next. Note that only a 32-byte boundary can be specified as a command chain address. This field specifies a value excluding the lower 5 bits.
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

16.2.8 DMA1 Channel Control Registers 0 to 3 (DMA1CHCR0 to DMA1CHCR3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CHE	Reserved	CCRE	Reserved			SASRE	DASRE	Reserved							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
31	CHE	0	R/W	DMA Channel Enable Enables or disables a channel. If this bit is set to 1, the data transfer of the corresponding channel is started. Note however that the data transfer is not performed while a bit indicating a transfer end (TE) or a transfer error (DE or SE) is set to 1. When this bit is cleared to 0, the data transfer will be aborted. If transfer has been aborted, even if this bit is set to 1 again, the aborted DMA transfer will not be resumed. When transfer has been aborted, the transfers up to two addresses before the values indicated in DMA1DAR0 to DMA1DAR3 will be completed, but the data values of the previous and current addresses cannot be guaranteed. This bit is not cleared to 0 when data transfer ends or is aborted, except for when an error response occurs in response to LOAD transfer performed for acquiring a command during command chain execution. 0: Disables data transfer 1: Enables data transfer
30	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
29	CCRE	0	R/W	Command Chain Enable Enables or disables a command chain. If a data transfer is requested while this bit is set to 1, the data transfer is performed by reading a command from an address specified by DCCAR. 0: Disables a command chain 1: Enables a command chain This bit is reserved in DMA1CHCR2 and DMA1CHCR3. In these registers, this bit is always read as 0 and the write value should always be 0.
28 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	SASRE	0	R/W	Transfer Source Address Stride Enable Enables or disables the stride registers for the transfer source address. 0: Disables SS bits in DMA1STRR0 and DMA1STRR1 1: Enables SS bits in DMA1STRR0 and DMA1STRR1 This bit is reserved in DMA1CHCR2 and DMA1CHCR3. In these registers, this bit is always read as 0 and the write value should always be 0.
24	DASRE	0	R/W	Transfer Destination Address Stride Enable Enables or disables the stride registers for the transfer destination address. 0: Disables DS bits in DMA1STRR0 and DMA1STRR1 1: Enables DS bits in DMA1STRR0 and DMA1STRR1 This bit is reserved in DMA1CHCR2 and DMA1CHCR3. In these registers, this bit is always read as 0 and the write value should always be 0.
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

16.2.9 DMA1 Channel Status Registers 0 to 3 (DMA1CHSR0 to DMA1CHSR3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved				SEE	Reserved	DEE	Reserved								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R/W	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				SE	Reserved	DE	Reserved					IE	Reserved		TE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W1C	R	R/W1C	R	R	R	R	R	R/W	R	R	R/W1C

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	SEE	0	R/W	Transfer Source Transfer Error Interrupt Enable Enables or disables an interrupt caused by a transfer error during data transfer from the transfer source. If this bit is set to 1, an interrupt is requested when the SE bit is set to 1. 0: Disables an interrupt request 1: Enables an interrupt request
26	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
25	DEE	0	R/W	Transfer Destination Transfer Error Interrupt Enable Enables or disables an interrupt caused by a transfer error during data transfer to the transfer destination. If this bit is set to 1, an interrupt is requested when the DE bit is set to 1. 0: Disables an interrupt request 1: Enables an interrupt request
24 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	Descriptions
11	SE	0	R/WC1	Transfer Source Transfer Error Flag Indicates that an address error has occurred during data transfer from the transfer source under the following conditions. • Value set in DMA1SAR0 to DMA1SAR3 does not match the transfer size boundary • Transfer source is an undefined space on the address map • Transfer source is in module stop mode When an error occurs, the DMA transfer being executed is halted. When this bit is set to 1, DMA transfers are disabled even if the CHE bit in DMA1CHCR0 to DMA1CHCR3 is set to 1. [Clearing condition] Writing 1 to this bit clears the flag. Writing 0 to this bit is ignored. 0 should be written to this bit except for when clearing this bit.
10	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
9	DE	0	R/WC1	Transfer Destination Transfer Error Flag Indicates that an address error has occurred during data transfer to the transfer destination under the following conditions. • Value set in DMA1DAR0 to DMA1DAR3 does not match the transfer size boundary • Transfer destination is an undefined space on the address map • Transfer destination is in module stop mode When an error occurs, the DMA transfer being executed is halted. When this bit is set to 1, DMA transfers are disabled even if the CHE bit in DMA1CHCR0 to DMA1CHCR3 is set to 1. [Clearing condition] Writing 1 to this bit clears the flag. Writing 0 to this bit is ignored. 0 should be written to this bit except for when clearing this bit.

Bit	Bit Name	Initial Value	R/W	Descriptions
8 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	IE	0	R/W	DMA Transfer End Interrupt Enable Enables or disables an interrupt caused by the end of a DMA transfer. If this bit is set to 1, an interrupt is requested when the TE bit is set to 1. 0: Disables an interrupt request 1: Enables an interrupt request
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TE	0	R/WC1	DMA Transfer End Interrupt Flag Indicates the transfer end flag. If a data transfer is completed after DMA1BCNTR0 to DMA1BCNTR3 are cleared to 0, this bit is set to 1. If a data transfer is terminated by a transfer error, or if a data transfer is forcibly terminated by clearing the CHE bit in DMA1CHCR0 to DMA1CHCR3 to 0, this bit will not be set to 1. In addition, if the next transfer is specified by using a command chain (CCRE = 1), this bit will not be set to 1 even if the current data transfer is completed normally. When this bit is set to 1, DMA transfers are disabled even if the CHE bit in DMA1CHCR0 to DMA1CHCR3 is set to 1. 0: Indicates that a data transfer is in progress or has been aborted 1: Indicates that a data transfer is completed (by DMA1BCNTR0 to DMA1BCNTR3 = 0) [Clearing condition] Writing 1 to this bit clears the flag. Writing 0 to this bit is ignored. 0 should be written to this bit except for when clearing this bit.

16.2.10 DMA1 Source Transfer Size Registers 2 and 3 (DMA1STRS2 and DMA1STRS3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													STRS		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	STRS	All 0	R/W	DMA Transfer Size for Transfer Source These bits set the DMA transfer size for the data transferred from the transfer source. 000: Byte units (Initial value) 001: Word (2-byte) units 010: Longword (4-byte) units 011: 8-byte units 100: Reserved (Setting prohibited) 101: 32-byte units 110: Reserved (Setting prohibited) 111: Reserved (Setting prohibited)

16.2.11 DMA1 Destination Transfer Size Registers 2 and 3 (DMA1DTRS2 and DMA1DTRS3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													DTRS		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	DTRS	All 0	R/W	DMA Transfer Size for Transfer Destination These bits set the DMA transfer size for the data transferred to the transfer destination. 000: Byte units (Initial value) 001: Word (2-byte) units 010: Longword (4-byte) units 011: 8-byte units 100: Reserved (Setting prohibited) 101: 32-byte units 110: Reserved (Setting prohibited) 111: Reserved (Setting prohibited)

16.3 Operation

Table 16.3 shows the transfer intervals supported by the DMAC1.

Table 16.3 DMAC1 Transfer Intervals

	Transfer Destination		
Transfer Source	**LBSC Space**	**DDR3-SDRAM Space**	**L Memory or L2C Memory**
LBSC space	Y	Y	Y
DDR3-SDRAM space	Y	Y	Y
L memory or L2C memory	Y	Y	Y

Legend:
Y: Transfer is enabled

Note: When performing DMA transfer to/from the LBSC space (when MPX or burst ROM interface is set), L memory, or L2C memory using ch0 or ch1, the registers must be set in accordance with the following conditions. A setting not complying with the following conditions is prohibited.
(n: Positive integer)

1. For contiguous region transfer
   - When L memory, L2C memory, or LBSC space is the transfer source, DMA1SAR0 and DMA1SAR1 must be set to a 32-byte boundary.
   - When L memory, L2C memory, or LBSC space is the transfer destination, DMA1DAR0 and DMA1DAR1 must be set to a 32-byte boundary.
   - DMA1BCNTR0 and DMA1BCNTR1 must be set to 32n bytes (bits 4 to 0 in DMA1BCNTR0 and DMA1BCNTR1 are 0).
2. For stride, gather, or scatter transfer
   - When L memory, L2C memory, or LBSC space is the transfer source, DMA1SAR0 and DMA1SAR1 must be set to a 32-byte boundary, and the SS bits in DMA1STRR0 and DMA1STRR1 must be set to 32n bytes (bits 20 to 16 in DMA1STRR0 and DMA1STRR1 are 0).
   - When L memory, L2C memory, or LBSC space is the transfer destination, DMA1DAR0 and DMA1DAR1 must be set to a 32-byte boundary, and the DS bits in DMA1STRR0 and DMA1STRR1 must be set to 32n bytes (bits 4 to 0 in DMA1STRR0 and DMA1STRR1 are 0).
   - DMA1BCNTR0 and DMA1BCNTR1 must be set to 32n bytes (bits 4 to 0 in DMA1BCNTR0 and DMA1BCNTR1 are 0).

- Bits SBCINI and SBCNT in DMA1SBCNTR0 and DMA1SBCNTR1 must be set to 32n bytes (bits 20 to 16 and 4 to 0 in DMA1SBCNTR0 and DMA1SBCNTR1 are 0).

When performing DMA transfer to/from the LBSC space (when MPX or burst ROM interface is set) using ch2 or ch3 and with the DMA transfer size set to 32 bytes, the registers must be set in accordance with the following condition. A setting not complying with the following condition is prohibited.

- When the LBSC space is the transfer source or transfer destination, DMA1SAR2 and DMA1SAR3 must be set to a 32-byte boundary.

Table 16.4 shows the relationship between the operation type and control register setting. The operation is controlled by four bits: bits 31, 29, 25, and 24 in DMA1 channel control registers 0 to 3 (DMA1CHCR0 to DMA1CHCR3). Ch2 and ch3 can only be used for contiguous region transfer.

Table 16.4 Relationship between Operation Type and Control Register Setting

	Bit 31: CHE	Bit 29: CCRE	Bit 25: SASRE	Bit 24: DASRE
Operation Type	**DMA Channel Enable**	**Command Chain Enable**	**Transfer Source Address Stride Enable**	**Transfer Destination Address Stride Enable**
No operation	0	—	—	—
Contiguous region transfer	1	0	0	0
Stride transfer*	1	0	1	1
Gather transfer*	1	0	1	0/1
Scatter transfer*	1	0	0/1	0/1
Command chain*	1	1	—	—
Setting prohibited	Combinations other than above			

Legend:

—: Don't care

Note: * Can be set only in ch0 and ch1.

In gather transfer, the transfer destination start address is the same in any block when the DASRE bit in DMA1CHCR is 0, and when the DASRE bit in DMA1CHCR is 1 and the DS bits in DMA1STRR is 0.

Similar in scatter transfer, the transfer source start address is the same in any block when the SASRE bit in DMA1CHCR is 0, and when the SASRE bit in DMA1CHCR is 1 and the SS bits in DMA1STRR is 0.

16.3.1 Channel Priority

When the DMAC1 receives transfer requests on two or more channels, every time data transfer for a read cycle or write cycle of one transfer unit (byte, word, longword, 8-byte, 16-byte, or 32-byte units) has finished, data transfer is started on the channel which has the highest priority among the channels in which transfer is enabled.

When the DMAC1 receives transfer requests on two or more channels simultaneously, data transfer is started according to the determined priority.

Simultaneous transfer requests are handled in the order of CH3 > CH2 > CH1 > CH0.

16.3.2 Contiguous Region Transfer (ch0 to ch3)

After transfer conditions have been set in registers and contiguous region transfer has been specified in a DMA1 channel control register, data transfer is performed with the following procedure. The registers for ch0 are set in the following procedure. Similar settings should be made when using ch1 to ch3.

1. Check if transfer is enabled.

   When DMA1OR.DMAE = 1, DMA1CHCR0.CHE = 1, DMA1CHSR0.SE = 0, DMA1CHSR0.DE = 0, and DMA1CHSR0.TE = 0, transfer is enabled. According to table 16.4, set DMA1 channel control register 0 to specify contiguous region transfer.
2. If transfer is enabled, data transfer starts. The DMA1BCNTR0 value is decremented for each transfer.
3. When the specified number of bytes has been transferred (when DMA1BCNTR0 = 0), the transfer ends normally and the TE bit in DMA1CHSR0 is set to 1. If the IE bit in DMA1CHSR0 is set to 1 at this time, a DMA transfer end interrupt is sent to the CPU.

   If a transfer error occurs, the data transfer is aborted. At this time, the SE bit or DE bit in DMA1CHSR0 is set to 1. Data transfers are also aborted when the CHE bit in DMA1CHCR0 is cleared to 0.

In ch0 and ch1, the transfer size is automatically determined by the addresses set in DMA1SAR0, DMA1SAR1, DMA1DAR0, and DMA1DAR1, and the remaining data size that is still to be transferred.

In ch2 and ch3, data transfer is performed with the transfer size set in DMA1STRS2, DMA1STRS3, DMA1DTRS2, and DMA1DTRS3. Note however that if an address that does not match the address boundary corresponding to the specified transfer size is set in DMA1SAR2, DMA1SAR3, DMA1DAR2, or DMA1DAR3, transfer is performed in 1-byte units until reaching the address boundary corresponding to the specified transfer size. If the size of the remaining transfer data is smaller than the specified transfer size, transfer is also performed in 1-byte units until the transfer finishes.

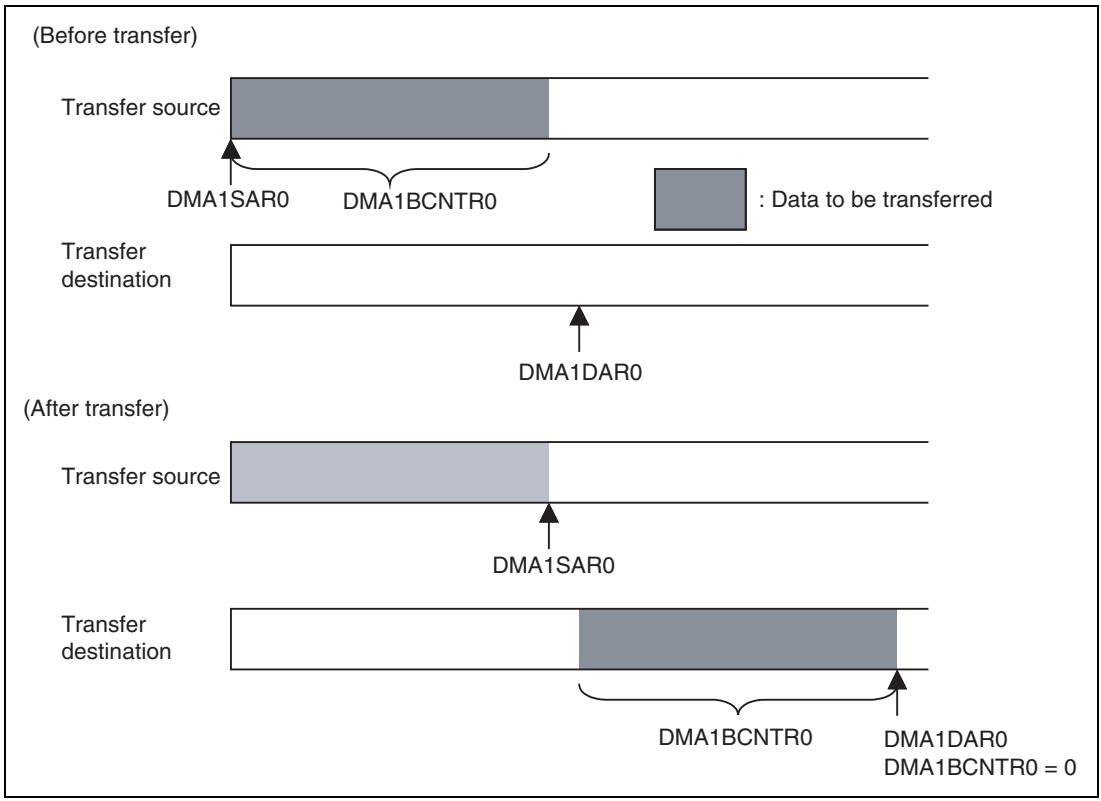


Figure 16.2 Contiguous Region Transfer

16.3.3 Stride, Gather, or Scatter Transfer (ch0 and ch1)

After transfer conditions have been set in registers and stride, gather, or scatter transfer has been specified in a DMA1 channel control register, data transfer is performed with the following procedure. The registers for ch0 are set in the following procedure. Similar settings should be made when using ch1.

1. Check if transfer is enabled.

   When DMA1OR.DMAE = 1, DMA1CHCR0.CHE = 1, DMA1CHSR0.SE = 0, DMA1CHSR0.DE = 0, and DMA1CHSR0.TE = 0, transfer is enabled. According to table 16.4, set DMA1 channel control register 0 to specify stride, gather, or scatter transfer.
2. If transfer is enabled, data transfer starts. The DMA1BCNTR0 value and the value of the SBCNT bits in DMA1SBCNTR0 are decremented for each transfer.
3. When DMA1BCNTR0 is not 0 and the SBCNT bits in DMA1SBCNTR0 are 0, the DS bit value in DMA1STRR0 is added to DMA1DAR0 and the SS bit value in DMA1STRR is added to DMA1SAR0 to determine the next transfer source and transfer destination addresses.

   Then, the value set in the SBCINI bits in DMA1SBCNTR0 is set to the SBCNT bits in DMA1SBCNTR0 and the processing returns to step 2.
4. When the specified number of bytes has been transferred (when DMA1BCNTR0 = 0), the transfer ends normally and the TE bit in DMA1CHSR0 is set to 1. If the IE bit in DMA1CHSR0 is set to 1 at this time, a DMA transfer end interrupt is sent to the CPU.

   If a transfer error occurs, the data transfer is aborted. At this time, the SE bit or DE bit in DMA1CHSR0 is set to 1. Data transfers are also aborted when the CHE bit in DMA1CHCR0 is cleared to 0.

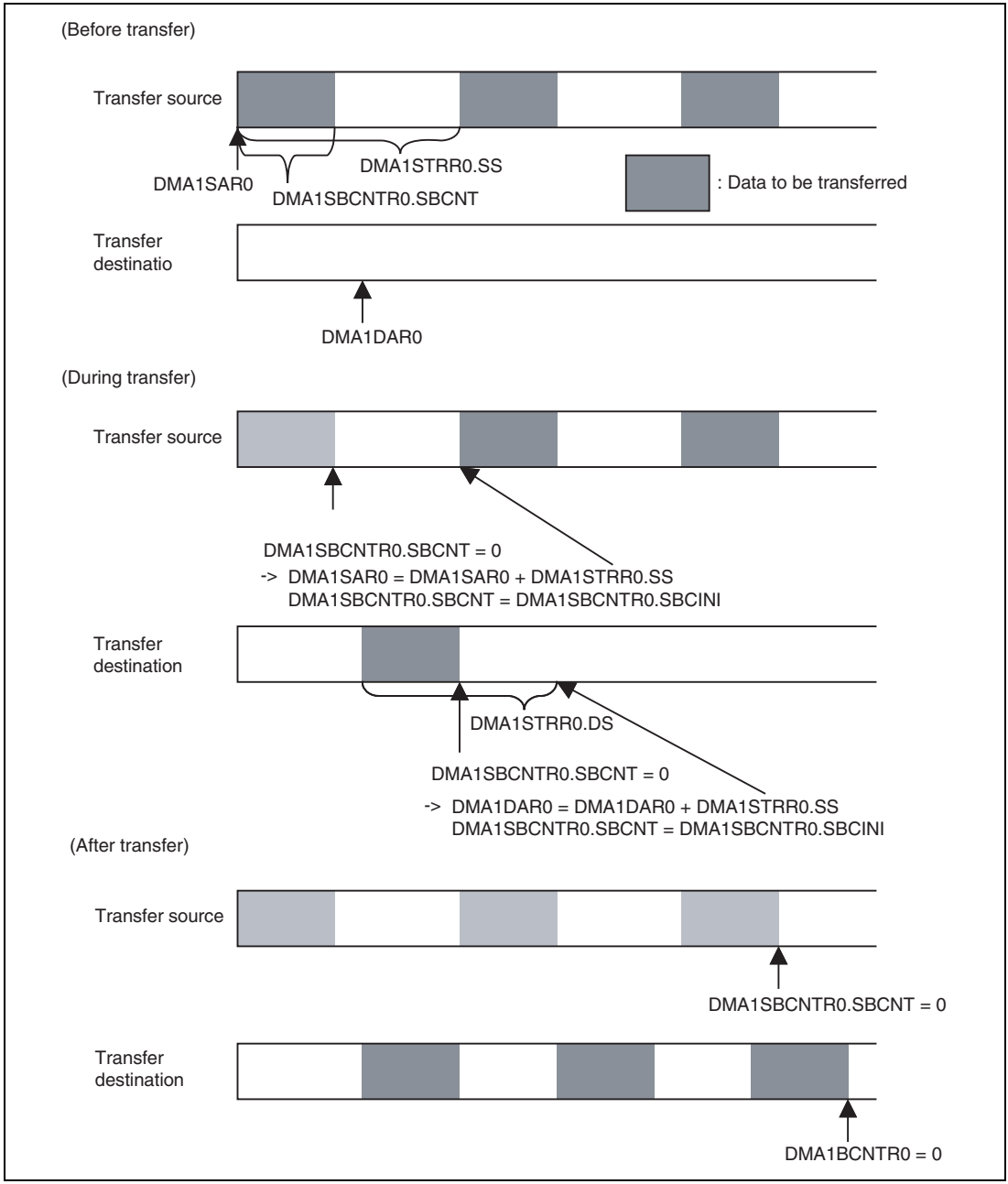


Figure 16.3 Stride Transfer

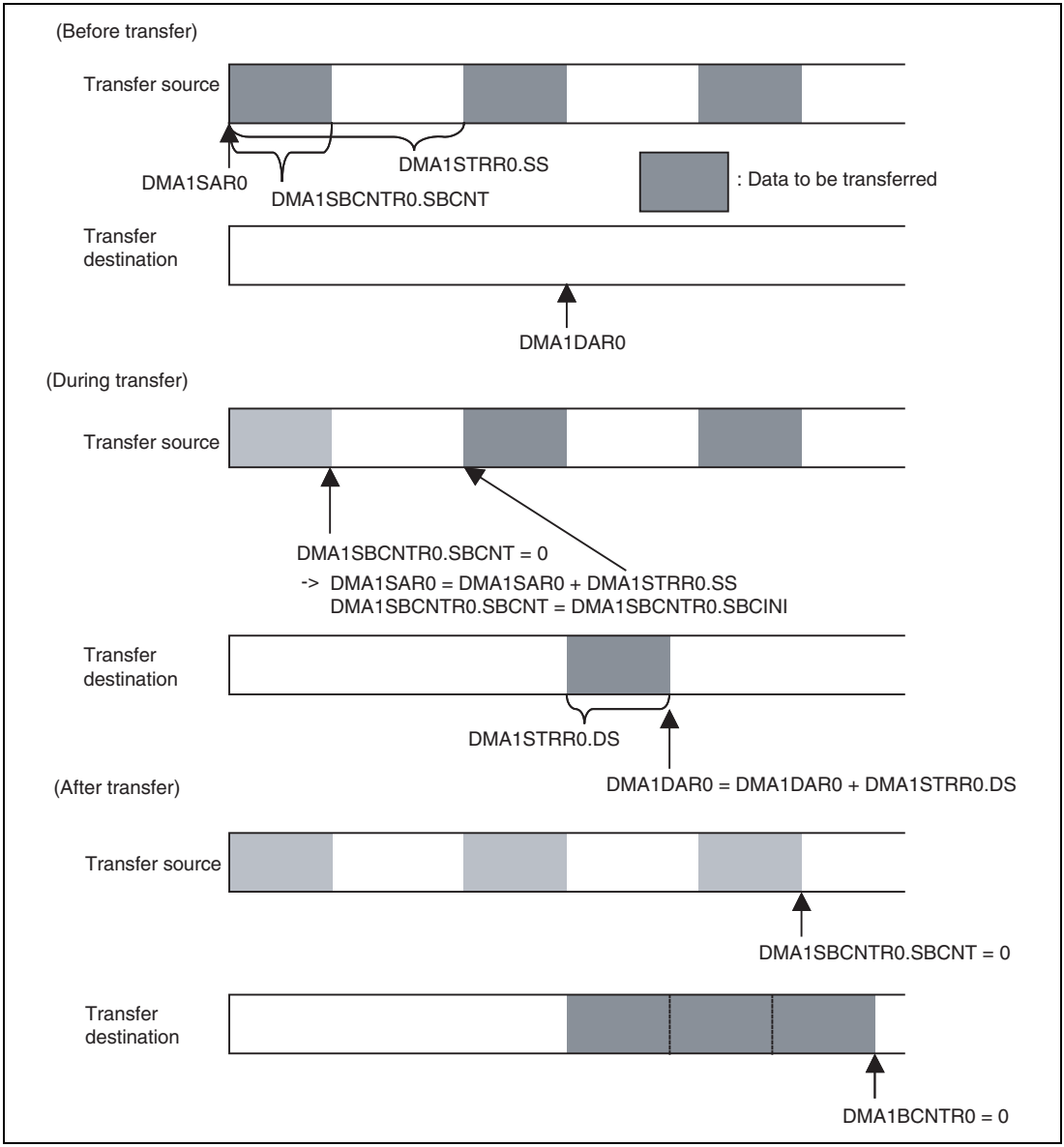


Figure 16.4 Gather Transfer

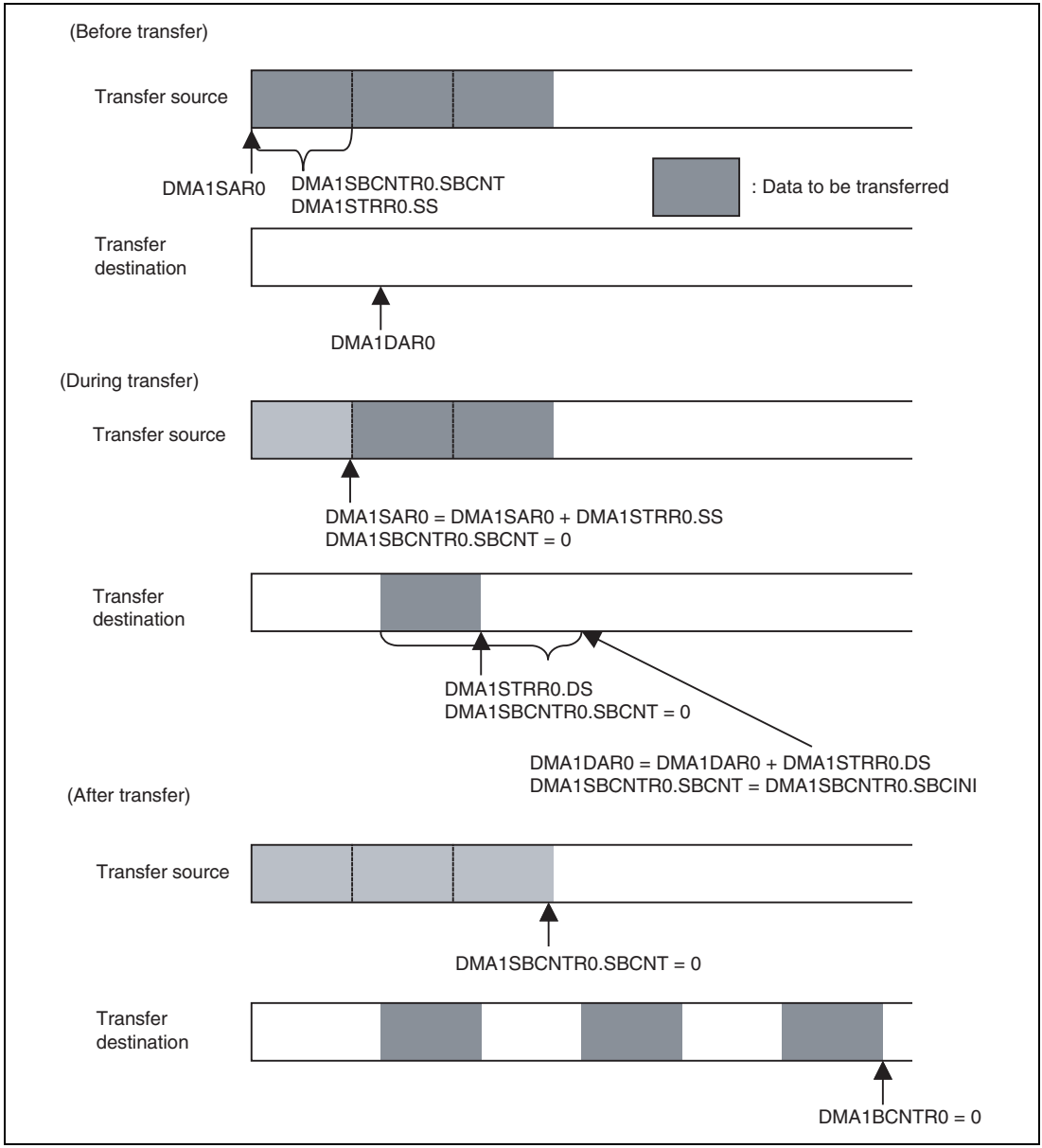


Figure 16.5 Scatter Transfer

16.3.4 Command Chain (ch0 and ch1)

After transfer conditions have been set in registers and a command chain has been specified in a DMA1 channel control register, data transfer is performed with the following procedure. The registers for ch0 are set in the following procedure. Similar settings should be made when using ch1.

1. Check if transfer is enabled.

   When DMA1OR.DMAE = 1, DMA1CHCR0.CHE = 1, DMA1CHSR0.SE = 0, DMA1CHSR0.DE = 0, and DMA1CHSR0.TE = 0, transfer is enabled. According to table 16.4, set DMA1 channel control register 0 to specify a command chain.
2. If transfer is enabled, read a data transfer command from the address set in DMA1CCAR and set the registers.
3. Execute the specified data transfer command.
4. Even if execution of the specified data transfer command ends normally, if the CCRE bit in DMA1CHCR0 is 1, there is a command that needs to be executed next. In such a case, the TE bit in DMA1CHSR0 is not set to 1 and no interrupt is generated even if the IE bit in DMA1CHSR0 is 1.
5. When DMA1CHCR0.CHE = 1, DMA1CHSR0.DE = 0, DMA1CHSR0.SE = 0, DMA1CHSR0.TE = 0, and DMA1CHCR0.CCRE = 1, the processing returns to step 2.
6. If the CCRE bit in DMA1CHCR0 is 0 when execution of the data transfer command has finished, data transfer by command chains is completed. At this time, the TE bit in DMA1CHSR0 is set to 1, and a DMA transfer end interrupt is generated if the IE bit in DMA1CHSR0 is 1.

   If a transfer error occurs, the data transfer is aborted. At this time, the SE bit or DE bit in DMA1CHSR0 is set to 1. Data transfers are also aborted when the CHE bit in DMA1CHCR0 is cleared to 0.

   If an error occurs while reading the data transfer command in step 2, the SE bit in DMA1CHSR0 is set to 1 and the CHE bit in DMA1CHCR0 is cleared to 0.

Figure 16.6 shows the command stream format.

The CHE bit that is set at H'00 must always be set to 1. In the last command stream of the command chain, 27'H0 must always be set to the CCA bits that are set at H'10.

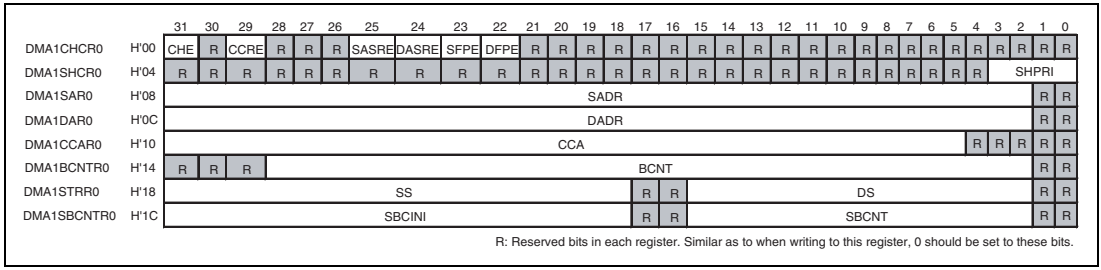


Figure 16.6 Command Stream Format in Command Chain

In a command chain, indicating the start address of the next command in the CCA bits enables DMA transfers to be performed continuously, as shown in figure 16.7.

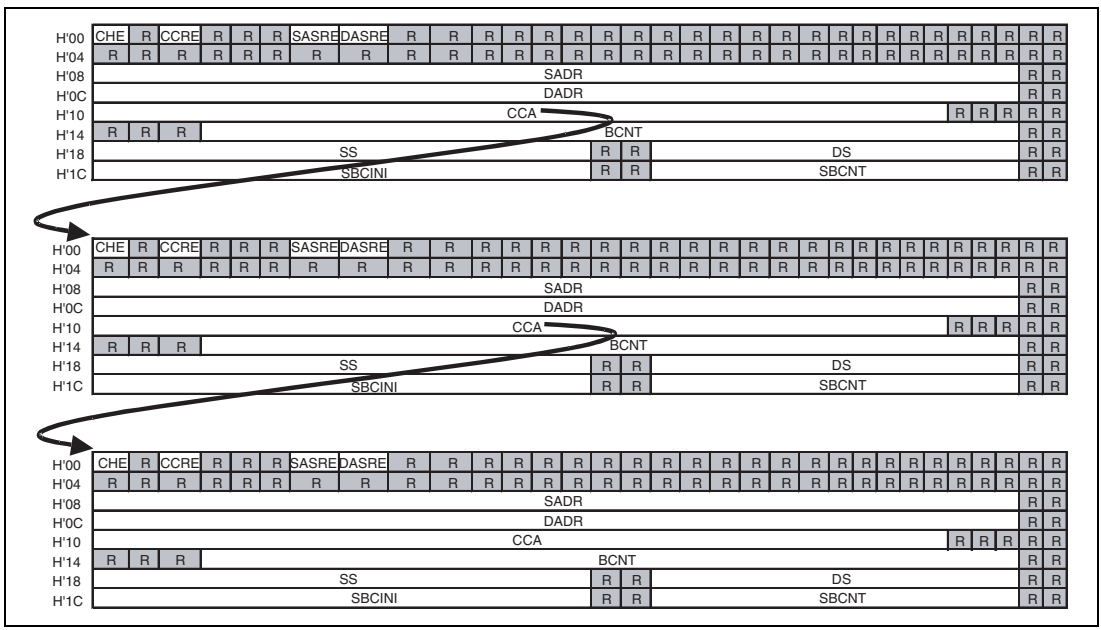


Figure 16.7 Command Chain

16.4 DMAC1 Interrupt Sources

The DMAC1 has three types of interrupt sources, one for each channel: DMA transfer end interrupt request (DMA1TEn), transfer source transfer error interrupt request (DMA1SEn), and transfer destination transfer error interrupt request (DMA1DEn) (n = 0 to 3).

Table 16.5 lists the interrupt sources. Interrupt sources are sent independently to the interrupt controller.

Table 16.5 DMAC1 Interrupt Sources

Interrupt Source	Description
DMA1TE0	Channel 0 DMA transfer end interrupt
DMA1TE1	Channel 1 DMA transfer end interrupt
DMA1TE2	Channel 2 DMA transfer end interrupt
DMA1TE3	Channel 3 DMA transfer end interrupt
DMA1SE0	Channel 0 transfer source transfer error interrupt
DMA1SE1	Channel 1 transfer source transfer error interrupt
DMA1SE2	Channel 2 transfer source transfer error interrupt
DMA1SE3	Channel 3 transfer source transfer error interrupt
DMA1DE0	Channel 0 transfer destination transfer error interrupt
DMA1DE1	Channel 1 transfer destination transfer error interrupt
DMA1DE2	Channel 2 transfer destination transfer error interrupt
DMA1DE3	Channel 3 transfer destination transfer error interrupt

16.5 Usage Notes

Note the following when using this DMAC1.

(1) Restart of DMA Transfer

In the following cases, subsequent DMA transfers cannot be resumed when a DMA transfer is finished in the middle of processing.

- When the DMAE bit in DMA1OR is cleared to 0
- When the CHE bit in DMA1CHCRn (n: 0 to 3) is cleared to 0
- When the value set in DMA1SARn or DMA1DARn (n: 0 to 3) does not match the transfer size boundary
- When the transfer destination is an undefined space on the address map
- When the transfer destination or transfer source is in module stop mode
- When the frequency is changed

To re-execute DMA transfer, set the registers again to start DMA transfer.

The procedure for setting the registers again is shown below. The values to be set again in the registers are determined from the DMA1DAR0 to DMA1DAR3 values at the time DMA transfer was aborted.

1. Confirm the DMA1DAR0 to DMA1DAR3 values, determine the values to be set again in DMA1SAR0 to DMA1SAR3 and DMA1DAR0 to DMA1DAR3, and set these registers.

   In ch0 and ch1, subtract 7'h20 (corresponding to 32-byte transfer data) from DMA1DAR0 and DMA1DAR1, and set the obtained values to DMA1DAR0 and DMA1DAR1. Set addresses that correspond to DMA1DAR0 and DMA1DAR1 to DMA1SAR0 and DMA1SAR1.

   In ch2 and ch3, subtract the number of bytes set in DMA1DTRS2 and DMA1DTRS3 from DMA1DAR2 and DMA1DAR23, and set the obtained values to DMA1DAR2 and DMA1DAR3. Set addresses that correspond to DMA1DAR2 and DMA1DAR3 to DMA1SAR2 and DMA1SAR3.
2. Set values to DMA1BCNTR0 to DMA1BCNTR3.
3. Check if transfer is enabled.

   When DMA1OR.DMAE = 1, DMA1CHCR0.CHE = 1, DMA1CHSR0.SE = 0, DMA1CHSR0.DE = 0, and DMA1CHSR0.TE = 0, transfer is enabled and data transfer starts.

(2) Module Stop Function and Frequency Change

While the DMAC1 is operating, if the standby control register (MSTPCR1) is used to start operation (clear the bit to 0) or stop operation (set the bit to 1) or if the frequency is changed, the operation of this LSI is not guaranteed.

Check that the DMAE bit (bit 31) in DMA1OR is 0 before setting the standby control register (MSTPCR1) or changing the frequency.

Section 17 HPB-DMAC

17.1 Overview

The HPB-DMAC performs DMA transfer between peripherals on the HPB bus and DDR3-SDRAM. The HPB-DMAC is assigned channel numbers 0 to 13, so that a total of fourteen channels are provided within the LSI. For each of these DMA channels, a different transfer destination can be selected, for independent parallel operation.

A separate data transfer mode is selectable for each channel. In this section, the character "n" refers to one of the fourteen DMA channels for the DMAC.

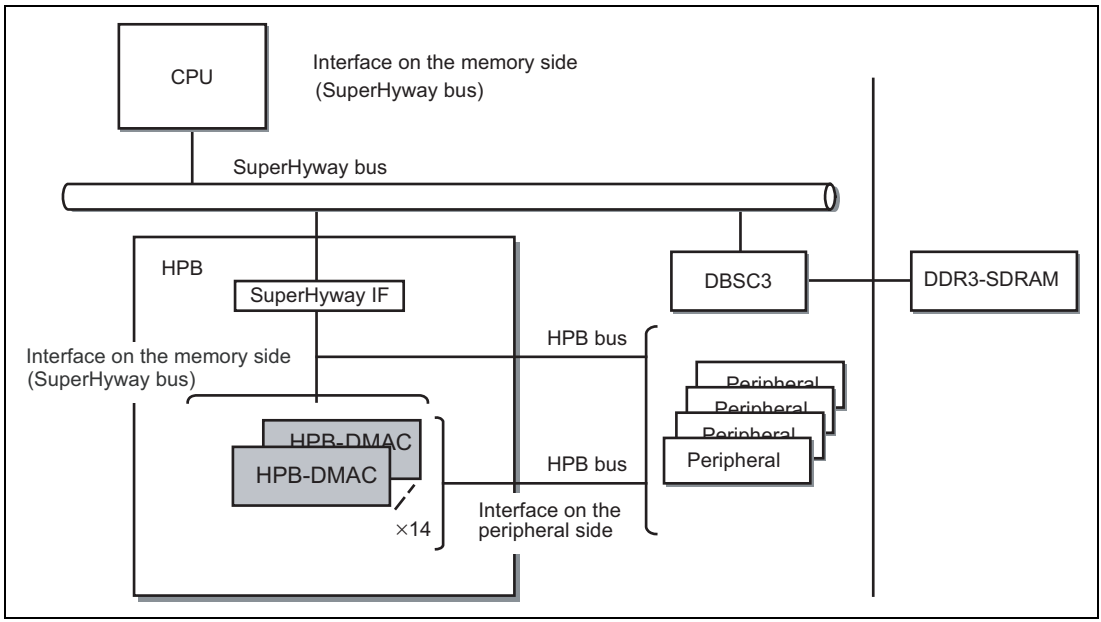


Figure 17.1 Context of HPB-DMAC

Figure 17.1 shows the context of the HPB-DMAC. The DMAC channels are connected to DDR3-SDRAM outside the chip via the DBSC3, connected via the memory (SuperHyway bus)-side interface, and to peripherals connected to the HPB bus via the peripheral-side interface. Selection of these peripherals is performed by setting registers within each DMAC channel.

17.2 Features

The HPB-DMAC has the following features.

- Fourteen channels (channels 0 to 13)
- Physical address space
- Capable of transfer from a peripheral module to a memory (SuperHyway bus), or from a memory (SuperHyway bus) to a peripheral module
- Transfer data length:
  - For peripherals: Selectable from 1, 2, or 4 bytes.
  - For memory (SuperHyway bus):

    When the SWMD bit in DCR register is set to 0: Memory (SuperHyway) access size (for access size of each channel, see section 17.4.10.)

    When the SWMD bit in DCR register is set to 1: Selectable from 1, 2, or 4 bytes

    When the PKMD bit in DCR register is set to 1 and final round of packing: Selectable from 1 byte to memory (SuperHyway) access size
- Transfer burst length: 1 or 8 (8-burst transfer is available only in HPB-DMAC channels 10 to 13)
- Transfer count
  - Maximum: 16 M (16,777,216) times
  - Minimum: One time
- Dual address mode

  Both the transfer source and transfer destination are accessed by using addresses.

  Values set in the DMAC internal registers indicate the addresses for both the transfer source and the transfer destination. (Registers: DMA source address register (DSAR0 or DSAR1), DMA destination address register (DDAR0 or DDAR1), and DMA control register (DCR; bits SPDAM and DPDAM control the addresses).
- Transfer requests: Peripheral requests, auto-requests, and timer requests are supported.
  - Peripheral request

    Transfer requests from on-chip peripheral modules.
  - Auto-request

    Initiates DMA transfer according to the DMAC internal timing.
  - Timer request

    A transfer request is generated at an interval specified by a timer in the DMAC.

- Transfer modes: Single transfer and continuous transfer modes are supported.
  - — Single transfer mode

    DMA transfer ends when transfer is completed for the transfer count specified by the DMA transfer count register.
  - — Continuous transfer mode

    Available in all channels. If there is a next DMA transfer request (DNXT) when transfer is completed for the transfer count specified by the DMA transfer count register, the next DMA transfer information is fetched and the next DMA transfer is continued. If no next DMA transfer request (DNXT) is found, the DMAC waits until a next DMA transfer request is specified. The continuous transfer mode is terminated by the DQEND bit setting in the DMA command register (DCMDR).

    DMA information can be specified in two modes: one mode uses one of two sets of DMA information registers repeatedly, and the other uses the two sets alternately.

    The DMAC also provides the automatic continuous transfer mode. The automatic continuous transfer mode is enabled by setting the ACMD bit to 1 in the DMA control register (DCR) while the continuous transfer mode is enabled (the CT bit = 1 in DCR). In this mode, when transfer is completed for the transfer count specified by the DMA transfer count register, the DMAC fetches the next DMA transfer information and continues DMA transfer regardless of whether there is a next DMA transfer request (DNXT). This mode is terminated by the DQEND bit setting in the DMA command register (DCMDR).
- Transfer end interrupt: An interrupt request can be sent to the CPU on completion of the number of transfers specified for each DMA information unit.

17.3 Specifications of DMA Transfer by HPB-DMAC

The relation of each DMA channel to peripherals is shown below. Transfers can be performed with various functional blocks over different channels, but in the case of memory as a transfer destination (SuperHyway bus), DDR3-SDRAM or similar is set as the transfer destination via the DBSC according to the address set.

Channel	Application	Communication Type	Selection of Communication Remote Device
HPB-DMA00	Communications with peripherals via the internal HPB bus	Dual address transfer, single transfer or 8-burst transfer * DREQ/DACK handshake	Internal peripheral for communication is selected using the internal registers of each DMAC. (SCIF or HSPI should be selected.)
HPB-DMA01			
HPB-DMA02			
HPB-DMA03			
HPB-DMA04			
HPB-DMA05			
HPB-DMA06			
HPB-DMA07	Communications with peripherals via the internal HPB bus		Internal peripheral for communication is selected using the internal registers of each DMAC. (SSI, HAC, and SDIF should be selected.)
HPB-DMA08			
HPB-DMA09			
HPB-DMA10			
HPB-DMA11			
HPB-DMA12	Communications with peripherals via the internal HPB bus	Dual address transfer, single/8-burst transfer DREQ/DACK handshake	Internal peripheral for communication is selected using the internal registers of each DMAC. (USB-FUNC)
HPB-DMA13			

Note: * Eight-burst transfer is supported for channels 10 and 11 only, and only when the transfer partner is SDIF.

17.4 Register Configuration

The following registers are all mapped to the SH register map space. The DMAC has fourteen channels; some registers are prepared for each individual channel and some are used by all channels in common.

17.4.1 List of HPB-DMAC Registers

Table 17.1 List of HPB-DMAC Registers (1)

Address (Bytes)	Name		Abbreviation	Access Type	Access Size
H'FFC00300	SD mode select register		SDMDR	R/W	32
H'FFC08000 + H'40 × [n]	[Individual] DMA source address register 0	DMA information register set 0	DSAR0	R/W	32
H'FFC08004 + H'40 × [n]	[Individual] DMA destination address register 0		DDAR0	R/W	32
H'FFC08008 + H'40 × [n]	[Individual] DMA transfer count register 0		DTCR0	R/W	32
H'FFC0800C + H'40 × [n]	[Individual] DMA source address register 1	DMA information register set 1	DSAR1	R/W	32
H'FFC08010 + H'40 × [n]	[Individual] DMA destination address register 1		DDAR1	R/W	32
H'FFC08014 + H'40 × [n]	[Individual] DMA transfer count register 1		DTCR1	R/W	32
H'FFC08018 + H'40 × [n]	[Individual] DMA source address status register		DSASR	R	32
H'FFC0801C + H'40 × [n]	[Individual] DMA destination address status register		DDASR	R	32
H'FFC08020 + H'40 × [n]	[Individual] DMA transfer count status register		DTCSR	R	32
H'FFC08024 + H'40 × [n]	[Individual] DMA port select register		DPTR	R/W	32
H'FFC08028 + H'40 × [n]	[Individual] DMA control register		DCR	R/W	32
H'FFC0802C + H'40 × [n]	[Individual] DMA command register		DCMDR	—/W	32
H'FFC08030 + H'40 × [n]	[Individual] DMA forced stop register		DSTPR	—/W	32

Address (Bytes)	Name	Abbreviation	Access Type	Access Size
H'FFC08034 + H'40 × [n]	[Individual] DMA status register	DSTSR	R	32
H'FFC08038 + H'40 × [n]	[Individual] DMA channel debug register	DDBGR	R/W	32
H'FFC0803C + H'40 × [n]	[Individual] DMA channel debug register 2	DDBGR2	R/W	32
H'FFC08800	[Common to HPB-DMAC] DMA timer control register	DTIMR	R/W	32
H'FFC0880C	[Common to HPB-DMAC] DMA transfer end interrupt status register	DINTSR	R	32
H'FFC08810	[Common to HPB-DMAC] DMA transfer end interrupt status clear register	DINTCR	R/WC1	32
H'FFC08814	[Common to HPB-DMAC] DMA transfer end interrupt enable register	DINTMR	R/W	32
H'FFC08818	[Common to HPB-DMAC] DMA activation status register	DACTSR	R	32
H'FFC0881C to H'FFC08850	[Common to HPB-DMAC] HPB-DMA00 to HPB-DMA13 channel software-reset register	HSRSTR0 to HSRSTR13	R/WC1	32
H'FFC08890, H'FFC08894	[Common to HPB-DMAC] HPB-DMA SuperHyway priority control registers 0, 1	HPB-DMASPR0	R/W	32

Legend:

[n]: HPB-DMAC Channel Number

Note: The above registers should be accessed in longword (32 bits) units; byte access and word access are prohibited.

Table 17.2 List of HPB-DMAC Registers (2)

Address (Bytes)	Name		Abbreviation	Power-On Reset by $\overline{\text{RESET}}$ pin, WDT, or H-UDI	Manual Reset by $\overline{\text{RESET}}$ pin or WDT or multiple exception
H' FFC00300	SD mode select register		SDMDR	H'0000_0000	H'0000_0000
H' FFC08000 + H'40 × [n]	[Individual] DMA source address register 0	DMA information register set 0	DSAR0	H'0000_0000	H'0000_0000
H'FFC08004 + H'40 × [n]	[Individual] DMA destination address register 0		DDAR0	H'0000_0000	H'0000_0000
H'FFC08008 + H'40 × [n]	[Individual] DMA transfer count register 0		DTCR0	H'0000_0000	H'0000_0000
H'FFC0800C + H'40 × [n]	[Individual] DMA source address register 1	DMA information register set 1	DSAR1	H'0000_0000	H'0000_0000
H'FFC08010 + H'40 × [n]	[Individual] DMA destination address register 1		DDAR1	H'0000_0000	H'0000_0000
H'FFC08014 + H'40 × [n]	[Individual] DMA transfer count register 1		DTCR1	H'0000_0000	H'0000_0000
H'FFC08018 + H'40 × [n]	[Individual] DMA source address status register		DSASR	H'0000_0000	H'0000_0000
H'FFC0801C + H'40 × [n]	[Individual] DMA destination address status register		DDASR	H'0000_0000	H'0000_0000
H'FFC08020 + H'40 × [n]	[Individual] DMA transfer count status register		DTCSR	H'0000_0000	H'0000_0000
H'FFC08024 + H'40 × [n]	[Individual] DMA port select register		DPTR	H'0000_0000	H'0000_0000
H'FFC08028 + H'40 × [n]	[Individual] DMA control register		DCR	H'0000_0000	H'0000_0000
H'FFC0802C + H'40 × [n]	[Individual] DMA command register		DCMDR	H'0000_0000	H'0000_0000
H'FFC08030 + H'40 × [n]	[Individual] DMA forced stop register		DSTPR	H'0000_0000	H'0000_0000
H'FFC08034 + H'40 × [n]	[Individual] DMA status register		DSTSR	H'0000_0020	H'0000_0020
H'FFC08038 + H'40 × [n]	[Individual] DMA channel debug register		DDBGR	H'0000_0000	H'0000_0000

Address (Bytes)	Name	Abbreviation	Power-On Reset by RESET pin, WDT, or H-UDI	Manual Reset by RESET pin or WDT or multiple exception
H'FFC0803C + H'40 × [n]	[Individual] DMA channel debug register 2	DDBGR2	H'0000_0000	H'0000_0000
H'FFC08800	[Common to HPB-DMAC] DMA timer control register	DTIMR	H'0000_0000	H'0000_0000
H'FFC0880C	[Common to HPB-DMAC] DMA transfer end interrupt status register	DINTSR	H'0000_0000	H'0000_0000
H'FFC08810	[Common to HPB-DMAC] DMA transfer end interrupt status clear register	DINTCR	H'0000_0000	H'0000_0000
H'FFC08814	[Common to HPB-DMAC] DMA transfer end interrupt enable register	DINTMR	H'0000_0000	H'0000_0000
H'FFC08818	[Common to HPB-DMAC] DMA activation status register	DACTSR	H'0000_0000	H'0000_0000
H'FFC0881C to H'FFC08850	[Common to HPB-DMAC] HPB-DMA00 to HPB-DMA13 channel software-reset register	HSRSTR0 to HSRSTR13	H'0000_0000	H'0000_0000
H' FFC08890	[Common to HPB-DMAC] HPB-DMA SuperHyway priority control register 0	HPB-DMASPR0	H'8888_8888	H'8888_8888
H' FFC08894	[Common to HPB-DMAC] HPB-DMA SuperHyway priority control register 1	HPB-DMASPR1	H'0088_8888	H'0088_8888

Legend:

[n]: HPB-DMAC Channel Number

Note: The above registers should be accessed in longword (32 bits) units; byte access and word access are prohibited.

- Notational conventions

  Initial value: Register value after a reset

  —: Undefined value

  R/W: Can be read from or written to; the written value can be read.

  R/WC1: Can be read from or written to; writing 1 initializes the bit but writing 0 is ignored.

  R: Read-only; the write value should always be 0.

  —/W: Write-only; an undefined value is read.

All the bits in the control register and status register are active high.

17.4.2 DMA Source Address Registers 0, 1 (DSAR0, DSAR1)

Each register specifies the DMA start address of the transfer source.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DSA[31:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSA[31:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DSA	All 0	R/W	These bits specify the DMA transfer source start address. The transfer source start address indicates a memory address or a peripheral module address according to the SMDL bit value in the DMA control register (DCR). When SMDL = 0: Transfer source address = memory (SuperHyway bus) address When SMDL = 1: Transfer source address = peripheral address

Notes: 1. When the address setting is a memory address, boundaries in the following table should be used.

	Boundary		
SWMD Bit in DCR	**0 to 6**	**7 to 11**	**12, 13**
0	8 bytes	16 bytes	32 bytes
1	4 bytes		

2. When the address setting is a peripheral module address, and in addition the SPDS or DPDS bit in DCR selects a 16-bit access size, up to a 16-bit boundary can be set. In this case, if an 8-bit boundary is set, upon writing the lowest 1 bit is ignored.
3. When the address set is a peripheral module address, and in addition the SPDS or DPDS bit in DCR selects an 8-bit access size, up to an 8-bit boundary can be set.
4. When an address setting is for a peripheral device, the upper address is provided for ease of understanding of the contents of software settings, and is not used for identification of access destination space specific to the peripheral.

17.4.3 DMA Destination Address Registers 0, 1 (DDAR0, DDAR1)

Each register specifies the DMA start address of the transfer destination.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DDA[31:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DDA[31:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DDA	All 0	R/W	These bits specify the DMA transfer destination start address. The transfer destination start address indicates a memory address or a peripheral module address according to the DMDL bit value in the DMA control register (DCR). When DMDL = 0: Transfer destination address = memory address When DMDL = 1: Transfer destination address = peripheral module address

Note: See the notes in section 17.4.2, DMA Source Address Register 0, 1. Those notes also apply to these registers.

17.4.4 DMA Transfer Count Registers 0, 1 (DTCR0, DTCR1)

Each register specifies the DMA transfer count.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									DTC[23:0]							
Initial value:	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTC[23:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	DTC	All 0	R/W	These bits specify the DMA transfer count (number of bytes, words, or longwords). The maximum count is 0h, which indicates 16 M (16,777,216) times.

Note: This register specifies the transfer count on the peripheral side for transfer from a peripheral to a memory (SuperHyway bus) or from a memory (SuperHyway bus) to a peripheral. For 8-burst DMA operation, one count for each 8-burst operation.

17.4.5 DMA Source Address Status Register (DSASR)

DSASR indicates the transfer source address.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DSAS[31:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSAS[31:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DSAS	All 0	R	These bits indicate the latest source address for which DMA transfer has been completed.

17.4.6 DMA Destination Address Status Register (DDASR)

DDASR indicates the transfer destination address.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DDAS[31:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DDAS[31:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DDAS	All 0	R	These bits indicate the latest destination address for which DMA transfer has been completed.

17.4.7 DMA Transfer Count Status Register (DTCSR)

DTCSR indicates the remaining count of the current transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									DTCS[23:0]							
Initial value:	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTCS[23:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	DTCS	All 0	R	These bits indicate the remaining count of the current DMA transfer (number of bytes, words, or longwords). The number of transferred bytes depends on the peripheral's data bus width.

Note: This register indicates the remaining transfer count on the peripheral side for transfer from a peripheral to a memory (SuperHyway bus) or from a memory (SuperHyway bus) to a peripheral. For 8-burst DMA operation, one count for each 8-burst operation.

17.4.8 DMA Port Select Register (DPTR)

DPTR selects the peripheral for DMA transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					SDPT								DDPT			
Initial value:	—	—	—	—	0	0	0	0	—	—	—	—	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	SDPT	All 0	R/W	Transfer Source Peripheral Access Port Select

Setting	HPB-DMAC Channels 0 to 6	7 to 11	12 and 13
H'0	SCIF0	SSI0	USB-FUNC0
H'1	SCIF1	SSI1	USB-FUNC1
H'2	SCIF2	SSI2	—
H'3	SCIF3	SSI3	—
H'4	SCIF4	HAC0	—
H'5	SCIF5	HAC1	—
H'6	HSPI	—	—
H'7	—	SD0-1	—
H'8	—	—	—
H'9	—	SD1-1	—
H'A to H'F	—	—	—

7 to 4	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	DDPT	All 0	R/W	Transfer Destination Peripheral Access Port Select

Setting	HPB-DMAC Channels 0 to 6	7 to 11	12 and 13
H'0	SCIF0	SSI0	USB-FUNC0
H'1	SCIF1	SSI1	USB-FUNC1
H'2	SCIF2	SSI2	—
H'3	SCIF3	SSI3	—
H'4	SCIF4	HAC0	—
H'5	SCIF5	HAC1	—
H'6	HSPI	SD0-0	—
H'7	—	—	—
H'8	—	SD1-0	—
H'9	—	—	—
H'A to H'F	—	—	—

Notes: 1. It is prohibited to specify the same module in multiple channels and to transfer data in the same direction in those channels.

2. When the SMDL bit is set to 0 in DCR (a memory is selected), the SDPT bit setting is ignored. When the DMDL bit is set to 0 in DCR (a memory is selected), the DDPT bit setting is ignored.
3. Setting values that are not assigned to any module is prohibited. If those values are set, operation cannot be guaranteed.

17.4.9 DMA Control Register (DCR)

DCR specifies the transfer mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						DTAMD	DTAC	DTAU	DTAU1	SWMD	BTMD	PKMD		CT	ACMD	DIP
Initial value:	—	—	—	—	—	0	0	0	0	0	0	0	—	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			SMDL	SPDAM	SDRMD		SPDS				DMDL	DPDAM	DDRMD		DPDS	
Initial value:	—	—	0	0	0	0	0	0	—	—	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
26	DTAMD	0	R/W	Specifies the data alignment conversion mode when a memory is accessed in DMA transfer (see section 17.5.7). 0: Conversion according to the combination of the input pin (little: endian mode) and the peripheral bus width. 1: Conversion according to the combination of DTAC (DMA data alignment conversion), DTAU (DMA data alignment unit), and DTAU1 (8-byte conversion in 4-byte units).
25	DTAC	0	R/W	Enables or disables data alignment conversion when a memory is accessed in DMA transfer (see section 17.5.7). This setting is valid when DTAMD = 1. 0: Disables data alignment conversion. 1: Enables data alignment conversion.

Bit	Bit Name	Initial Value	R/W	Description
24	DTAU	0	R/W	Specifies the unit for data alignment conversion (see section 17.5.7). This setting is valid when DTAMD = 1. 0: Byte units 1: Word units
23	DTAU1	0	R/W	Specifies whether 8-byte data alignment is performed in 4-byte units (see section 17.5.7). This setting is valid when DTAMD = 1. 0: Not performed. 1: Performed.
22	SWMD	0	R/W	Specifies memory (SuperHyway bus) access size. 0: *[1] (recommended to be cleared to 0 when DDR3 is specified) 1: 4 bytes (should be set to 1 when a peripheral device under the control of the HPB is specified)
21	BTMD	0	R/W	Specifies the burst DMA transfer. Burst DMA transfer is performed for peripherals. 0: Does not transfer in burst mode. 1: Transfers in burst mode (burst length is fixed to eight).
20	PKMD	0	R/W	Enables or disables packing of data read from a peripheral for DMA transfer from the peripheral to the SuperHyway bus. 0: Disables packing. 1: Enables packing.
19	—	—	R	Reserved This bit is always read as 0. The write value should always be 0.
18	CT	0	R/W	Specifies continuous DMA transfer. 0: Does not transfer in continuous mode. 1: Transfers in continuous mode.
17	ACMD	0	R/W	Specifies automatic continuous DMA transfer (valid only when CT = 1). 0: Does not transfer in automatic continuous mode (checks DNXT = 1 in DCMR). 1: Transfers in automatic continuous mode (regardless of the DNXT bit in DCMR).

Bit	Bit Name	Initial Value	R/W	Description
16	DIP	0	R/W	Specifies the valid DMA information set(s). 0: Uses one DMA information set repeatedly. 1: Uses two DMA information sets alternately.
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	SMDL	0	R/W	Selects the transfer source module. 0: Memory (SuperHyway bus) 1: Peripheral
12	SPDAM	0	R/W	Specifies whether to fix or increment the transfer source peripheral address (valid only when SMDL = 1). 0: Fixes the peripheral address at the value specified in DSAR0 or DSAR1. 1: Increments the peripheral address (increments by one for 8-bit transfer, by two for 16-bit transfer, or by 4 for 32-bit transfer).
11, 10	SDRMD	All 0	R/W	These bits specify the DMA request mode for the transfer source (valid only when SMDL = 1). 00: Module request (peripheral module request) 01: Auto-request 10: Timer request 11: Setting prohibited
9, 8	SPDS	All 0	R/W	These bits specify the data bus width for the transfer source peripheral (valid only when SMDL = 1). 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited
7, 6	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
5	DMDL	0	R/W	Selects the transfer destination module. 0: Memory (SuperHyway bus) 1: Peripheral

Bit	Bit Name	Initial Value	R/W	Description
4	DPDAM	0	R/W	Specifies whether to fix or increment the transfer destination peripheral address (valid only when DMDL = 1). 0: Fixes the peripheral address at the value specified in DSAR0 or DSAR1. 1: Increments the peripheral address (increments by one for 8-bit transfer, by two for 16-bit transfer, or by 4 for 32-bit transfer).
3, 2	DDRMD	All 0	R/W	These bits specify the DMA request mode for the transfer destination (valid only when DMDL = 1). 00: Module request (peripheral module request) 01: Auto-request 10: Timer request 11: Setting prohibited
1, 0	DPDS	All 0	R/W	These bits specify the data bus width for the transfer destination peripheral (valid only when DMDL = 1). 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited

Notes: 1. When the SWMD bit is set to 0, the access size of a memory (SuperHyway) of each channel is as follows.

	SuperHyway Access Size		
SWMD Bit in DCR	**0 to 6**	**7 to 11**	**12, 13**
0	8 bytes	16 bytes	32 bytes

2. When SMDL and DMDL = 1 and 0, data is transferred from a peripheral to a memory. When SMDL and DMDL = 0 and 1, data is transferred from a memory to a peripheral. Setting SMDL and DMDL = "1 and 1" or "0 and 0" is prohibited.
3. Not all DMAC functions can be applied to every peripheral. DMAC functions must be specified appropriately according to the functions and restrictions of each peripheral.

17.4.10 DMA Command Register (DCMDR)

DCMDR activates or stops DMA transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									BDOUT	DQSPD	DQSPC	DMSPD	DMSPC	DQEND	DNXT	DMEN
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
7	BDOUT	0	—/W	1: Forcibly writes the data read from a peripheral to the SuperHyway bus side (only for transfer from a peripheral to the SuperHyway bus when the peripheral is accessed in 8 or 16 bits). Writing 1 to this bit forcibly writes to the SuperHyway bus side, and then terminates DMA transfer.
6	DQSPD	0	—/W	1: Temporarily stops transfer in DMA information units.
5	DQSPC	0	—/W	1: Cancels temporary transfer stop in DMA information units.
4	DMSPD	0	—/W	1: Temporarily stops transfer in bus cycle units.
3	DMSPC	0	—/W	1: Cancels temporary transfer stop in bus cycle units.
2	DQEND	0	—/W	1: Terminates continuous DMA transfer mode. Only the DMA information specified before is transferred, and then continuous transfer mode is terminated.
1	DNXT	0	—/W	1: Requests the next DMA transfer. In continuous transfer mode, after the current DMA information is transferred, the next DMA information is transferred.
0	DMEN	0	—/W	1: Activates DMA transfer.

Note: For use of BDOUT, see sections 17.5.3 and 17.5.4.

17.4.11 DMA Forced Stop Register (DSTPR)

DSTPR stops DMA transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																DMSTP
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	—/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DMSTP	0	—/W	1: Forcibly terminates DMA transfer. After the current bus cycle is completed, DMA transfer is terminated. (Values set for the DMA transfer status registers (DSASR, DDASR, and DTCSR) are retained.)

17.4.12 DMA Status Register (DSTSR)

DSTSR indicates the DMA transfer status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

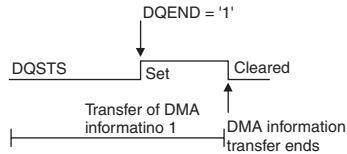
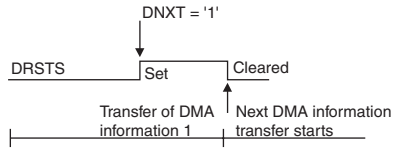
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										NDP1	NDP0	DQSPS	DMSPS	DQSTS	DRSTS	DMSTS
Initial value:	—	—	—	—	—	—	—	—	—	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
6	NDP1	0	R	Next DMA Transfer Information Register Status 1 0: Does not transfer DMA information in DMA information register set 1 in the next DMA information transfer. 1: Transfers DMA information in DMA information register set 1 in the next DMA information transfer.
5	NDP0	1	R	Next DMA Transfer Information Register Status 0 0: Does not transfer DMA information in DMA information register set 0 in the next DMA information transfer. 1: Transfers DMA information in DMA information register set 0 in the next DMA information transfer.
4	DQSPS	0	R	Temporary Stop Status of DMA Information Updating 0: Normal operation 1: DMA information updating is temporarily stopped.
3	DMSPS	0	R	Temporary Stop Status of DMA Transfer 0: Normal operation 1: DMA transfer is temporarily stopped.
2	DQSTS	0	R	DMA Acceptance End Status 0: DMA information can be accepted. 1: DMA information acceptance is stopped.
1	DRSTS	0	R	DMA Transfer Request Status 0: Next DMA transfer has not been requested. 1: Next DMA transfer has been requested.
0	DMSTS	0	R	DMA Status 0: DMA transfer has been completed. 1: DMA transfer is active.

Note: Either NDP0 or NDP1 (next DMA transfer information register set 0 or 1) is always set to 1.

The following shows the transition of each bit status in DSTSR.

		Conditions of Status Transition	
	0 →	1 →	0
NDP1	Initial state	Continuous transfer mode is selected and information register sets 0 and 1 are used alternately (DIP in DCR = 1): DMA transfer for information register set 0 is in progress.	Continuous transfer mode is selected and information register sets 0 and 1 are used alternately (DIP in DCR = 1): DMA transfer for information register set 1 is in progress.
NDP0	—	1. Initial state 2. Single transfer mode: Always 3. Continuous transfer mode is selected and information register set 0 is used repeatedly (DIP in DCR = 0): Always 4. Continuous transfer mode is selected and information register sets 0 and 1 are used alternately (DIP in DCR = 1): DMA transfer for information register set 1 is in progress.	Continuous transfer mode is selected and information register sets 0 and 1 are used alternately (DIP in DCR = 1): DMA transfer for information register set 0 is in progress.
DQSPS	Initial state	DMA transfer is temporarily stopped in DMA information units (DQSPD in DCMDR = 1) and the current information transfer is completed.	Temporary stop of DMA transfer in DMA information units is canceled (DQSPC in DCMDR = 1).

		Conditions of Status Transition	
	0 →	1 →	0
DMSPS	Initial state	DMA transfer is temporarily stopped in bus cycle units (DMSPD in DCMDR = 1) and the current bus cycle is completed.	Temporary stop of DMA transfer in bus cycle units is canceled (DMSPC in DCMDR = 1).
DQSTS	Initial state	Continuous transfer mode is selected (CT in DCR = 1) and DMA continuous transfer is terminated (DQEND = 1) during transfer of DMA information 1. 	Transfer of DMA transfer information ends.
DRSTS	Initial state	Continuous transfer mode is selected (CT in DCR = 1) and next DMA transfer information is requested (DNXT = 1) during transfer of DMA information 1. 	Next DMA information transfer is started.

Conditions of Status Transition

	0	→ 1	→ 0
DMSTS	Initial state	DMA is activated (DMEN in DCMDR = 1).	End state (remains in the idle state). 1. Transfer end a. Single transfer mode: Transfer of one DMA information set is completed. b. Continuous transfer mode: DRSTS = 0 and DQSTS = 1 in DSTSR and transfer of current DMA information is completed. 2. Forced stop DMSTP in DSTPR is set to 1 and DMA transfer is terminated after the current bus cycle is completed.

17.4.13 DMA Channel Debug Register (DDBGR)

DDBGR is used for debugging.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DBG02															
Initial value:	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										DBG01				DBG00		
Initial value:	—	—	—	—	—	—	—	—	—	0	0	0	—	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	DBG02	0	R/W	Test bit: This bit is a test bit, thus writing is prohibited. If this bit is written to, correct operation cannot be guaranteed.
30 to 7	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 4	DGB01	All 0	R	Test bit
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	DBG00	All 0	R	Test bit

17.4.14 DMA Channel Debug Register 2 (DDBGR2)

DDBGR2 is used for debugging.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DBG12	DBG11				DBG10									
Initial value:	—	0	0	0	—	—	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DBG10															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	—	R	Reserved This bit is always read as 0. The write value should always be 0.
30	DBG12	0	R/W	Test bit: This bit is a test bit, thus writing is prohibited. If this bit is written to, correct operation cannot be guaranteed.
29, 28	DBG11	All 0	R/W	Test bit: These bits are test bits, thus writing is prohibited. If these bits are written to, correct operation cannot be guaranteed.
27, 26	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 0	DBG10	All 0	R	Test bit

17.4.15 DMA Timer Control Register (DTIMR)

DTIMR specifies the timer cycle in the DMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTIM[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	DTIM	All 0	R/W	DMAC Internal Timer Cycle Set Request mode: Specify the DMA request interval in timer request mode. Request interval: DTIM × peripheral-side bus clock cycle (ns) • HPB-DMAC: 15 ns (66 MHz) Note: Even in timer request mode, operation is the same as that in auto-request mode when the initial value of DTIM is 0.

17.4.16 DMA Transfer End Interrupt Status Register (DINTSR)

DINTSR indicates the DMA transfer end interrupt status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			DTE13	DTE12	DTE11	DTE10	DTE9	DTE8	DTE7	DTE6	DTE5	DTE4	DTE3	DTE2	DTE1	DTE0
Initial value:	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	DTEn	All 0	R	DMA Transfer End Interrupt Status Each bit indicates the DMA transfer status (n: DMA channel number). 0: Initial state or data is being transferred before the count specified by DTCR is reached. 1: Transfer has been completed for the count specified by DTCR.

17.4.17 DMA Transfer End Interrupt Status Clear Register (DINTCR)

DINTCR clears the DMA transfer end interrupt status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			DTEC 13	DTEC 12	DTEC 11	DTEC 10	DTEC 9	DTEC 8	DTEC 7	DTEC 6	DTEC 5	DTEC 4	DTEC 3	DTEC 2	DTEC 1	DTEC 0
Initial value:	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	DTECn	All 0	—/W	DMA Transfer End Interrupt Status Clear Writing 1 to each bit clears the corresponding DMA transfer end interrupt status (n: DMA channel number) Writing 0 to these bits is ignored. Each bit is always read as 0.

17.4.18 DMA Transfer End Interrupt Enable Register (DINTMR)

DINTMR controls output of DMA transfer end interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			DTEM 13	DTEM 12	DTEM 11	DTEM 10	DTEM 9	DTEM 8	DTEM 7	DTEM 6	DTEM 5	DTEM 4	DTEM 3	DTEM 2	DTEM 1	DTEM 0
Initial value:	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	DTEMn	All 0	R/W	DMA Transfer End Interrupt Output Control An interrupt signal is output as a level signal (n: DMA channel number). 0: Does not output an interrupt on completion of a DMA transfer. 1: Outputs an interrupt on completion of a DMA transfer.

17.4.19 DMA Activation Status Register (DACTSR)

DACTSR indicates the activation status of each channel.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			DS13	DS12	DS11	DS10	DS9	DS8	DS7	DS6	DS5	DS4	DS3	DS2	DS1	DS0
Initial value:	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	DSn	All 0	R	DMA Channel n Status (n: DMA channel number) 0: Idle state 1: Active state

17.4.20 Software-Reset Register (HSRSTR0 and HSRSTR1)

HSRSTR0 and HSRSTR1 reset DMA channel n.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SRST
Initial value:	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SRST	0	R/WC1	Software Reset Resets DMA channel n. 0: Writing 0 is ignored. 1: Resets DMA channel n. This register is always read as 0.

Note: Writing 1 resets the DMAC regardless of the DMA transfer status. The same modules are reset as when a power-on reset or manual reset is input. Accordingly, a software reset should be used only while DMA transfer is not in progress (e.g., during system debugging). To stop operation, forced termination or temporary stop should be specified instead of software reset. In the registers used by all channels in common (DMA transfer end interrupt status register (DINTSR) and DMA transfer end interrupt enable register (DINTMR)), only the bits corresponding to the software-reset channel are initialized.

17.4.21 HPB-DMA SuperHyway Priority Control Register 0 (HPB-DMASPR0)

HPB-DMASPR0 specifies the SuperHyway bus access priority level for HPB-DMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPRR7				SPRR6				SPRR5				SPRR4			
Initial value:	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPRR3				SPRR2				SPRR1				SPRR0			
Initial value:	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SPRRn	H'8	R/W	Specifies the SuperHyway bus access priority level for each DMAC channel (n is the channel number). Priority: H'0 (lowest) to H'F (highest)

Note: Since SuperHyway bus access priority level settings relate to priority control for the SuperHyway bus overall, priority levels with other access modules must be confirmed when making settings.

17.4.22 HPB-DMA SuperHyway Priority Control Register 1 (HPB-DMASPR1)

HPB-DMASPR1 specifies the SuperHyway bus access priority level for HPB-DMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									SPRR13				SPRR12			
Initial value:	—	—	—	—	—	—	—	—	1	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPRR11				SPRR10				SPRR9				SPRR8			
Initial value:	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	SPRRn	H'8	R/W	Specifies the SuperHyway bus access priority level for each DMAC channel (n is the channel number). Priority: H'0 (lowest) to H'F (highest)

Note: Since SuperHyway bus access priority level settings relate to priority control for the SuperHyway bus overall, priority levels with other access modules must be confirmed when making settings.

17.4.23 SD Mode Select Register (SDMDR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															SDMD0	SDMD0
Initial value:	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 2	—	—	R/W	Reserved These bits are always read as 0. The write value should always be 0.
1	SDMD1	0	R/W	SD Data Transfer Mode Bit 0: Single-transfer 1: 8-burst transfer (HPB-DMAC11 only)
0	SDMD0	0	R/W	SD Data Transfer Mode Bit 0: Single-transfer 1: 8-burst transfer (HPB-DMAC10 only)

Notes:
- The SDMDR register specifies the operating mode of the HPB-DMAC. The mode settings do not affect the SD module. Use the registers in the SD module to make operation settings for the SD module.
- Burst transfer mode is supported for HPB-DMAC channels 10 and 11 only and can be enabled only when the SDPT and DDPT bits in the DMA port select register (DPTR) for HPB-DMAC channels 10 and 11 are set to SD0 or SD1. (Burst transfer mode is prohibited when the SDPT and DDPT bits in the DMA port select register (DPTR) for HPB-DMAC channels 10 and 11 are set to other than SD0 or SD1.)
- To enable burst transfer mode, set to 1 the PKMD bit in the DMA control register (DCR) for HPB-DMAC channels 10 and 11.
- Specify the transfer count when using 8-burst transfer mode or single-transfer mode.
- In 8-burst transfer mode, specify a transfer count of 8 × n (n = 0, 1, 2, …). It is not possible to use a fractional transfer count in 8-burst transfer mode.
- Module request is the only valid option for the DMA request mode setting specified by the SDRMD and DDRMD bits in the DMA control register (DCR) for HPB-DMAC channels 10 and 11. (Settings other than module request are prohibited.)

17.5 Operation

17.5.1 DMA Transfer Procedure

The following describes the DMA transfer procedure.

1. Making the initial setting for the DMAC

   Select the peripheral for DMA transfer in the DMA port select register (DPTR) (HPB-DMAC only).

   Specify transfer conditions in the DMA control register (DCR).
2. Specifying DMA transfer information

   Make appropriate settings in the following registers according to the DIP bit setting (input mode: using one information set repeatedly or two information sets alternately) in DCR:

   — DMA source address register (DSAR)

   — DMA destination address register (DDAR)

   — DMA transfer count register (DTCR)

   When the DIP bit specifies that one information set is used repeatedly, make the appropriate settings in DSAR0, DDAR0, and DTCR0.
3. Activating the DMAC

   Activate the DMAC by setting the DMEN bit in the DMA command register (DCMDR).
4. Reading the specified DMA transfer information

   The specified DMA transfer information is read from DMA information registers 0 and 1 in that order.
5. Clearing the DMA transfer request

   The DMA transfer request status signal is cleared.
6. Starting DMA transfer

   When the auto-request mode is selected as the transfer request mode, transfer automatically starts at the DMAC transfer timing after the transfer information is obtained.

   When the peripheral request mode is selected, DMA transfer is performed for one bus access cycle when a transfer request is accepted.

   When the timer request is selected, transfer automatically starts at the intervals specified in the DMAC internal timer after the transfer information is obtained.
7. Issuing an interrupt for the end of a specified number of transfers

   In single transfer mode, DMA transfer stops when transfer is completed for the specified number of times, and the CPU is notified of the end of transfer through an interrupt.

   In continuous transfer mode, the CPU is notified of the end of transfer in DMA transfer information units through an interrupt.

The interrupt signal is controlled according to the setting in the DMA transfer end interrupt enable register (DINTMR).

8. Reading the next DMA transfer information (continuous transfer mode))

   If the next DMA transfer request is specified, the information of the transfer is read and data is transferred in the same way as described in step 6.

   If no additional DMA transfer request is specified (DRSTS = 0), the continuous DMA transfer mode is terminated when DQSTS = 1, or the next DMA transfer request is waited for when DQSTS = 0.

9. Adding DMA transfer information (continuous transfer mode))

   If new DMA transfer information should be added, specify the information in the DMA transfer information set that will be used for the next transfer (the next DMA transfer information set can be checked with the NDP1 and NDP0 bits in the DMA status register (DSTSR)).

   If no DMA transfer information should be added, write 1 to the DQEND bit in the DMA command register (DCMDR) to terminate the continuous transfer mode.

Note: Actually, the source bus and destination bus operate independently.

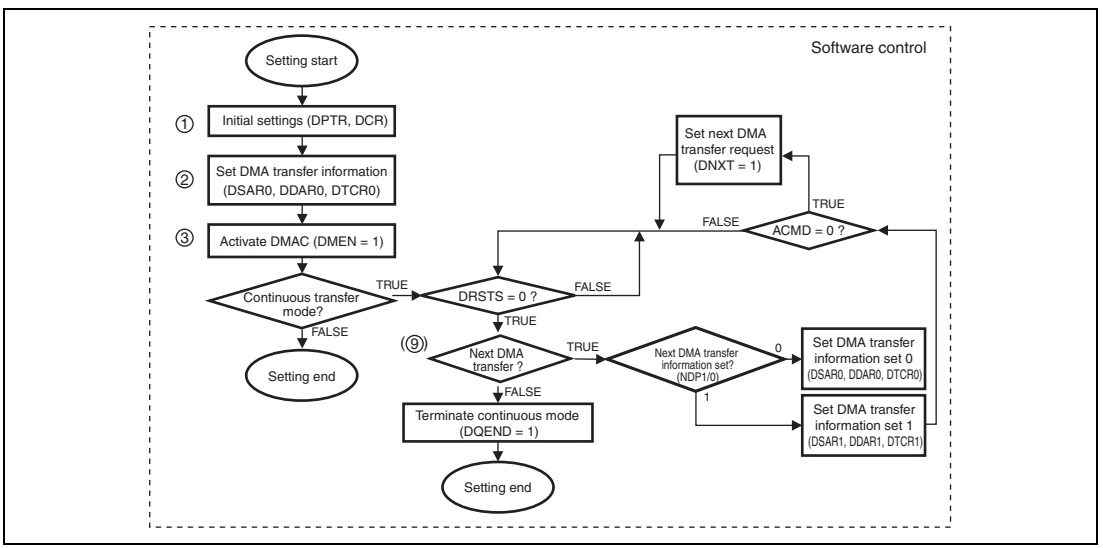


Figure 17.2 DMA Transfer Flowchart (1)

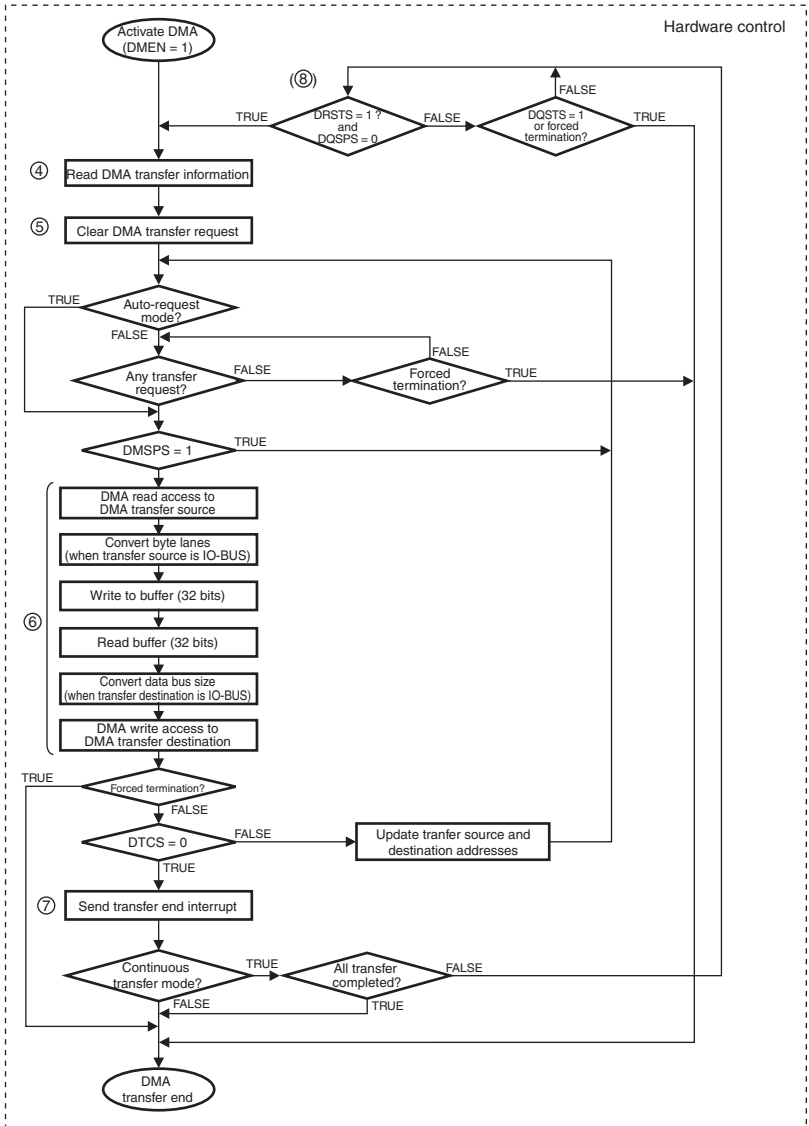


Figure 17.2 DMA Transfer Flowchart (2)

17.5.2 Continuous DMA Transfer Operation

The following shows the relationship between addition of DMA transfer information by software and DMA transfer information read and data transfer operation by hardware, which are described in step 9 in section 17.5.1, DMA Transfer Procedure. Figures 17.3 and 17.4 show transfer using DMA information set 0 repeatedly, and figures 17.5 and 17.6 show transfer using DMA information sets 0 and 1 alternately.

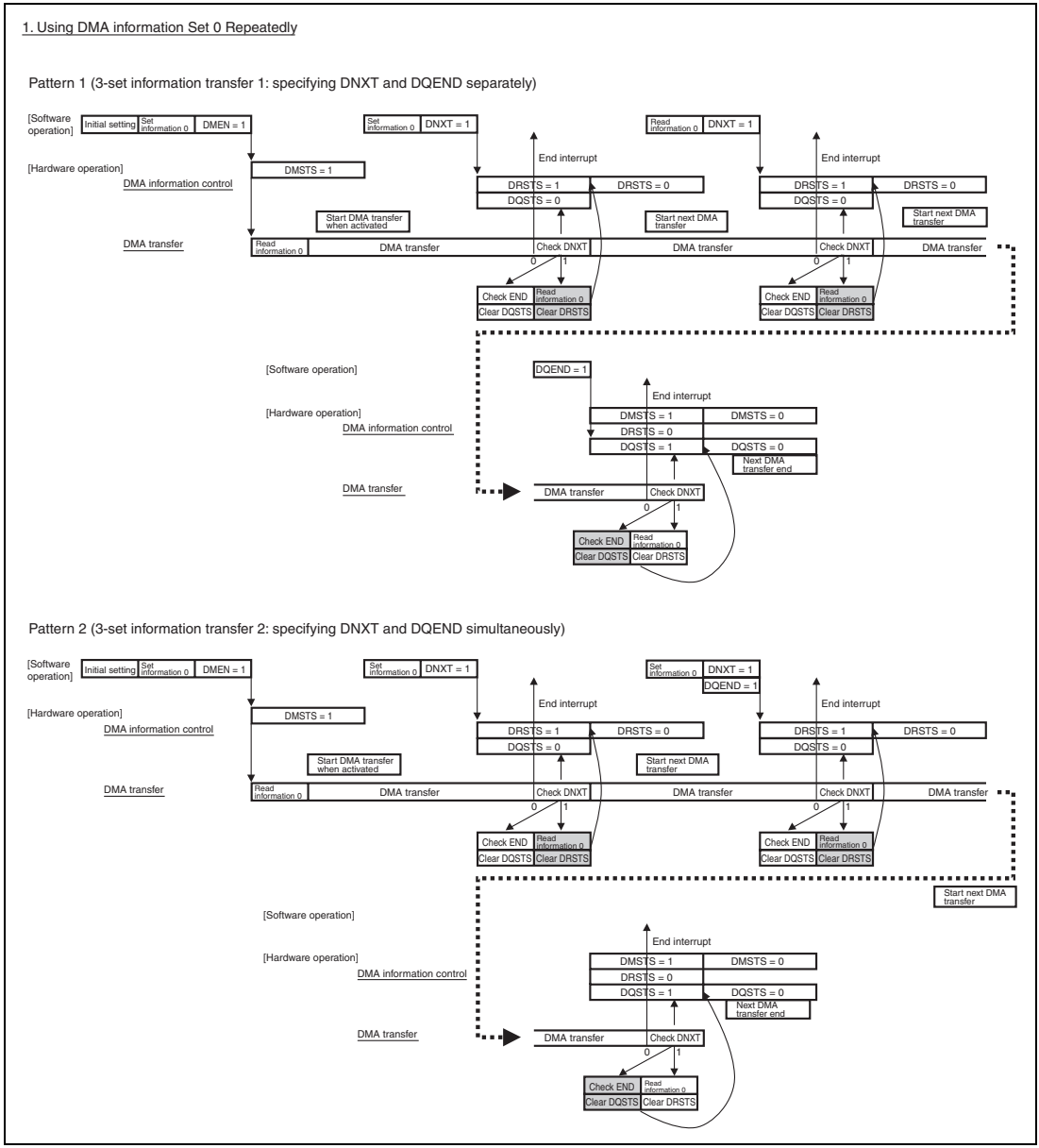


Figure 17.3 Using DMA Information Set 0 Repeatedly (1)

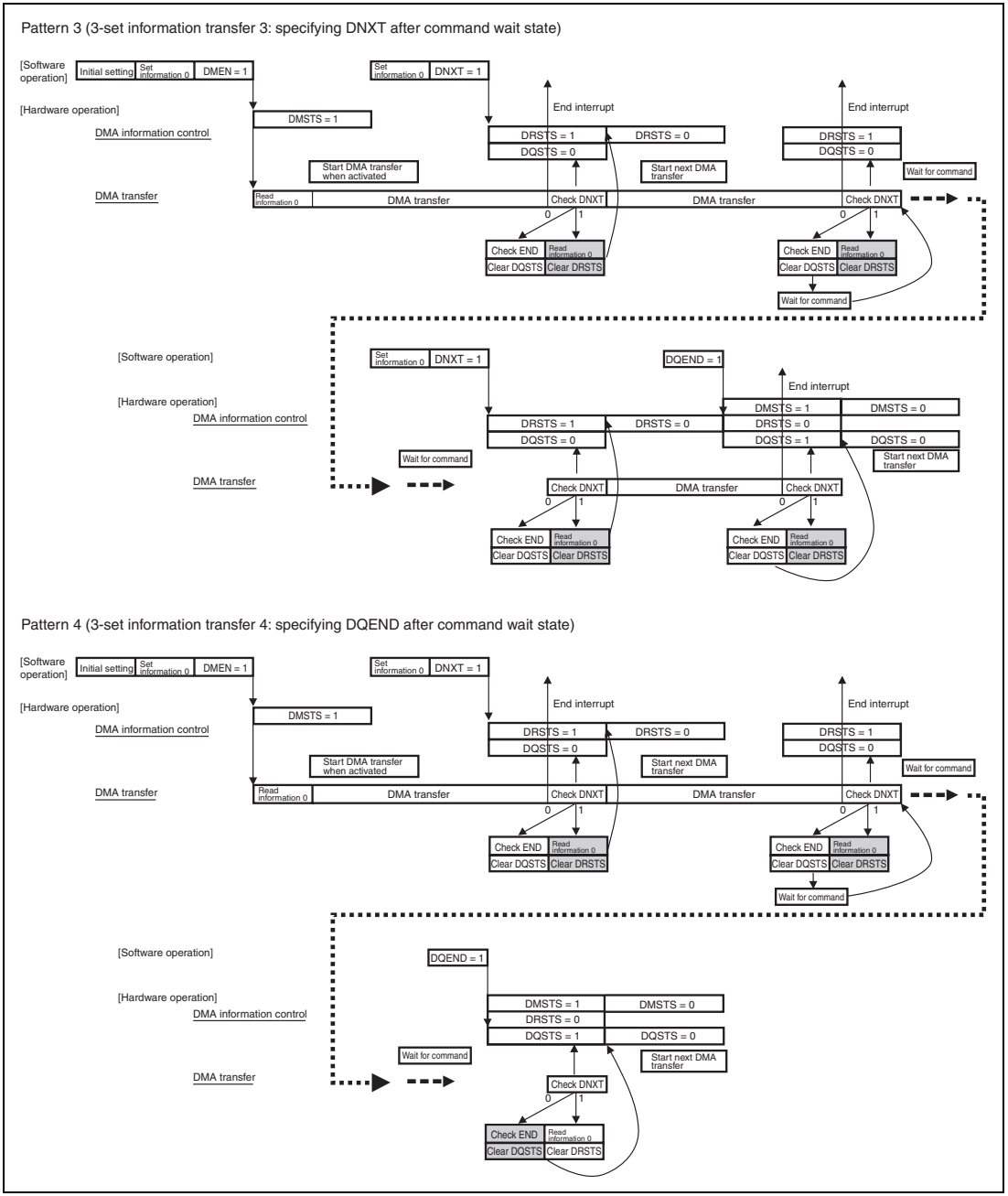


Figure 17.4 Using DMA Information Set 0 Repeatedly (2)

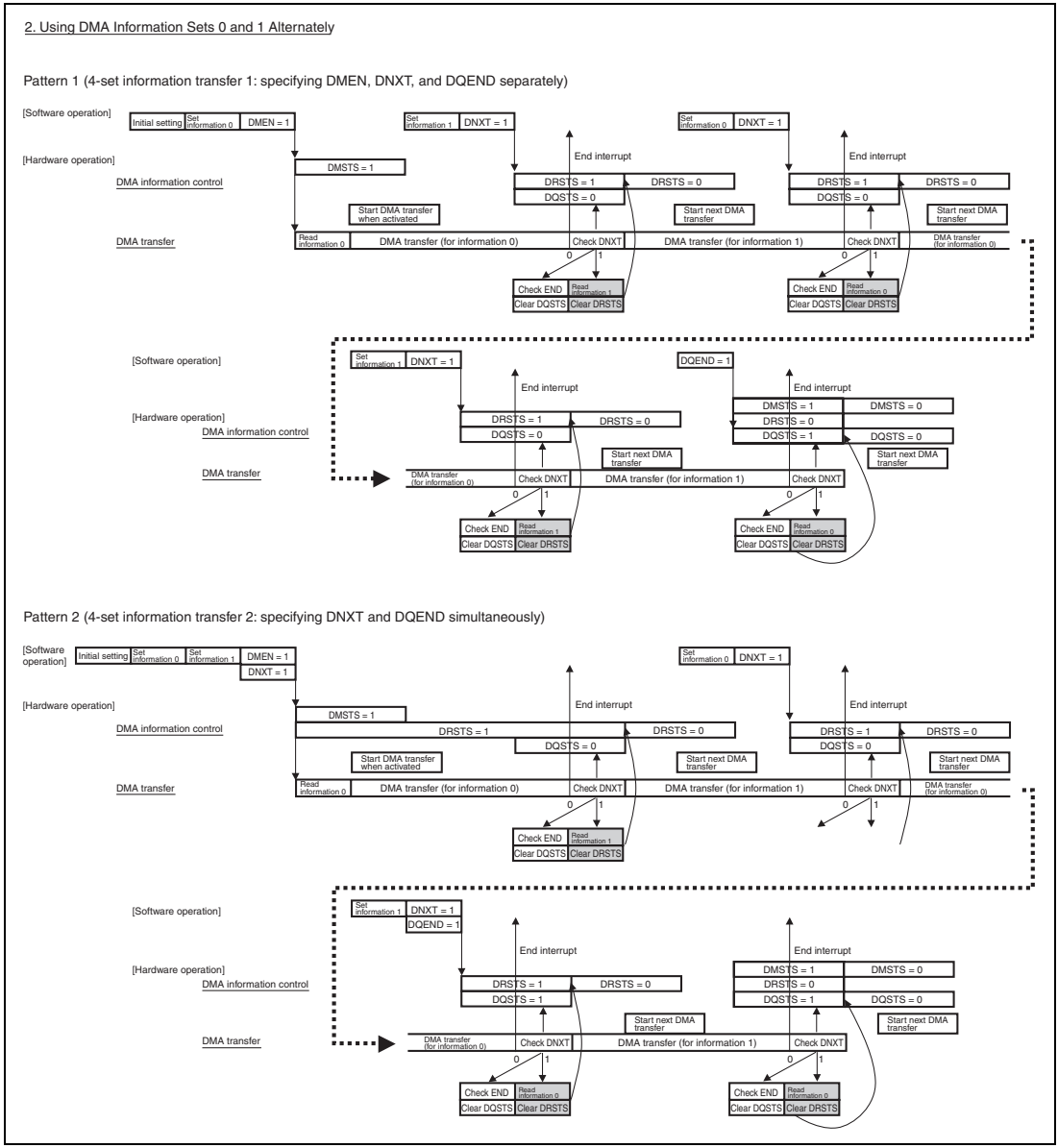


Figure 17.5 Using DMA Information Sets 0 and 1 Alternately (1)

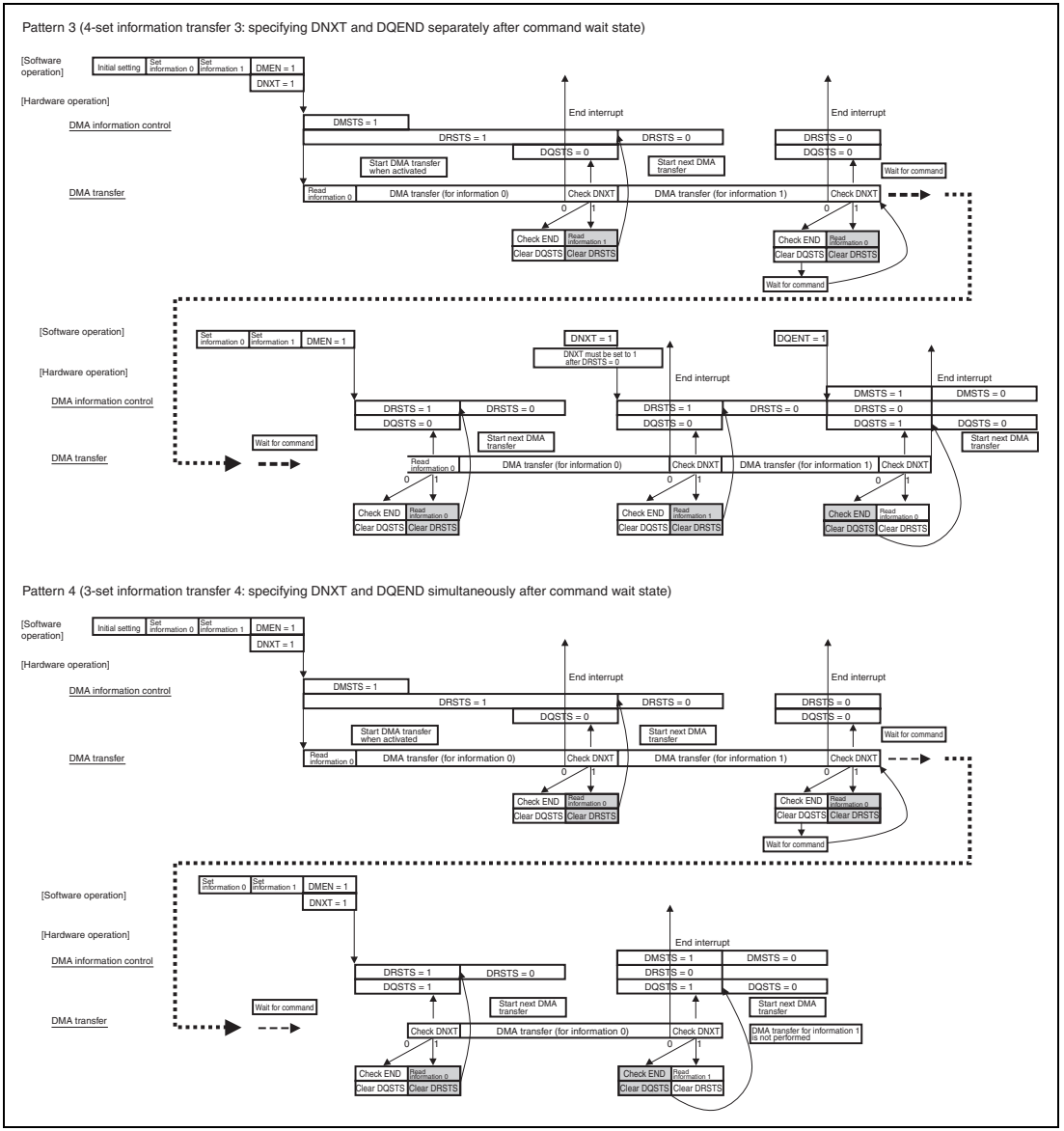


Figure 17.6 Using DMA Information Sets 0 and 1 Alternately (2)

17.5.3 Packing Data Read from Peripheral

Writing 1 to the PKMD bit when making necessary settings in the DMA control register (DCR) in the usual DMA activation procedure enables packing of data read from a peripheral module and then writing the data to memory (the SuperHyway bus). Packing size can be specified to either 4 bytes or a byte unit specified by the DCR register by the SWMD bit in DCR. However, when the transfer destination on the SuperHyway side is memory (DDR3-SDRAM), packing size in a byte unit specified by the DCR register is recommended in order to use memory and the SuperHyway efficiently. HPB-DMAC can transfer data in the units specified when the SWMD in the DCR register is 0, if the transfer destination on the SuperHyway side allows 32-byte access and the destination device is SRAM etc.

During the packing operation, even if the DMAC holds data less than the specified packing size when the DMAC completes the specified count of transfer, the DMAC writes fetched data to memory and indicates the end of DMA transfer by setting the DTEn bit in the DMA transfer end interrupt status register (DINTSR). If the peripheral module completes a DMA request before the specified transfer count is reached, DMA transfer can be terminated by writing 1 to the BDOUT bit in the DMA command register (DCMDR). In this case, if data which is being packed remains in the DMAC, the DMAC writes the data to memory (SuperHyway). The DTE[n] bit in the DMA transfer end interrupt status register (DINTSR) is set and a transfer end interrupt is generated. Note that zero padding does not occur during write to memory. If no data remains in the DMAC, the DMAC terminates DMA transfer without accessing memory and generates a transfer end interrupt in the same way as when data remains in the DMAC.

When transfer is terminated by the BDOUT bit setting in continuous transfer mode, the DMAC transfers the next DMA information if the next DMA information transfer is requested through the DNXT bit in DCMDR, and then terminates DMA transfer in the continuous transfer mode termination procedure. In this mode, DMA requests (dreq) from peripheral modules are masked (not accepted) before a transfer end interrupt occurs after the BDOUT bit is set to 1.

17.5.4 Limitations on Packing of Data Read from Peripheral

- The transfer count is specified in DMA transfer count registers 0 and 1 (DTCR0 and DTCR1) in the DMAC.
- If DMA transfer from a peripheral module is completed before the specified transfer count is reached, the DMAC cannot distinguish whether data is being transferred or the transfer has been completed, and data below the packing size may remain in the DMAC internal buffer. When the size of remaining data is the same as the specified packing size, the data is transferred to memory.
- Data remaining in the DMAC internal buffer is written to memory through a forced write executed by setting the BDOUT bit in the DMA command register (DCMDR). (Zero padding

does not occur because the transfer destination is memory. For example, if 3-byte data remains in the DMAC untransferred, the 3-byte data is written to memory as is.)

- A forced write can be triggered by a peripheral or its communication end interrupt. Note that whether DMA transfer from the peripheral is completed when the communication end interrupt occurs depends on the specifications of the peripheral. Accordingly, check the specifications of the module before executing a forced write.

17.5.5 Notification of the End of DMA Transfer

The DMAC notifies the CPU via the INTC of the end of transfer through transfer end interrupt signal (a level signal) when transfer is completed for the transfer count specified in DMA transfer information in single transfer mode. In continuous transfer mode, the DMAC outputs transfer end interrupt signal every time transfer is completed for the transfer count specified in one DMA transfer information set.

The transfer end interrupt signal is controlled according to the setting in the DMA transfer end interrupt enable register (DINTMR). Writing 1 to the DMA transfer status clear register clears the transfer end interrupt signal.

17.5.6 DMA Transfer Start, Stop, and Resume Procedures

This section describes the procedures for starting, stopping, and resuming DMA transfer.

- To Stop (Cancel) DMA Transfer during Operation

Step	Overview	Register Operation	Operation after Register Write
1 ↓	Specify forced termination for DMAC.	Write 1 to DMSTP in DSTPR of the DMAC.	The DMAC stops DMA transfer as soon as the current DMA bus cycle is completed, and then enters the idle state. Untransferred data remaining in the buffer is discarded. The registers retain the values. No end interrupt is issued.
2 ↓	Check that the DMAC has entered the idle state.	If DMSTS in DSTSR of the DMAC is 0, the DMAC is in the idle state.	—
3	Specify forced termination for peripheral devices.	(Depends on the peripheral devices.)	The peripheral devices stop sending DMA requests.

Note: Step 2 can be done after step 3.

- To Temporarily Stop (Pause) DMA Transfer during Operation and Then Resume It

Step	Overview	Register Operation	Operation after Register Write
1 ↓	Specify temporary stop for DMAC.	Write 1 to DMSPD in DCMDR of the DMAC.	The DMAC temporarily stops DMA transfer as soon as the current DMA bus cycle is completed. The DMAC retains its internal status, including untransferred data remaining in the buffer, without change.
2 ↓	Check that the DMAC has entered the temporary stop state.	If DMSPS in DSTSR of the DMAC is 1, the DMAC is in the suspended state.	—
3 ↓	A certain period of time has elapsed.		
4	Specify cancel of temporary stop (resume transfer).	Write 1 to DMSPC in DCMDR of the DMAC.	The paused state is canceled, and the DMAC resumes its operation with the DMA transfer for the DREQ detected by the DMAC.

Note: Step 2 can be done between steps 3 and 4.

- To Temporarily Stop (Pause) DMA Transfer during Operation and Then Terminate (Cancel) Operation

Step	Overview	Register Operation	Operation after Register Write
1 ↓	Specify temporary stop for DMAC.	Write 1 to DMSPD in DCMDR of the DMAC.	The DMAC temporarily stops DMA transfer as soon as the current DMA bus cycle is completed. The DMAC retains its internal status, including untransferred data remaining in the buffer, without change.
2 ↓	Check that the DMAC has entered the temporary stop state.	If DMSPS in DSTSR of the DMAC is 1, the DMAC is in the suspended state.	—
3 ↓	A certain period of time has elapsed.		

Step	Overview	Register Operation	Operation after Register Write
4 ↓	Specify forced termination for DMAC.	Write 1 to DMSTP in DSTPR of the DMAC.	The DMAC exits the temporary stop state and enters the idle state. Untransferred data remaining in the buffer is discarded. The registers retain the values. No end interrupt is issued.
5 ↓	Check that the DMAC has entered the idle state.	If DMSTS in DSTSR of the DMAC is 0, the DMAC is in the idle state.	—
6	Specify forced termination for peripheral devices.	(Depends on the peripheral devices.)	The peripheral devices stop sending DMA requests.

Notes: Step 2 can be done between steps 3 and 4.
Step 5 can be done after step 6.

17.5.7 Data Alignment in SuperHyway Bus Interface

The SuperHyway bus is always accessed through a handshake using an access request and a request acknowledge. Alignment of data read or written during the access to memory through the SuperHyway bus is always converted.

When the DTAMD bit in the DMA control register (DCR) is 0, the SuperHyway bus data alignment is converted according to the endian mode signal (DMAC input signal: little) and the peripheral data bus width (SPDS[1:0] or DPDS[1:0] in DCR). When the DTAMD bit in DCR is 1, data alignment is converted according to the DTAC, DTAU, and DTAU1 bit settings in DCR.

The following table shows the SuperHyway bus data alignment control according to the DTAMD bit in DCR, endian mode signal (DMAC input signal: little), peripheral data bus width (SPDS1 and SPDS0 or DPDS1 and DPDS0 in DCR), and DTAC, DTAU, and DTAU1 bits in DCR.

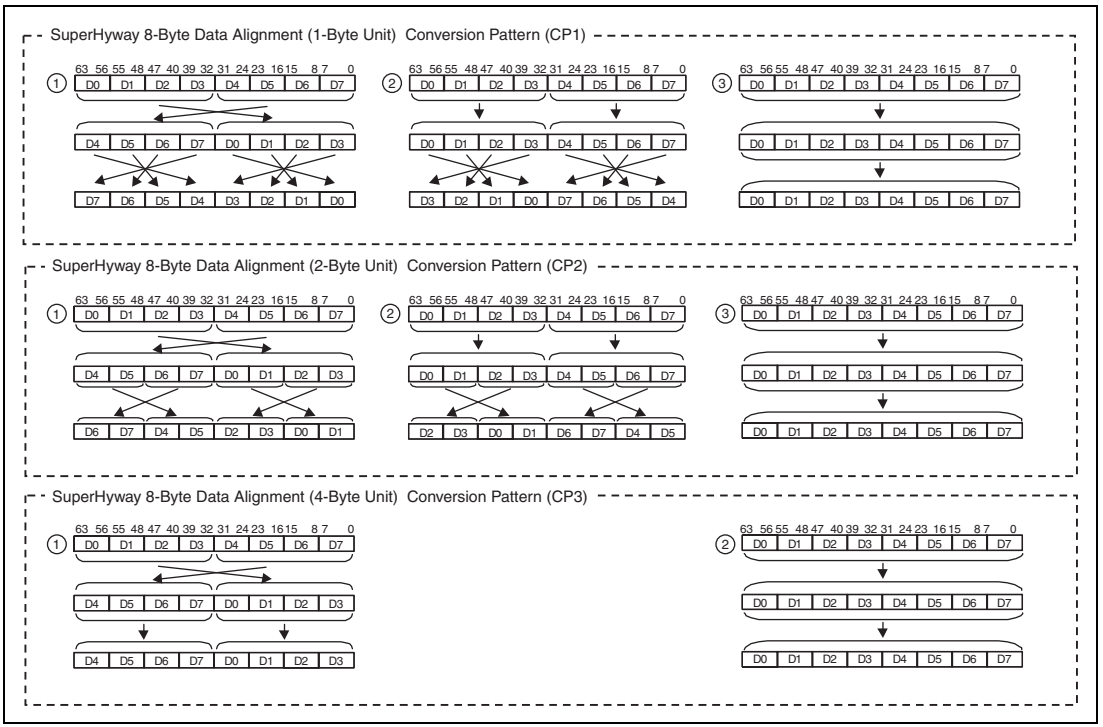
Table 17.3 Data Alignment Control According to Bit and Signal Settings

No.	DTAMD	little (MD[8])	PDS [1:0]	DTAC	DTAU	DTAU1	Data Alignment of 4 Bytes	Unit for 4-Byte Data Alignment	8-Byte Data Alignment in 4-Byte Unit	Conversion Pattern	Remarks
1	0	0	00 (8 bits)	*	*	*	Not controlled	8 bits	Not controlled	CP1 (3)	Standard conversion (Software must specify only PDS[1:0])
2	0	0	01 (16 bits)	*	*	*	Not controlled	16 bits	Not controlled	CP2 (3)	
3	0	0	10 (32 bits)	*	*	*	Not controlled	8 bits	Not controlled	CP3 (2)	
4	0	1	00 (8 bits)	*	*	*	Controlled	8 bits	Controlled	CP1 (1)	
5	0	1	01 (16 bits)	*	*	*	Controlled	16 bits	Controlled	CP2 (1)	
6	0	1	10 (32 bits)	*	*	*	Not controlled	8 bits	Controlled	CP3 (1)	
7	1	*	*	0	0	0	Not controlled	8 bits	Not controlled	CP1 (3)	Special conversion (Software must specify the alignment mode)
8	1	*	*	0	0	1	Not controlled	8 bits	Controlled	CP3 (1)	
9	1	*	*	0	1	0	Not controlled	16 bits	Not controlled	CP2 (3)	
10	1	*	*	0	1	1	Not controlled	16 bits	Controlled	CP3 (1)	
11	1	*	*	1	0	0	Controlled	8 bits	Not controlled	CP1 (2)	
12	1	*	*	1	0	1	Controlled	8 bits	Controlled	CP1 (1)	
13	1	*	*	1	1	0	Controlled	16 bits	Not controlled	CP2 (2)	
14	1	*	*	1	1	1	Controlled	16 bits	Controlled	CP2 (1)	

Legend:

*: Don't care

The following shows data alignment conversion in the DMAC. Conversion pattern numbers in the table above correspond to the conversion numbers below.



17.5.8 Data Alignment in HPB Bus Interface

The HPB bus is accessed in the data bus width specified in the SPDS or DPDS bit in the DMA control register (DCR), and big endian is always assumed.

17.6 Usage Notes

Note the following when using the HPB-DMAC.

17.6.1 Modifying Operating Frequency

Do not modify the operating frequency through CPG register settings during DMA transfer operation.

Before modifying the frequency, be sure to check that the DMSTS bit (bit 0) of the DMA status register (DSTSR) is 0.

To restart processing, use the procedure described in section 17.5.1, DMA Transfer Procedure.

Section 18 Clock Pulse Generator (CPG)

The CPG generates clocks provided to the internal and external bus interfaces of the SH7786, and controls power-down mode. The CPG consists of a crystal oscillator circuit, PLLs, divider, and the control unit.

18.1 Features

The CPG has the following features.

- Generates SH7786 internal clocks*

  Generates the CPU clock (Ick) used in the CPU, FPU, cache, and TLB; the SuperHyway clock (SHck) used in the SuperHyway, the DU clock (DUck) used in the display unit; and the peripheral clock (Pck) used in the interface with on-chip peripheral modules.
- Generates SH7786 external clocks

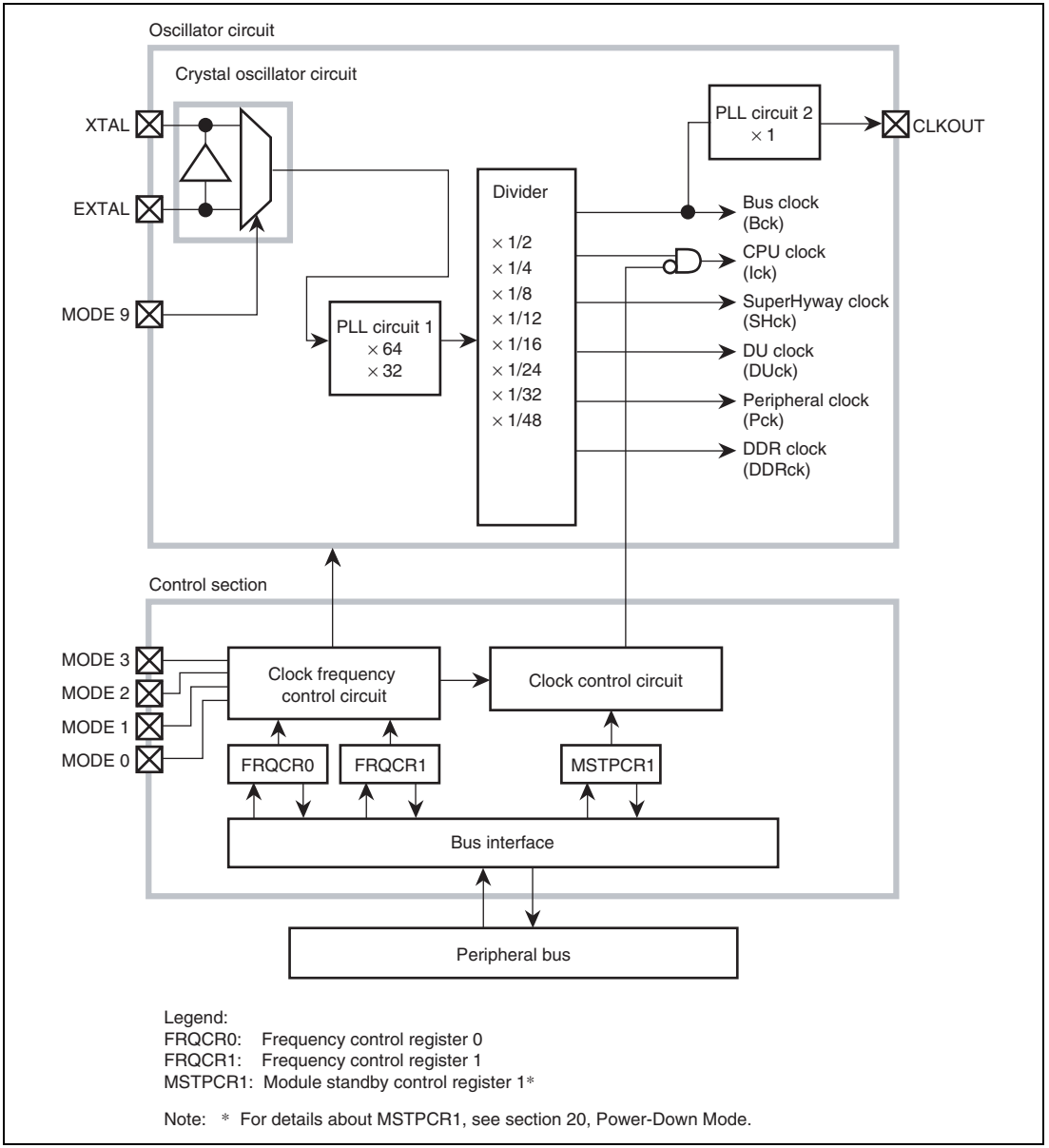
  Generates the bus clock (Bck) used in the interface with the external devices, and the DDR clock (DDRck) for the memory clock used in the DDRIF.
- Clock operating modes

  Selects a crystal resonator or an external clock input for the clock input to the CPG
- Controls power-down mode

  Can stop the CPU in sleep mode and specific modules in module standby mode. For details, see section 20, Power-Down Mode.

Figure 18.1 shows a block diagram of the CPG.

Note: * For description of the clock used by each module, see the sections on individual modules.



Legend:
FRQCR0: Frequency control register 0
FRQCR1: Frequency control register 1
MSTPCR1: Module standby control register 1*

Note: * For details about MSTPCR1, see section 20, Power-Down Mode.

Figure 18.1 Block Diagram of the CPG

The function of each block in the CPG is as follows.

- PLL circuit 1

  PLL circuit 1 multiplies the input clock frequency on the PLL circuit by 32 or 64.
- PLL circuit 2

  PLL circuit 2 matches the phases of the bus clock (Bck) and the clock of the CLKOUT pin that is used in the local bus.
- Crystal oscillator circuit

  The crystal oscillator circuit is used when a crystal resonator is connected to the XTAL and EXTAL pins. The crystal oscillator circuit can be used by the MODE9 pin setting.
- Divider

  Divider divides the input clock frequency from the crystal oscillator circuit or the EXTAL pin and generates the CPU clock (Ick), SuperHyway clock (SHck), DU clock (DUck), peripheral clock (Pck), DDR clock (DDRck), and external bus clock (Bck). The division ratio is selected by mode setting pins MODE0, MODE1, MODE2, and MODE3.

18.2 Input/Output Pins

Table 18.1 shows the CPG pin configuration.

Table 18.1 CPG Pin Configuration

Pin Name	Function	I/O	Description
MODE0, MODE1, MODE2, MODE3,	Mode Pins 0,1,2,3 Clock operating mode*1	Input	Select the clock operating mode These pins are multiplexed with the following pins. MODE0: the SCIF0_TXD (SCIF channel 0), $\overline{\text{IRL4}}$ (INTC), SDIF0D0 (SDIF channel 0), and PH0 (GPIO) pins MODE1: the SCIF0_RXD (SCIF channel 0), $\overline{\text{IRL5}}$ (INTC), SDIF0D1 (SDIF channel 0), and PH1 (GPIO) pins MODE2: the SCIF0_SCK (SCIF channel 0), $\overline{\text{IRL6}}$ (INTC), SDIF0D2 (SDIF channel 0), and PH2 (GPIO) pins MODE3: the $\overline{\text{SCIF0\_RTS}}$ (SCIF channel 0), $\overline{\text{IRL7}}$ (INTC), SDIF0D3 (SDIF channel 0), and PH3 (GPIO) pins
MODE9	Mode Pin 9 Clock input mode*1	Input	Selects whether the crystal resonator is used When MODE9 is set to the low level, the external clock is input from the EXTAL pin. When MODE9 is set to the high level, the crystal resonator is connected directly to the EXTAL and XTAL pins. MODE9 is multiplexed with the SCIF4_RXD (SCIF channel 4), DRAK1(DMAC), SSI3_SDATA(SSI3), PJ2(GPIO)pins.
XTAL	Clock Pins	Output	Connected to a crystal resonator
EXTAL		Input	Used to input an external clock or connected to a crystal resonator.
CLKOUT*2		Output	Used to output a local bus clock
CLKOUTENB	Clock Output Enabled	Output	The low level is output when the output clock of the CLKOUT is unstable. When the input to the $\overline{\text{PRESET}}$ pin is the low level, the high level is output regardless of the status of the output clock on the CLKOUT pin.

Note: The clock operating mode and the clock input mode depend on the states of the mode pins on a power-on reset via the $\overline{\text{PRESET}}$ pin.

18.3 Clock Operating Modes

Table 18.2 shows the relationship between setting of the mode pins (MODE0 to MODE3) and the clock operating modes.

Table 18.2 Clock Operating Modes and Operations of the Divider and PLLs

Clock Operating Mode	Setting of Mode Control Pins*1*2					
	MODE3	MODE2	MODE1	MODE0	PLL1	PLL2
0	L	L	L	L	On (× 64)	On
1	L	L	L	H	On (× 64)	On
2	L	L	H	L	On (× 64)	On
3	L	L	H	H	On (× 32)	On
4	L	H	L	L	On (× 32)	On
5	L	H	L	H	On (× 32)	On

Notes: 1. For the MODE0 to MODE3 pins, setting except the above mode pins (MODE0 to MODE3) is prohibited.
2. L stands for low level, and H stands for high level.

Table 18.3 Clock Operating Modes and Frequency Multiplication Ratio for Each Clock

Clock Operating Mode	FRQMR1 Initial Value	Frequency Multiplication Ratio (for Input Clock)					
		CPU Clock Ick	SuperHyway Clock SHck	DU Clock DUck	Bus Clock Bck	Peripheral Clock Pck	DDR Clock DDRck
0	H'1F25 1F48	× 32	× 16	× 8	× 16/3	× 8/3	× 32
1	H'1F26 1F46	× 32	× 16	× 8	× 4	× 4	× 32
2	H'1F29 1F49	× 32	× 16	× 8	× 2	× 2	× 32
3	H'1F25 1F48	× 16	× 8	× 4	× 8/3	× 4/3	× 16
4	H'1F26 1F46	× 16	× 8	× 4	× 2	× 2	× 16
5	H'1F29 1F49	× 16	× 8	× 4	× 1	× 1	× 16

18.4 Register Descriptions

Table 18.4 lists the registers. Table 18.5 shows the register states in each processing mode.

Table 18.4 Register Configuration

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync Clock
Frequency control register 0	FRQCR0	R/W	H'FFC4 0000	H'1FC4 0000	32	Pck
Frequency control register 1	FRQCR1	R/W	H'FFC4 0004	H'1FC4 0004	32	Pck
Frequency display register 1	FRQMR1	R	H'FFC4 0014	H'1FC4 0014	32	Pck
PLL control register	PLLCR	R/W	H'FFC4 0024	H'1FC4 0024	32	Pck
Standby control register 0*	MSTPCR0	R/W	H'FFC4 0030	H'1FC4 0030	32	Pck
Standby control register 1*	MSTPCR1	R/W	H'FFC4 0034	H'1FC4 0034	32	Pck
Standby display register*	MSTPMR	R	H'FFC4 0044	H'1FC4 0044	32	Pck

Note: 1. For details of these registers, see section 20, Power-Down Mode.

Table 18.5 Register State in Each Processing Mode

Register Name	Abbreviation	Power-on Reset by the $\overline{\text{PRESET}}$ Pin, WDT, or H-UDI	Manual Reset by WDT or Multiple Exception	Sleep or Light Sleep by SlEEP Instruction
Frequency control register 0	FRQCR0	H'0000 0000	Retained	Retained
Frequency control register 1	FRQCR1	H'0000 0000	Retained	Retained
Frequency display register 1	FRQMR1	H'1F2x 1F4x*[2]	Retained	Retained
PLL control register	PLLCR	H'0000 0000	Retained	Retained
Standby control register 0*[1]	MSTPCR0	H'0000 0000	Retained	Retained
Standby control register 1*[1]	MSTPCR1	H'0000 0000	Retained	Retained
Standby display register*[1]	MSTPMR	H'00x8 0000*[3]	Retained	Retained

Notes: 1. For details of these registers, see section 20, Power-Down Mode.
2. The state of this register depends on the settings of mode pins MODE0 to MODE3 obtained on a power-on reset via the $\overline{\text{PRESET}}$ pin. See table 18.3.
3. The state of this register depends on the setting of the mode pin MPMD obtained on a power-on reset via the $\overline{\text{PRESET}}$ pin.

18.4.1 Frequency Control Register 0 (FRQCR0)

FRQCR0 is a 32-bit readable and partially writable register that executes a sequence for changing the frequency of each clock. After the sequence is executed, FRQCR0 is automatically cleared to 0. FRQCR0 can only be accessed in longwords.

To write to FRQCR0, set the code value (H'CF) in the upper byte and use the longword. No other code values can be written. The code value is always read as 0.

FRQCR0 is initialized by only a power-on reset via the $\overline{\text{PRESET}}$ pin, a WDT overflow, or the H-UDI.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Code value (H'CF)								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRQE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R/W	Code value (H'CF) These bits are always read as 0. The write value should always be H'CF.
23 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FRQE	0	R/W	Frequency Change Sequence Enabled Enables the execution of a sequence that changes the frequency of each clock according to the value set in FRQCR1. After executing the sequence, this bit is automatically cleared to 0. 0: Execution of a sequence that changes the frequency is disabled. 1: Execution of a sequence that changes the frequency is enabled. Note: Some division ratio settings are prohibited. When a value that is not shown in Tables 18.7 to 18.8 is set in FRQCR1, do not set 1 in FRQE.

18.4.2 Frequency Control Register 1 (FRQCR1)

FRQCR1 is a 32-bit readable/writable register that can select the division ratio of divider for the CPU clock (lck), the peripheral clock (Pck), and the bus clock (Bck). To check the division ratio of divider for each clock, read FRQMR1. FRQCR1 can only be accessed in longwords.

FRQCR1 only changes the division ratio of a clock to which a value other than H'0 has been written. Therefore, set a value other than H'0 in the bit corresponding to the clock for which you want to change the division ratio. Other bits should be set to H'0.

To change the division ratio of each clock to the value set in FRQCR1, you must set 1 in the FRQE bit in FRQCR0 to execute the sequence that changes the frequency. After the sequence is executed, this register is automatically cleared to H'0000 0000.

FRQCR1 is initialized by only a power-on reset via the PRESET# pin or a WDT overflow.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IFC3	IFC2	IFC1	IFC0	—	—	—	—	—	—	—	—	BFC3	BFC2	BFC1	BFC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PFC3	PFC2	PFC1	PFC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	IFC3	0	R/W	Frequency division ratio of the CPU clock (Ick)
30	IFC2	0	R/W	0000: No change
29	IFC1	0	R/W	0001: × 1/2
28	IFC0	0	R/W	0010: × 1/4
				Others: Setting prohibited
27 to 20	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, the operation is not guaranteed.
19	BFC3	0	R/W	Frequency division ratio of the bus clock (Bck)
18	BFC2	0	R/W	0000: No change
17	BFC1	0	R/W	0101: × 1/12
16	BFC0	0	R/W	0110: × 1/16
				1000: × 1/24
				1001: × 1/32
				Others: Setting prohibited
15 to 4	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, the operation is not guaranteed.
3	PFC3	0	R/W	Frequency division ratio of the peripheral clock (Pck)
2	PFC2	0	R/W	
1	PFC1	0	R/W	0000: No change
0	PFC0	0	R/W	0110: × 1/16
				1000: × 1/24
				1001: × 1/32
				1011: × 1/48
				Others: Setting prohibited

18.4.3 Frequency Display Register 1 (FRQMR1)

FRQMR1 is a 32-bit readable register that reads the division ratio of divider for the CPU clock (lck), the SuperHyway clock (SHck), the peripheral clock (Pck), the DDR clock (DDRck),the bus clock (Bck), and the DU clock (DUck). FRQMR1 can only be accessed in longwords.

This register is initialized by only a power-on reset via the $\overline{\text{PRESET}}$ pin, a WDT overflow, and the H-UDI.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IFST3	IFST2	IFST1	IFST0	—	—	—	—	SFST3	SFST2	SFST1	SFST0	BFST3	BFST2	BFST1	BFST0
Initial value:	0	0	0	1	1	1	1	1	0	0	1	x	x	x	x	x
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MFST3	MFST2	MFST1	MFST0	—	—	—	—	S3FST3	S3FST2	S3FST1	S3FST0	PFST3	PFST2	PFST1	PFST0
Initial value:	0	0	0	1	1	1	1	1	0	1	0	0	x	x	x	x
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: The initial value (x: a bit whose value is undefined) depends on the settings of mode pins MODE0 to MODE3 on a power-on reset via the $\overline{\text{PRESET}}$ pin. See Table 18.3.

Bit	Bit Name	Initial Value	R/W	Description
31	IFST3	0	R	Frequency division ratio of the CPU clock (Ick)
30	IFST2	0	R	0001: × 1/2
29	IFST1	0	R	0010: × 1/4
28	IFST0	1	R	
27 to 24	—	All 1	R	Reserved These bits are always read as 1.
23	SFST3	0	R	Frequency division ratio of the SuperHyway clock (SHck) 0010: × 1/4
22	SFST2	0	R	
21	SFST1	1	R	
20	SFST0	0	R	
19	BFST3	x	R	Frequency division ratio of the bus clock (Bck)
18	BFST2	x	R	0101: × 1/12
17	BFST1	x	R	0110: × 1/16
16	BFST0	x	R	1000: × 1/24 1001: × 1/32

Bit	Bit Name	Initial Value	R/W	Description
15	MFST3	0	R	Frequency division ratio of the DDR clock (DDRck)
14	MFST2	0	R	0010: ×1/4
13	MFST1	0	R	
12	MFST0	1	R	
11 to 8	—	All 1	R	Reserved These bits are always read as 1.
7	S3FST3	0	R	Frequency division ratio of the DU clock (DUck)
6	S3FST3	1	R	0100: × 1/8
5	S3FST3	0	R	
4	S3FST3	0	R	
3	PFST3	x	R	Frequency division ratio of the peripheral clock (Pck) 0110: × 1/16 1000: × 1/24 1001: × 1/32 1011: × 1/48
2	PFST2	x	R	
1	PFST1	x	R	
0	PFST0	x	R	

18.4.4 PLL Control Register (PLLCR)

PLLCR is a 32-bit readable/writable register that controls the clock output on the CLKOUT pin. This register can only be accessed in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CKOFF	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

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	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, the operation is not guaranteed.
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	CKOFF	0	R/W	CLKOUT Output Enabled Stops clock output on the CLKOUT pin 0: Clock is output on the CLKOUT pin 1: The CLKOUT pin is placed in the high impedance state.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

18.5 Calculating the Frequency

Table 18.6 shows the relationship between the division ratio of divider described for frequency control register FRQCR1 and frequency display register FRQMR1, and the EXTAL input.

Table 18.6 Relationship Between the Division Ratio of Divider and the Frequency

	Frequency (for an input clock)	
Division ratio of divider	Clock operating mode 0 to 2	Clock operating mode 3 to 5
× 1/2	× 32	× 16
× 1/4	× 16	× 8
× 1/8	× 8	× 4
× 1/12	× 16/3	× 8/3
× 1/16	× 4	× 2
× 1/24	× 8/3	× 4/3
× 1/32	× 2	× 1
× 1/48	× 4/3	× 2/3

18.6 How to Change the Frequency

To change the frequency of the internal clock and the local bus clock (CLKOUT) with software, set frequency control registers FRQCR0 and FRQCR1 according to the following procedure. Tables 18.7 to 18.8 list the selectable combinations of frequencies.

18.6.1 Changing the Frequency of Clocks Other than the Bus Clock

When changing the frequency of a clock except the bus clock, disable counting-up by the WDT.

The following describes the procedure for changing the frequency.

1. In FRQCR1, set a value (other than H'0) in the bit corresponding to the clock for which you want to change the division ratio.*
2. Set H'CF000001 in FRQCR0 to enable execution of the sequence that changes the frequency. The sequence that changes the frequency starts.
3. When H'00000000 is read from FRQCR0, the sequence that changes the frequency has finished. The internal clock has been changed to the clock with the specified division ratio.

18.6.2 Changing the Bus Clock Frequency

When changing the bus clock frequency, start counting-up by the WDT after the oscillation of PLL circuit 2 is stable. When a WDT overflow occurs during counting, this LSI resumes operation.

Figures 18.2 and 18.3 show the timing of the CLKOUT and CLKOUTENB pins when the bus clock frequency is changed.

The following describes the procedure for changing the frequency.

1. Write 0 to the TME bit in WDTCSR to stop the WDT.
2. In WDTBST, after the oscillation of PLL circuit 2 is stable, set the time that can elapse before the LSI resumes operation. Writing H'55000001 sets the minimum value. Writing H'55000000 sets the maximum value.
3. In FRQCR1, set a value (except H'0) in the bit corresponding to the clock for which you want to change the division ratio.*
4. Set H'CF000001 in FRQCR0 to enable execution of the sequence that changes the frequency. The sequence that changes the frequency starts.
5. The CLKOUTENB pin output changes to low level. After ten cycles of the peripheral clock (Pck), an unstable clock is output to the CLKOUT pin.

6. When the oscillation of PLL circuit 2 is stable, wait for 24 cycles of the peripheral clock (Pck). Then output a high level signal to the CLKOUTENB pin.
7. When the WDT starts counting up and the value of WDTBCNT is equal to the value of WDTBST, the LSI resumes operation.
8. When H'00000000 is read from FRQCR0, the sequence that changes the frequency has finished. The internal clock has been changed to the clock with the specified division ratio.

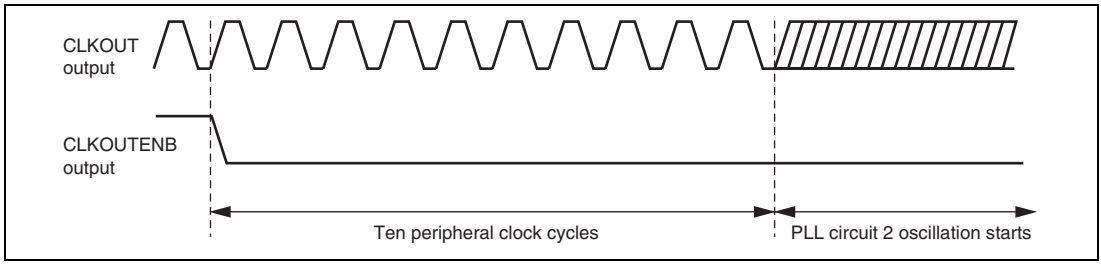


Figure 18.2 Beginning of the Change of the Bus Clock Frequency

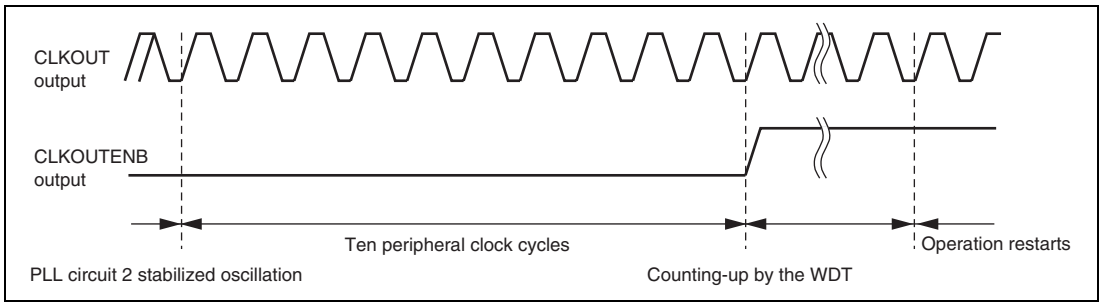


Figure 18.3 End of the Change of the Bus Clock Frequency

Table 18.7 Selectable Combinations of Clock Frequency (CPU Clock: × 1/2, DDR Clock: × 1/4)

	Division ratio of divider					
FRQMR1 read value	CPU clock Ick	SuperHyway clock SHck	DU clock DUck	Bus clock Bck	Peripheral clock Pck	DDR clock DDRck
H'1F25 1F48	× 1/2	× 1/4	× 1/8	× 1/12	× 1/24	× 1/2
H'1F25 1F4B	× 1/2	× 1/4	× 1/8	× 1/12	× 1/48	× 1/2
H'1F26 1F46	× 1/2	× 1/4	× 1/8	× 1/16	× 1/16	× 1/2
H'1F26 1F49	× 1/2	× 1/4	× 1/8	× 1/16	× 1/32	× 1/2
H'1F26 1F4B	× 1/2	× 1/4	× 1/8	× 1/16	× 1/48	× 1/2
H'1F28 1F48	× 1/2	× 1/4	× 1/8	× 1/24	× 1/24	× 1/2
H'1F28 1F4B	× 1/2	× 1/4	× 1/8	× 1/24	× 1/48	× 1/2
H'1F29 1F49	× 1/2	× 1/4	× 1/8	× 1/32	× 1/32	× 1/2

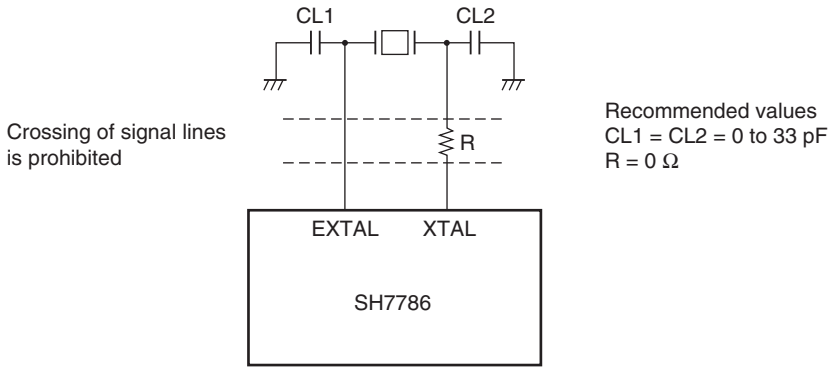
Table 18.8 Selectable Combinations of Clock Frequency (CPU Clock: × 1/4, DDR Clock: × 1/4)

	Division ratio of divider					
FRQMR1 read value	CPU clock Ick	SuperHyway clock SHck	DU clock DUck	Bus clock Bck	Peripheral clock Pck	DDR clock DDRck
H'2F25 1F48	× 1/4	× 1/4	× 1/8	× 1/12	× 1/24	× 1/2
H'2F25 1F4B	× 1/4	× 1/4	× 1/8	× 1/12	× 1/48	× 1/2
H'2F26 1F46	× 1/4	× 1/4	× 1/8	× 1/16	× 1/16	× 1/2
H'2F26 1F49	× 1/4	× 1/4	× 1/8	× 1/16	× 1/32	× 1/2
H'2F26 1F4B	× 1/4	× 1/4	× 1/8	× 1/16	× 1/48	× 1/2
H'2F28 1F48	× 1/4	× 1/4	× 1/8	× 1/24	× 1/24	× 1/2
H'2F28 1F4B	× 1/4	× 1/4	× 1/8	× 1/24	× 1/48	× 1/2
H'2F29 1F49	× 1/4	× 1/4	× 1/8	× 1/32	× 1/32	× 1/2

18.7 Notes on Designing Board

1. Note on Using a Crystal Resonator

   Place the crystal resonator and capacitors close to the EXTAL and XTAL pins as much as possible. No other signal lines should cross the signal line of these pins. Induction may prevent correct oscillation.



Note: The values of CL1, CL2, and the dumping resistance should be determined through evaluation and consultation with the manufacturer of the crystal resonator to be used.

Figure 18.4 Note on Using a Crystal Resonator

2. Note on Inputting the External Clock from the EXTAL Pin

   Do not connect anything to the XTAL pin.

3. Note on Using a PLL Oscillator Circuit

   Place VREFA, VSSAI, VDDA, and VSSA away from other VDDs and VSSs on the power supply of the board. Insert resistors, RCB , and bypass capacitors, CPB and CB, as noise filters near the pins.

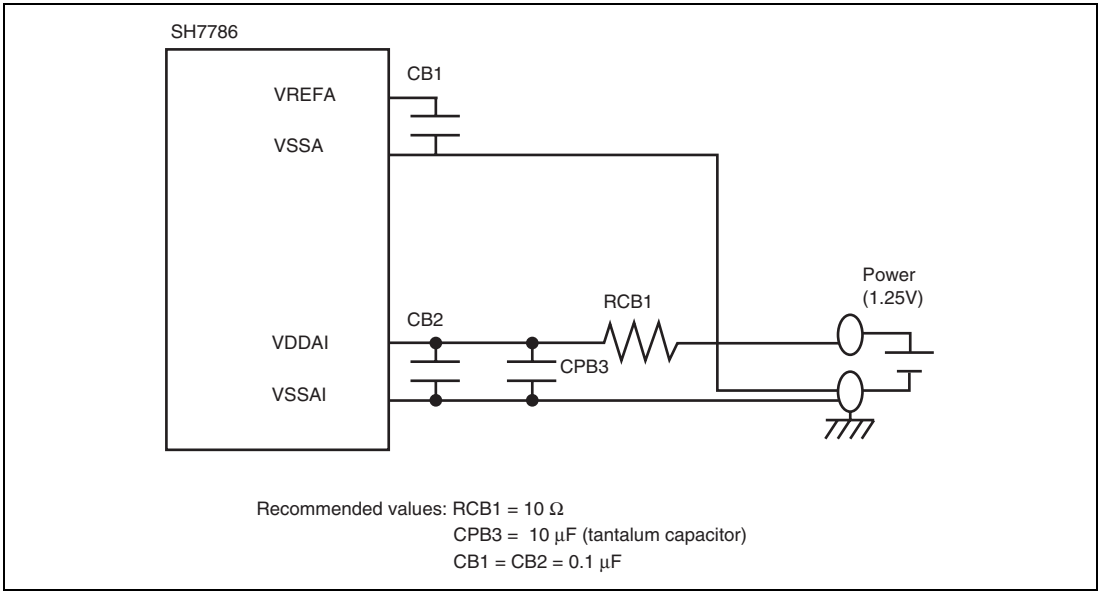


Figure 18.5 Note on Using a PLL Oscillator Circuit

Section 19 Watchdog Timer and Reset (WDT)

The watchdog timer and reset module (WDT) comprises a reset control unit and a watchdog timer control unit, and controls the power-on reset sequence and internal reset of the LSI.

The WDT is a single-channel timer that can be used either as a watchdog timer or interval timer.

In addition to the above-mentioned WDT, each of the CPUs has an individual watchdog timer that can issue manual resets: WDT (CPU0) and WDT (CPU1).

19.1 Features

19.1.1 WDT Features

- The watchdog timer unit monitors for system runaway using a timer counting at regular time intervals.
- Two operating modes:
  - In watchdog timer mode, internal reset of the chip is initiated on counter overflow and on-chip modules are reset.
  - In interval timer mode, an interrupt is generated on counter overflow.
- Supports power-on reset.
- In order to prevent accidental writing to the WDT-related registers, writing to them is only possible when a certain code is set in the uppermost eight bits in the data for writing.

19.1.2 Features of WDT (CPU0) and WDT (CPU1)

- These watchdog timers, which are incremented at a fixed intervals, are used to monitor for system runaway.
- WDT (CPU0) issues a manual reset to CPU0 when counter overflow occurs.
- WDT (CPU1) issues a manual reset to CPU1 when counter overflow occurs.
- To prevent the registers related to WDT (CPU0) and WDT (CPU1) from being written to accidentally, a specific code must be set in the upper eight bits when writing to them.

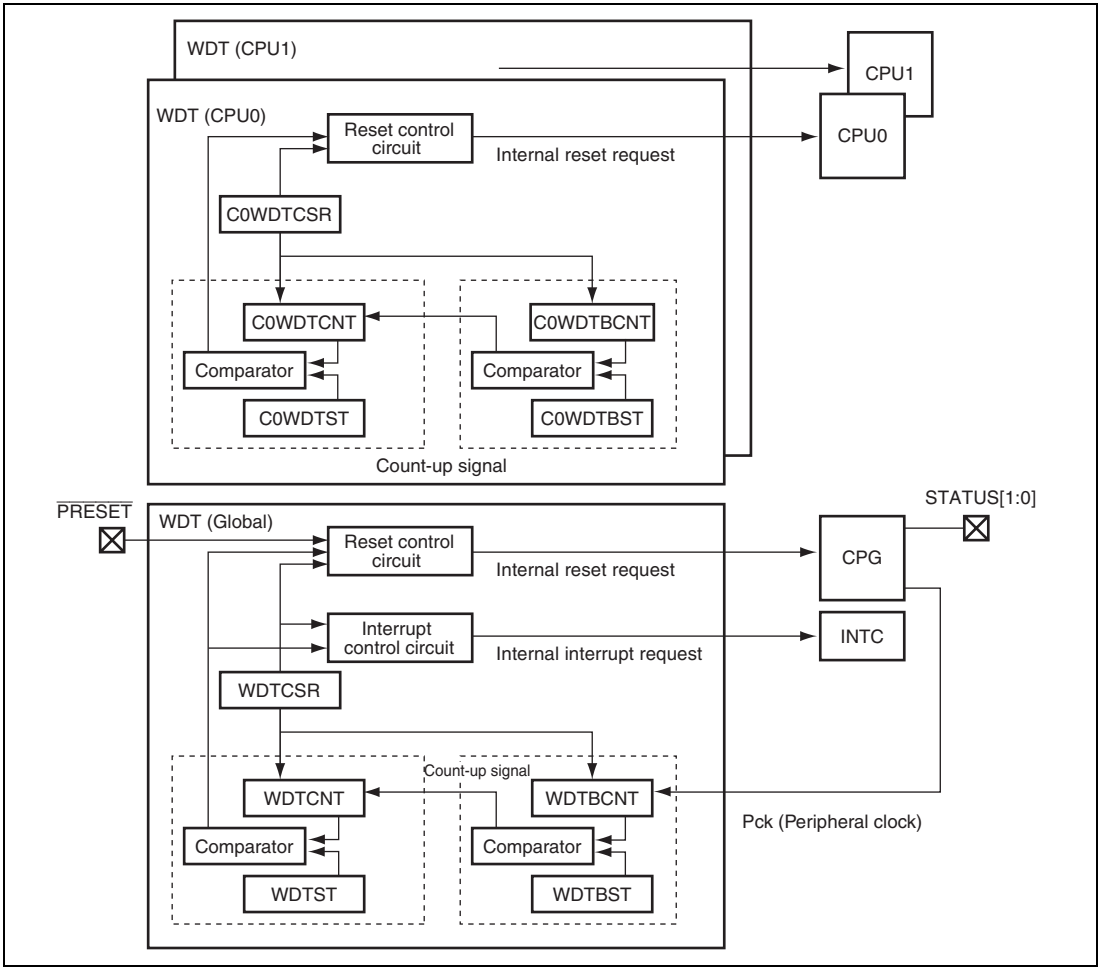


Figure 19.1 Block Diagram of WDT

19.2 Input/Output Pins

Table 19.1 shows the pin configuration of the WDT module.

Table 19.1 Pin Configuration

Pin name	Function	I/O	Description
$\overline{\text{PRESET}}$	Power-on reset input	Input	A low level input to this pin places the LSI in the power-on reset state.
STATUS[1:0]	Status output	Output	Indicate CPU0's operating status STATUS1 / STATUS0 / Operating Status High / High / Reset High / Low / Sleep mode Low / Low / Normal operation The STATUS0 pin is multiplexed with the SSI2_CLK (SSI channel 2) pin. The STATUS1 pin is multiplexed with the SSI3_CLK (SSI channel 3) pin.

19.3 Register Descriptions

Table 19.2 shows the registers of the WDT module and reset vector. Table 19.3 shows the register states in each operating mode. The abbreviation of the registers for each CPU is described as CnXXX; C0XXX is for CPU 0 and C1XXX for CPU 1.

Table 19.2 Register Configuration

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync Clock
Watchdog timer stop time register	WDTST	R/W	H'FFC8 0000	H'1FC8 0000	32	Pck
Watchdog timer control/status register	WDTCSR	R/W	H'FFC8 0004	H'1FC8 0004	32	Pck
Watchdog timer base stop time register	WDTBST	R/W	H'FFC8 0008	H'1FC8 0008	32	Pck
Watchdog timer counter	WDTCNT	R	H'FFC8 0010	H'1FC8 0010	32	Pck
Watchdog timer base counter	WDTBCNT	R	H'FFC8 0018	H'1FC8 0018	32	Pck
CPU0 watchdog timer stop time register	C0WDTST	R/W	H'FE40 0080	H'1E40 0080	32	Pck

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync Clock
CPU0 watchdog timer control/status register	C0WDTCSR	R/W	H'FE40 0084	H'1E40 0084	32	Pck
CPU0 watchdog timer base stop time register	C0WDTBST	R/W	H'FE40 0088	H'1E40 0088	32	Pck
CPU0 watchdog timer counter	C0WDTCNT	R	H'FE40 0090	H'1E40 0090	32	Pck
CPU0 watchdog timer base counter	C0WDTBCNT	R	H'FE40 0098	H'1E40 0098	32	Pck
CPU0 reset vector set register	C0RESETVEC	R/W	H'FE40 0008	H'1E40 0008	32	Pck
CPU1 watchdog timer stop time register	C1WDTST	R/W	H'FE40 1080	H'1E40 1080	32	Pck
CPU1 watchdog timer control/status register	C1WDTCSR	R/W	H'FE40 1084	H'1E40 1084	32	Pck
CPU1 watchdog timer base stop time register	C1WDTBST	R/W	H'FE40 1088	H'1E40 1088	32	Pck
CPU1 watchdog timer counter	C1WDTCNT	R	H'FE40 1090	H'1E40 1090	32	Pck
CPU1 watchdog timer base counter	C1WDTBCNT	R	H'FE40 1098	H'1E40 1098	32	Pck
CPU1 reset vector set register	C1RESETVEC	R/W	H'FE40 1008	H'1E40 1008	32	Pck

Table 19.3 Register States in Each Operating Mode

Register Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin	Power-on Reset by WDT/H-UDI	Manual Reset by WDT/ Multiple Exception	Sleep/Light Sleep Mode by SLEEP Instruction
Watchdog timer stop time register	WDTST	H'0000 0000	Retained	Retained	Retained
Watchdog timer control/status register	WDTCSR	H'0000 0000	Retained	Retained	Retained
Watchdog timer base stop time register	WDTBST	H'0000 0000	Retained	Retained	Retained
Watchdog timer counter	WDTCNT	H'0000 0000	H'0000 0000	Retained	Retained
Watchdog timer base counter	WDTBCNT	H'0000 0000	H'0000 0000	Retained	Retained
CPU0 watchdog timer stop time register	C0WDTST	H'0000 0000	Retained	Retained	Retained
CPU0 watchdog timer control/status register	C0WDTCSR	H'0000 0000	Retained	Retained	Retained
CPU0 watchdog timer base stop time register	C0WDTBST	H'0000 0000	Retained	Retained	Retained
CPU0 watchdog timer counter	C0WDTCNT	H'0000 0000	Retained	Retained	Retained
CPU0 watchdog timer base counter	C0WDTBCNT	H'0000 0000	Retained	Retained	Retained
CPU0 reset vector set register	C0RESETVEC	H'A000 0000	Retained	Retained	Retained
CPU1 watchdog timer stop time register	C1WDTST	H'0000 0000	Retained	Retained	Retained
CPU1 watchdog timer control/status register	C1WDTCSR	H'0000 0000	Retained	Retained	Retained
CPU1 watchdog timer base stop time register	C1WDTBST	H'0000 0000	Retained	Retained	Retained
CPU1 watchdog timer counter	C1WDTCNT	H'0000 0000	Retained	Retained	Retained
CPU1 watchdog timer base counter	C1WDTBCNT	H'0000 0000	Retained	Retained	Retained
CPU1 reset vector set register	C1RESETVEC	H'A000 0000	Retained	Retained	Retained

19.3.1 Watchdog Timer Stop Time Register (WDTST)

WDTST is a 32-bit readable/writable register that specifies the time until watchdog timer counter WDTCNT overflows. The time until WDTCNT overflows becomes minimum when H'5A00 0001 is set, and maximum when H'5A00 0000 is set.

WDTST should be written as a longword unit, with H'5A in the most significant byte. The value read from this byte is always H'00. WDTST is only rest by a power-on reset caused by the $\overline{\text{PRESET}}$ pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Code for writing (H'5A)								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTST											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Code for writing)	All 0	R/W	Code for writing (H'5A) These bits are always read as H'00. When writing to this register, the value written to these bits must be H'5A.
23 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	WDTST	All 0	R/W	Timer Stop These bits set the counter value at which WDTCNT overflows. H'001: Minimum overflow value H'000: Maximum overflow value

19.3.2 Watchdog Timer Control/Status Register (WDTCSR)

WDTCSR is a 32-bit readable/writable register comprising timer mode-selecting bits and overflow flags.

WDTCSR should be written to as a longword unit, with H'5A in the most significant byte. The value read from this byte is always H'00. WDTCSR is only rest by a power-on reset caused by the $\overline{\text{PRESET}}$ pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Code for writing (H'5A)								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TME	WT/$\overline{\text{IT}}$	RSTS	WOVF	IOVF	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Code for writing)	All 0	R/W	Code for writing (H'5A) These bits are always read as H'00. When writing to this register, the value written to these bits must be H'5A.
23 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TME	0	R/W	Timer Enable Starts or stops the timer operation. 0: Stops counting up. 1: Starts counting up.
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Specifies whether the WDT is used as a watchdog timer or interval timer. Up counting may not be performed correctly if this bit is modified while the WDT is running. 0: Interval timer mode 1: Watchdog timer mode

Bit	Bit Name	Initial Value	R/W	Description
5	RSTS	0	R/W	Reset Select Specifies the type of reset on WDTCNT overflow in watchdog timer mode. This setting is ignored in interval timer mode. 0: Power-on reset 1: Setting prohibited
4	WOVF	0	R/W	Watchdog Timer Overflow Flag Indicates that WDTCNT has overflowed in watchdog timer mode. This flag is not set in interval timer mode. 0: WDTCNT has not overflowed. 1: WDTCNT has overflowed.
3	IOVF	0	R/W	Interval Timer Overflow Flag Indicates that WTCNT has overflowed in interval timer mode. This flag is not set in watchdog timer mode. 0: WDTCNT has not overflowed. 1: WDTCNT has overflowed.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

19.3.3 Watchdog timer Base Stop Time Register (WDTBST)

WDTBST is a 32-bit readable/writable register that specifies the time until counter WDTBCNT overflows when the bus clock frequency has been changed. The time until WDTBCNT overflows becomes minimum when H'5500 0001 is set, and maximum when H'5500 0000 is set.

WDTBST should be written to as a longword unit, with H'55 in the most significant byte. The value read from this byte is always H'00. WDTBST is only rest by a power-on reset caused by the $\overline{\text{PRESET}}$ pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Code for writing (H'55)								—	—	—	—	—	—	WDTBST	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTBST															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Code for writing)	All 0	R/W	Code for writing (H'55) These bits are always read as H'00. When writing to this register, the value written to these bits must be H'55.
23 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17 to 0	WDTBST	All 0	R/W	Base Timer Stop These bits set the counter value at which WDTBCNT overflows. H'00001: Minimum overflow value H'00000: Maximum overflow value

19.3.4 Watchdog Timer Counter (WDTCNT)

WDTCNT is a 32-bit read-only register comprising a 12-bit counter that is incremented by the WDTBCNT overflow signal. When WDTCNT overflows, a reset of the selected type is initiated in watchdog timer mode, or an interrupt is generated in interval timer mode.

WDTCNT is only reset by a power-on reset. Writing to this register is invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTCNT											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	WDTCNT	All 0	R	Counter value

19.3.5 Watchdog Timer Base Counter (WDTBCNT)

WDTBCNT is a 32-bit read-only register comprising an 18-bit counter that is incremented by the peripheral clock (Pck). When WDTBCNT overflows, WDTCNT is incremented and WDTBCNT is cleared to H'0000 0000.

WDTBCNT is only reset by a power-on reset. Writing to this register is invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTBCNT	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTBCNT															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17 to 0	WDTBCNT	All 0	R	Base counter value

19.3.6 CPUn watchdog timer stop time register (CnWDTST) (n = 0 to 1)

CnWDTST is a 32-bit readable/writable register that specifies the time until CnWDTCNT overflows. The time until CnWDTCNT overflows becomes minimum when H'5A00 0001 is set to the WDTST bits, and maximum when H'5A00 0000 is set.

CnWDTST should be written as a longword unit, with H'5A in the most significant byte. The value read from this byte is always H'00.

This register is reset only by the power-on reset signal of the $\overline{\text{PRESET}}$ pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Code for writing (H'5A)								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTST											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Code for writing)	All 0	R/W	Code for writing (H'5A) These bits are always read as H'00. When writing to this register, the value written to these bits must be H'5A.
23 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	WDTST	All 0	R/W	Timer Stop These bits set the counter value at which CnWDTCNT overflows. H'001: Minimum overflow value H'000: Maximum overflow value

19.3.7 CPUn Watchdog Timer Control/Status Register (CnWDTCSR) (n = 0 to 1)

CnWDTCSR is a 32-bit readable/writable register comprising timer enable-selecting bits and overflow flags.

CnWDTCSR should be written to as a longword unit, with H'5A in the most significant byte. The value read from this byte is always H'00.

This register is reset only by the power-on reset signal of the $\overline{\text{PRESET}}$ pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Code for writing (H'5A)								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TME	—	—	WOVF	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Code for writing)	All 0	R/W	Code for writing (H'5A) These bits are always read as H'00. When writing to this register, the value written to these bits must be H'5A.
23 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TME	0	R/W	Timer Enable Starts or stops the timer operation. 0: Stops counting up. 1: Starts counting up.
6 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	WOVF	0	R/W	Watchdog Timer Overflow Flag Indicates that CnWDTCNT has overflowed. This flag is not set in interval timer mode. 0: CnWDTCNT has not overflowed. 1: CnWDTCNT has overflowed.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

19.3.8 CPUn Watchdog timer Base Stop Time Register (CnWDTBST) (n = 0 to 1)

CnWDTST is a 32-bit readable/writable register that specifies the time until CnWDTBCNT overflows. The time until CnWDTBCNT overflows becomes minimum when H'5500 0001 is set to the WDTBST bits, and maximum when H'5500 0000 is set.

CnWDTST should be written as a longword unit, with H'55 in the most significant byte. The value read from this byte is always H'00.

This register is reset only by the power-on reset signal of the $\overline{\text{PRESET}}$ pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Code for writing (H'55)								WDTBST							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTBST															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Code for writing)	All 0	R/W	Code for writing (H'55) These bits are always read as H'00. When writing to this register, the value written to these bits must be H'55.
23 to 0	WDTBST	All 0	R/W	Timer Stop These bits set the counter value at which CnWDTBST overflows. H'001: Minimum overflow value H'000: Maximum overflow value

19.3.9 CPUn Watchdog Timer Counter (CnWDTCNT) (n = 0 to 1)

CnWDTCNT is a 32-bit read-only register that is incremented by the CnWDTBCNT overflow. When CnWDTCNT overflows, a manual reset to the CPUn (n=0, 1) is generated.

This register is reset only by the power-on reset. Writing to CnWDTCNT is invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTCNT											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	WDTCNT	All 0	R	Counter value of CnWDTCNT

19.3.10 CPUn Watchdog Timer Base Counter (CnWDTBCNT) (n = 0 to 1)

CnWDTBCNT is a 32-bit read-only register that is incremented by the peripheral clock (Pck). When CnWDTBCNT overflows, CnWDTCNT is incremented and CnWDTBCNT is cleared to H'0000 0000.

This register is reset only by the power-on reset. Writing to CnWDTBCNT is invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	WDTBCNT							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTBCNT															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	WDTBCNT	All 0	R	Base counter value of CnWDTBCNT

19.3.11 CPUn Reset Vector Set Register (CnRESETVEC) (n = 0 to 1)

Each CPU branches to the address of the reset vector (CnRESETVEC) when a reset is generated.

The reset vector is initialized to H'A000 0000 at the power-on reset by the $\overline{\text{PRESET}}$ pin, though not initialized by a manual reset by WDT.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RESTVEC [31:16]															
Initial value:	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESTVEC [15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RESTVEC	H'A000 0000	R/W	Reset Vector of CPUn (n = 0 to 1)

19.4 Operation

19.4.1 Reset Request

Power-on reset and manual reset are available. Their requesting sources are described below.

(1) Power-on reset

- Requesting sources
  - A low level input on the $\overline{\text{PRESET}}$ pin
  - WDTCNT overflow when the WT/$\overline{\text{IT}}$ bit is 1 and the RSTS bit is 0 in WDTCSR
  - The H-UDI reset (for details, see section 32, User Debugging Interface)

    For details, see section 32, User Debugging Interface (H-UDI).
- Branch address: H'A000 0000
- Operation until branching

  The exception code H'000 is set in EXPEVT. After initializing VBR and SR, the processing branches by setting PC = H'A000 0000.
  During initialization, the VBR register is rest to H'0000 0000. The SR register is initialized such that the MD, RB, and BL bits are set to 1, the FD bit is cleared to 0, and the interrupt mask level bits (IMASK3 to IMASK0) are set to B'1111.
  Then, the CPU and peripheral modules are initialized. For details, refer to the register descriptions in the corresponding sections.

  At power-on, ensure that a low level is input to the $\overline{\text{PRESET}}$ pin. A low level input is also needed on the $\overline{\text{TRST}}$ pin to initialize the H-UDI.

```
Power_on_reset()
{
        EXPEVT = H'0000 0000;
        VBR = H'0000 0000;
        SR.MD = 1;
        SR.RB = 1;
        SR.BL = 1;
        SR.(I0-I3) = B'1111;
        SR.FD = 0;
        Initialize_CPU();
        Initialize_Module(PowerOn);
        PC = H'A000 0000;
}
```

(2) Manual reset

- Requesting sources
  - A general exception other than a user break while the BL bit in SR is set to 1.
  - CnWDTCNT (n=0,1) overflow
- Branch address: CnRESETVEC (n=0,1) address
- Operation until branching

  The exception code H'020 is set in EXPEVT. After initializing VBR and SR, the processing branches by setting PC = CnRESETVEC.
  During initialization, the VBR register is rest to H'0000 0000. The SR register is initialized such that the MD, RB, and BL bits are set to 1, the FD bit is cleared to 0, and the interrupt mask level bits (IMASK3 to IMASK0) are set to B'1111.
  Then, the CPU and peripheral modules are initialized. For details, refer to the register descriptions in the corresponding sections.

```
Manual_reset()
{
    EXPEVT = H'0000 0020;
    VBR = H'0000 0000;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    SR.(I0-I3) = B'1111;
    SR.FD = 0;
    Initialize_CPU();
    Initialize_Module(Manual);
    PC = CnRESETVEC;
}
```

19.4.2 Using Watchdog Timer Mode

1. Set the WDTCNT/CnWDTCNT overflow time in WDTST/CnWDTST.
2. If WDTCNT is set, set the WT/$\overline{IT}$ bit in WDTCSR to 1, and select the type of reset with the RSTS bit.
3. When the TME bit in WTCSR/CnWTCSR is set to 1, the WDT counter starts.
4. In watchdog timer mode, clear the WDTCNT/CnWDTCNT or WDTBCNT/CnWDTBCNT periodically so that WDTCNT/CnWDTCNT does not overflow. See section 19.4.6, Clearing WDT Counter, for how to clear the WDT counter.
5. When the WDTCNT/CnWDTCNT overflows, the WDT sets the WOVF flag in WDTCSR/CnWDTCSR to 1, and generates a reset of the type specified by the RSTS bit. After the reset state is exited, WDTCNT/CnWDTCNT and WDTBCNT/CnWDTBCNT start counting again.

19.4.3 Using Interval Timer Mode

In interval timer mode, the WDT generates an interval timer interrupt each time the counter overflows. This allows interrupt generation at regular intervals.

1. Set the WDTCNT overflow time in WDTST.
2. Clear the WT/$\overline{IT}$ bit in WDTCSR to 0.
3. When the TME bit in WDTCSR is set to 1, the WDT counter starts.
4. When the WDTCNT overflows, the WDT sets the IOVF flag in WDTCSR to 1, generating an interval timer interrupt (ITI) request. WDTCNT and WDTBCNT continue counting.

19.4.4 Time Until WDT Counters Overflow

The relationship between WDTCNT and WDTBCNT is shown in figure 19.2. The example shown in the figure is the operation in interval timer mode, where WDTCNT restarts counting after it has overflowed. In watchdog timer mode, WDTCNT and WDTBCNT are cleared to 0 after the reset state is exited and start counting up again.

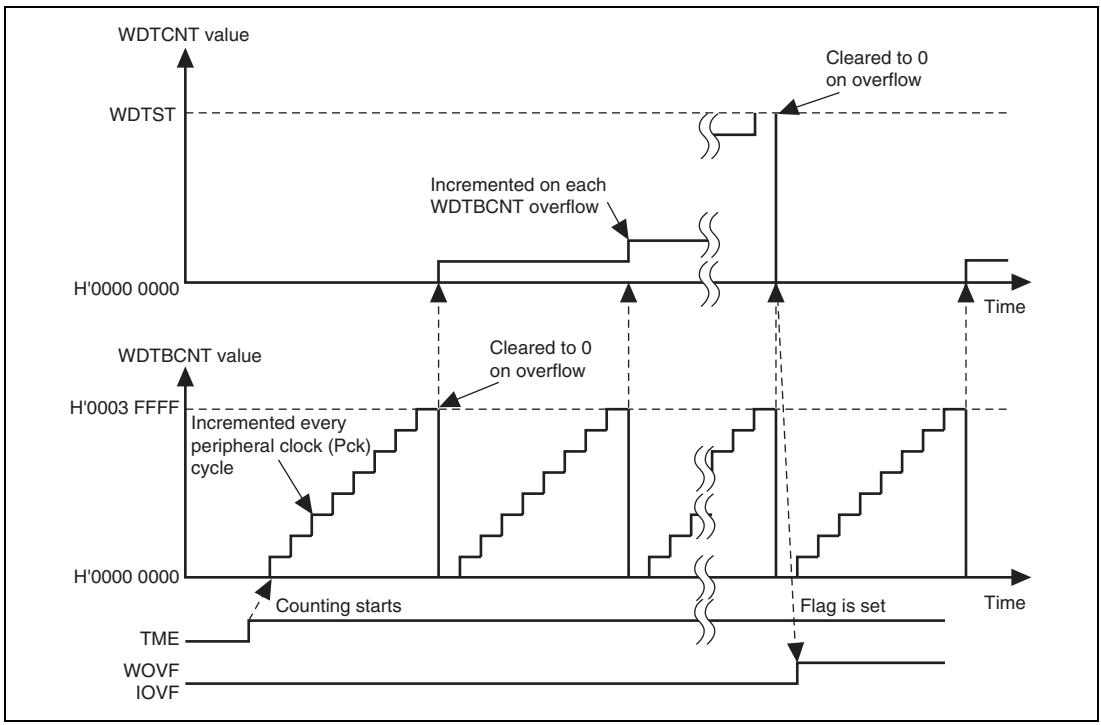


Figure 19.2 WDT Counting Operations (Example in Interval Timer Mode)

WDTBCNT is an 18-bit counter that is incremented by the peripheral clock. If the period of peripheral clock Pck is represented as tPck (ns), the overflow time of WDTBCNT is expressed as follows.

$$2^{18} \text{ [bit]} \times \text{tPck [ns]} = 0.262 \times \text{tPck [ms]}$$

WDTCNT is a 12-bit counter that is incremented each time WDTBCNT overflows. The time until WDTCNT/CnWDTCNT overflows becomes maximum when 0 is written to all the bits in WDTST. If the period of peripheral clock Pck is represented as tPck (ns), the maximum overflow time of WDTCNT is expressed as follows.

$$2^{12} \text{ [bit]} \times (0.262 \times \text{tPck}) \text{ [ms]} = 1.073 \times \text{tPck [s]}$$

The time until WDTCNT overflows becomes minimum when H'5A00 0001 is written to WDTST. In this case, the overflow time is equal to that of WDTBCNT.

For example, if the peripheral clock frequency is 66 MHz, tPck is 15 ns and the overflow times are as follows.

Overflow time of WDTBCNT: $0.262 \times 15 = 3.93$ [ms]
Maximum overflow time of WDTCNT: $1.073 \times 15 = 16.10$ [s]

19.4.5 Time Until WDT (CPU0) and WDT (CPU1) Counters Overflow

The relationship between CnWDTCNT and CnWDTBCNT is shown in figure 19.3. After manual reset state by the overflow of CnWDTCNT, CnWDTCNT and CnWDTBCNT are cleared to 0 and start counting up again.

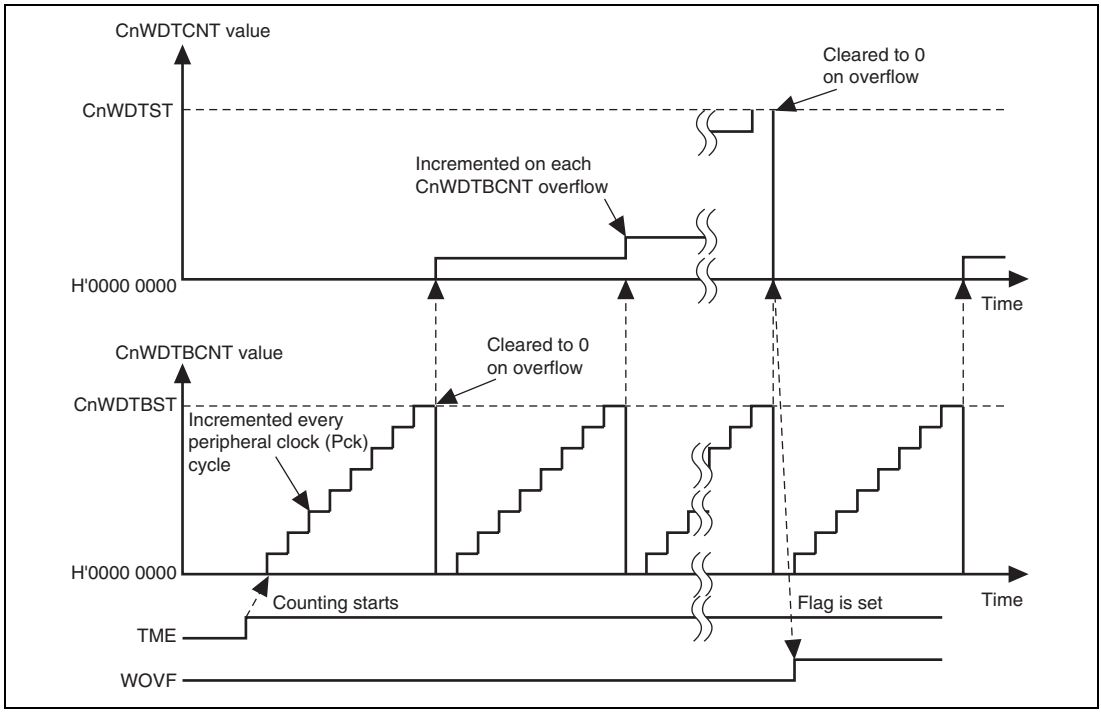


Figure 19.3 WDT (CPU0) and WDT (CPU1) Counting Operations (Only Watchdog Timer Mode)

CnWDTBCNT is a 24-bit counter that is incremented by the peripheral clock. The time until it overflows becomes maximum when H'5500 0000 is written to CnWDTBST. If the period of peripheral clock Pck is represented as tPck (ns), the maximum overflow time of CnWDTBCNT is expressed as follows.

$$2^{24}\ [\text{bit}] \times \text{tPck}\ [\text{ns}] = 16.777 \times \text{tPck}\ [\text{ms}]$$

CnWDTCNT is a 12-bit counter that is incremented each time CnWDTBCNT overflows. The time until CnWDTCNT overflows becomes maximum when 0 is written to all the bits in CnWDTST. If the period of peripheral clock Pck is represented as tPck (ns), the maximum overflow time of WDTCNT/CnWDTCNT is expressed as follows.

$$2^{12}\ [\text{bit}] \times (16.777 \times \text{tPck})\ [\text{ms}] = 68.718 \times \text{tPck}\ [\text{s}]$$

For example, if the peripheral clock frequency is 66 MHz, tPck is 15 ns and the overflow times are as follows.

Maximum overflow time of CnWDTCNT: $68.718 \times 15 = 1030$ [s]

19.4.6 Clearing WDT Counters

Setting the overflow value in WDTBST clears WDTBCNT to 0, and setting the overflow value in WDTST clears WDTCNT to 0.

19.5 Status Pin Change Timing during Reset

19.5.1 Power-On Reset by $\overline{\text{PRESET}}$ pin

Since the PLL circuit is initialized when the LSI enters the power-on reset state, the PLL oscillation settling time needs to be ensured. This means that a high level must not be input to the $\overline{\text{PRESET}}$ pin during the PLL oscillation settling time. The PLL oscillation settling time is the sum of the settling times for PLL1 and PLL2.

After the state on the $\overline{\text{PRESET}}$ pin input is changed from a low level to high level, the internal reset state continues until the reset holding time elapses. The reset holding time is equal to or more than 40 cycles of the peripheral clock (Pck).

(1) When the power is turned on

When the power is turned on, ensure that a low level is input to the $\overline{\text{PRESET}}$ pin. A low level input is also needed on the $\overline{\text{TRST}}$ pin to initialize the H-UDI.

The timing of reset state indication on the STATUS[1:0] pins is asynchronous. On the other hand, the timing of indicating normal operation is synchronous with the peripheral clock (Pck), and is therefore asynchronous with the clocks input from the EXTAL pin and the CLKOUT pin.

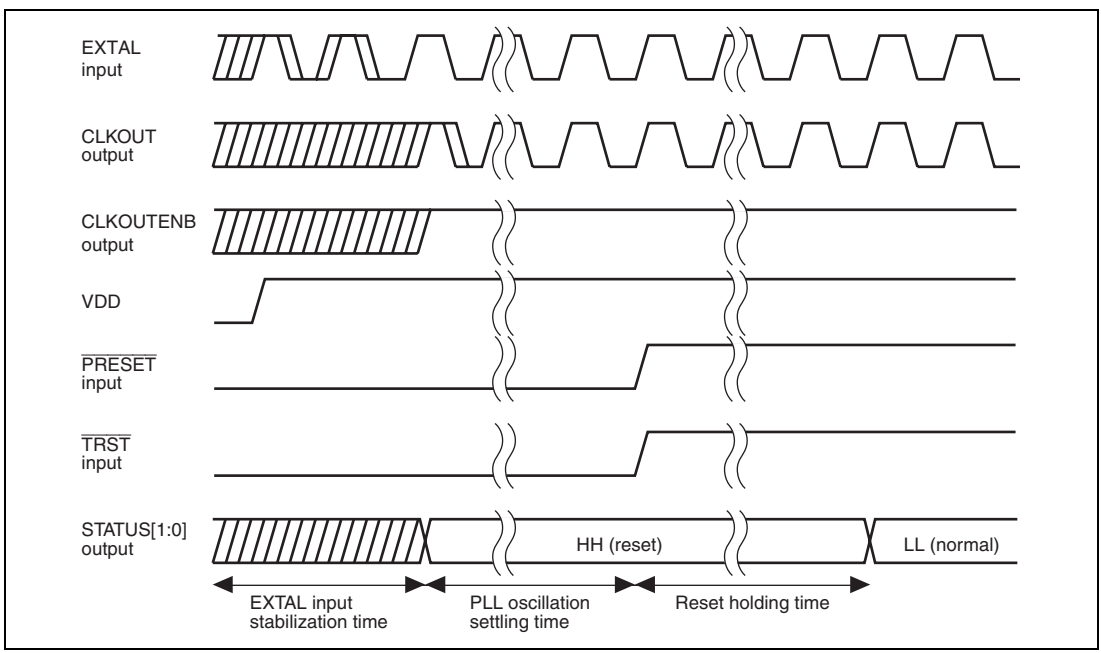


Figure 19.4 STATUS Output when Power is Turned On

(2) Power-on reset caused by $\overline{\text{PRESET}}$ input during normal operation

It is necessary to ensure the PLL oscillation settling time when a power-on reset is initiated by low level input on the $\overline{\text{PRESET}}$ pin during normal operation.

The timing of reset state indication on the STATUS[1:0] pins is asynchronous. The timing of indicating normal operation is synchronous with the peripheral clock (Pck), and is therefore asynchronous with the clocks input from the EXTAL pin and the CLKOUT pin.

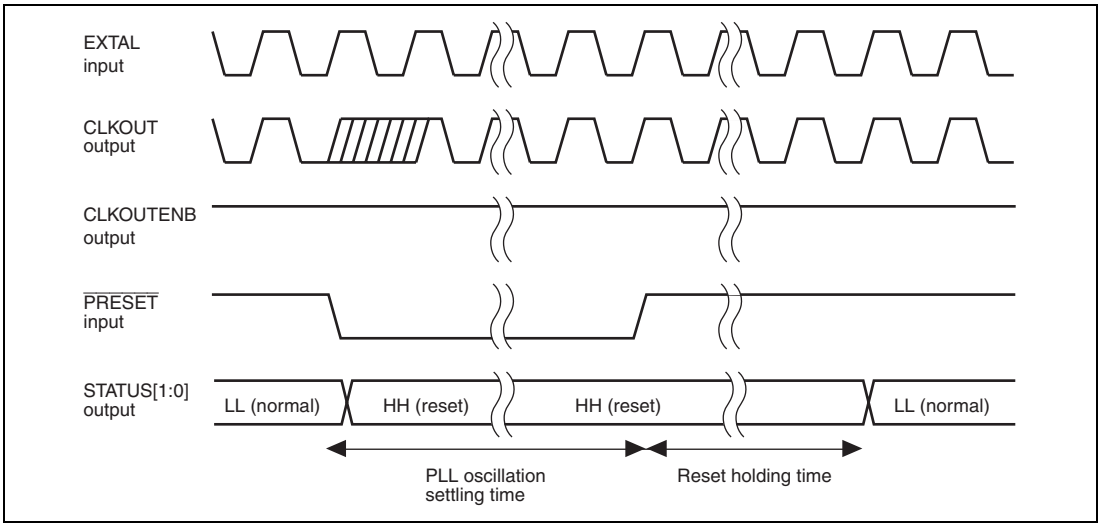


Figure 19.5 STATUS Output by Power-On Reset Caused by $\overline{\text{PRESET}}$ Input during Normal Operation

(3) Power-on reset caused by $\overline{\text{PRESET}}$ input in sleep mode

It is necessary to ensure the PLL oscillation settling time when a power-on reset is initiated by a low level input on the $\overline{\text{PRESET}}$ pin during sleep mode.

The timing of reset state indication on the STATUS[1:0] pins is asynchronous. The timing of indicating normal operation is synchronous with the peripheral clock (Pck), and is therefore asynchronous with the clocks input from the EXTAL pin and the CLKOUT pin.

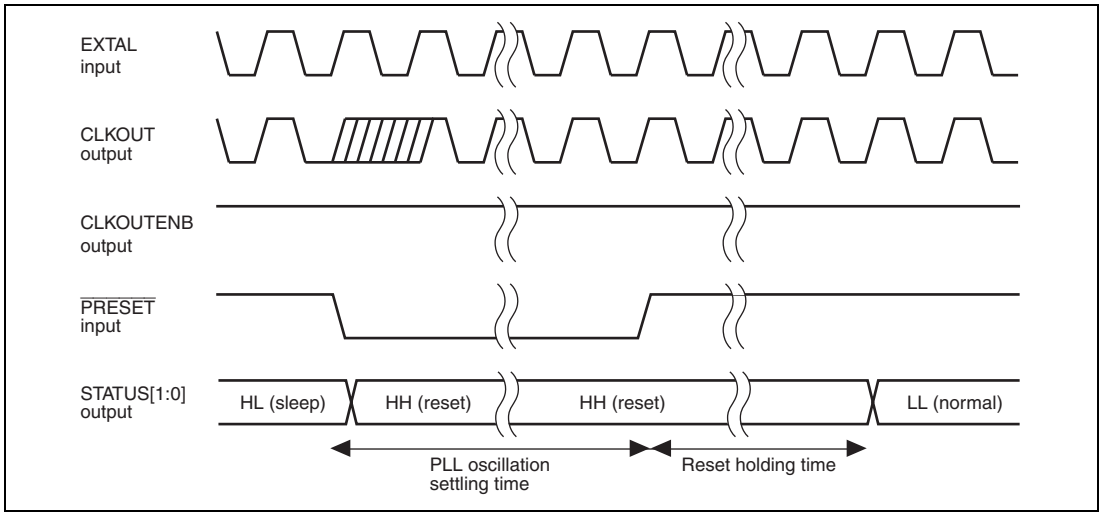


Figure 19.6 STATUS Output by Power-On Reset Caused by $\overline{\text{PRESET}}$ Input in Sleep Mode

19.5.2 Power-On Reset by Watchdog Timer Overflow

The time period taken by power-on reset on watchdog timer overflow (WDT reset holding time) is equal to or more than 40 cycles of the peripheral clock (Pck).

The transition time from watchdog timer overflow to the power-on reset state (WDT reset setup time) is equal to or more than 40 cycles of the peripheral clock (Pck).

If the bus clock frequency has been changed from the initial value, the oscillation settling time of PLL2 circuit and the time until the LSI resumes operation (WDT count up) are also required. In this case, the WDT reset holding time is two peripheral clock (Pck) cycles or more.

(1) Power-on reset caused by watchdog timer overflow during normal operation

The timing of indicating the reset state or normal operation on the STATUS[1:0] pins is synchronous with the peripheral clock (Pck), and is therefore asynchronous with the clocks input from the EXTAL pin and the CLKOUT pin.

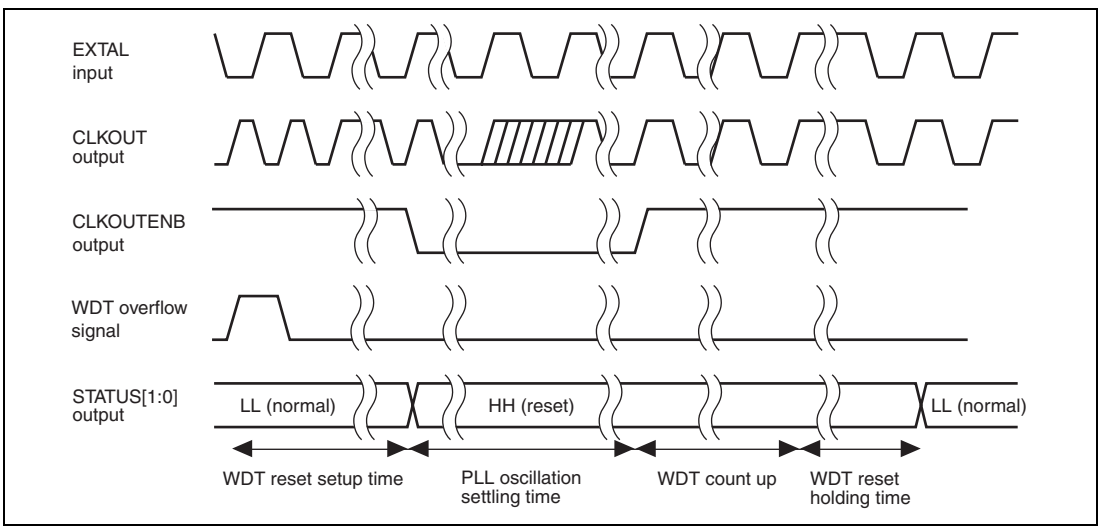


Figure 19.7 STATUS Output by Power-On Reset Caused by WDT Overflow during Normal Operation

(2) Power-on reset caused by watchdog timer overflow in sleep mode

The timing of indicating the reset state or normal operation on the STATUS[1:0] pins is synchronous with the peripheral clock (Pck), and is therefore asynchronous with the clocks input from the EXTAL pin and the CLKOUT pin.

If the bus clock frequency has been changed from the initial value, the oscillation settling time of PLL2 circuit and the time until the LSI resumes operation (WDT count up) are also required. In this case, the WDT reset holding time is two peripheral clock (Pck) cycles or more.

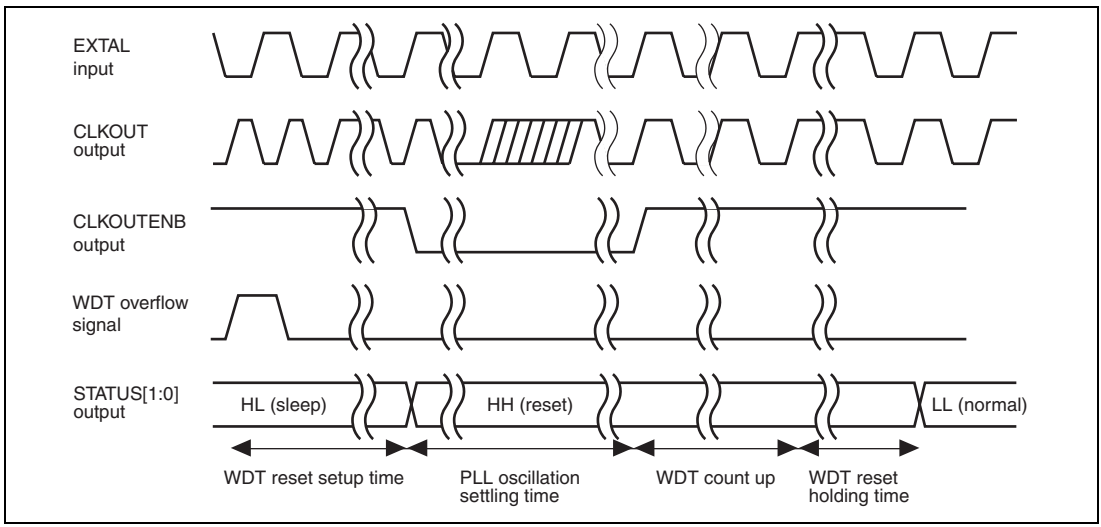


Figure 19.8 STATUS Output by Power-On Reset Caused by WDT Overflow in Sleep Mode

Section 20 Power-Down Mode

In power-down mode, some of the on-chip modules and the CPU are stopped. This enables to reduce power consumption.

20.1 Features

- Supports sleep mode and module standby mode
- Supports DDR3-SDRAM power supply backup mode that turns off the power supplies the 1.8-V power supply

20.1.1 Types of Power-Down Modes

Power-down modes have the following modes and functions.

- Sleep mode
- Light sleep mode
- Module standby function
- DDR3-SDRAM power supply backup

Table 20.1 shows the conditions of transition to each mode, the states of the CPU, on-chip modules, etc. in each mode, and the methods for releasing each mode.

Table 20.1 States of Power-Down Modes

Power-Down Mode	Conditions of Transition	State							
		CPG	CPU	On-Chip Memory	On-Chip Peripheral Module		Pin	DDR3-SDRAM	Releasing Methods
					DMAC	Others			
Sleep mode	SLEEP instruction executed when the LTSLP bit in CnSTBCR is cleared 0.	Operate	Stopped (contents of registers retained)	Retained	Operate	Operate	Operation retained	Auto-refresh or self-refresh*	1. Interrupt 2. Power-on reset 3. Manual reset
Light sleep mode	SLEEP instruction executed when the LTSLP bit in CnSTBCR is set to 1.	Operate	Stopped (contents of registers retained)	Operate	Operate	Operate	Operation retained	Self-refresh	1. Interrupt 2. Power-on reset 3. Manual reset
Module standby function	Clear the corresponding bits in MSTPCR0 and MSTPCR1 to 0	Operate	Operate	Retained	Stopped in six-channel units: channels 0 to 5 and channels 6 to 11.	Specified modules stopped	Operation retained	Auto-refresh or self-refresh	Clear the corresponding bits in MSTPCR0 and MSTPCR1 to 0
DDR3-SDRAM power supply backup*	See section 12.7	Stopped	Stopped	Undefined	Stopped	Stopped	High-impedance state except for the 1.8-V power supply interface	Self-refresh	Power-on reset

Note: * Power supplies (1.25 V, 3.3 V) except the 1.5 V power supply are stopped in DDR3-SDRAM power supply backup mode. Therefore, all circuitry other than the pad of DDR3 interface are stopped, including DDR3 interface control unit, and the contents of registers are not retained.

20.2 Input/Output Pins

Table 20.2 shows the pins related to power-down mode.

Table 20.2 Pin Configuration

Pin name	Function	I/O	Description		
STATUS1	Processing state 1	Output	Indicate the operating states of this LSI		
STATUS0	Processing state 2		STATUS1	STATUS0	Operating states
			H	H	Reset
			H	L	Sleep mode
			L	L	Normal operation
			The STATUS0 pin is multiplexed with the SSI2_CLK (SSI channel 2) pin.		
			The STATUS1 pin is multiplexed with the SSI3_CLK (SSI channel 3) pin.		

Note: L means low level, and H means high level.

20.3 Register Descriptions

Table 20.3 shows the list of registers. Table 20.4 shows the register states in each processing mode.

Table 20.3 Register configuration

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync clock
Standby control register 0	MSTPCR0	R/W	H'FFC4 0030	H'1FC4 0030	32	Pck
Standby control register 1	MSTPCR1	R/W	H'FFC4 0034	H'1FC4 0034	32	Pck
Standby display register	MSTPMR	R	H'FFC4 0044	H'1FC4 0044	32	Pck
CPU0 Ick frequency setting register	C0IFC	R/W	H'FE40 0000	H'1E40 0000	32	Sck2
CPU1 Ick frequency setting register	C1IFC	R/W	H'FE40 1000	H'1E40 1000	32	Sck2
CPU0 standby control register	C0STBCR	R/W	H'FE40 0004	H'1E40 0004	32	Sck2
CPU1 standby control register	C1STBCR	R/W	H'FE40 1004	H'1E40 1004	32	Sck2

Table 20.4 Register States of CPG in Each Processing Mode

		Power-on Reset	Manual Reset	Sleep/ Light Sleep
Register Name	**Abbreviation**	**By $\overline{\text{PRESET}}$ Pin/WDT/H-UDI**	**By WDT/Multiple Exceptions**	**By SLEEP Instruction**
Standby control register 0	MSTPCR0	H'0000 0000	Retained	Retained
Standby control register 1	MSTPCR1	H'0000 0000	Retained	Retained
Standby display register	MSTPMR	H'00x8 0000*[1]	Retained	Retained
CPU0 Ick frequency setting register	C0IFC	H'0000 0000	Retained	Retained
CPU1 Ick frequency setting register	C1IFC	H'0000 0000	Retained	Retained
CPU0 standby control register	C0STBCR	H'0000 0000	Retained	Retained
CPU1 standby control register	C1STBCR	H'0000 0001	Retained	Retained

Note: 1. The initial value after a power-on reset depends on the mode pin state (MPMD).
If a low level signal is input to MPMD pin, the initial value is H'0008 0000.
If a high level signal is input to the MPMD, the initial value is H'0088 0000.

20.3.1 Standby Control Register 0 (MSTPCR0)

MSTPCR0 is a 32-bit readable/writable register that can specify whether each peripheral module operates or is stopped. MSTPCR can be accessed only in longword.

This register is only initialized by a power-on reset by the $\overline{\text{PRESET}}$ pin or power-on reset by WDT overflow.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MSTP[29:24]						MSTP[23:20]				—	—	MSTP[17:16]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSTP[15:14]		—	—	MSTP[11:10]		MSTP[9:8]		—	—	MSTP[5:4]		—	MSTP2	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29 to 24	MSTP[29:24]	All 0	R/W	Module Stop Bit [29:24] Specify that the clock supply to the module of the corresponding bit is stopped [29]: SCIF channel 5, [28]: SCIF channel 4, [27]: SCIF channel 3, [26]: SCIF channel 2, [25]: SCIF channel 1, [24]: SCIF channel 0 0: The corresponding module operates 1: The clock to the corresponding module is stopped
23 to 20	MSTP[23:20]	All 0	R/W	Module Stop Bit [23:20] Specify that the clock supply to the module of the corresponding bit is stopped [23]: SSI channel 3, [22]: SSI channel 2, [21]: SSI channel 1, [20]: SSI channel 0 0: The corresponding module operates 1: The clock to the corresponding module is stopped
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	MSTP[17:16]	All 0	R/W	Module Stop Bit [17:16] Specify that the clock supply to the module of the corresponding bit is stopped [17]: HAC channel 1, [16] HAC channel 0 0: The corresponding module operates 1: The clock to the corresponding module is stopped
15, 14	MSTP[15:14]	All 0	R/W	Module Stop Bit [15:14] Specify that the clock supply to the module of the corresponding bit is stopped [15]: I2C channel 1, [14] I2C channel 0 0: The corresponding module operates 1: The clock to the corresponding module is stopped

Bit	Bit Name	Initial Value	R/W	Description
13, 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	MSTP[11:8]	All 0	R/W	Module Stop Bit [11:8] Specify that the clock supply to the module of the corresponding bit is stopped [11]: TMU channels 9 to 11 [10]: TMU channels 6 to 8 [9]: TMU channels 3 to 5 [8]: TMU channels 2 to 0 0: The corresponding module operates 1: The clock to the corresponding module is stopped
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	MSTP[5] MSTP[4]	All 0	R/W	Module Stop Bit [5:4] Specify that the clock supply to the module of the corresponding bit is stopped [5]: SDIF channel 1, [4]: SDIF channel 0 0: The corresponding module operates 1: The clock to the corresponding module is stopped
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	MSTP[2]	0	R/W	Module Stop Bit [2] Specify that the clock supply to the module of the corresponding bit is stopped [2]: HSPI 0: The corresponding module operates 1: The clock to the corresponding module is stopped
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

20.3.2 Standby Control Register 1 (MSTPCR1)

MSTPCR1 is a 32-bit readable/writable register that each module of H-UDI, UBC, DMAC, and GDTA operates or is stopped. MSTPCR1 can be accessed only in longword.

This register is only initialized by a power-on reset by the $\overline{\text{PRESET}}$ pin or power-on reset by WDT overflow.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MSTP112	—	MSTP[110:108]			—	—	MSTP[105:102]				—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R	R

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	MSTP112	0	R/W	Module Stop Bit 112 Specifies that the clock supply to the USB module is stopped 0: USB operates 1: USB stopped
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	MSTP [110:108]	All 0	R/W	Module Stop Bit [110:108] Specifies that the clock supply to the PCI Express channels of the corresponding bit is stopped [110]: PCI Express channel 2, [109]: PCI Express channel 1, [108]: PCI Express channel 0 0: PCI Express operates 1: PCI Express stopped
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, 4	MSTP [105:104]	All 0	R/W	Module Stop Bit [105:104] Specifies that the clock supply to the DMAC channels of the corresponding bit is stopped [5]: DMAC channels 11 to 6, [4] DMAC channels 5 to 0 0: DMAC operates 1: DMAC stopped
3	MSTP103	0	R/W	Module Stop Bit 103 Specifies that the clock supply to the DU module is stopped 0: DU operates 1: DU stopped
2	MSTP102	0	R/W	Module Stop Bit 102 Specifies that the clock supply to the Ether module is stopped 0: Ether operates 1: Ether stopped
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

20.3.3 Standby Display Register (MSTPMR)

MSTPMR is a 32-bit readable register that indicates whether the USB/PCI Express/display unit (DU)/DMAC/Ether Mac controller (Ether) modules are in the module standby state. MSTPMR can be accessed only in longword.

This register is initialized by a power-on reset by the $\overline{\text{PRESET}}$ pin, power-on reset by WDT overflow, or H-UDI reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	x	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MSTP MUSB	—	MSTPMPCI[2:0]			—	—	MSTP S105	MSTP S104	MSTP MDU	MSTP METH	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	x	x	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Notes: The initial value of the bit 23 depends on the detected level of the MPMD pin at a power-on reset by the $\overline{\text{PRESET}}$ pin. It becomes 1 when the level of the MPMD is H, and becomes 0 when the level of the MPMD is L.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0.
23	—	x	R	If a high level signal is input to MPMD pin, This bit is read as 1. If a low level signal is input to MPMD pin, This bit is read as 0.
22 to 20	—	All 0	R	Reserved These bits are always read as 0.
19	—	1	R	Reserved This bit is always read as 1.
18 to13	—	All 0	R	Reserved These bits are always read as 0.
12	MSTPMUSB	0	R	Module Stop Display Bit USB Indicates the state of clock supply to the USB module 0: USB operates 1: USB stopped
11	—	0	R	Reserved This bit is always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	MSTPMPCI [2:0]	All 0	R	Module Stop Display Bit PCI [2:0] Indicates the state of clock supply to the PCIexpress channel of the corresponding bit. [10]: PCIexpress channel 2 [9]: PCIexpres channel 1 [8]: PCIexpress channel 0 0: PCIexpress operates 1: PCIexpress stopped
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5, 4	MSTPS105 MSTPS104	All 0	R	Module Stop Display Bit 105, 104 Indicates the state of clock supply to the DMAC channels of the corresponding bit MSTPS105: DMAC channels 11 to 6 MSTPS104: DMAC channels 5 to 0 0: The DMAC channels operate 1: The DMAC channels stopped
3	MSTPMDU	x	R	Module Stop Display Bit DU Indicates the state of clock supply to the DU module. When a low level signal is input to the MODE12 or MODE11 pin, the clock supply to the DU is stopped 0: DU operates 1: DU stopped
2	MSTPMETH	x	R	Module Stop Display Bit Ether Indicates the state of clock supply to the Ether module. When a low level signal is input to the MODE12 or a high level signal is input to the MODE11 pin, the clock supply to the Ether is operated. Other than above, it is stopped. 0: Ether operates 1: Ether stopped
1, 0	—	All 0	R	Reserved These bits are always read as 0.

20.3.4 CPU0 Ick Frequency Setting Register (C0IFC)

C0IFC is a 32-bit register that sets the ratio of the supply clock to CPU core 0. C0IFC can be accessed only in longword.

This register is initialized by a power-on reset by the $\overline{\text{PRESET}}$ pin, power-on reset by WDT overflow, or H-UDI reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	IIFC0[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	IIFC0[2:0]	000	R/W	These bits set the frequency ratio of the CPU clock that is set by FRQCR1 and the clock that is supplied internally to the CPU. 000: x1 001: x1/2 010: x1/4 or x1/3* Other than above: Setting prohibited If the frequencies of the input Ick and SHck are equal, setting 010 is prohibited.

Note: * If the ratio of the input Ick and SHck is 1:3 or 1:6, the frequency ratio will be 1/3 or 1/6, respectively.

20.3.5 CPU1 Ick Frequency Setting Register (C1IFC)

C1IFC is a 32-bit register that sets the ratio of the supply clock to CPU core 1. C1IFC can be accessed only in longword.

This register is initialized by a power-on reset by the $\overline{\text{PRESET}}$ pin, power-on reset by WDT overflow, or H-UDI reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	IIFC1[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	IIFC1[2:0]	000	R/W	These bits set the frequency ratio of the CPU clock that is set by FRQCR1 and the clock that is supplied internally to the CPU. 000: x1 001: x1/2 010: x1/4 or x1/3* Other than above: Setting prohibited If the frequencies of the input Ick and SHck are equal, setting 010 is prohibited.

Note: * If the ratio of the input Ick and SHck is 1:3 or 1:6, the frequency ratio will be 1/3 or 1/6, respectively.

20.3.6 CPU0 Standby Control Register (C0STBCR)

C0STBCR is a 32-bit register that sets the module stop state of CPU core 0. C0STBCR can be accessed only in longword.

This register is initialized by a power-on reset by the $\overline{\text{PRESET}}$ pin, power-on reset by WDT overflow, or H-UDI reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LTSLP0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SLEEP0	RESET0	MSTP0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	LTSLP0	0	R/W	Decides whether the operation changes to the light sleep mode or sleep mode, when the SLEEP instruction is executed at CPU core 0. 0: Sleep mode 1: Light Sleep mode
30 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	SLEEP0	0	R/W	Indicates that CPU core 0 is in the sleep/light sleep state. 0: CPU core 0 is not in the sleep/light sleep state. 1: CPU core 0 is in the sleep/light sleep state.
1	RESET0	0	R/W	Reset Bit For details, see 20.5.3, CPU Core Module Stop Control. 0: Don't operate a power-on reset when MSTP is cleared 1: Operate a power-on reset when MSTP is cleared

Bit	Bit Name	Initial Value	R/W	Description
0	MSTP0	0	R/W	Module Stop Bit Indicates that CPU core 0 is in the module stop state. For details, see 20.5.3, CPU Core Module Stop Control. 0: CPU core 0 is operating. 1: CPU core 0 is in the module stop state.

20.3.7 CPU1 Standby Control Register (C1STBCR)

C1STBCR is a 32-bit register that sets the module stop state of CPU core 1. C1STBCR is can be accessed only in longword.

This register is initialized by a power-on reset by the $\overline{\text{PRESET}}$ pin, power-on reset by WDT overflow, or H-UDI reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LTSLP1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SLEEP1	RESET1	MSTP1
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	LTSLP1	0	R/W	Decides whether the operation changes to the light sleep mode or sleep mode, when the SLEEP instruction is executed at CPU core 1. 0: Sleep mode 1: Light Sleep mode
30 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	SLEEP1	0	R/W	Indicates that CPU core 1 is in the sleep/light sleep state. 0: CPU core 1 is not in the sleep/light sleep state. 1: CPU core 1 is in the sleep/light sleep state.

Bit	Bit Name	Initial Value	R/W	Description
1	RESET1	1	R/W	Reset Bit For details, see 20.5.3, CPU Core Module Stop Control. 0: Don't operate a power-on reset when MSTP is cleared 1: Operate a power-on reset when MSTP is cleared
0	MSTP1	1	R/W	Module Stop Bit Indicates that CPU core 1 is in the module stop state. For details, see 20.5.3, CPU Core Module Stop Control. 0: CPU core 1 is operating. 1: CPU core 1 is in the module stop state.

20.4 Sleep Mode

20.4.1 Transition to Sleep Mode

When the SLEEP instruction is executed, the state is changed from the program execution state to sleep mode. Although the CPU is stopped after the instruction is executed, the contents of the CPU register are retained.

On-chip modules other than the CPU continue to operate. The clock is output to the CLKOUT pin.

In sleep mode of CPU core 0, a high level signal is output to the STATUS1 pin, and a low level signal is output to the STATUS0 pin.

20.4.2 Releasing Sleep Mode

Sleep mode is released by interrupts (NMI, $\overline{\text{IRQ}}$/$\overline{\text{IRL}}$[7:0], and on-chip module) and reset.

In sleep mode, interrupts are accepted even if the BL bit in the SR register is 1. If needed, put SPC, SSR, etc to stack before executing the SLEEP instruction.

(1) Release by Interrupts

When the NMI, $\overline{\text{IRQ}}$/$\overline{\text{IRL}}$[7:0], and on-chip module interrupts are generated, sleep mode is released and exception handling of interrupts are performed. The code corresponding to the interrupt sources is set to the INTEVT register.

For details of the timing of the changes in the STATUS pin, see section 20.6.2, Releasing Sleep Mode.

(2) Release by a Reset

Sleep mode is released by a power-on reset by the $\overline{\text{PRESET}}$ pin, power-on reset by WDT overflow, H-UDI reset, and manual reset. For details of the timing of the changes in the STATUS pin, see section 19.5, Status Pin Change Timing during Reset.

20.5 Module Standby Functions

This LSI supports the module standby state, where the clock supplied to on-chip modules is stopped.

20.5.1 Transition to Module Standby Mode

By setting the MSTP bits in MSPTCR, the clock supply can be stopped to the corresponding module.

In each module that is in the module standby state, the state right before transition to module standby state is retained. After register setting, the state right before stop is retained. The state of the external pin is retained. When the module is released from the module standby state, the module resumes operation.

Note: See the description of the individual MSTP bits in the standby control registers.

The MSTP bits should be set to 1 while the module is in the idle state after completing its operation and there is no possible activation sources from external pins or other modules.

20.5.2 Releasing Module Standby Functions

The module standby functions are released by clearing the MSTP bits in MSTPCR0 to 0 or a power-on reset.

20.5.3 CPU Core Module Stop Control

CPU module stop control is performed by writing to MSTPn in CnSTBCR (n = 0 or 1), in the following cases.

(1) Initial settings after a power-on reset (excluding a power-on reset triggered by RESETn in CnSTBCR)

(2) Operation stop/restart processing in case of a CPU core malfunction

Do not use CPU module stop control in cases other than the above.

(1) Initial settings after a power-on reset (excluding a power-on reset triggered by RESETn in CnSTBCR)

Immediately after cancelation of a power-on reset, CPU core 0 operates alone and CPU core 1 is in the module stop state. Do not change the value of RESET1 in C1STBCR until the CPU core 1 module stop state is canceled (MSTP1 in C1STBCR is cleared to 0).

(2) Operation stop/restart processing in case of a CPU core malfunction

When one CPU core detects a malfunction in the operation of the other CPU core, the affected CPU core can be stopped and restarted by following the sequence below. Note that if a communication state exists between the stopped CPU core and the emulator, communication may be cut off. In addition, the restart procedure is no longer available if all the CPU cores are stopped, so do not stop all the CPU cores at the same time.

1. Set the IIFC field in the CnIFC register (n= 0, 1) corresponding to the CPU core to be stopped to B'000.

2. Write B'1 to the MSTPn bit in the CnSTBCR register corresponding to the CPU core to be stopped. The CPU core stops operating at this point.

3. Write B'1 to the RESETn bit in the CnSTBCR register corresponding to the stopped CPU. It is necessary to trigger a power-on reset in order to restart the CPU core.

4. Write B'0 to the MSTPn bit in the CnSTBCR register corresponding to the stopped CPU. The CPU core restarts at this point. Before proceeding with other processing, read the CnSTBCR register once.

20.6 Timing of the Changes on the STATUS Pins

20.6.1 Reset

For details, see section 19.5, Status Pin Change Timing during Reset.

20.6.2 Releasing Sleep Mode

(1) When Sleep Mode is Released by an Interrupt

Figure 20.1 shows the timing of the changes in the STATUS pin.

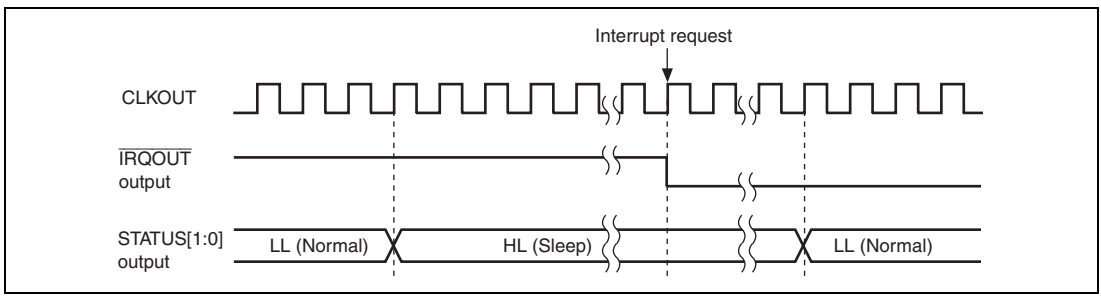


Figure 20.1 Status Pins Output when an Interrupt Occurs in Sleep Mode

20.7 DDR3-SDRAM Power Supply Backup

For details, see section 12.5.8, DDR3-SDRAM Power Supply Backup Function.

Section 21 Timer Unit (TMU)

This LSI includes an on-chip 32-bit timer unit (TMU), which has six channels (channels 0 to 11).

21.1 Features

The TMU has the following features.

- Auto-reload type 32-bit down-counter provided for each channel
- Input capture function provided only for channel 2
- Selection of rising edge or falling edge as external clock input edge when external clock is selected or input capture function is used: Channels 0 to 2
- 32-bit timer constant register for auto-reload use, readable/writable at any time, and 32-bit down-counter provided for each channel
- Selection of seven counter input clocks: Channels 0 to 2

  External clock (TCLK) and five internal clocks (Pck/4, Pck/16, Pck/64, Pck/256, and Pck/1024) obtained by dividing the peripheral clock (Pck is the peripheral clock).
- Selection of five counter input clocks: Channels 3 to 11

  Five peripheral clocks (Pck/4, Pck/16, Pck/64, Pck/256, and Pck/1024) (Pck is the peripheral clock).
- Two interrupt sources

  One underflow source (each channel) and one input capture source (channel 2).

Figure 21.1 shows a block diagram of the TMU.

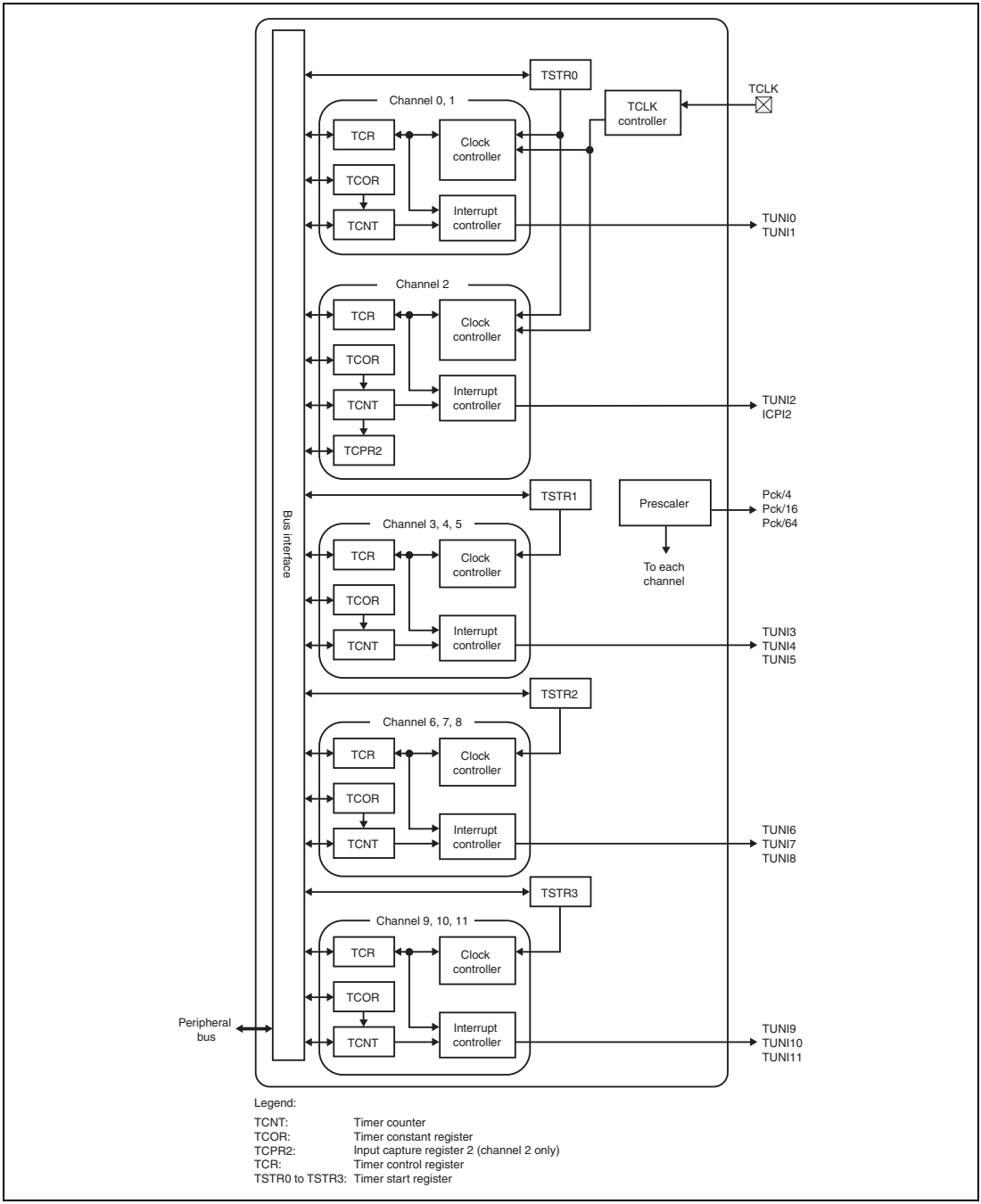


Figure 21.1 Block Diagram of TMU

21.2 Input/Output Pins

Table 21.1 shows the TMU pin configuration.

Table 21.1 Pin Configuration

Pin Name	Abbreviation	I/O	Description
Clock input	TCLK	I/O	External clock input pin for channels 0, 1 and 2 /input capture control input pin for channel 2

21.3 Register Descriptions

Tables 21.2 and 21.3 show the TMU register configuration.

Table 21.2 Register Configuration (1)

Channel	Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Common to 0, 1, 2	Timer start register 0	TSTR0	R/W	H'FFD8 0004	H'1FD8 0004	8	Pck
0	Timer constant register 0	TCOR0	R/W	H'FFD8 0008	H'1FD8 0008	32	Pck
	Timer counter 0	TCNT0	R/W	H'FFD8 000C	H'1FD8 000C	32	Pck
	Timer control register 0	TCR0	R/W	H'FFD8 0010	H'1FD8 0010	16	Pck
1	Timer constant register 1	TCOR1	R/W	H'FFD8 0014	H'1FD8 0014	32	Pck
	Timer counter 1	TCNT1	R/W	H'FFD8 0018	H'1FD8 0018	32	Pck
	Timer control register 1	TCR1	R/W	H'FFD8 001C	H'1FD8 001C	16	Pck
2	Timer constant register 2	TCOR2	R/W	H'FFD8 0020	H'1FD8 0020	32	Pck
	Timer counter 2	TCNT2	R/W	H'FFD8 0024	H'1FD8 0024	32	Pck
	Timer control register 2	TCR2	R/W	H'FFD8 0028	H'1FD8 0028	16	Pck
	Input capture register 2	TCPR2	R	H'FFD8 002C	H'1FD8 002C	32	Pck
Common to 3, 4, 5	Timer start register 1	TSTR1	R/W	H'FFDA 0004	H'1FDA 0004	8	Pck
3	Timer constant register 3	TCOR3	R/W	H'FFDA 0008	H'1FDA 0008	32	Pck
	Timer counter 3	TCNT3	R/W	H'FFDA 000C	H'1FDA 000C	32	Pck
	Timer control register 3	TCR3	R/W	H'FFDA 0010	H'1FDA 0010	16	Pck

Channel	Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Size	Sync Clock
4	Timer constant register 4	TCOR4	R/W	H'FFDA 0014	H'1FDA 0014	32	Pck
	Timer counter 4	TCNT4	R/W	H'FFDA 0018	H'1FDA 0018	32	Pck
	Timer control register 4	TCR4	R/W	H'FFDA 001C	H'1FDA 001C	16	Pck
5	Timer constant register 5	TCOR5	R/W	H'FFDA 0020	H'1FDA 0020	32	Pck
	Timer counter 5	TCNT5	R/W	H'FFDA 0024	H'1FDA 0024	32	Pck
	Timer control register 5	TCR5	R/W	H'FFDA 0028	H'1FDA 0028	16	Pck
Common to 6,7,8	Timer start register 2	TSTR2	R/W	H'FFDC0004	H'1FDC0004	8	Pck
6	Timer constant register 6	TCOR6	R/W	H'FFDC0008	H'1FDC0008	32	Pck
	Timer counter 6	TCNT6	R/W	H'FFDC000C	H'1FDC000C	32	Pck
	Timer control register 6	TCR6	R/W	H'FFDC0010	H'1FDC0010	16	Pck
7	Timer constant register 7	TCOR7	R/W	H'FFDC0014	H'1FDC0014	32	Pck
	Timer counter 7	TCNT7	R/W	H'FFDC0018	H'1FDC0018	32	Pck
	Timer control register 7	TCR7	R/W	H'FFDC001C	H'1FDC001C	16	Pck
8	Timer constant register 8	TCOR8	R/W	H'FFDC0020	H'1FDC0020	32	Pck
	Timer counter 8	TCNT8	R/W	H'FFDC0024	H'1FDC0024	32	Pck
	Timer control register 8	TCR8	R/W	H'FFDC0028	H'1FDC0028	16	Pck
Common to 9,10,11	Timer start register 3	TSTR3	R/W	H'FFDE0004	H'1FDE0004	8	Pck
9	Timer constant register 9	TCOR9	R/W	H'FFDE0008	H'1FDE0008	32	Pck
	Timer counter 9	TCNT9	R/W	H'FFDE000C	H'1FDE000C	32	Pck
	Timer control register 9	TCR9	R/W	H'FFDE0010	H'1FDE0010	16	Pck
10	Timer constant register 10	TCOR10	R/W	H'FFDE0014	H'1FDE0014	32	Pck
	Timer counter 10	TCNT10	R/W	H'FFDE0018	H'1FDE0018	32	Pck
	Timer control register 10	TCR10	R/W	H'FFDE001C	H'1FDE001C	16	Pck
11	Timer constant register 11	TCOR11	R/W	H'FFDE0020	H'1FDE0020	32	Pck
	Timer counter 11	TCNT11	R/W	H'FFDE0024	H'1FDE0024	32	Pck
	Timer control register 11	TCR11	R/W	H'FFDE0028	H'1FDE0028	16	Pck

Table 21.3 Register Configuration (2)

Channel	Register Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/ WDT/H-UDI	Manual Reset by WDT/ Multiple Exceptions	Sleep/ Light Sleep by SLEEP Instruction	Module Standby
Common to 0, 1, 2	Timer start register 0	TSTR0	H'00	Retained	Retained	Retained
0	Timer constant register 0	TCOR0	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 0	TCNT0	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 0	TCR0	H'0000	Retained	Retained	Retained
1	Timer constant register 1	TCOR1	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 1	TCNT1	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 1	TCR1	H'0000	Retained	Retained	Retained
2	Timer constant register 2	TCOR2	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 2	TCNT2	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 2	TCR2	H'0000	Retained	Retained	Retained
	Input capture register 2	TCPR2	Retained	Retained	Retained	Retained
Common to 3, 4, 5	Timer start register 1	TSTR1	H'00	Retained	Retained	Retained
3	Timer constant register 3	TCOR3	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 3	TCNT3	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 3	TCR3	H'0000	Retained	Retained	Retained
4	Timer constant register 4	TCOR4	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 4	TCNT4	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 4	TCR4	H'0000	Retained	Retained	Retained
5	Timer constant register 5	TCOR5	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 5	TCNT5	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 5	TCR5	H'0000	Retained	Retained	Retained
Common to 6, 7, 8	Timer start register 2	TSTR2	H'00	Retained	Retained	Retained

Channel	Register Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/ WDT/H-UDI	Manual Reset by WDT/ Multiple Exceptions	Sleep/ Light Sleep by SLEEP Instruction	Module Standby
6	Timer constant register 6	TCOR6	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 6	TCNT6	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 6	TCR6	H'0000	Retained	Retained	Retained
7	Timer constant register 7	TCOR7	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 7	TCNT7	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 7	TCR7	H'0000	Retained	Retained	Retained
8	Timer constant register 8	TCOR8	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 8	TCNT8	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 8	TCR8	H'0000	Retained	Retained	Retained
Common to 9,10,11	Timer start register 3	TSTR3	H'00	Retained	Retained	Retained
9	Timer constant register 9	TCOR9	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 9	TCNT9	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 9	TCR9	H'0000	Retained	Retained	Retained
10	Timer constant register 10	TCOR10	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 10	TCNT10	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 10	TCR10	H'0000	Retained	Retained	Retained
11	Timer constant register 11	TCOR11	H'FFFF FFFF	Retained	Retained	Retained
	Timer counter 11	TCNT11	H'FFFF FFFF	Retained	Retained	Retained
	Timer control register 11	TCR11	H'0000	Retained	Retained	Retained

21.3.1 Timer Start Registers (TSTRn) (n = 0 to 3)

The TSTR registers are 8-bit readable/writable registers that specifies whether TCNT of the corresponding channel is operated or stopped.

- TSTR0

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR2	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	STR2	0	R/W	Counter Start 2 Specifies whether TCNT2 is operated or stopped. 0: TCNT2 count operation is stopped 1: TCNT2 performs count operation
1	STR1	0	R/W	Counter Start 1 Specifies whether TCNT1 is operated or stopped. 0: TCNT1 count operation is stopped 1: TCNT1 performs count operation
0	STR0	0	R/W	Counter Start 0 Specifies whether TCNT0 is operated or stopped. 0: TCNT0 count operation is stopped 1: TCNT0 performs count operation

- TSTR1

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR5	STR4	STR3
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	STR5	0	R/W	Counter Start 5 Specifies whether TCNT5 is operated or stopped. 0: TCNT5 count operation is stopped 1: TCNT5 performs count operation
1	STR4	0	R/W	Counter Start 4 Specifies whether TCNT4 is operated or stopped. 0: TCNT4 count operation is stopped 1: TCNT4 performs count operation
0	STR3	0	R/W	Counter Start 3 Specifies whether TCNT3 is operated or stopped. 0: TCNT3 count operation is stopped 1: TCNT3 performs count operation

- TSTR2

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR8	STR7	STR6
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	STR8	0	R/W	Counter Start 8 Specifies whether TCNT8 is operated or stopped. 0: TCNT8 count operation is stopped 1: TCNT8 performs count operation
1	STR7	0	R/W	Counter Start 7 Specifies whether TCNT7 is operated or stopped. 0: TCNT7 count operation is stopped 1: TCNT7 performs count operation
0	STR6	0	R/W	Counter Start 6 Specifies whether TCNT6 is operated or stopped. 0: TCNT6 count operation is stopped 1: TCNT6 performs count operation

- TSTR3

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR11	STR10	STR9
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	STR11	0	R/W	Counter Start 11 Specifies whether TCNT11 is operated or stopped. 0: TCNT11 count operation is stopped 1: TCNT11 performs count operation
1	STR10	0	R/W	Counter Start 10 Specifies whether TCNT10 is operated or stopped. 0: TCNT10 count operation is stopped 1: TCNT10 performs count operation
0	STR9	0	R/W	Counter Start 9 Specifies whether TCNT9 is operated or stopped. 0: TCNT9 count operation is stopped 1: TCNT9 performs count operation

21.3.2 Timer Constant Registers (TCORn) (n = 0 to 11)

The TCOR registers are 32-bit readable/writable registers. When a TCNT counter underflows while counting down, the TCOR value is set in that TCNT, which continues counting down from the set value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

21.3.3 Timer Counters (TCNTn) (n = 0 to 11)

The TCNT registers are 32-bit readable/writable registers. Each TCNT counts down on the input clock selected by the TPSC2 to TPSC0 bits in TCR.

When a TCNT counter underflows while counting down, the UNF flag is set in TCR of the corresponding channel. At the same time, the TCOR value is set in TCNT, and the count-down operation continues from the set value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

21.3.4 Timer Control Registers (TCRn) (n = 0 to 11)

The TCR registers are 16-bit readable/writable registers. Each TCR selects the count clock, specifies the edge when an external clock is selected, and controls interrupt generation when the flag indicating TCNT underflow is set to 1. TCR2 is also used for input capture control and control of interrupt generation in the event of input capture.

- TCR0, TCR1, TCR3 to TCR11

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UNF	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- TCR2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ICPF	UNF	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	ICPF*[1]	0	R/W	Input Capture Interrupt Flag Status flag, provided in channel 2 only, which indicates the occurrence of input capture. 0: Input capture has not occurred [Clearing condition] When 0 is written to ICPF 1: Input capture has occurred [Setting condition] When input capture occurs*[2]
8	UNF	0	R/W	Underflow Flag Status flag that indicates the occurrence of TCNT underflow. 0: TCNT has not underflowed [Clearing condition] When 0 is written to UNF 1: TCNT has underflowed [Setting condition] When TCNT underflows*[2]
7	ICPE1*[1]	0	R/W	Input Capture Control
6	ICPE0*[1]	0	R/W	These bits, provided in channel 2 only, specify whether the input capture function is used, and control enabling or disabling of interrupt generation when the function is used. The CKEG bits specify whether the rising edge or falling edge of the TCLK pin is used to set the TCNT2 value in TCPR2. The TCNT2 value is set in TCPR2 only when the ICPF bit in TCR2 is 0. When the ICPF bit is 1, TCPR2 is not set in the event of input capture. 00: Input capture function is not used. 01: Reserved (setting prohibited) 10: Input capture function is used, but interrupt due to input capture (TICPI2) is not enabled. 11: Input capture function is used, and interrupt due to input capture (TICPI2) is enabled.

Bit	Bit Name	Initial Value	R/W	Description
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling or disabling of interrupt generation when the UNF status flag is set to 1, indicating TCNT underflow. 0: Interrupt due to underflow (TUNI) is disabled 1: Interrupt due to underflow (TUNI) is enabled
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the external clock input edge when an external clock is selected or the input capture function is used. 00: Count/input capture register set on rising edge 01: Count/input capture register set on falling edge 1X: Count/input capture register set on both rising and falling edges
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT count clock.
0	TPSC0	0	R/W	000: Counts on Pck/4 001: Counts on Pck/16 010: Counts on Pck/64 011: Counts on Pck/256 100: Counts on Pck/1024 101, 110: Setting prohibited 111: Counts on external clock (TCLK) *[3]

Notes: X: Don't care

1. Reserved bit in channels 0, 1, 3 to 11 (initial value is 0, and can only be read).
2. Writing 1 does not change the value; the previous value is retained.
3. Do not set in channels 3 to 11.

21.3.5 Input Capture Register 2 (TCPR2)

TCPR2 is a 32-bit read-only register for use with the input capture function, provided only in channel 2. The input capture function is controlled by means of the ICPE and CKEG bits in TCR2. When input capture occurs, the TCNT2 value is copied into TCPR2. The value is set in TCPR2 only when the ICPF bit in TCR2 is 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

21.4 Operation

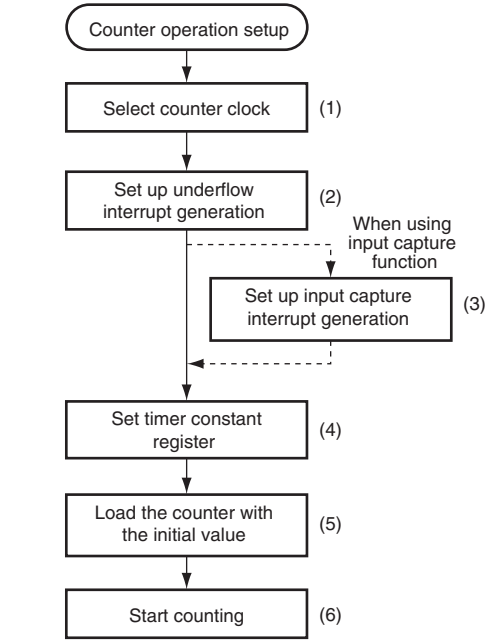
Each channel has a 32-bit timer counter (TCNT) and a 32-bit timer constant register (TCOR). Each TCNT performs count-down operation. The channels have an auto-reload function that allows cyclic count operations, and can also perform external event counting. Channel 2 also has an input capture function.

21.4.1 Counter Operation

When one of bits STR11 to STR0 in TSTR0 to TSTR3 is set to 1, the TCNT for the corresponding channel starts counting. When TCNT underflows, the UNF flag in TCR is set. If the UNIE bit in TCR is set to 1 at this time, an interrupt request is sent to the CPU. At the same time, the value is copied from TCOR into TCNT, and the count-down continues (auto-reload function).

(1) Example of Counter Operation Setting Procedure

Figure 21.2 shows an example of the counter operation setting procedure.



(1) Select the counter clock with the TPSC2 to TPSC0 bits in TCR. When the external clock (TCLK) is selected, specify the external clock edge with the CKEG1 and CKEG0 bits in TCR.

(2) Specify whether an interrupt is generated on TCNT underflow with the UNIE bit in TCR.

(3) When the input capture function is used, set the ICPE bits in TCR, which also specify the use of the interrupt function.

(4) Set a value in TCOR.

(5) Set the initial value in TCNT.

(6) Set the STR bit in TSTR to 1 to start counting.

Note: When an interrupt is generated, clear the source flag in the interrupt handler processing. If the interrupt is enabled without clearing the flag, another interrupt will be generated.

Figure 21.2 Example of Count Operation Setting Procedure

(2) Auto-Reload Count Operation

Figure 21.3 shows the TCNT auto-reload operation.

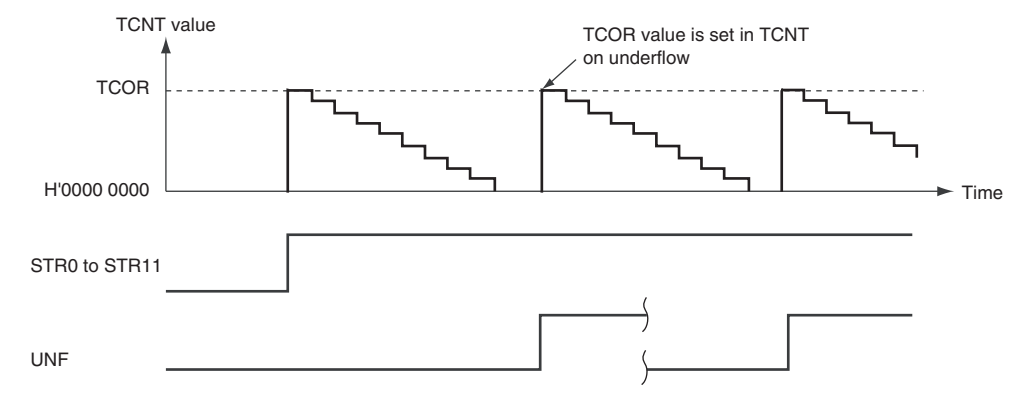


Figure 21.3 TCNT Auto-Reload Operation

(3) TCNT Count Timing

- Operating on internal clock

  Any of five internal count clocks (Pck/4, Pck/16, Pck/64, Pck/256, or Pck/1024) scaled from the peripheral clock can be selected as the count clock by means of the TPSC2 to TPSC0 bits in TCR.

  Figure 21.4 shows the timing in this case.

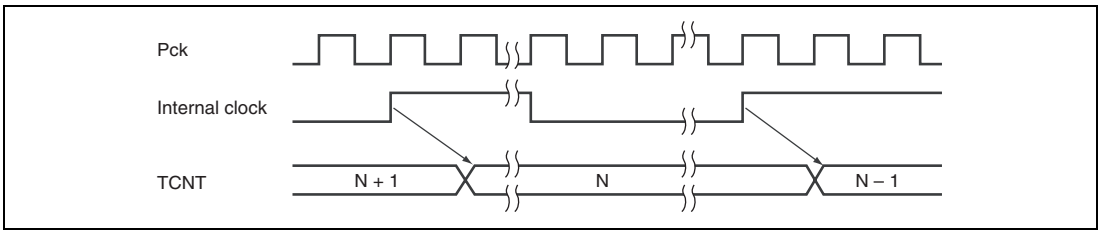


Figure 21.4 Count Timing when Operating on Internal Clock

- Operating on external clock

  In channels 0, 1, and 2, the external clock input pin (TCLK) input can be selected as the timer clock by means of the TPSC2 to TPSC0 bits in TCR. The detected edge (rising, falling, or both edges) can be selected with the CKEG1 and CKEG0 bits in TCR.

  Figure 21.5 shows the timing for both-edge detection.

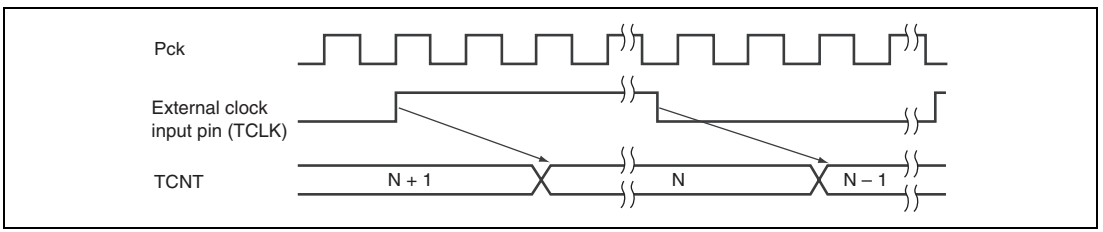


Figure 21.5 Count Timing when Operating on External Clock Input

21.4.2 Input Capture Function

Channel 2 has an input capture function.

The procedure for using the input capture function is as follows:

1. Use bits TPSC2 to TPSC0 in TCR2 to set an internal clock as the timer operating clock.
2. Use bits IPCE1 and IPCE0 in TCR2 to specify use of the input capture function, and whether interrupts are to be generated when this function is used.

3. Use bits CKEG1 and CKEG0 in TCR2 to specify whether the rising or falling edge of the TCLK pin is to be used to set the TCNT value in TCPR2.

When input capture occurs, the TCNT2 value is set in TCPR2 only when the ICPF bit in TCR2 is 0. A new DMAC transfer request is not generated until processing of the previous request is finished.

Figure 21.6 shows the operation timing when the input capture function is used (with TCLK rising edge detection).

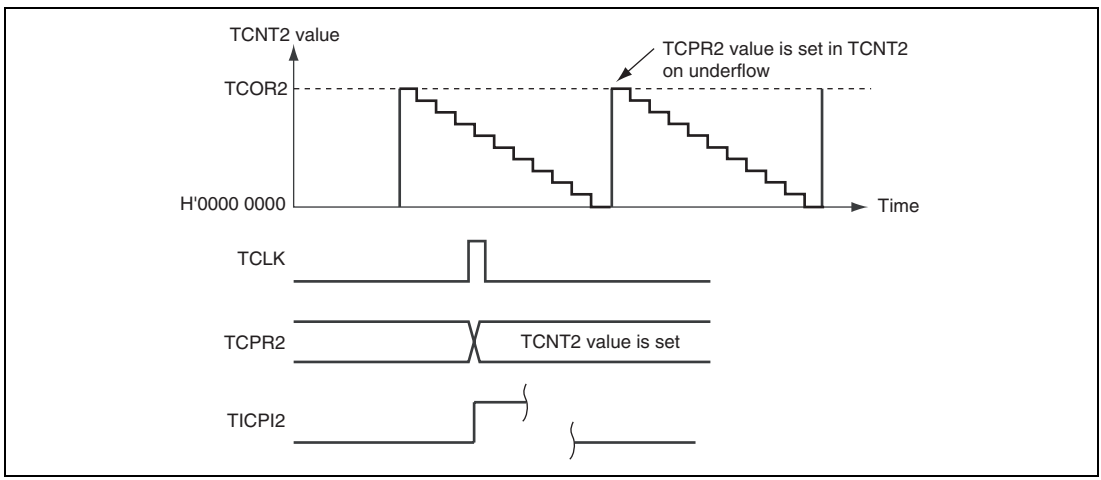


Figure 21.6 Operation Timing when Using Input Capture Function

21.5 Interrupts

There are seven TMU interrupt sources: underflow interrupts and the input capture interrupt when the input capture function is used. Underflow interrupts are generated on each of the channels, and input capture interrupts on channel 2 only.

An underflow interrupt request is generated (for each channel) when both the UNF bit and the interrupt enable bit (UNIE) for that channel are set to 1.

When the input capture function is used and an input capture request is generated, an interrupt is requested if the ICPF bit in TCR2 is 1 and the input capture control bits (ICPE1 and ICPE0) in TCR2 are both set to 11.

The TMU interrupt sources are summarized in table 21.4.

Table 21.4 TMU Interrupt Sources

Channel	Interrupt Source	Description
0	TUNI0	Underflow interrupt 0
1	TUNI1	Underflow interrupt 1
2	TUNI2	Underflow interrupt 2
	TICPI2	Input capture interrupt 2
3	TUNI3	Underflow interrupt 3
4	TUNI4	Underflow interrupt 4
5	TUNI5	Underflow interrupt 5
6	TUNI6	Underflow interrupt 6
7	TUNI7	Underflow interrupt 7
8	TUNI8	Underflow interrupt 8
9	TUNI9	Underflow interrupt 9
10	TUNI10	Underflow interrupt 10
11	TUNI11	Underflow interrupt 11

21.6 Power-Down Mode

When using following modes or functions to enter power-down mode, it is necessary to stop operations of counting before enter power-down mode.

(1) Sleep mode

The timer unit (TMU) continues operation.

(2) Light Sleep mode

The timer unit (TMU) continues operation.

(3) Modules stand-by

Supplying clock to the timer unit (TMU) stops. When the module state transition to the stand-by occurs, any operations of counting or requests of interrupt should be prohibited. Or reconfigure the operations of count after releasing the stand-by state.

(4) Changing clock frequencies

When changing the clock frequency on running the counter, correct operation cannot be guaranteed. Before changing the clock frequency, any operations of counting or requests of interrupt should be prohibited. Or set the counter again after the sequence of changing the clock frequency.

21.7 Usage Notes

21.7.1 Register Writes

When writing to a TMU register, timer count operation must be stopped by clearing the start bit (STR11 to STR0) for the relevant channel in TSTR.

Note that TSTR can be written to, and the UNF and ICPF bits in TCR can be cleared while the count is in progress. When the flags (UNF and ICPF) are cleared while the count is in progress, make sure not to change the values of bits other than those being cleared.

21.7.2 Reading from TCNT

Reading from TCNT is performed synchronously with the timer count operation. Note that when the timer count operation is performed simultaneously with reading from a register, the synchronous processing causes the TCNT value before the count-down operation to be read as the TCNT value.

21.7.3 External Clock Frequency

Ensure that the external clock (TCLK) input frequency for channels 0, 1 and 2 does not exceed Pck/4.

Section 22 Ethernet MAC Controller (Ether)

22.1 Overview of Ether

The Ether is a control unit which incorporates a function core conforming to the IEEE802.3u MAC (media access control) layer standard.

22.1.1 Features of Ether

- IEEE802.3u MAC (Ether) function
- Transmits and receives data frames
- Supports transfer at 10 and 100 Mbps
- Supports MII (media independent interface) interface conforming to IEEE802.3u
- Magic packet detection
- Flow control conforming to IEEE802.3x or back pressure system
- On-chip DMA controller

22.1.2 Logic Structure

Figure 22.1 shows a block diagram of the Ether. Table 22.1 lists the functions of each block.

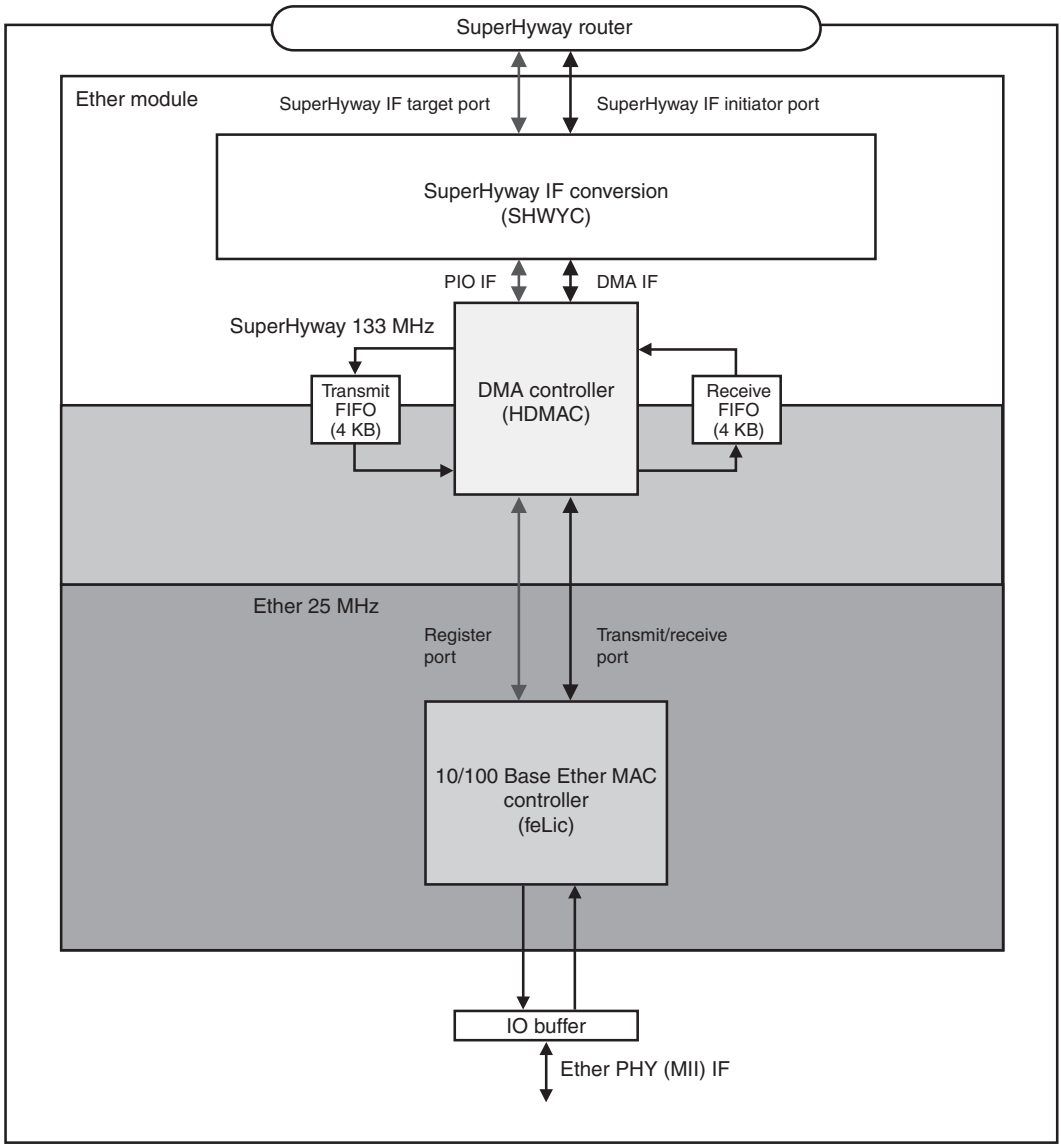


Figure 22.1 Block Diagram of Ether

Table 22.1 Functions of Each Block

Block	Function
DMA controller (HDMAC)	• DMA transfer of data transmitted from and received in the 10/100 Base Ethernet MAC controller between the transmit/receive buffer on memory and internal FIFOs For details, see section 22.8, HDMAC Function Specifications.
10/100 Base Ether MAC controller (feLic)	• Conforms to the IEEE802.3u MAC layer standard. • Transmits and receives data frames. • Supports transfer at 10/100 Mbps. • Supports MII interface conforming to IEEE802.3u. • Magic packet detection • Flow control conforming to IEEE802.3x or back pressure system For details, see section 22.9, feLic Function Specifications.
Transmit/receive FIFO (TFIFO/RFIFO)	• Transmit/receive FIFO for DMA controller Transmit FIFO depth: 4 kbytes Receive FIFO depth: 4 kbytes

22.2 Pin Definition

22.2.1 Pin Function

Table 22.2 lists the external pin functions.

Table 22.2 External Pin Function

Type	Pin Name	Polarity	I/O	Function	Number of Pins
Ether MII, etc	ETH_TX_CLK	P	Input	Transmit clock	1
	ETH_TXD[3:0]	P	Output	Transmit data	4
	ETH_TX_EN	P	Output	Transmit data enable	1
	ETH_TX_ER	P	Output	Transmit error	1
	ETH_CRS	P	Input	Carrier sense	1
	ETH_RX_COL	P	Input	Collision detection	1
	ETH_RX_CLK	P	Input	Receive clock	1
	ETH_RX_ER	P	Input	Receive error	1
	ETH_RX_DV	P	Input	Receive data valid	1
	ETH_RXD[3:0]	P	Input	Receive data	4
	ETH_MDC	P	Output	Management data clock	1
	ETH_mdio	P	I/O	Management data	1
	ETH_LINK	P	Input	PHY output LINK signal	1
	ETH_MAGIC	P	Output	Magic packet detection	1

22.3 Endian

The Ether has a 32-bit interface. Big or little endian is selected for the SuperHyway 64-bit interface through the endian mode of the LSI and the DE bit in the CXR0 register of the HDMAC. Figure 22.2 shows data arrangement.

Big endian mode: DE bit in CXR0 = 0

Little endian mode: DE bit in CXR0 = 1

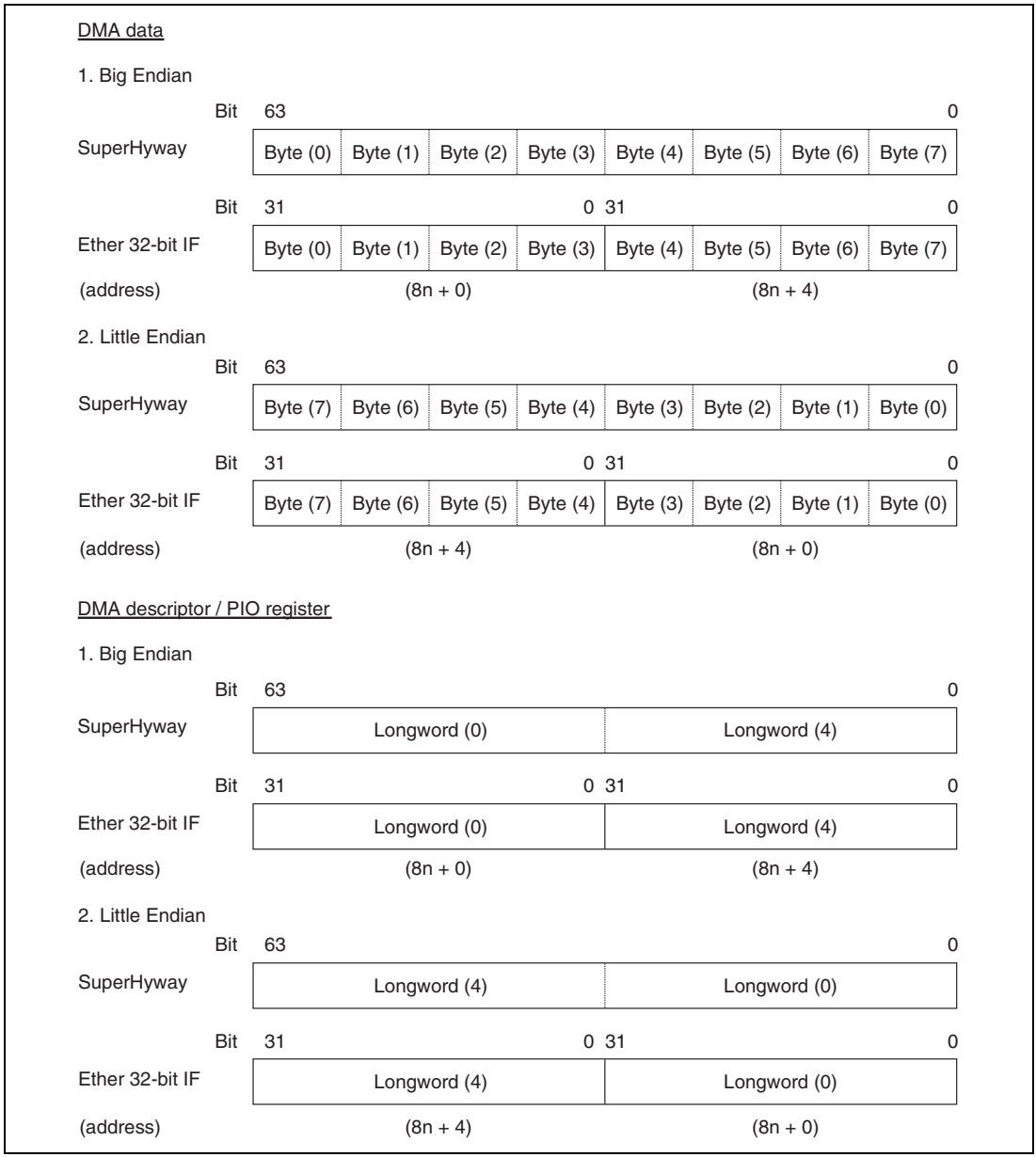


Figure 22.2 Data Arrangement

The Ether allows endian conversion for DMA data through the setting of the CXR0 register of the HDMAC in little endian mode. Figure 22.3 shows endian conversion for DMA data.

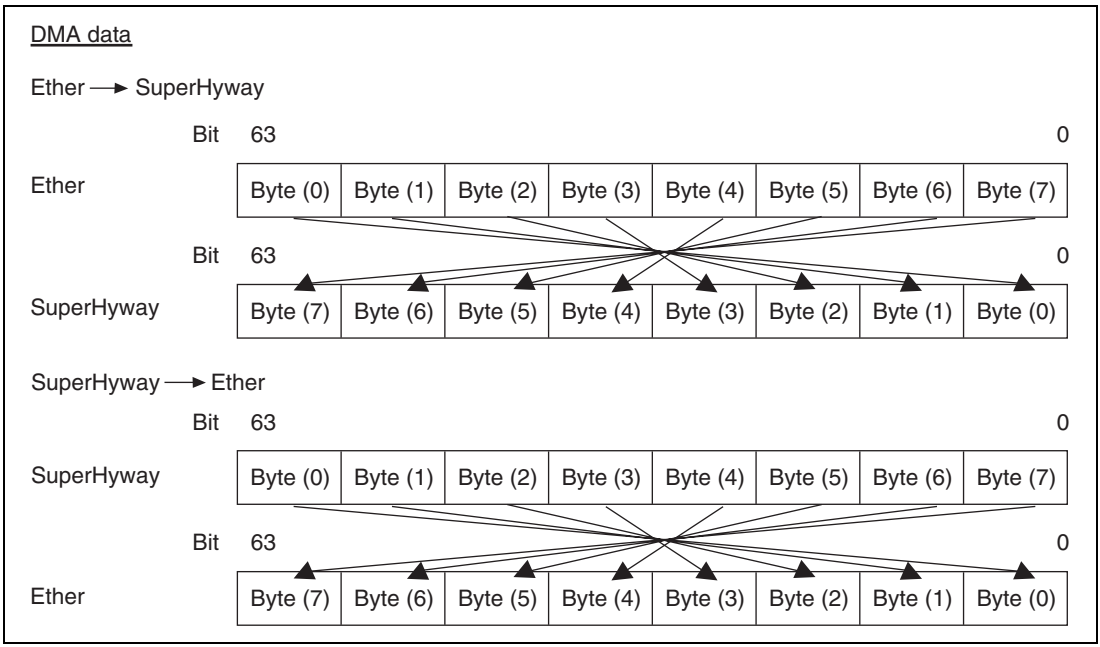


Figure 22.3 Endian Conversion for DMA Data

22.4 Register Descriptions

22.4.1 Notes on Register Access

To access the registers defined in this specifications document by upper software, the following restrictions and notes should be obeyed.

1. For the register bits not defined implicitly in this specifications document, the write value should always be 0, and the read value should always be handled as undefined.
2. Even when there are no restrictions on writing to each register, the undefined bits should be written to as described in the above restriction.
3. Each register is a 32-bit long and should be accessed in 32-bit units. Accordingly, any register cannot be partially read from or written to.

22.4.2 Register Configuration

Table 22.3 shows the configuration of HDMAC registers. Table 22.4 shows the configuration of feLic registers. Table 22.5 shows the state of HDMAC registers in each processing mode. Table 22.6 shows the state of feLic registers in each processing mode.

Table 22.3 HDMAC Register Configuration

Name	Abbreviation	R/W	P4 Area Address*[1]	Area 7 Address*[1]	Access Size	Synchronous Clock
HDMAC operating mode setting register	CXR0	R/W*[2]	H'FEE0 0200	H'1EE0 0200	32	clks3
Transmit activation register	CXR1	R/W*[2]	H'FEE0 0208	H'1EE0 0208	32	clks3
Receive activation register	CXR2	R/W	H'FEE0 0210	H'1EE0 0210	32	clks3
Transmit descriptor start address setting register	CXR3	R/W*[2]	H'FEE0 0218	H'1EE0 0218	32	clks3
Receive descriptor start address setting register	CXR4	R/W*[2]	H'FEE0 0220	H'1EE0 0220	32	clks3
Status register	CXR5	R/W	H'FEE0 0228	H'1EE0 0228	32	clks3
Interrupt mask setting register	CXR6	R/W	H'FEE0 0230	H'1EE0 0230	32	clks3
Error mask setting register	CXR7	R/W	H'FEE00238	H'1EE0 0238	32	clks3
Discarded frame counter register	CXR8	R/W	H'FEE00240	H'1EE0 0240	32	clks3
Transmit FIFO threshold setting register	CXR9	R/W*[2]	H'FEE0 0248	H'1EE0 0248	32	clks3
External FIFO depth setting register	CXR10	R/W*[2]	H'FEE0 0250	H'1EE0 0250	32	clks3
Receive activation reset method setting register	CXR11	R/W*[2]	H'FEE0 0258	H'1EE0 0258	32	clks3
Reserved	CXR12	—	H'FEE0 0260	H'1EE0 0260	32	clks3
Transmit FIFO underrun counter register	CXR13	R/W	H'FEE0 0264	H'1EE0 0264	32	clks3
Receive FIFO overflow counter register	CXR14	R/W	H'FEE0 0268	H'1EE0 0268	32	clks3
Reserved	CXR15	—	H'FEE0 026C	H'1EE0 026C	32	clks3

Name	Abbreviation	R/W	P4 Area Address*[1]	Area 7 Address*[1]	Access Size	Synchronous Clock
Receive FIFO busy transmit threshold setting register	CXR16	R/W	H'FEE0 0270	H'1EE0 0270	32	clks3
Reserved	CXR17	—	H'FEE0 0274	H'1EE0 0274	32	clks3
Reserved	CXRS2	—	H'FEE0 0278	H'1EE0 0278	32	clks3
Transmit interrupt mode setting register	CXR18	R/W	H'FEE0 027C	H'1EE0 027C	32	clks3
Reserved	CXR19	—	H'FEE0 0280	H'1EE0 0280	32	clks3
Reserved	CXRD0	—	H'FEE0 02C8	H'1EE0 02C8	32	clks3
Reserved	CXRD1	—	H'FEE0 02CC	H'1EE0 02CC	32	clks3
Reserved	CXRD2	—	H'FEE0 02D0	H'1EE0 02D0	32	clks3
Reserved	CXRD3	—	H'FEE0 02D4	H'1EE0 02D4	32	clks3
Reserved	CXRD4	—	H'FEE0 02D8	H'1EE0 02D8	32	clks3
Reserved	CXRD5	—	H'FEE0 02DC	H'1EE0 02DC	32	clks3
Reserved	CXRD6	—	H'FEE0 02E0	H'1EE0 02E0	32	clks3

Notes: 1. P4 area address indicates the register address when P4 area in the virtual address space is used. Area 7 address indicates the register address when accessed using TLB via area 7 in the physical address space.

2. There are restrictions on write access.

Table 22.4 feLic Register Configuration

Name	Abbreviation	R/W	P4 Area Address*[1]	Area 7 Address*[1]	Access Size	Synchronous Clock
feLic operating mode setting register	CXR20	R/W*[2]	H'FEE0 0300	H'1EE0 0300	32	clks3
Long frame length check value setting register	CXR2A	R/W*[2]	H'FEE0 0308	H'1EE0 0308	32	clks3
Status register	CXR21	R/W	H'FEE0 0310	H'1EE0 0310	32	clks3
Interrupt mask setting register	CXR22	R/W	H'FEE0 0318	H'1EE0 0318	32	clks3
MII control register	CXR23	R/W	H'FEE0 0320	H'1EE0 0320	32	clks3
PHY status register	CXR2B	R	H'FEE0 0328	H'1EE0 0328	32	clks3
Random number generating counter upper limit setting register	CXR30	R/W	H'FEE0 0340	H'1EE0 0340	32	clks3
IPG counter setting register	CXR70	R/W*[2]	H'FEE0 0350	H'1EE0 0350	32	clks3
Auto PAUSE parameter setting register	CXR71	R/W	H'FEE0 0354	H'1EE0 0354	32	clks3
Manual PAUSE parameter setting register	CXR72	W	H'FEE0 0358	H'1EE0 0358	32	clks3
Receive PAUSE frame counter register	CXR80	R	H'FEE0 0360	H'1EE0 0360	32	clks3
PAUSE frame retransmit count setting register	CXR81	R/W*[2]	H'FEE0 0364	H'1EE0 0364	32	clks3
PAUSE frame retransmit counter register	CXR82	R	H'FEE0 0368	H'1EE0 0368	32	clks3
Reserved	CXR83	—	H'FEE0 036C	H'1EE0 036C	32	clks3
MAC address high register	CXR24	R/W*[2]	H'FEE0 03C0	H'1EE0 03C0	32	clks3
MAC address low register	CXR25	R/W*[2]	H'FEE0 03C8	H'1EE0 03C8	32	clks3
TINT1 count register	CXR40	R/W	H'FEE0 03D0	H'1EE0 03D0	32	clks3
TINT2 count register	CXR41	R/W	H'FEE0 03D4	H'1EE0 03D4	32	clks3
TINT3 count register	CXR42	R/W	H'FEE0 03D8	H'1EE0 03D8	32	clks3
TINT4 count register	CXR43	R/W	H'FEE0 03DC	H'1EE0 03DC	32	clks3
RINT1 count register	CXR50	R/W	H'FEE0 03E4	H'1EE0 03E4	32	clks3
RINT2 count register	CXR51	R/W	H'FEE0 03E8	H'1EE0 03E8	32	clks3
RINT3 count register	CXR52	R/W	H'FEE0 03EC	H'1EE0 03EC	32	clks3

Name	Abbreviation	R/W	P4 Area Address*[1]	Area 7 Address*[1]	Access Size	Synchronous Clock
RINT4 count register	CXR53	R/W	H'FEE0 03F0	H'1EE0 03F0	32	clks3
RINT5 count register	CXR54	R/W	H'FEE0 03F4	H'1EE0 03F4	32	clks3
RINT8 count register	CXR55	R/W	H'FEE0 03F8	H'1EE0 03F8	32	clks3

Notes: 1. P4 area address indicates the register address when P4 area in the virtual address space is used. Area 7 address indicates the register address when accessed using TLB via area 7 in the physical address space.
2. There are restrictions on write access.

Table 22.5 HDMAC Register States in Each Operating Mode

Register Name	Abbreviation	Power-On Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiplex Exception	Sleep by Sleep Instruction	Module Standby	Light Sleep
HDMAC operating mode setting register	CXR0	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Transmit activation register	CXR1	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Receive activation register	CXR2	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Transmit descriptor start address setting register	CXR3	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Receive descriptor start address setting register	CXR4	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Status register	CXR5	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Interrupt mask setting register	CXR6	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Error mask setting register	CXR7	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Discarded frame counter register	CXR8	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Transmit FIFO threshold setting register	CXR9	H'0000 0000	H'0000 0000	Retained	Retained	Retained
External FIFO depth setting register	CXR10	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Receive activation reset method setting register	CXR11	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Reserved	CXR12	—	—	—	—	—
Transmit FIFO underrun counter register	CXR13	H'0000 0000	H'0000 0000	Retained	Retained	Retained

Register Name	Abbreviation	Power-On Reset by RESET Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiplex Exception	Sleep by Sleep Instruction	Module Standby	Light Sleep
Receive FIFO overflow counter register	CXR14	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Reserved	CXR15	—	—	—	—	—
Receive FIFO busy transmit threshold setting register	CXR16	H'0007 0007	H'0007 0007	Retained	Retained	Retained
Reserved	CXR17	—	—	—	—	—
Reserved	CXRS2	—	—	—	—	—
Transmit interrupt mode setting register	CXR18	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Reserved	CXR19	—	—	—	—	—
Reserved	CXRD0	—	—	—	—	—
Reserved	CXRD1	—	—	—	—	—
Reserved	CXRD2	—	—	—	—	—
Reserved	CXRD3	—	—	—	—	—
Reserved	CXRD4	—	—	—	—	—
Reserved	CXRD5	—	—	—	—	—
Reserved	CXRD6	—	—	—	—	—

Table 22.6 feLic Register States in Each Operating Mode

Register Name	Abbreviation	Power-On Reset by RESET Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiplex Exception	Sleep by Sleep Instruction	Module Standby	Light Sleep
feLic operating mode setting register	CXR20	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Long frame length check value setting register	CXR2A	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Status register	CXR21	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Interrupt mask setting register	CXR22	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MII control register	CXR23	H'0000 000x	H'0000 000x	Retained	Retained	Retained
PHY status register	CXR2B	H'0000 000x	H'0000 000x	Retained	Retained	Retained

Register Name	Abbreviation	Power-On Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiplex Exception	Sleep by Sleep Instruction	Module Standby	Light Sleep
Random number generating counter upper limit setting register	CXR30	H'0000 0000	H'0000 0000	Retained	Retained	Retained
IPG counter setting register	CXR70	H'0000 0014	H'0000 0014	Retained	Retained	Retained
Auto PAUSE parameter setting register	CXR71	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Manual PAUSE parameter setting register	CXR72	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Receive PAUSE frame counter register	CXR80	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PAUSE frame retransmit count setting register	CXR81	H'0000 0000	H'0000 0000	Retained	Retained	Retained
PAUSE frame retransmit count register	CXR82	H'0000 0000	H'0000 0000	Retained	Retained	Retained
Reserved	CXR83	—	—	—	—	—
MAC address high register	CXR24	H'0000 0000	H'0000 0000	Retained	Retained	Retained
MAC address low register	CXR25	H'0000 0000	H'0000 0000	Retained	Retained	Retained
TINT1 counter register	CXR40	H'0000 0000	H'0000 0000	Retained	Retained	Retained
TINT2 counter register	CXR41	H'0000 0000	H'0000 0000	Retained	Retained	Retained
TINT3 counter register	CXR42	H'0000 0000	H'0000 0000	Retained	Retained	Retained
TINT4 counter register	CXR43	H'0000 0000	H'0000 0000	Retained	Retained	Retained
RINT1 counter register	CXR50	H'0000 0000	H'0000 0000	Retained	Retained	Retained
RINT2 counter register	CXR51	H'0000 0000	H'0000 0000	Retained	Retained	Retained
RINT3 counter register	CXR52	H'0000 0000	H'0000 0000	Retained	Retained	Retained
RINT4 counter register	CXR53	H'0000 0000	H'0000 0000	Retained	Retained	Retained
RINT5 counter register	CXR54	H'0000 0000	H'0000 0000	Retained	Retained	Retained
RINT8 counter register	CXR55	H'0000 0000	H'0000 0000	Retained	Retained	Retained

22.4.3 HDMAC Operating Mode Setting Register (CXR0)

CXR0 is used for specifying the operating mode of the HDMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DE	DL1	DL0	—	—	—	SWR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 7	—	All 0	R	Reserved
6	DE	0	R/W	DMA Data Endian Conversion 0: Does not perform endian conversion for DMA data. (initial value) 1: Performs endian conversion for DMA data. Endian conversion for descriptors and registers is not performed.
5, 4	DL1 and DL0	00	R/W	Transmit/Receive Descriptor Length Setting — Descriptor length 00 — 16 bytes (initial value) 01 — 32 bytes 10 — 64 bytes 11 — Reserved
3 to 1	—	All 0	R	Reserved
0	SWR	0	R/W	HDMAC/feLic Software Reset Writing 1 resets the HDMAC/feLic module except for CXR3, CXR4, CXR8, CXR13 and CXR14. Note: Registers should not be accessed until 64 clock cycles have elapsed after the SWR bit is set to 1. When writing to registers (except for the SWR bit), the transmitting and receiving function should be disabled.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.4 Transmit Activation Register (CXR1)

CXR1 initiates transmission by the HDMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRNS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 1	—	All 0	R	Reserved
0	TRNS	0	R/W	Transmit Activation If 1 is written to this bit, the HDMAC accesses the transmit descriptor ring and transmits frames for all the valid descriptors. This bit is automatically cleared when transmission of all the valid frames has been completed, or the transmit descriptor depletion occurs. Note: Writing 0 is disabled.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.5 Receive Activation Register (CXR2)

CXR2 initiates reception by the HDMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	R
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 1	—	All 0	R	Reserved
0	R	0	R/W	Receive Ready If 1 is written to this bit, the HDMAC accesses the receive descriptor ring and enters the receive wait state. For operation of this bit after completion of reception, which depends on the CXR11 settings, see section 22.4.14, Receive Activation Reset Method Setting Register (CXR11). If 0 is written to this bit, the HDMAC completes reception of the frame being processed, and then disables the receiving function. This bit is reset to 0 when the RACT bit of the fetched descriptor is 0 (when the receive descriptor depletion occurs).

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.6 Transmit Descriptor Start Address Setting Register (CXR3)

CXR3 specifies the start address of the transmit descriptor ring. The start address of the descriptor ring should be aligned to the boundary consistent with the descriptor length specified by the DL bits in CXR0.

Writing to CXR3 should be performed before transmission is started. Writing to CXR3 after transmission has been started is prohibited and operation according to the written value cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDPA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDPA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDPA [31:0]	All 0	R/W	Transmit Descriptor Start Address Bits 3 to 0 should be set to 0 for alignment of the descriptor. Note: A value other than 0 must not be set to bits 3 to 0. Writing is prohibited after transmission has been started.

22.4.7 Receive Descriptor Start Address Setting Register (CXR4)

CXR4 specifies the start address of the receive descriptor ring. The start address of the descriptor ring should be aligned to the boundary consistent with the descriptor length specified by the DL bits in CXR0.

Writing to CXR4 should be performed before reception is started. Writing to CXR4 after reception has been started is prohibited and operation according to the written value cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDPA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDPA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDPA [31:0]	All 0	R/W	Transmit Descriptor Start Address Bits 3 to 0 should be set to 0 for alignment of the descriptor. Note: A value other than 0 must not be set to bits 3 to 0. Writing is prohibited after transmission has been started.

22.4.8 Status Register (CXR5)

CXR5 is a register that indicates HDMAC interrupt sources.

The status sources are output to the INTC module by a signal of each source.

Each bit is cleared by writing 1 to the corresponding bit. The MINT bit is set to 0 by clearing CXR21. Generation of interrupts can be masked by the corresponding bits in CXR6.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TWB	BINT	AINT	LKON	TABT	RABT	RFRMER	BER	MINT	FTC	TDE	TFE	FRC	RDE	RFE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TINT8	TINT7	TINT6	TINT5	TINT4	TINT3	TINT2	TINT1	RINT8	RINT7	RINT6	RINT5	RINT4	RINT3	RINT2	RINT1
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Bit	Bit Name	Initial Value*2	R/W	Description
31	—	0	R	Reserved
30	TWB	0	R/W*1	Indicates that write-back to the transmit descriptor whose TWBI bit is set has been performed. The interrupt can be generated every frame by so setting CXR18. <Interrupt source>
29	BINT	0	R/W*1	Not used.
28	AINT	0	R/W*1	Not used.
27	LKON	0	R/W*1	Not used.
26	TABT	0	R/W*1	Transmit Abort Detect <Interrupt source>
25	RABT	0	R/W*1	Receive Abort Detect <Interrupt source>
24	RFRMER	0	R/W*1	Receive Frame Count Overflow Occurrence <Interrupt source>
23	BER	0	R/W*1	Indicates that a DMA error is input.
22	MINT	0	R	M Port Interrupt Occurrence <Interrupt source> Indicates that an interrupt signal from the feLic is asserted.

Bit	Bit Name	Initial Value*[2]	R/W	Description
21	FTC	0	R/W*[1]	Frame Transmit Completion <Interrupt source> Indicates that transmission of all the frames specified by the transmit descriptor has been completed. This bit is not set for each frame.
20	TDE	0	R/W*[1]	Transmit Descriptor Depletion <Interrupt source> Indicates that the fetched transmit descriptor is invalid. However, the following cases are excluded. • Immediately after the information in the descriptor includes the end of a frame. • Immediately after the information in the descriptor includes one whole frame.
19	TFE	0	R/W*[1]	Transmit FIFO Underflow Error Occurrence <Interrupt source> Indicates that the transmit FIFO becomes empty during frame transmission.
18	FRC	0	R/W*[1]	Frame Receive Completion <Interrupt source> Indicates that the receive descriptor is updated by frame reception. This bit is set when reception of one frame has been completed regardless of the setting of the RR bit in CXR11.
17	RDE	0	R/W*[1]	Receive Descriptor Depletion <Interrupt source> Indicates that the fetched receive descriptor is invalid. Occurrence of this source resets the R bit in CXR2 to 0.
16	RFE	0	R/W*[1]	Receive FIFO Overflow Error Occurrence <Interrupt source>
15 to 8	TINT8 to TINT1	All 0	R/W*[1]	Transmit Port Interrupt
7 to 0	RINT8 to RINT1	All 0	R/W*[1]	Receive Port Interrupt

Notes: 1. These bits are cleared by writing 1 to the corresponding bits. Writing 0 has no meaning.
2. This register is initialized by an HDMAC/feLic software reset.

22.4.9 Interrupt Mask Setting Register (CXR6)

CXR6 is a register that masks the interrupts indicated by CXR5.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TWB	BINT	AINT	LKON	TABT	RABT	RFRMER	BER	MINT	FTC	TDE	TFE	FRC	RDE	RFE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TINT8	TINT7	TINT6	TINT5	TINT4	TINT3	TINT2	TINT1	RINT8	RINT7	RINT6	RINT5	RINT4	RINT3	RINT2	RINT1
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31	—	0	R	Reserved
30	TWB	0	R/W	Interrupt Enable 0: Disables an interrupt indicated by the corresponding bit in CXR5. 1: Enables an interrupt indicated by the corresponding bit in CXR5. Note: Bits 29 to 27 should always be set to 0.
29	BINT	0	R/W	
28	AINT	0	R/W	
27	LKON	0	R/W	
26	TABT	0	R/W	
25	RABT	0	R/W	
24	RFRMER	0	R/W	
23	BER	0	R/W	
22	MINT	0	R/W	
21	FTC	0	R/W	
20	TDE	0	R/W	
19	TFE	0	R/W	
18	FRC	0	R/W	
17	RDE	0	R/W	
16	RFE	0	R/W	
15 to 8	TINT8 to TINT1	All 0	R/W	
7 to 0	RINT8 to RINT0	All 0	R/W	

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.10 Error Mask Setting Register (CXR7)

CXR7 is a register that specifies whether or not to allow the status of the descriptor information (indicated by TFS[7:0] bits in TD0 or RFS[7:0] bits in RD0) to be reflected in the TFE bit in the transmit descriptor 0 (TD0) or the RFE bit in the receive descriptor 0 (RD0) as a summary.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TINT8	TINT7	TINT6	TINT5	TINT4	TINT3	TINT2	TINT1	RINT8	RINT7	RINT6	RINT5	RINT4	RINT3	RINT2	RINT1
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 16	—	All 0	R	Reserved
15 to 8	TINT8 to TINT1	All 0	R/W	Transmit Descriptor Error Flag (TFE) Mask 0: Allows TINT<8:1> (TFS[7:0] bits in TD0) to be reflected in TFE. 1: Does not allow TINT<8:1> (TFS[7:0] bits in TD0) to be reflected in TFE.
7 to 0	RINT8 to RINT1	All 0	R/W	Receive Descriptor Error Flag (RFE) Mask 0: Allows RNT<8:1> (RFS[7:0] bits in RD0) to be reflected in RFE. 1: Does not allow RINT<8:1> (RFS[7:0] bits in RD0) to be reflected in RFE.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.11 Discarded Frame Counter Register (CXR8)

CXR8 is a register that indicates the number of discarded frames.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MIS[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved
15 to 0	MIS15 to MIS0	All 0	R/W	Discarded Frame Counter These bits indicate the number of frames which are discarded because the receive descriptor is disabled, for example. The counter stops counting when all bits are set to 1 (H'FFFF). Note: Writing any data clears these bits.

22.4.12 Transmit FIFO Threshold Setting Register (CXR9)

CXR9 is a register that specifies the threshold for the transmit FIFO.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FO[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 11	—	All 0	R	Reserved
10 to 0	FO10 to FO0	All 0	R/W	Transmit FIFO Threshold Setting — Threshold H'000 — Store and forward (initial value) H'001 — 4 bytes H'002 — 8 bytes : — : H'0FE — 1016 bytes H'0FF — 1020 bytes H'100 — 1024 bytes : — : H'1FE — 1040 bytes H'1FF — 2044 bytes H'200 — 2048 bytes : — : H'3FE — 4088 bytes 3 H'FF — 4092 bytes H'400 to H'7FF — Setting prohibited Reading (transmission) is started when one frame data has been written to the FIFO or when data in the FIFO exceeds the above setting. Using on the initial value (store and forward) is recommended to prevent from occurring underflow error on sending. Note: Writing is disabled when the TRNS bit in CXR1 = 1.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.13 External FIFO Depth Setting Register (CXR10)

CXR10 is a register that specifies the depth of the transmit and receive FIFOs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TA[4:0]					—	—	—	RA[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 13	—	All 0	R	Reserved
12 to 8	TA4 to TA0	All 0	R/W	Transmit FIFO Depth Setting — Depth H'00 — 256 bytes (initial value) H'01 — 512 bytes H'02 — 768 bytes H'03 — 1024 bytes : — : H'07 — 2048 bytes : — : H'0F — 4096 bytes H'10 to H'1F — Setting prohibited Note: Writing to bits 12 to 8 is disabled when the TRNS bit in CXR1 = 1.
7 to 5	—	All 0	R	Reserved

Bit	Bit Name	Initial Value*[1]	R/W	Description
4 to 0	RA4 to RA0	All 0	R/W	Receive FIFO Depth Setting: Depth H'00: 256 bytes (initial value) H'01: 512 bytes H'02: 768 bytes H'03: 1024 bytes : H'07: 2048 bytes : H'0F: 4096 bytes H'10 to H'1F: Setting prohibited Note: Writing to bits 4 to 0 is disabled when the R bit in CXR2 = 1.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.14 Receive Activation Reset Method Setting Register (CXR11)

CXR11 is a register that specifies a method for resetting the receive ready bit (R bit in CXR2).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RNR	RR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 2	—	All 0	R	Reserved
1	RNR	0	R/W	Not used. This bit should always be set to 0.
0	RR	0	R/W	Receive Ready Reset 0: (initial value) Automatically clears the receive ready bit (R bit in CXR2) by hardware. The receive ready bit (R bit in CXR2) is automatically cleared by receiving one frame, thus disabling reception of the following frames. In this mode, interrupt control for each frame is possible. 1: Normal operating mode Resets the receive ready bit (R bit in CXR2 by software. After 1 is written to the receive ready bit (R bit in CXR2), the HDMAC automatically fetches the receive descriptor and receives frames until 0 is written to the receive ready bit (R bit in CXR2). Continuous reception of multiple frames is possible. Note that the receive ready bit (R bit in CXR2) is reset to 0 when the RACT bit in the fetched descriptor is 0. Note: Writing to this bit is disabled when the R bit in CXR2 = 1.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.15 Transmit FIFO Underrun Counter Register (CXR13)

CXR13 is a register that indicates the number of times the transmit FIFO underflow occurs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TFUF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved
15 to 0	TFUF15 to TFUF0	All 0	R/W	Transmit FIFO Underflow Counter The counter stops counting when all bits are set to 1 (H'FFFF). Note: Writing any data clears these bits.

22.4.16 Receive FIFO Overflow Counter Register (CXR14)

CXR14 is a register that indicates the number of times the receive FIFO overflow occurs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFOF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved
15 to 0	RFOF15 to RFOF0	All 0	R/W	Receive FIFO Overflow Counter The counter stops counting when all bits are set to 1 (H'FFFF). Note: Writing any data clears these bits.

22.4.17 Receive FIFO Busy Transmit Threshold Setting Register (CXR16)

CXR16 is a register that specifies the threshold value for sending the receive FIFO busy to the feLic based on the number of frames and number of data bytes stored in the receive FIFO.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFF[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 19	—	All 0	R	Reserved
18 to 16	RFF2 to RFF0	All 1	R/W	Busy Transmit Threshold based on the Number of Receive Frames Setting Threshold 000 Transmits the busy when two frames have been stored. 001 Transmits the busy when four frames have been stored. 010 Transmits the busy when six frames have been stored. : 110 Transmits the busy when 14 frames have been stored. 111 Transmits the busy when 16 frames have been stored. (initial value)
15 to 3	—	All 0	R	Reserved

Bit	Bit Name	Initial Value*[1]	R/W	Description
2 to 0	RFD2 to RFD0	All 1	R/W	Busy Transmit Threshold based on the Amount of Data in the Receive FIFO Setting Threshold 000 Transmits the busy when 224 (256 – 32) bytes have been stored. 001 Transmits the busy when 480 (512 – 32) bytes have been stored. 010 Transmits the busy when 736 (768 – 32) bytes have been stored. 011 Transmits the busy when 992 (1024 – 32) bytes have been stored. : 110 Transmits the busy when 1760 (1792 – 32) bytes have been stored. 111 See the following note. (initial value) Note: The set value to these bits should be smaller than the set value to the RA bits in CXR10. If the former and latter values are equal, the busy will be sent out when (receive FIFO depth − 64 bytes) have been stored in the receive FIFO, not when (receive FIFO depth − 32 bytes) have been stored. For example, when the RA bits in CXR10 and RFD bits in CXR16 are both 7, the busy will be sent out when 1984 (= 2048 − 64) bytes have been stored in the receive FIFO.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.18 Transmit Interrupt Mode Setting Register (CXR18)

CXR18 is used for setting transmit interrupt operating mode for the HDMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TIM	—	—	—	TIS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 5	—	All 0	R	Reserved
4	TIM	0	R/W	Transmit Interrupt Mode 0: Mode in which the interrupt is issued each time a frame has been transmitted. 1: Mode in which the interrupt is issued when write-back to the descriptor whose TWBI bit is set has been completed.
3 to 1	—	All 0	R	Reserved
0	TIS	0	R/W	Interrupt Mode Enable 0: TIM setting is disabled. (Initial value) 1: TIM setting is enabled.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.19 feLic Operating Mode Setting Register (CXR20)

CXR20 is a register that specifies the operating mode of the FELIC. Some mode bits in this register should not be modified while the transmitting and receiving functions are enabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	TPC	ZPF	PFR	RXF	TXF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	CER	—	—	MPM	—	—	RPE	TPE	—	ILB	OLB	DPM	PRM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R	R	R/W	R/W	R	R/W*1	R/W*1	R/W*1	R/W*1

Bit	Bit Name	Initial Value*2	R/W	Description
31 to 21	—	All 0	R	Reserved
20	TPC	0	R/W	PAUSE Frame Transmission 0: Transmission of a PAUSE frame is enabled even during a PAUSE period. 1: Transmission of a PAUSE frame is disabled during PAUSE period.
19	ZPF	0	R/W	PAUSE Frame Enable with TIME = 0 0: Control of a PAUSE frame whose TIME parameter value is 0 is disabled. 1: Control of a PAUSE frame whose TIME parameter value is 0 is enabled.
18	PFR	0	R/W	PAUSE Frame Receive Mode 0: PAUSE frame is not transferred to the upper module. 1: PAUSE frame is transferred to the upper module.
17	RXF	0	R/W	Operating Mode for Reception Flow Control 0: PAUSE frame detection is disabled. 1: Reception flow control is enabled.
16	TXF	0	R/W	Operating Mode for Transmission Flow Control 0: Transmission flow control is disabled. 1: Transmission flow control is enabled.
15 to 13	—	All 0	R	Reserved

Bit	Bit Name	Initial Value*[2]	R/W	Description
12	CER	0	R/W	CRC Error Frame Receive Mode 0: Normal mode (an error frame is regarded as an error). 1: Mode in which the received frame with a CRC error is not regarded as an error.
11, 10	—	All 0	R	Reserved
9	MPM	0	R/W	Magic Packet Detection Enable 0: Magic packet detection is disabled. 1: Magic packet detection is enabled.
8, 7	—	All 0	R	Reserved
6	RPE	0	R/W	Reception Enable 0: Receiving function is disabled after completion of current frame reception. 1: Frame receiving function is enabled and receive data is transferred to the HDMAC.
5	TPE	0	R/W	Transmission Enable 0: Transmitting function is disabled after completion of current frame transmission. 1: A frame transmit request from the HDMAC is enabled.
4	—	0	R	Reserved
3	ILB	0	R/W*[1]	FELIC Loopback Mode 0: The FELIC enters normal mode. 1: The FELIC enters loopback mode.
2	OLB	0	R/W*1	Not used.
1	DPM	0	R/W*[1]	Duplex Mode 0: The FELIC enters half-duplex mode. Note that FELIC loopback mode should not be specified during this mode. 1: The FELIC enters full-duplex mode.
0	PRM	0	R/W*[1]	Promiscuous Mode 0: The FELIC enters normal mode. (When the MPM bit is set, the FELIC operates in normal mode regardless of the status of this bit.) 1: The FELIC enters promiscuous mode.

Notes: 1. Bits 3 to 0 should not be modified while the transmitting and receiving functions are enabled.
2. This register is initialized by an HDMAC/feLic software reset.

22.4.20 Long Frame Length Check Value Setting Register (CXR2A)

CXR2A is a register that specifies the upper limit of the receive frame length used for checking the length.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FLUL[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 12	—	All 0	R	Reserved
11 to 0	FLUL11 to FLUL0	All 0	R/W	Frame Length Upper Limit The specified value is used as the frame length check value. When the received frame length value exceeds this value, it is regarded as RINT4 (frame length error). (See below.) Setting Check Value H'000 to H'5EE 1518 bytes H'5EF 1519 bytes : : H'7FF 2047 bytes H'800 to H'FFF 2048 bytes Note: The CXR20 register should not be modified while the receiving function is enabled.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.21 Status Register (CXR21)

CXR21 is a register that indicates the FELIC status.

The status sources are output to the INTC module by a signal of each source.

Each bit is cleared by writing 1 to the corresponding bit. Generation of interrupts can be masked by the corresponding bits in CXR22.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	BCR	PRO	—	LNK	MPR	FCD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R/W*1	R/W*1	R/W*1

Bit	Bit Name	Initial Value*2	R/W	Description
31 to 6	—	All 0	R	Reserved
5	BCR	0	R/W*1	Reserved
4	PRO	0	R/W*1	PAUSE Frame Retransmit Retry Over <Interrupt Source> Indicates that the PAUSE frame retransmission count has reached the upper limit of retransmission set in the CXR81 register.
3	—	0	R	Reserved
2	LNK	0	R/W*1	LINK Signal Change Interrupt <Interrupt Source> Indicates that the LINK signal output from the PHY has changed from high to low or low to high.
1	MPR	0	R/W*1	Magic Packet Receive Interrupt <Interrupt Source> Indicates that a magic packet has been received. (This bit is cleared by writing 1 to the bit; however, the MAGIC signal is not cleared because it is an external output signal.)
0	FCD	0	R/W*1	Illegal Carrier Detection Interrupt <Interrupt Source> Indicates that an illegal carrier is detected on the line.

Notes: 1. These bits are cleared by writing 1 to the corresponding bits. Writing 0 has no meaning.
2. This register is initialized by an HDMAC/feLic software reset.

22.4.22 Interrupt Mask Setting Register (CXR22)

CXR22 is a register that masks the interrupts indicated by CXR21.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	BCRE	PROE	—	LNKE	MPRE	FCDE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 6	—	All 0	R	Reserved
5	BCRE	0	R/W	Reserved
4	PROE	0	R/W	PAUSE Frame Retransmit Retry Over Interrupt Enable 0: Disables an interrupt indicated by the corresponding bit in CXR21. 1: Enables an interrupt indicated by the corresponding bit in CXR21.
3	—	0	R	Reserved
2	LKNE	0	R/W	LINK Signal Change Interrupt Enable 0: Disables an interrupt indicated by the corresponding bit in CXR21. 1: Enables an interrupt indicated by the corresponding bit in CXR21.
1	MPRE	0	R/W	Magic Packet Receive Interrupt Enable 0: Disables an interrupt indicated by the corresponding bit in CXR21. 1: Enables an interrupt indicated by the corresponding bit in CXR21.
0	FCDE	0	R/W	Illegal Carrier Detection Interrupt Enable 0: Disables an interrupt indicated by the corresponding bit in CXR21. 1: Enables an interrupt indicated by the corresponding bit in CXR21.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.23 MII Control Register (CXR23)

CXR23 is a register that controls the MII interface to access the PHY register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	×	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 4	—	All 0	R	Reserved
3	MDI	×	R	MII Management Data In Read data from MII
2	MDO	0	R/W	MII Management Data Out Write data to MII
1	MMD	0	R/W	MII Management Mode 1: Data is written to MII. 0: Data is read from MII.
0	MDC	0	R/W	MII Management Clock Clock for MII

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.24 PHY Status Register (CXR2B)

CXR2B is a register that can monitor the PHY status signal.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LINK
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 1	—	All 0	R	Reserved
0	LINK	×	R	The status of the PHY output LINK signal can be monitored. See the specifications of the PHY to be connected.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.25 Random Number Generating Counter Upper Limit Setting Register (CXR30)

CXR30 is a register that can set the upper limit of the counter used for the random number generator.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RDM[19:16]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDM[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 20	—	All 0	R	Reserved
19 to 0	RDM19 to RDM0	All 0	R/W	Upper limit of the Counter used for the Random Number Generator H'00000: Setting for normal operation H'00001 to H'FFFFE: Counter upper limit Note: Since the operation of the feLic random number generator depends on the value set to this register, take care when setting a value other than 0.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.26 IPG Counter Setting Register (CXR70)

CXR70 is a register that specifies the IPG (inter packet gap) value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	IPG[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 5	—	All 0	R	Reserved
4 to 0	IPG4 to IPG0	10100	R/W	These bits set the IPG value in 40 ns units. Setting IPG value H'00 to H' 06 400 ns H'07 440 ns : : H'13 920 ns H'14 960 ns (default) : : H'1F 1440 ns Note: The CXR20 register should not be modified while the transmitting and receiving functions are enabled.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.27 Automatic PAUSE Parameter Setting Register (CXR71)

CXR71 is used to set the TIME parameter value of an automatic PAUSE frame. When an automatic PAUSE frame is transmitted, the value set in this register is used as the TIME parameter of the PAUSE frame.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	APAUSE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 16	—	All 0	R	Reserved
15 to 0	APAUSE15 to APAUSE0	All 0	R/W	TIME Parameter Value of an Automatic PAUSE Frame One bit is equivalent to 512 bit-time.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.28 Manual PAUSE Parameter Setting Register (CXR72)

CXR72 is used to set the TIME parameter value of a manual PAUSE frame. When a manual PAUSE frame is transmitted, the value set in this register is used as the TIME parameter of the PAUSE frame.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPAUSE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 16	—	All 0	W	Reserved
15 to 0	MPAUSE15 to MPAUSE0	All 0	W	TIME Parameter Value of an Manual PAUSE Frame One bit is equivalent to 512 bit-time.

Notes: This register can't be read.
1. This register is initialized by an HDMAC/feLic software reset.

22.4.29 Receive PAUSE Frame Counter Register (CXR80)

CXR80 is used to the counter of received the PAUSE frame.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RPAUSE[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 8	—	All 0	R	Reserved
7 to 0	RPAUSE 7 to RPAUSE 0	All 0	R	Received PAUSE frame Counter

Notes: Writing to this register has no meaning.

1. This register is initialized by an HDMAC/feLic software reset.

22.4.30 PAUSE Frame Retransmit Count Setting Register (CXR81)

CXR81 is used to set the upper limit of the automatic PAUSE frame retransmission count. For details of this register, see 22.9.2 (6), Flow Control Conforming to IEEE802.3.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPAUSE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 16	—	All 0	R	Reserved
15 to 0	TXPAUSE15 to TXPAUSE0	All 0	R/W	Upper Limit of PAUSE Frame Retransmission Setting — Retransmit count H'0000 — Unlimited*[2] H'0001 — 1 : — : H'FFFF — 65535 Note: The CXR20 register should not be modified while the transmitting function is enabled.

Notes: 1. This register is initialized by an HDMAC/feLic software reset.

2. When using in unlimited setting, The time parameter MPAUSE of CXR71 should be set other than zero.

22.4.31 PAUSE Frame Retransmit Counter Register (CXR82)

CXR82 is used to the counter of retransmitting of the PAUSE frame. For details of this register, see 22.9.2 (6), Flow Control Conforming to IEEE802.3.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXP[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 8	—	All 0	R	Reserved
15 to 0	TXP15 to TAP0	All 0	R	PAUSE Frame Retransmit Counter

Notes: Writing to this register has no meaning.

1. This register is initialized by an HDMAC/feLic software reset.

22.4.32 MAC Address High Register (CXR24)

CXR24 is used to set the upper 32 bits of the universal MAC address.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MACH[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MACH[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	MACH31 to MACH0	All 0	R/W	MAC Address Upper 32 Bits Note: These bits should not be written to while the transmission enable and reception enable bits in CXR20 are set.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.33 MAC Address Low Register (CXR25)

CXR25 is used to set the lower 16 bits of the universal MAC address.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MACL[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 16	—	All 0	R	Reserved
15 to 0	MACL15 to MACL0	All 0	R/W	MAC Address Lower 16 Bits Note: These bits should not be written to while the transmission enable and reception enable bits in CXR20 are set.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.34 TINT1 Count Register (CXR40)

CXR40 is a register with which the TINT1 source signal count value can be read and refer to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TINT1C[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TINT1C[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	TINT1C31 to TINT1C0	All 0	R/W	TINT1 (Transmit Timeout) Counter This counter is not automatically updated after incremented to H'FFFFFFFF. To update the counter continuously and restart increment from H'00000000, write-access to the counter and clear it. Note: Cleared by write access.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.35 TINT2 Count Register (CXR41)

CXR41 is a register with which the TINT2 source signal count value can be read and refer to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TINT2C[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TINT2C[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	TINT2C31 to TINT2C0	All 0	R/W	TINT2 (Collision Detection during Frame Transmission) Counter This counter is not automatically updated after incremented to H'FFFFFFFF. To update the counter continuously and restart increment from H'00000000, write-access to the counter and clear it. Note: Cleared by write access.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.36 TINT3 Count Register (CXR42)

CXR42 is a register with which the TINT3 source signal count value can be read and refer to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TINT3C[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TINT3C[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	TINT3C31 to TINT3C0	All 0	R/W	TINT3 (Carrier Loss during Frame Transmission) Counter This counter is not automatically updated after incremented to H'FFFFFFFF. To update the counter continuously and restart increment from H'00000000, write-access to the counter and clear it. Note: Cleared by write access.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.37 TINT4 Count Register (CXR43)

CXR43 is a register with which the TINT4 source signal count value can be read and refer to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TINT4C[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TINT4C[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	TINT4C31 to TINT4C0	All 0	R/W	TINT4 (Carrier not Detected) Counter This counter is not automatically updated after incremented to H'FFFFFFFF. To update the counter continuously and restart increment from H'00000000, write-access to the counter and clear it. Note: Cleared by write access.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.38 RINT1 Count Register (CXR50)

CXR50 is a register with which the RINT1 source signal count value can be read and refer to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RINT1C[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RINT1C[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	RINT1C31 to RINT1C0	All 0	R/W	RINT1 (CRC Error) Counter This counter is not automatically updated after incremented to H'FFFFFFFF. To update the counter continuously and restart increment from H'00000000, write-access to the counter and clear it. Note: Cleared by write access.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.39 RINT2 Count Register (CXR51)

CXR51 is a register with which the RINT2 source signal count value can be read and refer to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RINT2C[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RINT2C[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	RINT2C31 to RINT2C0	All 0	R/W	RINT2 (Frame Receive Error) Counter This counter is not automatically updated after incremented to H'FFFFFFFF. To update the counter continuously and restart increment from H'00000000, write-access to the counter and clear it. Note: Cleared by write access.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.40 RINT3 Count Register (CXR52)

CXR52 is a register with which the RINT3 source signal count value can be read and refer to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RINT3C[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RINT3C[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	RINT3C31 to RINT3C0	All 0	R/W	RINT3 (Frame Length Error: less than 64 bytes) Counter This counter is not automatically updated after incremented to H'FFFFFFFF. To update the counter continuously and restart increment from H'00000000, write-access to the counter and clear it. Note: Cleared by write access.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.41 RINT4 Count Register (CXR53)

CXR53 is a register with which the RINT4 source signal count value can be read and refer to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RINT4C[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RINT4C[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	RINT4C31 to RINT4C0	All 0	R/W	RINT4 (Frame Length Error: 1518 bytes or more) Counter This counter is not automatically updated after incremented to H'FFFFFFFF. To update the counter continuously and restart increment from H'00000000, write-access to the counter and clear it. Note: Cleared by write access.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.42 RINT5 Count Register (CXR54)

CXR54 is a register with which the RINT5 source signal count value can be read and refer to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RINT5C[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RINT5C[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	RINT5C31 to RINT5C0	All 0	R/W	RINT5 (Fractional Number Bit Error) Counter This counter is not automatically updated after incremented to H'FFFFFFFF. To update the counter continuously and restart increment from H'00000000, write-access to the counter and clear it. Note: Cleared by write access.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.4.43 RINT8 Count Register (CXR55)

CXR55 is a register with which the RINT8 source signal count value can be read and refer to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RINT8C[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RINT8C[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value*[1]	R/W	Description
31 to 0	RINT8C31 to RINT8C0	All 0	R/W	RINT8 (Multicast Frame Reception) Counter This counter is not automatically updated after incremented to H'FFFFFFFF. To update the counter continuously and restart increment from H'00000000, write-access to the counter and clear it. Note: Cleared by write access.

Note: 1. This register is initialized by an HDMAC/feLic software reset.

22.5 Data Format

22.5.1 Ether Packet

(1) Transmit Packet

Figure 22.4 shows the data format of a transmit packet. Table 22.7 shows the field definition of a transmit packet.

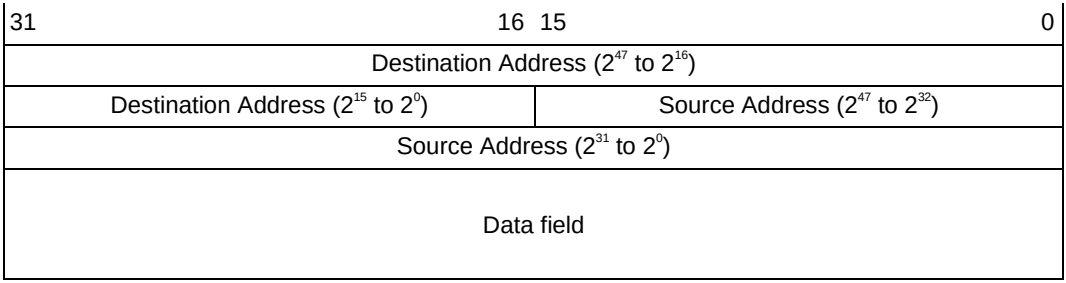


Figure 22.4 Data Format of a Transmit Packet

Table 22.7 Field Definition of a Transmit Packet

Field	Definition
Destination Address	See IEEE Std 802.3.
Source Address	See IEEE Std 802.3.
Data field	See IEEE Std 802.3.

(2) Receive Packet

Figure 22.5 shows the data format of a receive packet. Table 22.8 shows the field definition of a receive packet.

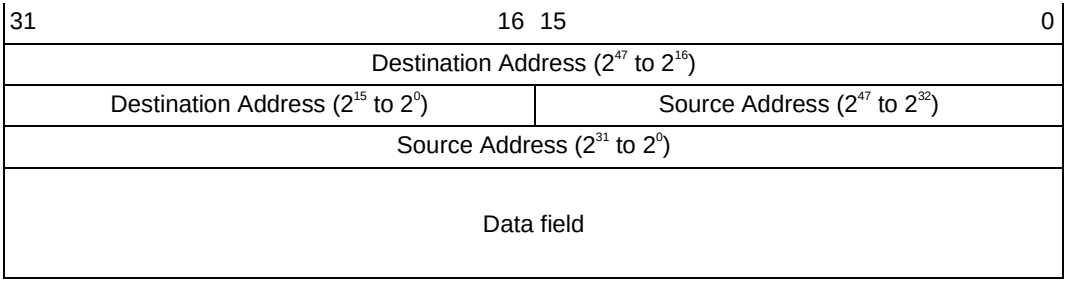


Figure 22.5 Data Format of a Receive Packet

Table 22.8 Field Definition of a Receive Packet

Field	Definition
Destination Address	See IEEE Std 802.3.
Source Address	See IEEE Std 802.3.
Data field	See IEEE Std 802.3.

22.6 Software Control Flow

22.6.1 Ether Software Control Flow

(1) Example of Ethernet Transmission and Reception Procedure

Figure 22.6 shows an example of initialization procedure for ethernet transmission and reception. Figure 22.7 shows an example of ethernet transmission and reception procedure.

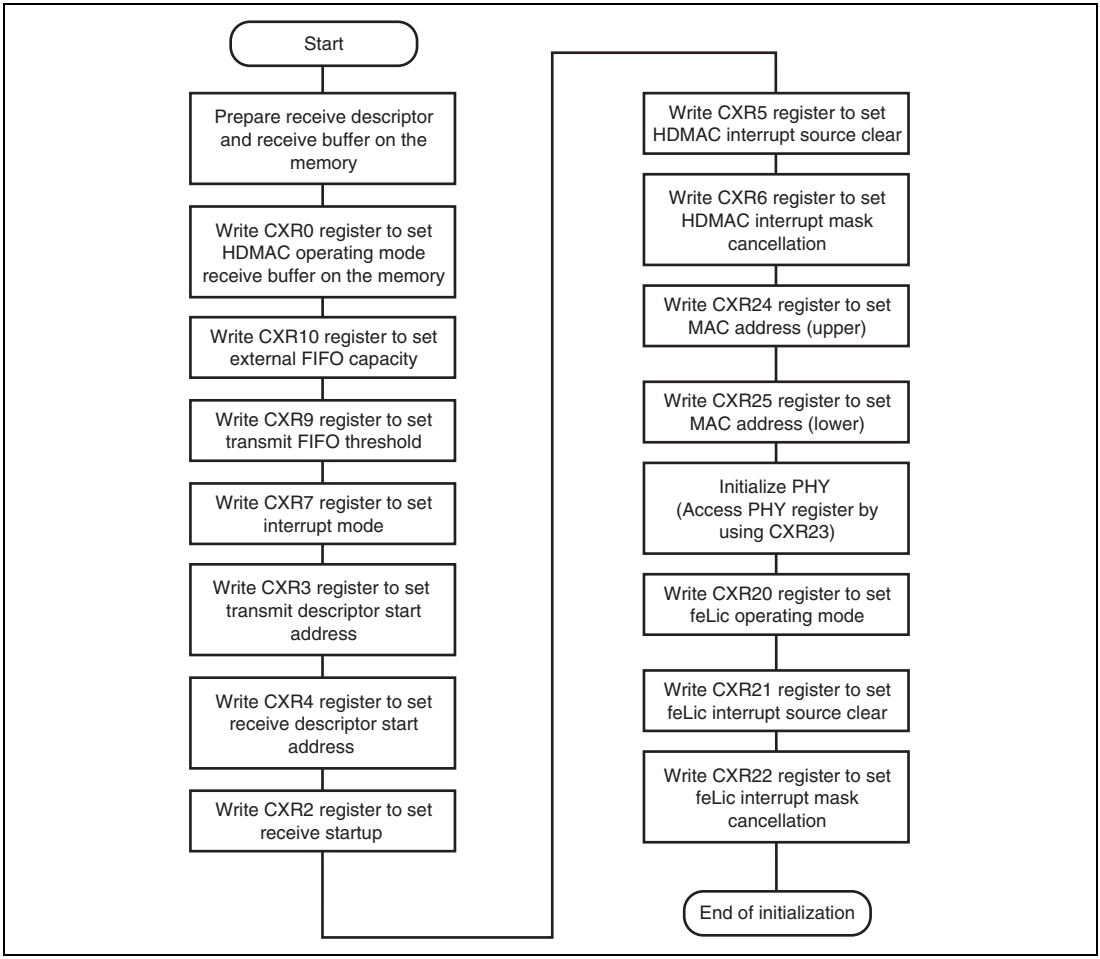


Figure 22.6 Initialization Procedure for Ethernet Transmission and Reception

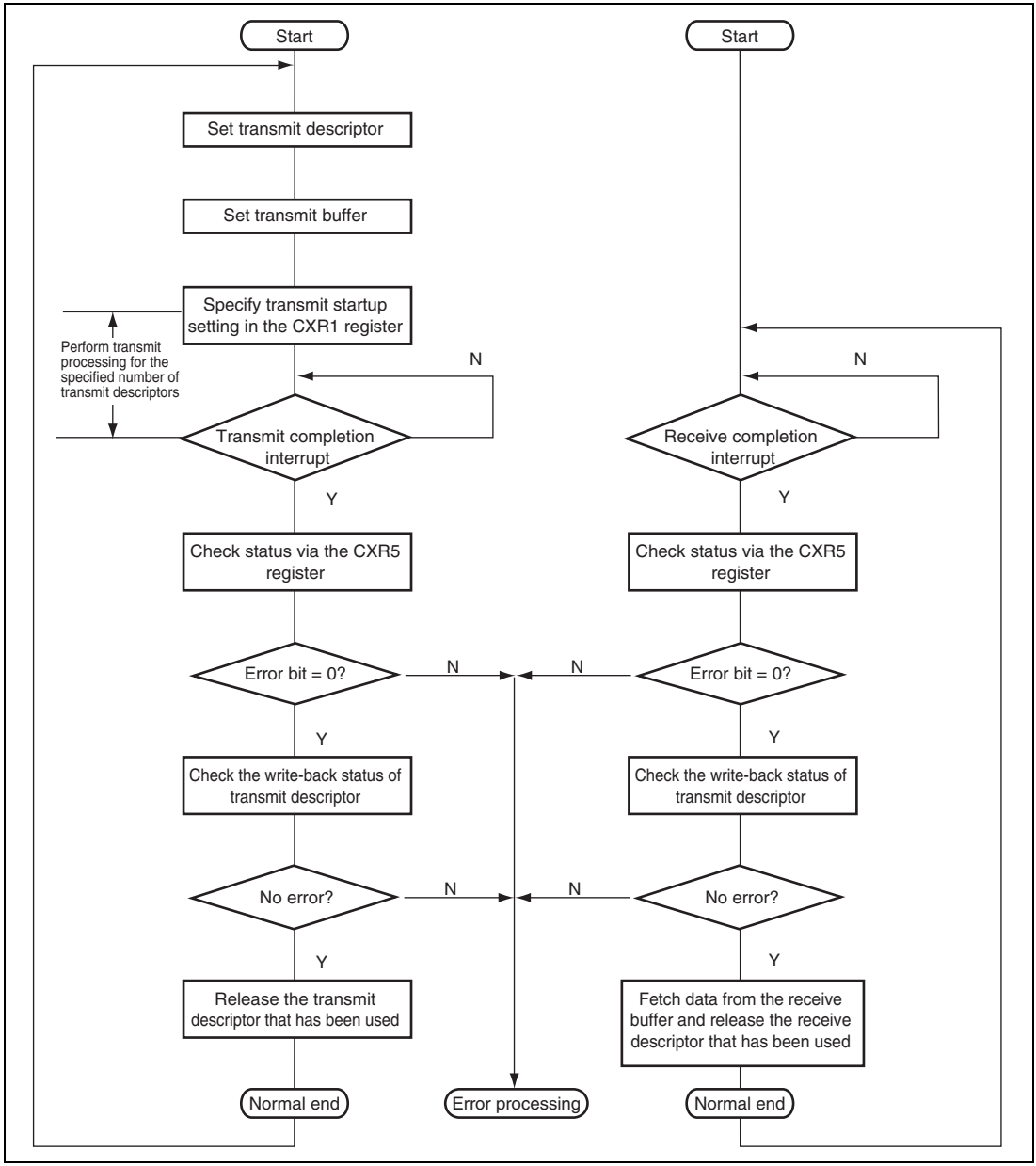


Figure 22.7 Ethernet Transmission and Reception Procedure

(2) Example of PHY Register Accessing Procedure

The PHY register is accessed by using the CXR23 register. This access should be performed using the MII management frame format as shown in figure 22.8. Figure 22.9 shows an example of 1-bit data writing procedure to achieve the MII management frame. Figure 22.10 shows an example of bus releasing procedure. Figure 22.11 shows an example of 1-bit data reading procedure. Figure 22.12 shows an example of single bus releasing procedure.

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of Bits	32	2	2	5	5	2	16	
Read	1..1	01	10	AAAA	RRRR	Z0	D....D	
Write	1..1	01	10	AAAA	RRRR	Z0	D....D	X

Legend:

PRE: 32 consecutive 1s
ST: Write 01 to indicate the start of a frame
OP: Write a code to indicate an access type
PHYAD: Write 00001 if a PHY-LSI address is 1 (write sequentially from MSB). The values of these bits change according to the PHY-LSI address.
REGAD: Write 00001 when a register address is 1 (write sequentially from MSB). The value of these bits change according to the PHY-LSI address.
TA: Data transmit source switch time on the MII interface.
(a) Write 10 during write
(b) Release bus (indicated as Z0) during read
DATA: 16-bit data. Written to or read sequentially from MSB.
(a) Write 16-bit data during write
(b) Read 16-bit data during read
IDLE: Wait time until next MII management format is input
(a) Release a single bus (indicated as X) during write
(b) Control not required because a bus has been released using the TA bits during read

Figure 22.8 MII Management Frame Format

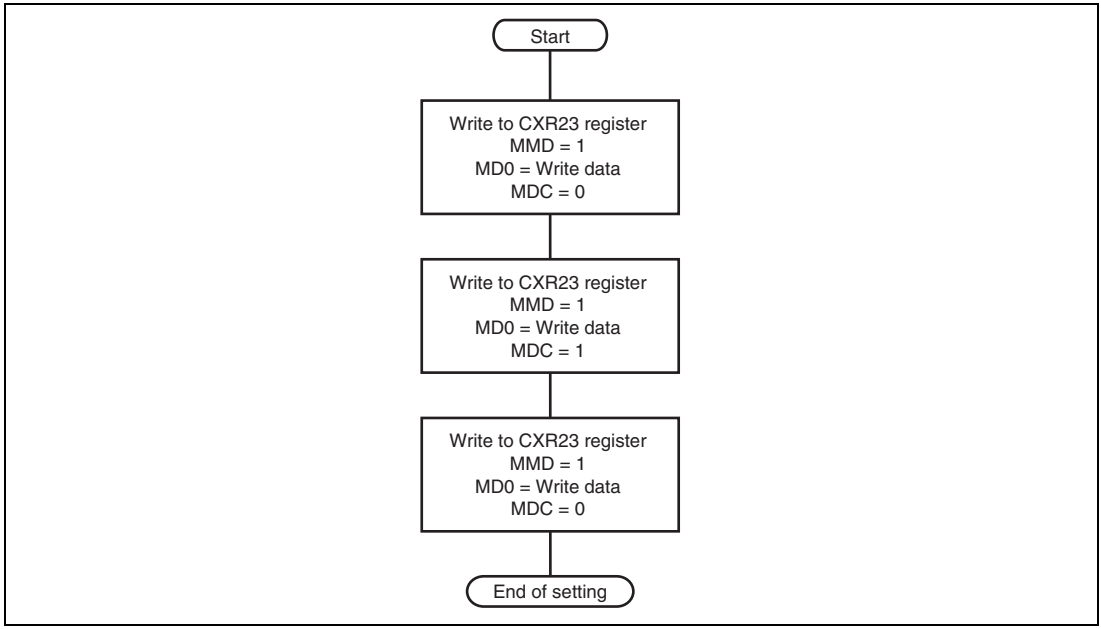


Figure 22.9 1-Bit Data Writing Procedure

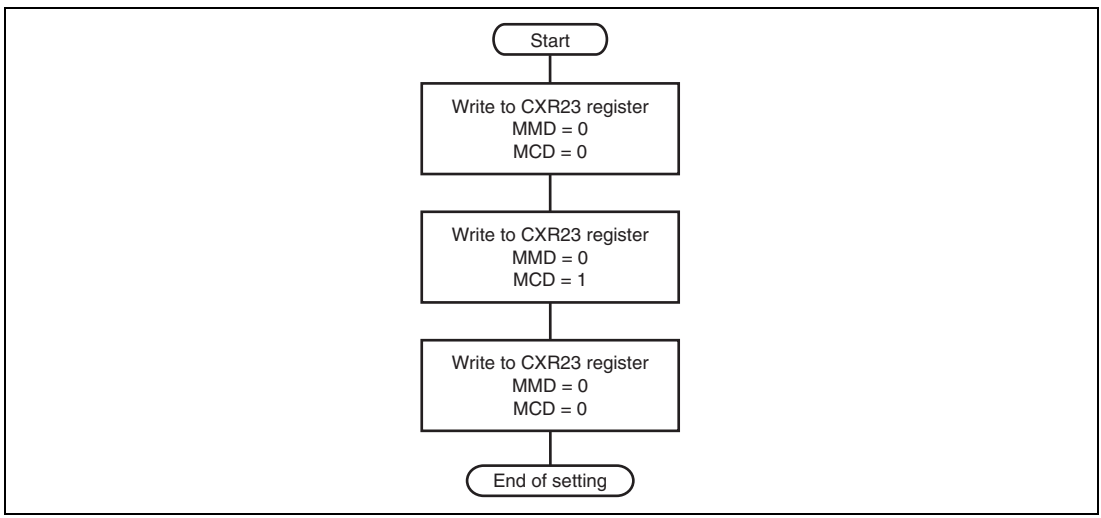


Figure 22.10 Bus Releasing Procedure

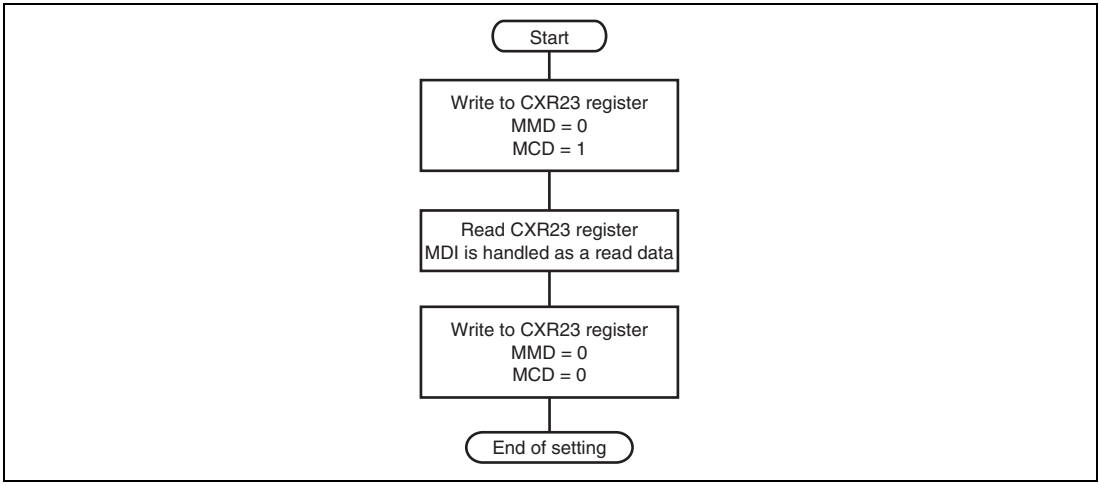


Figure 22.11 1-Bit Data Reading Procedure

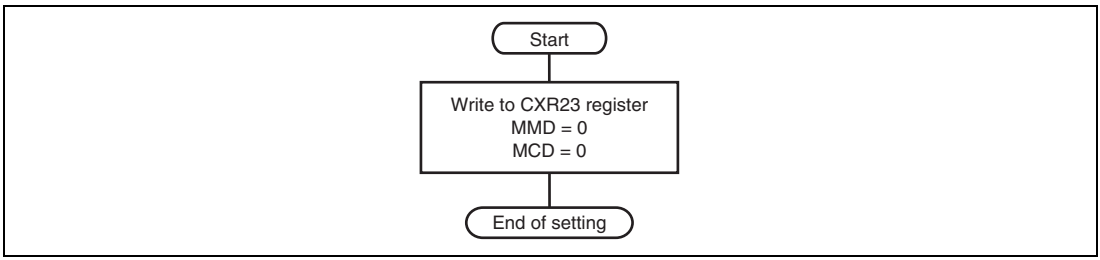


Figure 22.12 Single Bus Releasing Procedure

22.7 Notes

22.7.1 Software Reset

The ether can be software-reset by setting the SWR bit in the CXR0 register of the HDMAC to 1. If software reset is applied during DMA transfer, the current DMA transfer can be completed but the data transferred by DMA transfer cannot be guaranteed.

22.7.2 Standby

If the ether is requested to enter the standby state, the ether first completes the current SuperHyway initiator and target operation, and then enters the standby state. After returning from the standby state, the ether should be reset and initialized.

22.8 HDMAC Function Specifications

22.8.1 Operation

(1) Basic Operation

Figure 22.13 shows an image of control between this logic core (HDMAC) and device driver (software).

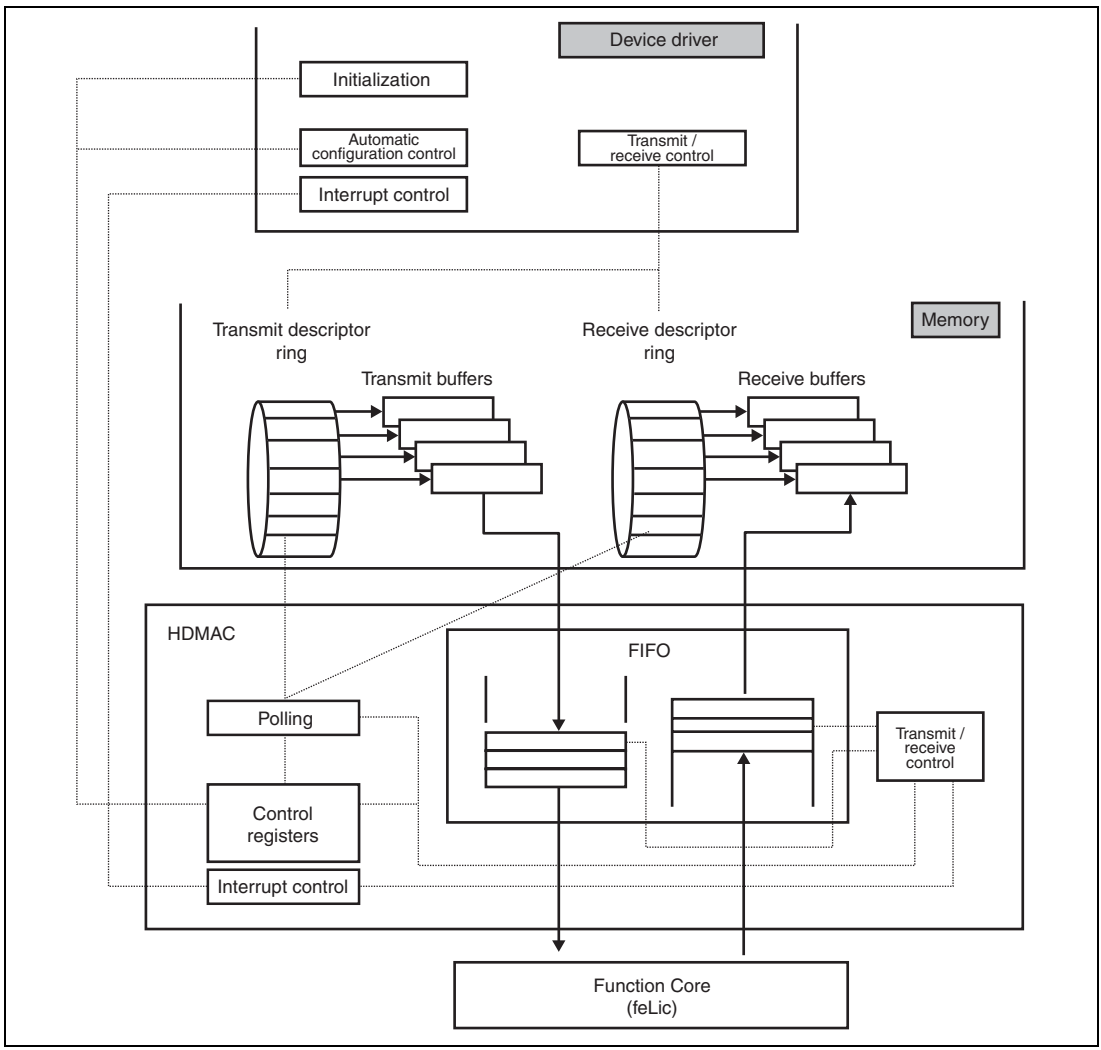


Figure 22.13 Image of Control between Device Driver and HDMAC

The device driver should create the transmit and receive descriptor rings and their corresponding transmit and receive buffers on the memory.

During transmission, this logic core fetches the transmit descriptor to obtain the transmit buffer address and the number of transmit bytes, and transmits data from the transmit buffer to the function core via the transmit FIFO. During reception, this logic core fetches the receive descriptor to obtain the receive buffer address and buffer size, and stores data received from the function core in the receive buffer via the receive FIFO.

(2) Descriptor Rings

There is no restriction on the number of descriptors. It is recommended, however, that multiple descriptors are provided.

This logic core uses the descriptor from the start descriptor to the last descriptor sequentially. If it reaches the last descriptor, it returns to use the start descriptor.

The start descriptor address is specified by the start address setting registers (CXR3 and CXR4). The last descriptor is specified by the descriptor ring end bits (TDL, RDL) = 1 in the descriptor. The descriptor boundary is specified by the descriptor length bits (DL) in the operating mode setting register (CXR0).

There is no restriction on the number of receive descriptors in terms of a ring configuration. Note that if only one receive descriptor is used, a receive descriptor depletion occurs when one frame has been received. (For details, refer to section 22.8.1 (8) (g), Receive Descriptor Depletion.)

For details on the descriptor description format, refer to section 22.8.2, Transmit Descriptors, and section 22.8.3, Receive Descriptors.

(3) Multiple Frames and Multiple Buffers

Multiple frames and multiple buffers enable consecutive frame transmission and reception.

In the receiver, a frame can be divided and stored in multiple buffers separately by specifying the frame in multiple descriptors.

In the transmitter, a frame is specified by a descriptor.

During transmission, after having transmitted data (frame) specified by a transmit descriptor, this logic core fetches the next descriptor. This logic core then determines that there is data (frame) to be transmitted next if the TACT bit in the fetched transmit descriptor is 1 and transmits the data (frame). If the TACT bit in the fetched transmit descriptor is 0, this logic core completes the transmit operation.

Note that bits 29 and 28 in the transmit descriptor TD0 should be set to 11. Otherwise, correct operation cannot be guaranteed.

During reception, after having received a frame, this logic core fetches the next receive descriptor. This logic core determines that the reception is ready if the RACT bit in the fetched receive descriptor is 1 and stores the received frame. If the RACT bit in the fetched receive descriptor is 0, this logic core determines that the receive descriptors have been depleted and completes the receive operation.

If this logic core receives a frame longer than the buffer length specified in the receive descriptor, it stores the frame up to the specified buffer length and then fetches the next receive descriptor. If this logic core further receives a frame longer than the buffer length specified in the fetched receive descriptor, it stores the frame up to the specified buffer length and then fetches the following descriptor. If this logic core receives a frame whose frame length is within the buffer size specified by the fetched receive descriptor, it stores the last byte of the frame and writes back the number of bytes so far stored in the receive buffer to the receive frame length (RFL) in the receive descriptor. The logic core then fetches the next descriptor to prepare for the next frame reception.

Note that correct operation cannot be guaranteed if the receive buffer length is specified as 0 in the receive descriptor.

(4) DMA Operation

(a) Burst Operation

This logic core accesses the descriptors and buffers in longword units via DMA transfer.

During descriptor (16 bytes as standard) fetch, 16-byte DMA transfer is performed per bus acquisition.

During transmit descriptor write back, 4-byte DMA transfer is performed per bus acquisition.

During receive descriptor write back, 8-byte DMA transfer is performed per bus acquisition.

During buffer data transfer, a maximum of 32-byte DMA transfer is performed per bus acquisition.

If the remaining data in a frame to be transferred is 17 to 32 bytes, 32-byte DMA transfer is performed per bus acquisition.

If the remaining data in a frame to be transferred is 9 to 16 bytes, 16-byte DMA transfer is performed per bus acquisition.

If the remaining data in a frame to be transferred is 5 to 8 bytes, 8-byte DMA transfer is performed per bus acquisition.

If the remaining data in a frame to be transferred is 1 to 4 bytes, 4-byte DMA transfer is performed per bus acquisition.

(b) DMA Operation Error Report

If a DMA error is detected during DMA operation, the BER bit in the status register (CXR5) is set to 1 to generate an interrupt. After interrupt, DMA operation should be restarted by a reset (system reset or software reset).

(5) Transmit and Receive Buffers

A transmit buffer address and transmit buffer length to be specified by a transmit descriptor can be set in byte units. However, if 1 to 16 bytes is specified as the transmit buffer length, the transmit buffer address should be aligned on a 32-byte boundary. Note that correct operation cannot be guaranteed if 0 byte is specified as the transfer buffer length.

A receive buffer address to be specified by a receive descriptor should be aligned on a 32-byte boundary to prevent the DMA burst operation from being performed beyond SDRAM boundary.

In addition, if the number of data bytes remaining in a frame to be transferred cannot be divided by DMA burst bytes (32 bytes), the receive buffer length to be specified by the receive descriptor should be specified on a 32-byte boundary since extra data is transmitted.

For example, if the number of last data bytes remaining in a receive frame is 17 bytes, invalid data is written in the 18th to 32nd bytes in the corresponding address of the receive buffer since 32-byte DMA burst transfer is performed.

(6) Endian

Little endian can be used for the data arrangement in the transmit and receive buffers according to the DE bit setting in the operating mode setting register (CXR0). (The initial value is big endian.)

Note that only the endian of the data in the transmit and receive buffers can be changed. Endian of descriptors and registers cannot be changed.

(7) Register Access

Registers are read- and write-accessed via the SuperHyway. The registers in the function core connected to the HDMAC are accessed via the HDMAC and SuperHyway.

(8) Transmit Operation

(a) Normal Transmit Operation

This logic core starts the transmit processing when 1 is written to the TRNS bit in the transmit activation register (CXR1).

This logic core fetches a descriptor next to the previous descriptor from the transmit descriptor ring.

If the TACT bit in the fetched transmit descriptor is 1, data in the transmit buffer is read out and written in the transmit FIFO sequentially according to the specified transmit buffer address and the number of bytes.

If the TACT bit in the fetched transmit descriptor is 0, the TRNS bit in the transmit activation register (CXR1) is cleared to 0 and transmit processing is completed without any operation.

When data of the buffer length indicated by the descriptor has been transmitted to the function core via the transmit FIFO (normally completed or aborted), the transmit descriptor is written back to, and the next descriptor is fetched.

While the TACT bit in the fetched transmit descriptor is 1, the transmit descriptor fetch and DMA transfer are performed continuously.

If the TACT bit in the fetched transmit descriptor is 0, the FTC bit in the status register (CXR5) is set to 1 to indicate the completion of transmission thus requesting an interrupt to the CPU, and simultaneously the TRNS bit in the transmit activation register (CXR1) is cleared to 0 to stop transmit processing.

If data in the transmit FIFO exceeds the threshold (changes according to the register setting), if data of a frame has been stored, or if the transmit FIFO becomes full, the function core is activated and data in the transmit FIFO is transmitted to the function core synchronously with the RDY signal in the function core.

When a transmit completion signal or abort signal is input from the function core, the above transmit operation is ended.

(b) Continuous Frame Transmit Operation

Multiple frames can be specified in a transmit descriptor.

This logic core clears the TACT bit in the transmit descriptor to 0 when data transmission indicated in the transmit descriptor has been completed, and fetches the next transmit descriptor.

While the TACT bit in the fetched descriptor is 1, the frame transmission is performed continuously. When the TACT bit in the transmit descriptor is 0, the FTC bit in the status register (CXR5) is set to 1 to indicate the transmission completion thus requesting an interrupt to the CPU, and the TRNS bit in the transmit activation register (CXR1) is cleared to 0 to stop transmit processing.

Generation of a transmit completion interrupt (the FTC bit in CXR1 is set to 1) informs the CPU that all the specified frames have been transmitted.

(c) Transmit Abort

If this logic core receives an abort signal (ATABT) from the function core during frame transmission, it aborts data transmission of the corresponding frame from transmit FIFO to the function core.

When the transmission is aborted, this logic core clears the TACT bit in the transmit descriptor to 0, sets bit 8 in the transmit frame status (TFS) to 1.

The following operations are the same as (1) Normal Transmit Operation or (2) Continuous Frame Transmit Operation described above.

When detected the TACT bit in the fetched transmit descriptor is 0, the FTC bit in the status register (CXR5) is set to 1 to indicate the transmission completion, and TABT bit is also set to 1 to indicate the transmission abort to requesting an interrupt to the CPU.

(d) Transmit FIFO Underflow

This logic core does not transmit data to the function core until data in the transmit FIFO exceeds the specified threshold (changes according to the register setting), data of a frame is stored in the transmit FIFO, or the transmit FIFO becomes full to prevent the transmit FIFO from underflow.

An underflow will occur if the transmit FIFO becomes empty (there is no data to be transmitted) because the bus mastership cannot be obtained.

In this case, the TFE bit in the status register (CXR5) is set to 1 to indicate transmit FIFO underflow and an interrupt is requested to the CPU.

The frame where an underflow occurred is transmitted to the function core in the same way as the normal frame. After the frame transmission, the TACT bit in the transmit descriptor is cleared to 0 and transmit frame status (TFS) is set to 1. The following operations are the same as (1) Normal Transmit Operation or (2) Continuous Frame Transmit Operation described above.

(e) Transmit Frame Retry

If this logic core receives an ATRTRY signal from the function core after having transmitted the last byte of a frame to the function core, this logic core fetches the start descriptor of the current frame again, and re-transmits the data of the current frame from the start to last bytes to the transmit FIFO.

(9) Receive Operation

(a) Normal Receive Operation

This logic core starts receive processing when 1 is written to the R bit in the receive activation register (CXR2).

This logic core fetches the receive descriptor (Nth receive descriptor) next to the previous receive descriptor ((N – 1)th receive descriptor) from the receive descriptor ring, and enters the receive wait state.

In the receive wait state, if the receive FIFO contains data of 32 bytes or more or the last byte, data is transferred from the receive FIFO to the receive buffer.

When the receive buffer specified by the receive descriptor becomes full, this logic core writes 0 to the RACT bit in the receive descriptor and fetches the (N + 1)th receive descriptor. If the RACT bit in the (N + 1)th receive descriptor is 1, this logic core enters the receive wait state again.

If the last byte of a frame has been transferred, this logic core writes 0 and 1 in the RACT bit and the RFP[0] bit in the receive descriptor, respectively, sets the FRC bit in the status register (CXR5) to 1 to indicate the completion of frame reception, requests an interrupt to the CPU, and finally fetches the (N + 2)th receive descriptor.

If the RACT bit in the (N + 2)th receive descriptor is 1, this logic core enters the receive wait state. Contrarily, if the RACT bit in the (N + 2)th receive descriptor is 0, a receive descriptor depletion occurs. (For details, refer to section 22.8.1 (8) (g), Receive Descriptor Depletion.)

(b) Continuous Frame Receive Operation

As described in (1) Normal Receive Operation, when a frame has been received and the last byte of the frame has been transferred, this logic core writes 0 and 1 in the RACT bit and RFP[0] bit in the Nth receive descriptor, respectively, sets the FRC bit in the status register (CXR5) to 1 to indicate the completion of frame reception, generates an interrupt to the CPU, fetches the (N + 1)th receive descriptor, and enters the receive wait state.

When the last byte in the (N + 1)th receive descriptor has been transferred, this logic core writes 0 and 1 in the RACT bit and RFP[0] bit in the (N + 1)th receive descriptor, respectively, sets the FRC bit in the status register (CXR5) to 1 to indicate the completion of frame reception, generates an interrupt to the CPU, fetches the (N + 2)th receive descriptor, and enters the receive wait state.

This logic core then receives frames continuously until the RACT bit in the fetched descriptor is cleared to 0 (a receive descriptor depletion occurs).

(c) Receive Operation of Multiple Frames and Multiple Buffers

If the byte count of the received Mth frame is greater than the buffer length specified by the Nth receive descriptor, this logic core transfers data of the specified receive buffer length and then writes 0s in the RACT bit and RFP[0] bit in the Nth receive descriptor and fetches the (N + 1)th receive descriptor.

With the (N + 1)th receive descriptor, this logic core performs the data transfer of the same frame (Mth frame). If the byte count of the received Mth frame is greater than the buffer length specified by the (N + 1)th receive descriptor, this logic core transfers data of the specified receive buffer length and writes 0s in the RACT bit and RFP[0] bit in the (N + 1)th receive descriptor and fetches the (N + 2)th receive descriptor as described above. Contrarily, if the byte count of the received Mth frame is smaller than the buffer length specified by the (N + 1)th receive descriptor, this logic core, after having transferred the last byte, writes 0 and 1 in the RACT bit and RFP[0] bit in the (N + 1)th receive descriptor, sets the FRC bit in the status register (CRX5) to 1 to indicate the completion of frame reception, generates an interrupt to the CPU, fetches the (N + 2)th receive descriptor, and enters the receive wait state, as described in (1) Normal Receive Operation.

(d) Receive Abort

If this logic core receives a receive abort (ARABT) signal from the function core during frame reception, it aborts reception of the corresponding frame data from the function core.

This logic core transfers data stored in the receive FIFO before receive abortion to the receive buffer. When the last byte has been transferred, this logic core clears the RACT bit in the receive descriptor to 0, sets bit 8 of receive frame status in the receive descriptor to 1, sets the FRC and RABT bits in the status register (CRX5) to 1s to indicate the frame transmission completion and receive abort detection, respectively, and finally generates an interrupt to the CPU.

If this logic core receives an ARABT signal from the function core while the number of data bytes of the stored frame in the receive FIFO is less than 16 bytes, this logic core deletes the corresponding frame.

(e) Receive FIFO Overflow

If the receive FIFO becomes full because the bus mastership cannot be obtained, an overflow error occurs because the next data cannot be stored in the receive FIFO.

If an overflow error occurs, the receive FIFO terminates the data reception and sets the RFE bit in the status register (CXR5) to 1 to indicate the receive FIFO overflow, and generates an interrupt to the CPU. When the transfer of the byte immediately before the overflow has been completed, this logic core writes 0 and 1 to the RACT bit and bit 9 of receive frame status (RFS) in the receive descriptor, respectively, sets the FRC and RFE bits in the status register (CXR5) to 1s to indicate the frame reception completion and receive FIFO overflow, respectively, and finally generates an interrupt to the CPU.

While the receive FIFO is full, the next frame is received but discarded. Simultaneously, the discarded frame counter register (CXR8) is incremented.

If the receive FIFO becomes not full after data has been transferred from the receive FIFO to the receive buffer, this logic core restarts the reception from the start byte of the next receive frame.

The number of frames that can be simultaneously stored in the receive FIFO is limited to 16 frames due to the frame management restriction. Accordingly, this logic core cannot receive a frame if 16 frames have been stored in the receive FIFO. If this logic core receives the start data of the 17th frame, it sets the RFRMER bit in the status register (CXR5) to 1 to indicate the receive frame count overflow, and generates an interrupt to the CPU.

This logic core receives the next frame but discards it, and simultaneously increments the discarded frame counter register (CXR8).

If a space of 1 frame or more is produced in the receive FIFO after data has been transferred from the receive FIFO to the receive buffer, this logic core restarts the reception from the start byte of the next receive frame.

(f) Flow Control Support

This function core provides an ARBSY signal to inform the function core of the receive FIFO status. It can output the ARBSY signal based on the data amount stored in the receive FIFO according to the setting of the receive FIFO busy transmit threshold setting register (CXR16).

Accordingly, the flow can be controlled by sending "BUSY" to the function core before the receive FIFO overflows.

The BUSY status can be cancelled when the data byte count or the frame count in the receive FIFO become (the byte count value causing a transition to the BUSY state − 32 bytes) or smaller

and (the frame count value causing a transition to the BUSY state – 1 frame) or smaller, respectively.

(g) Receive Descriptor Depletion

When this logic core has transferred data of the Mth frame to the receive buffer specified by the Nth receive descriptor, it fetches the (N + 1)th receive descriptor to prepare for the reception of the (M + 1)th frame.

If the RACT bit of the fetched descriptor ((N + 1)th descriptor) is 0, this logic core regards it as receive descriptor depletion, sets the RDE bit in the status register (CXR5) to 1 to indicate the receive descriptor depletion, and generates an interrupt to the CPU.

Simultaneously, this logic core clears the RR bit in the receive activation register (CXR2) to 0 and terminates the receive processing. In this case, data transfer from the function core to the receive FIFO continues.

In this situation, the CPU should clear the interrupt source and reset the receive descriptor correctly.

If the valid receive descriptor is set before a receive FIFO overflow occurs, a frame reception can be continued without data loss.

22.8.2 Transmit Descriptors

Figure 22.14 shows a transmit descriptor format.

If 1 to 16 bytes is specified as the transmit buffer length, the transmit buffer address should be aligned on a 32-byte boundary. Note that correct operation cannot be guaranteed if 0 byte is specified as the transfer buffer length.

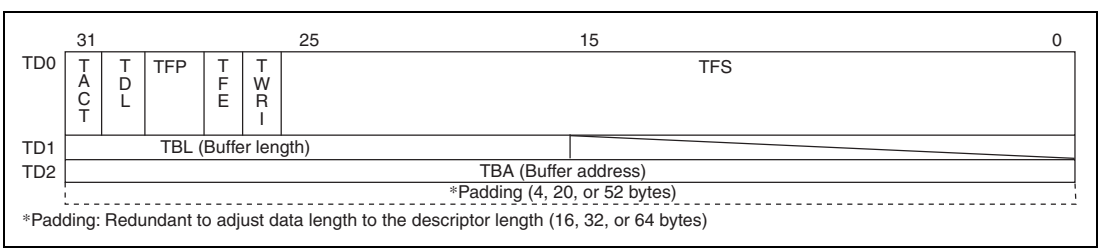


Figure 22.14 Transmit Descriptor Format

(1) Transmit Descriptor 0 (TD0)

Table 22.9 shows a definition of transmit descriptor 0. (A bit to be written back to is underlined.)

Table 22.9 Definition of Transmit Descriptor 0

Bit	Bit Name	Initial Value	R/W	Definition
31	TACT	Undefined	R/W	Descriptor Valid Indicates that the corresponding descriptor is valid. This bit is set to 1 by a driver, and reset (cleared to 0) by this logic core when a frame transmission has been completed or a frame transmission has been aborted for some reason.
30	TDL	Undefined	R/W	Descriptor Ring End Set to 1 to indicate that the corresponding descriptor is an end of the transmit descriptor ring.
29, 28	TFP	Undefined	R/W	Position in a Frame 11: Indicate that the information in this descriptor includes one whole frame. This field should be set to 11.
27	TFE	Undefined	R/W	Transmit Frame Error Set to 1 to indicate that an error occurs in TFS. (For bits 7 to 0 in TFS, this bit can be prevented from being set to 1 by the CXR7 register. Note, however, that this bit cannot be prevented from being set if the sources corresponding to bits 7 to 0 in TFS are used to set bit 8 in TFS.)
26	TWBI	Undefined	R/W	Write-Back Completion Interrupt Request (Enabled according to CXR18 settings) 0: nop 1: Requests an interrupt after a write-back to this descriptor has been completed.
25 to 0	TFS	Undefined	R/W	Transmit Frame Status Bit 8: Set to 1 to indicate that the underflow occurs shown as an abort signal is set to 1 during transmission of the frame, (TFE set source). Bits 7 to 0: Set to 1 to indicate that a corresponding bit among TINT8 to TINT1 is set to 1 (TFE setting sources specified by CXR7).

(2) Transmit Descriptor 1 (TD1)

Table 22.10 shows a definition of transmit descriptor 1.

Table 22.10 Definition of Transmit Descriptor 1

Bit	Bit Name	Initial Value	R/W	Definition
31 to 16	TBL	Undefined	R/W	Buffer length: Indicates the valid byte count in the target transmit buffer.
15 to 0	—	Undefined	R/W	Reserved

(3) Transmit Descriptor 2 (TD2)

Table 22.11 shows a definition of transmit descriptor 2.

Table 22.11 Definition of Transmit Descriptor 2

Bit	Bit Name	Initial Value	R/W	Definition
31 to 0	TBA	Undefined	R/W	Buffer address: Indicates the start address of the transmit buffer.

22.8.3 Receive Descriptors

Figure 22.15 shows a receive descriptor format.

A receive buffer address specified by a receive descriptor should be aligned on a 32-byte boundary. Note that correct operation cannot be guaranteed if 0 byte is specified as the receive buffer length (RBL).

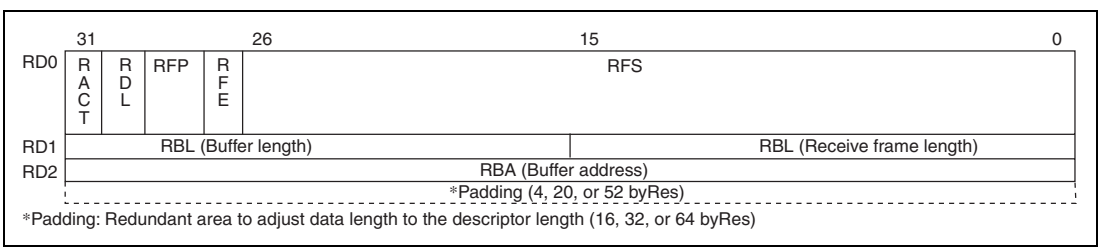


Figure 22.15 Receive Descriptor Format

(1) Receive Descriptor 0 (RD0)

Table 22.12 shows a definition of receive descriptor 0. (A bit to be written back to is underlined.)

Table 22.12 Definition of Receive Descriptor 0

Bit	Bit Name	Initial Value	R/W	Definition
31	RACT	Undefined	R/W	Descriptor Valid Indicates that the corresponding descriptor is valid. This bit is set to 1 by a driver, and reset (cleared to 0) by this logic core when a frame transmission has been completed or the receive buffers have become full.
30	RDL	Undefined	R/W	Descriptor Ring End Set to 1 to indicate that the corresponding descriptor is an end of the receive descriptor ring.
29, 28	RFP	Undefined	R/W	Position in a Frame 10: Indicate that the information in this descriptor includes the start of a frame. 11: Indicate that the information in this descriptor includes one whole frame. 01: indicate that the information in this descriptor includes the end of a frame. 00: Indicate that the information in this descriptor is other than above.
27	RFE	Undefined	R/W	Receive Frame Error Set to 1 to indicate that an error occurs in RFS. (For bits 7 to 0 in RFS, this bit can be prevented from being set to 1 by the CXR7 register. Note, however, that this bit cannot be prevented from being set if the sources corresponding to bits 7 to 0 in TFS are used to set bit 8 in RFS.)
26 to 0	RFS	Undefined	R/W	Receive Frame Status Bit 9: Set to 1 to indicate that an aborted frame is written back due to a receive FIFO overflow (FRE setting source). Bit 8: Set to 1 to indicate that an abort signal is set to 1 during frame reception (RFE set source). Bits 7 to 0: Set to 1 to indicate that a corresponding bit among RINT8 to TINT1 is set to 1 (RFE setting sources specified by CXR7).

(2) Receive Descriptor 1 (RD1)

Table 22.13 shows a definition of receive descriptor 1. (A bit to be written back to is underlined.)

Table 22.13 Definition of Receive Descriptor 1

Bit	Bit Name	Initial Value	R/W	Definition
31to 16	RBL	Undefined	R/W	Buffer length Indicate the byte count in the target receive buffer. The buffer length should be specified by 32 bytes x n.
15 to 0	RFL	Undefined	R/W	Receive Frame Length Indicate the receive frame length (bytes) stores in the receive buffer. This receive frame length does not include the number of bytes for padding specified in the CXRS5. This receive frame length is written back in the receive descriptor that includes the last of a frame.

(3) Receive Descriptor 1 (RD2)

Table 22.14 shows a definition of receive descriptor 2.

Table 22.14 Definition of Receive Descriptor 1

Bit	Bit Name	Initial Value	R/W	Definition
31 to 0	RBA	Undefined	R/W	Buffer Address These bits indicate the start address of a receive buffer. The buffer address should be aligned on a 32-byte boundary.

22.8.4 Error Detection and Report

The error sources are classified into two types: those detected by the HDMAC and those detected and reported by the feLic.

Each error is described below. If an error is detected, the detected error is indicated in the status register (CXR5) and an interrupt signal is output.

(1) Error Sources Detected by the HDMAC

(a) Descriptor Depletion

If no descriptor whose descriptor valid bit is set can be detected during frame reception, a receive descriptor depletion error occurs and the R bit in the receive activation register (CXR2) is reset to complete the receive processing.

(b) External FIFO Underflow/Overflow

If a transmit FIFO becomes empty during frame transmission, an underflow error occurs because there is no data to be transmitted.

If a receive FIFO becomes full during frame reception, an overflow error occurs because the next receive data cannot be stored in the receive FIFO.

If an underflow error in a transmit FIFO or an overflow error in a receive FIFO is detected, it is indicated in the status register (CXR5).

(2) Error Sources Detected and Reported by the Function Core (feLic)

(a) Transmit Interrupt

Interrupt sources detected by the function core (feLic) during frame transmission are reported to the transmit descriptor of the corresponding frame and status register (CXR5) as TINT1 to TINT8, or transmit abort detect (TABT).

(b) Receive Interrupt

Interrupt sources detected by the function core (feLic) during frame reception are reported to the receive descriptor of the corresponding frame and status register (CXR5) as RINT1 to RINT8, and receive abort detect (RABT).

(c) MPORT Interrupt (MINT)

If an error is detected other than during transmission or reception, it is reported as an MPORT interrupt.

For details on TINT, RINT, and MINT bits, see the Function Core Function Specifications.

(3) Error Log Information

This logic core provides the following log collection registers.

1. Discarded frame counter register (CXR8)

2. Transmit FIFO underrun counter register (CXR13)
3. Receive FIFO overflow counter register (CXR14)

22.8.5 Firmware/Software Interface

(1) Interrupts

Interrupts sources in this logic module are as follows.

1. RINT1 to RINT8
2. TINT1 to TINT8
3. Receive FIFO overflow
4. Receive descriptor depletion
5. Frame receive completion
6. Transmit FIFO underflow
7. Transmit descriptor depletion (not used)
8. Frame transmit completion
9. M port interrupt
10. DMA error
11. Receive frame count overflow
12. Receive abort detection
13. Transmit abort detection
14. External EXIN signal assertion detection (not used)
15. Transmit descriptor write-back

The above interrupt sources are indicated in the status register (CXR5). An interrupt source (other than M port interrupt signal (MINT) bit) can be reset by writing 1 to the corresponding bit in the CXR5 register. An M port interrupt signal (MINT) can be reset by resetting the corresponding source in the function core. The above interrupt sources can be separately masked by setting the interrupt mask register (CXR6).

22.9 feLic Function Specifications

22.9.1 Configuration

Figure 22.16 shows a block diagram of feLic.

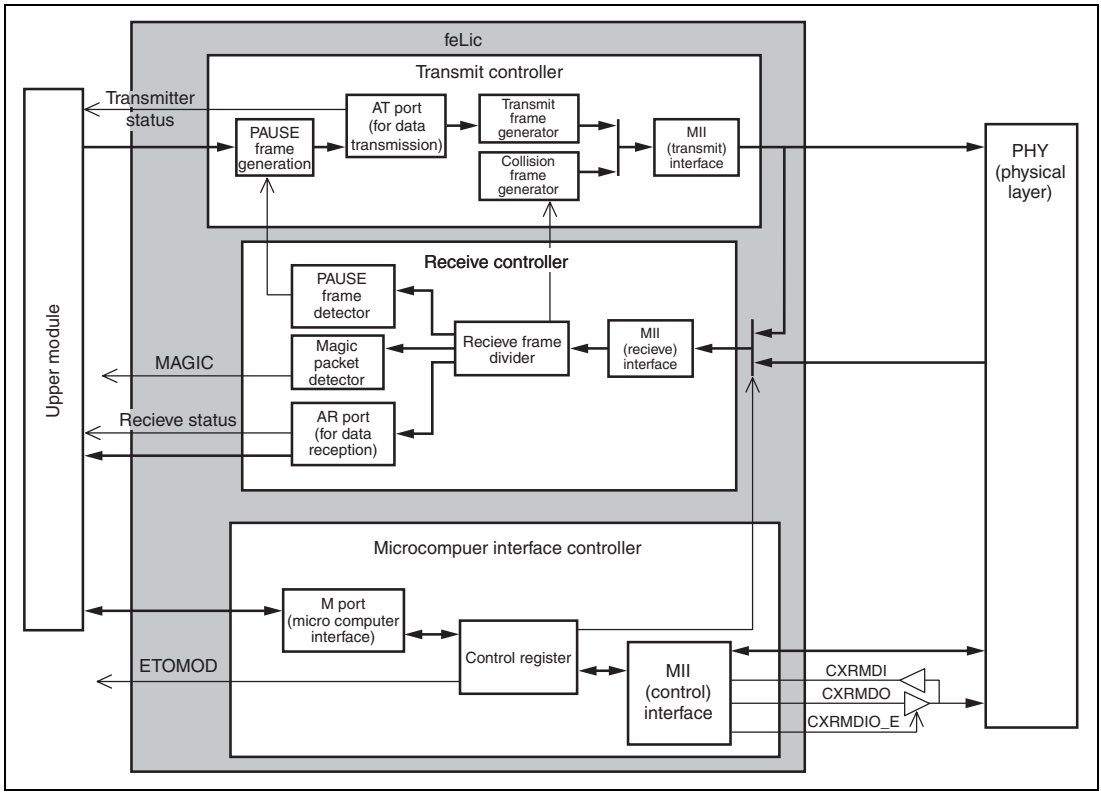


Figure 22.16 feLic Block Diagram

22.9.2 feLic Function

(1) Transmit Controller

The transmit controller assembles data received from AT port into the IEEE802.3 format frames, and transmits them via the MII interface.

The main functions of the transmit controller are listed below.

1. Assembles data received from the AT port into the IEEE802.3 format frames and transmits them.
2. Performs CRC calculation and adds it to the frame.
3. Retransmits frames (up to 15 times) when a collision occurs.
4. Provides the MII interface function conforming to the IEEE802.3u.
5. Performs serial and parallel conversion according to the PHY chip speed.
6. Provides transmit frame data padding function.
7. Generates a PAUSE frame.

(2) Receive Controller

Upon receiving a frame via the MII interface, the receive controller checks the frame address to see if the address matches the self-node; it also checks CRC and the frame length of the data and transfers the data to the AR port.

The main functions of the receive controller are listed below.

1. Checks the format of the received frame header.
2. Checks the CRC and frame length of the received frame data.
3. Transfers frames addressed to itself, multicast frames, and broadcast frames to the AR port.
4. Provides the MII interface function conforming to the IEEE802.3u.
5. Performs serial and parallel conversion according to the PHY chip speed.
6. Monitors a magic packet.
7. Analyzes a PAUSE frame.

(3) Microcomputer Interface Controller

The microcomputer interface controller incorporates a function to interface to the upper microprocessor. The data width of the M port is 32 bits. The M port operates synchronously with the external clock. The upper microprocessor controls the registers in this microcomputer interface controller block and the registers in PHY via the MII interface.

The main functions of the microcomputer interface controller are listed below.

1. Controls the operation of transmitter and receiver.
2. Accesses the PHY registers via the serial bus.

(4) Magic Packet Monitor Function

The receive controller monitors the magic packet and report the magic packet detection to an external device in magic packet monitor mode. This function becomes effective by the MPM bit in the CXR20 register. When the magic packet is detected, "eth magic" signal that is an output is asserted and the CXR21 register is updated. Their detecting functions are explained as follows.

(a) Assertion by the eth magic signal

Eth magic signal that is an output signal of the feLic is asserted synchronizing with the rising edge of the received clock from PHY. Detection of the magic packet can be informed to external circuits by using this signal. The eth magic signal is negated by the hardware reset and the software reset of the HDMAC/feLic using CXR0 register.

(b) Assertion by the CXR21 register

Updating CXR21 can assert as software interruption. The detecting information is cleared by writing a 1 to the corresponding bit. Note that if a magic packet is received again, CXR21 register is not updated while the eth magic signal is asserting. Set the feLic again after clearing by the procedure described (1).

(5) Transmit and Receive Status Statistical Function

The interrupt source signals (TINT/RINT) to be output to the upper modules can be counted and referenced as statistical information by accessing a register via the M port.

The interrupt source signal counting starts after reset (includes a soft reset by CXR0 register) and the count value can be cleared only by a write access.

(6) Flow Control Conforming to IEEE802.3

As the flow control during full-duplex operation, the flow control conforming to IEEE802.3x is supported. Transmission of a PAUSE frame used in flow control is performed in the following methods. These methods can be used by mutually combining.

(a) Automatic PAUSE Frame Transmission

A PAUSE frame is transmitted automatically by asserting the ARBSY signal. Note that as the Timer value included in the PAUSE frame, the value set in the parameter setting register (see CX71) is used. If the ARBSY signal is not negated when the time indicated by the Timer value has passed after transmitting a PAUSE frame, a PAUSE frame is transmitted again. The number of PAUSE frame re-transmissions (upper limit) can be specified from 1 to 65535 times by setting the PAUSE frame retransmit count setting register (CXR81). If the number of PAUSE frame re-

transmissions reaches the upper limit, the following PAUSE frame re-transmissions will not be performed.

Transmission can be restarted when ARBSY signal is negated once and then asserted again thus resetting the PAUSE frame re-transmission counter (CXR82). In addition, it is possible not to set the upper limit of the re-transmissions (unlimited).

(b) Manual PAUSE Frame Transmission

This method can send the PAUSE frame by calling of software. Set the Timer value to the CXR72 register to sending the PAUSE frame. It is sent just one time (one frame) in this method.

(c) PAUSE Timer Value

For the PAUSE frame whose Timer value is 0, it is possible to enable or disable the PAUSE frame control.

- During transmission

  When control of a PAUSE frame whose Timer value is 0 is enabled, a PAUSE frame with Timer value 0 is transmitted if the ARBSY signal is negated before the time indicated by the Timer value has passed.

  When control of a PAUSE frame whose Timer value is 0 is disabled, the next frame will not be transmitted until the time indicated by the Timer value has passed.
- During reception

  When control of a PAUSE frame whose Timer value is 0 is enabled, a PAUSE frame is set in the receive wait counter even if the time indicated by Timer value is 0. (The receive wait state is cancelled.)

  When control of a PAUSE frame whose Timer value is 0 is disabled, the PAUSE frame is discarded if the time indicated by Timer value is 0.

(d) PAUSE Frame Reception

When a PAUSE frame is received, this logic function suspends transmission of the next frame until the time indicated by the Timer value has passed. However, transmission of the frame being transmitted continues. A number of the PAUSE frame receiving is counted to CXR80.

(7) Pack-Pressure Flow Control

As the flow control during half-duplex operation, the back-pressure flow control is supported. This method generates a pseudo collision (transmits a collision frame) to stop the frame reception if the remaining space in the FIFO is not sufficient when the frame is received. This method dose not refer to the DA.

22.9.3 Detailed Description of Transmit and Receive Controllers

(1) Transmit Controller

The transmit controller assembles the transmit data from the AT port into the frames based on the appropriate format and transmits them via the MII interface. Here, an error may occur according to the status of the communication lines. If an error of TINT1 to RINT8 occurs, the transmission is aborted and the error source is reported to the upper block using the abort signal (ARABT). If the transmission is completed normally, it is reported via the transmit completion signal (ATCOMP).

The conditions of generating each error source and operation are described below.

- TINT1: Transmit timeout
  When a collision occurs (the COL signal assertion) during frame transmission, the transmission is retried after the back-off time has passed. A transmit timeout is generated if a collision occurs again during the 15th transmission retry.
- TINT2: Collision detection during frame transmission
  Generated if a delay collision described in the IEEE802.3 standard occurs.
  Description in the standard: It is regarded as a delay collision if a collision occurs after the collision window at 512-bit time from the transmission start.
- TINT3: Carrier loss during frame transmission
  Generated when a carrier is lost (the CRS signal is negated) during frame transmission.
- TINT4: Carrier not detected
  Generated when a carrier is not detected (the CRS signal is not asserted) during preamble field transmission.
- TINT5: Not assigned
- TINT6: Not assigned
- TINT7: Not assigned
- TINT8: Not assigned

If any transmit error described above occurs, the feLic enters the wait state for the next frame transmission request and starts the next frame transmission when the next frame transmission is requested.

(2) Receive Controller

The receive controller transfers a frame received via the MII interface to the AR port. Here, an error may occur according to the status of the communication lines. If an error of RINT1 to RINT4 occurs, the reception is aborted and the error source is reported to the upper block using the abort signal (ARABT). RINT5 indicates that a fractional number bit error has occurred. Note,

however, that a frame with an RINT5 error but with no CRC error can be received normally. A frame with RINT8 can also be received normally. RINT8 is an error source allowing normal frame reception and is reported by using the receive completion signal (ARCOMP) when the frame destination address is a multicast address.

The conditions of generating each error source and operation are described below.

- RINT1: CRC error

  Generated when an error is detected in the FCS field in the receive frame.
- RINT2: Frame receive error

  Reported when the RX_ER signal in the MII interface is asserted during frame reception. For details on the RX_ER signal assertion, see the Data Book of the PHY to be connected.
- RINT3: Frame length error

  Reported when the receive frame length is less than 64 bytes.
- RINT4: Frame length error

  Reported when the receive frame length exceeds the value specified in the CXR2A register. An excess of data field over the specified frame length is discarded in the feLic and will not be transferred to the upper block.
- RINT5: Fractional number bit error

  Reported when the received frame length is not the multiple of an octet. In this case, the last 1 to 4 octets in the data field will not be transferred to the upper block.
- RINT6: Not assigned
- RINT7: Not assigned
- RINT8: Multicast frame reception

  Reported when the frame destination address is a multicast address. This error source is reported with a receive completion signal only when other error source is not generated.

If any receive error described above occurs, the feLic enters the wait state for the next frame reception and starts the next frame reception when the next frame is received.

Section 23 Display Unit (DU)

23.1 Overview

23.1.1 Features

Plane: The display surfaces normally called the foreground, background, and cursor, are called planes in this document. The maximum display resolution of planes is WXGA (1280 × 768). Parameters for each plane can be set independently through the settings of an internal register. The internal register settings can also be used to set the display priority order. Combined display of up to four planes is possible when the plane size is WVGA (832 × 496).

- Display size
- Display position
- Display data format (8 bits/pixel, 16 bits/pixel, ARGB, or YC)
- Plane superpositioning
- Scrolling
- Wrapping-around
- Blinking
- Buffering control

The internal register settings can be used to select four different control modes.

- Auto rendering mode (double buffering)
- Manual display change mode (double buffering)
- Auto display change mode (double buffering)
- Video capture mode (triple buffering)

Alpha-Ratio Plane: There are four display planes which are used for display data. Each of these can have an alpha value. There is also an alpha-ratio plane, which is exclusively used for alpha values.

Synchronization Method: Internal register settings can be used to select any of three synchronization modes for the display output timing.

- Master mode (internal sync mode)
- TV sync mode (external sync mode)
- Sync method switching mode

CRT Scan Mode (CRT Scan Method): Internal register settings can be used to select from among three scan modes.

- Non-interlaced mode
- Interlaced sync mode
- Interlaced sync & video mode

YC→RGB Colorspace Conversion Functions: Image data stored in YC format can be converted into the RGB colorspace and displayed in a window.
(However, two or more pixels cannot be converted into the RGB colorspace at the same time.)

Color Palette: An internal color palette plane is provided, capable of simultaneously displaying 256 colors out of a possible 260,000 colors.

Eight-bit blending ratios are provided for every 256 colors.

Display Capture: The RGB-666 digital data for output to the pins is convertible into RGB-565 data for storage in external memory.

Register Access Control: The module has internal control registers; these are accessible by the HPB protocol over the HPB. The access size is fixed as 32 bits.

23.1.2 Block Diagram

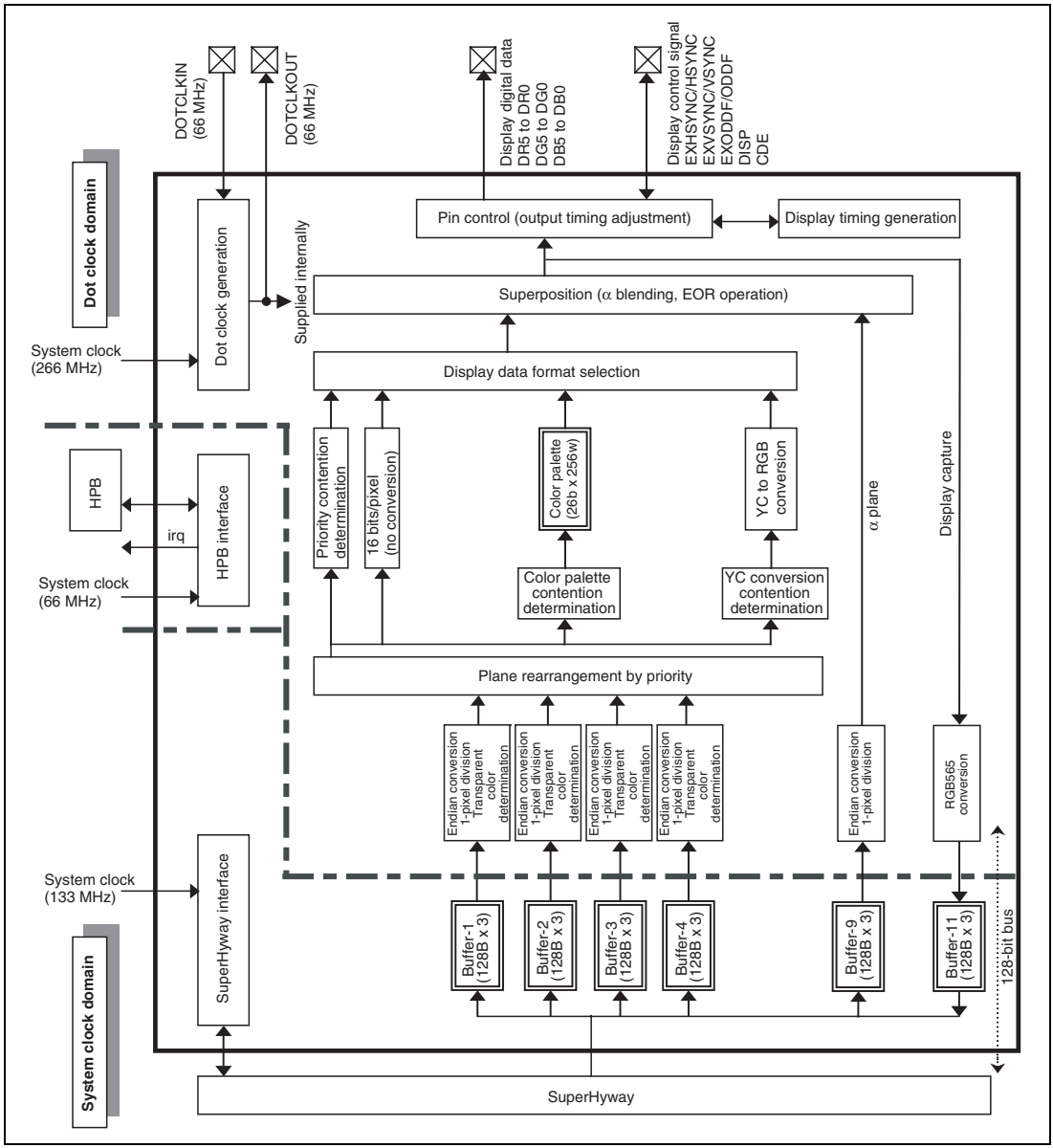


Figure 23.1 Block Diagram of the Display Unit (DU)

23.1.3 External Pins

Table 23.1 Pin Configuration

Name	Pin Name	I/O	Function	Signal Name Used in This Document
Input dot clock	DOTCLKIN	I/O	Input dot clock	DCLKIN
Output dot clock	DOTCLKOUT	Output	Output dot clock	DCLKOUT
Horizontal synchronous output/ external horizontal synchronous input	EXHSYNC/ HSYNC/ CSYNC	I/O	Composite synchronous output signal (Initial value)	CSYNC
			Horizontal synchronous output/ External horizontal synchronous input	HSYNC (output)/ EXHSYNC (input)
Vertical synchronous output/ external vertical synchronous input	EXVSYNC/ VSYNC/ CSYNC	I/O	Vertical synchronous output/ External vertical synchronous input (Initial value)	VSYNC (output)/ EXVSYNC (input)
			Composite synchronous output signal	CSYNC
Odd/even field	EXODDF/ ODDF/ CSYNC	I/O	Odd/even field (Initial value)	ODDF
			CLAMP output signal	CLAMP
			Composite synchronous output signal	CSYNC
Display interval	DISP/ CSYNC	Output	Display interval (Initial value)	DISP
			Composite synchronous output signal	CSYNC
			DE output signal	DE
Color detection	CDE	Output	Color detection	CDE
Display data	DR0	Output	Digital red 0	Digital RGB
	DR1	Output	Digital red 1	
	DR2	Output	Digital red 2	
	DR3	Output	Digital red 3	
	DR4	Output	Digital red 4	
	DR5	Output	Digital red 5	

Name	Pin Name	I/O	Function	Signal Name Used in This Document
Display data	DG0	Output	Digital green 0	Digital RGB
	DG1	Output	Digital green 1	
	DG2	Output	Digital green 2	
	DG3	Output	Digital green 3	
	DG4	Output	Digital green 4	
	DG5	Output	Digital green 5	
	DB0	Output	Digital blue 0	
	DB1	Output	Digital blue 1	
	DB2	Output	Digital blue 2	
	DB3	Output	Digital blue 3	
	DB4	Output	Digital blue 4	
	DB5	Output	Digital blue 5	

Note: In this document, unless otherwise noted, "dot clock" refers to the output dot clock.

23.1.4 Register Configuration

In the display unit (DU), register update methods include external update and internal update.

(1) External Update

An "external update" is an update which reflects the address-mapped register settings made by the CPU after the end of CPU access. Registers related to display control (for example, the display unit system control register) and the settings of which are updated through external updates can be overwritten without display flicker by using the VBK flag and FRM flag in the display status register (DSSR) indicating the start position of the vertical blanking interval.

(2) Internal Update

An "internal update" is an update which reflects the address-mapped register settings with the internal update timing of the display unit (DU). Hence in the case of a register with an internal update function, even when the CPU overwrites address-mapped registers related to display operation without being aware of the display timing, display flicker can be prevented.

An internal update is performed during the interval in which the display reset (DRES) bit in the display unit system control register (DSYSR) is 1 and at the beginning of each frame. The internal update performed at the beginning of each frame is disabled using the internal update disable (IUPD) bit in the display unit system control register (DSYSR).

Internal updates are performed on the following bits by setting them to the display reset (DRES) bit in the display unit system control register (DSYSR):

- Display mode register (DSMR)
  - ODPM2 (bit 29): ODDF pin mode 2
  - VSPM (bit 28): VSYNC pin mode
  - ODPM (bit 27): ODDF pin mode
  - DIPM (bits 26 and 25): DISP pin mode
  - CSPM (bit 24): CSYNC pin mode
  - DIL (bit 19): polarity reversal bit of the DISP signal
  - VSL (bit 18): polarity reversal bit of the VSYNC signal
  - HSL (bit 17): polarity reversal bit of the HSYNC signal
- Output signal timing adjustment register (OTAR)
  - All bits

The registers for the X and Y start positions for plane n in the interlaced sync & video mode (PnSPXR, PnSPYR) are also internally updated at the beginning of a field.

Updates are performed at the falling edge of VSYNC output when the sync method of the display unit system control register (DSYSR) is master mode (TVM1 = 0, TVM0 = 0), or at the falling edge of EXVSYNC detected in TV sync mode (TVM1 = 1, TVM0 = 0). In sync transition mode (TVM1 = 0, TVM0 = 1), internal updates are not performed.

However, plane n display area start address 0 register, plane n display area start address 1 register, and plane n display area start address 2 register are internally updated in display operation and externally updated when the video data and rendering data are written to addresses specified for these registers.

The address-mapped registers with an internal update function are shown in table 23.2, Register Configuration. The initial settings for these registers should be made during the interval in which the DRES bit is 1.

Table 23.2 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size	Bits with Internal Update Function
Display unit system control register	DSYSR	R/W	FFF80000	32 bits	DSEC (bit 20) DEN (bit 8)
Display mode register	DSMR	R/W	FFF80004	32 bits	All bits except the following bits which are updated by the display reset (DRES) bit in the display unit system control register (DSYSR): ODPM2 (bit 29) VSPM (bit 28) ODPM (bit 27) DIPM (bits 26 and 25) CSPM (bit 24) DIL (bit 19) VSL (bit 18) HSL (bit 17)
Display status register	DSSR	R/W	FFF80008	32 bits	None
Display unit status register clear register	DSRCR	R/W	FFF8000C	32 bits	None
Display unit interrupt enable register	DIER	R/W	FFF80010	32 bits	None
Color palette control register	CPCR	R/W	FFF80014	32 bits	All bits
Display plane priority register	DPPR	R/W	FFF80018	32 bits	All bits
Display unit extensional function enable register	DEFR	R/W	FFF80020	32 bits	All bits are updated by DRES.
Display alpha ratio plane control register	DAPCR	R/W	FFF80024	32 bits	All bits
Display capture control register	DCPCR	R/W	FFF80028	32 bits	All bits
Display unit extensional function enable register 2	DEFR2	R/W	FFF80034	32 bits	All bits are updated by DRES.
Horizontal display start register	HDSR	R/W	FFF80040	32 bits	All bits

Register Name	Abbreviation	R/W	Address	Access Size	Bits with Internal Update Function
Horizontal display end register	HDER	R/W	FFF80044	32 bits	All bits
Vertical display start register	VDSR	R/W	FFF80048	32 bits	All bits
Vertical display end register	VDER	R/W	FFF8004C	32 bits	All bits
Horizontal cycle register	HCR	R/W	FFF80050	32 bits	All bits
Horizontal sync width register	HSWR	R/W	FFF80054	32 bits	All bits
Vertical cycle register	VCR	R/W	FFF80058	32 bits	All bits
Vertical sync point register	VSPR	R/W	FFF8005C	32 bits	All bits
Equal pulse width register	EQWR	R/W	FFF80060	32 bits	All bits
Serration width register	SPWR	R/W	FFF80064	32 bits	All bits
CLAMP signal start register	CLAMPSR	R/W	FFF80070	32 bits	All bits
CLAMP signal width register	CLAMPWR	R/W	FFF80074	32 bits	All bits
DE signal start register	DESR	R/W	FFF80078	32 bits	All bits
DE signal width register	DEWR	R/W	FFF8007C	32 bits	All bits
Color palette transparent color register	CP1TR	R/W	FFF80080	32 bits	All bits
Display-off output register	DOOR	R/W	FFF80090	32 bits	All bits
Color detection register	CDER	R/W	FFF80094	32 bits	All bits
Background plane output register	BPOR	R/W	FFF80098	32 bits	All bits
Raster interrupt offset register	RINTOFSR	R/W	FFF8009C	32 bits	All bits

Register Name	Abbreviation	R/W	Address	Access Size	Bits with Internal Update Function
Plane 1 mode register	P1MR	R/W	FFF80100	32 bits	All bits
Plane 1 memory width register	P1MWR	R/W	FFF80104	32 bits	All bits
Plane 1 blending ratio register	P1ALPHAR	R/W	FFF80108	32 bits	All bits
Plane 1 display size X register	P1DSXR	R/W	FFF80110	32 bits	All bits
Plane 1 display size Y register	P1DSYR	R/W	FFF80114	32 bits	All bits
Plane 1 display position X register	P1DPXR	R/W	FFF80118	32 bits	All bits
Plane 1 display position Y register	P1DPYR	R/W	FFF8011C	32 bits	All bits
Plane 1 display area start address 0 register	P1DSA0R	R/W	FFF80120	32 bits	All bits
Plane 1 display area start address 1 register	P1DSA1R	R/W	FFF80124	32 bits	All bits
Plane 1 display area start address 2 register	P1DSA2R	R/W	FFF80128	32 bits	All bits
Plane 1 start position X register	P1SPXR	R/W	FFF80130	32 bits	All bits
Plane 1 start position Y register	P1SPYR	R/W	FFF80134	32 bits	All bits
Plane 1 wrap-around start position register	P1WASPR	R/W	FFF80138	32 bits	All bits
Plane 1 wrap-around memory width register	P1WAMWR	R/W	FFF8013C	32 bits	All bits
Plane 1 blinking time register	P1BTR	R/W	FFF80140	32 bits	All bits
Plane 1 transparent color 1 register	P1TC1R	R/W	FFF80144	32 bits	All bits
Plane 1 transparent color 2 register	P1TC2R	R/W	FFF80148	32 bits	All bits
Plane 1 memory length register	P1MLR	R/W	FFF80150	32 bits	All bits

Register Name	Abbreviation	R/W	Address	Access Size	Bits with Internal Update Function
Plane 2 mode register	P2MR	R/W	FFF80200	32 bits	All bits
Plane 2 memory width register	P2MWR	R/W	FFF80204	32 bits	All bits
Plane 2 blending ratio register	P2ALPHAR	R/W	FFF80208	32 bits	All bits
Plane 2 display size X register	P2DSXR	R/W	FFF80210	32 bits	All bits
Plane 2 display size Y register	P2DSYR	R/W	FFF80214	32 bits	All bits
Plane 2 display position X register	P2DPXR	R/W	FFF80218	32 bits	All bits
Plane 2 display position Y register	P2DPYR	R/W	FFF8021C	32 bits	All bits
Plane 2 display area start address 0 register	P2DSA0R	R/W	FFF80220	32 bits	All bits
Plane 2 display area start address 1 register	P2DSA1R	R/W	FFF80224	32 bits	All bits
Plane 2 display area start address 2 register	P2DSA2R	R/W	FFF80228	32 bits	All bits
Plane 2 start position X register	P2SPXR	R/W	FFF80230	32 bits	All bits
Plane 2 start position Y register	P2SPYR	R/W	FFF80234	32 bits	All bits
Plane 2 wrap-around start position register	P2WASPR	R/W	FFF80238	32 bits	All bits
Plane 2 wrap-around memory width register	P2WAMWR	R/W	FFF8023C	32 bits	All bits
Plane 2 blinking time register	P2BTR	R/W	FFF80240	32 bits	All bits
Plane 2 transparent color 1 register	P2TC1R	R/W	FFF80244	32 bits	All bits
Plane 2 transparent color 2 register	P2TC2R	R/W	FFF80248	32 bits	All bits
Plane 2 memory length register	P2MLR	R/W	FFF80250	32 bits	All bits

Register Name	Abbreviation	R/W	Address	Access Size	Bits with Internal Update Function
Plane 3 mode register	P3MR	R/W	FFF80300	32 bits	All bits
Plane 3 memory width register	P3MWR	R/W	FFF80304	32 bits	All bits
Plane 3 blending ratio register	P3ALPHAR	R/W	FFF80308	32 bits	All bits
Plane 3 display size X register	P3DSXR	R/W	FFF80310	32 bits	All bits
Plane 3 display size Y register	P3DSYR	R/W	FFF80314	32 bits	All bits
Plane 3 display position X register	P3DPXR	R/W	FFF80318	32 bits	All bits
Plane 3 display position Y register	P3DPYR	R/W	FFF8031C	32 bits	All bits
Plane 3 display area start address 0 register	P3DSA0R	R/W	FFF80320	32 bits	All bits
Plane 3 display area start address 1 register	P3DSA1R	R/W	FFF80324	32 bits	All bits
Plane 3 display area start address 2 register	P3DSA2R	R/W	FFF80328	32 bits	All bits
Plane 3 start position X register	P3SPXR	R/W	FFF80330	32 bits	All bits
Plane 3 start position Y register	P3SPYR	R/W	FFF80334	32 bits	All bits
Plane 3 wrap-around start position register	P3WASPR	R/W	FFF80338	32 bits	All bits
Plane 3 wrap-around memory width register	P3WAMWR	R/W	FFF8033C	32 bits	All bits
Plane 3 blinking time register	P3BTR	R/W	FFF80340	32 bits	All bits
Plane 3 transparent color 1 register	P3TC1R	R/W	FFF80344	32 bits	All bits
Plane 3 transparent color 2 register	P3TC2R	R/W	FFF80348	32 bits	All bits
Plane 3 memory length register	P3MLR	R/W	FFF80350	32 bits	All bits
Plane 4 mode register	P4MR	R/W	FFF80400	32 bits	All bits

Register Name	Abbreviation	R/W	Address	Access Size	Bits with Internal Update Function
Plane 4 memory width register	P4MWR	R/W	FFF80404	32 bits	All bits
Plane 4 blending ratio register	P4ALPHAR	R/W	FFF80408	32 bits	All bits
Plane 4 display size X register	P4DSXR	R/W	FFF80410	32 bits	All bits
Plane 4 display size Y register	P4DSYR	R/W	FFF80414	32 bits	All bits
Plane 4 display position X register	P4DPXR	R/W	FFF80418	32 bits	All bits
Plane 4 display position Y register	P4DPYR	R/W	FFF8041C	32 bits	All bits
Plane 4 display area start address 0 register	P4DSA0R	R/W	FFF80420	32 bits	All bits
Plane 4 display area start address 1 register	P4DSA1R	R/W	FFF80424	32 bits	All bits
Plane 4 display area start address 2 register	P4DSA2R	R/W	FFF80428	32 bits	All bits
Plane 4 start position X register	P4SPXR	R/W	FFF80430	32 bits	All bits
Plane 4 start position Y register	P4SPYR	R/W	FFF80434	32 bits	All bits
Plane 4 wrap-around start position register	P4WASPR	R/W	FFF80438	32 bits	All bits
Plane 4 wrap-around memory width register	P4WAMWR	R/W	FFF8043C	32 bits	All bits
Plane 4 blinking time register	P4BTR	R/W	FFF80440	32 bits	All bits
Plane 4 transparent color 1 register	P4TC1R	R/W	FFF80444	32 bits	All bits
Plane 4 transparent color 2 register	P4TC2R	R/W	FFF80448	32 bits	All bits
Plane 4 memory length register	P4MLR	R/W	FFF80450	32 bits	All bits

Register Name	Abbreviation	R/W	Address	Access Size	Bits with Internal Update Function
Alpha plane 1 mode register	AP1MR	R/W	FFF8A100	32 bits	All bits
Alpha plane 1 memory width register	AP1MWR	R/W	FFF8A104	32 bits	All bits
Alpha plane 1 display size X register	AP1DSXR	R/W	FFF8A110	32 bits	All bits
Alpha plane 1 display size Y register	AP1DSYR	R/W	FFF8A114	32 bits	All bits
Alpha plane 1 display position X register	AP1DPXR	R/W	FFF8A118	32 bits	All bits
Alpha plane 1 display position Y register	AP1DPYR	R/W	FFF8A11C	32 bits	All bits
Alpha plane 1 display area start address 0 register	AP1DSA0R	R/W	FFF8A120	32 bits	All bits
Alpha plane 1 display area start address 1 register	AP1DSA1R	R/W	FFF8A124	32 bits	All bits
Alpha plane 1 display area start address 2 register	AP1DSA2R	R/W	FFF8A128	32 bits	All bits
Alpha plane 1 start position X register	AP1SPXR	R/W	FFF8A130	32 bits	All bits
Alpha plane 1 start position Y register	AP1SPYR	R/W	FFF8A134	32 bits	All bits
Alpha plane 1 wrap-around start position register	AP1WASPR	R/W	FFF8A138	32 bits	All bits
Alpha plane 1 wrap-around memory width register	AP1WAMWR	R/W	FFF8A13C	32 bits	All bits
Alpha plane 1 blinking time register	AP1BTR	R/W	FFF8A140	32 bits	All bits
Alpha plane 1 memory length register	AP1MLR	R/W	FFF8A150	32 bits	All bits
Display capture memory width register	DCMWR	R/W	FFF8C104	32 bits	All bits
Display capture area start address register	DCSAR	R/W	FFF8C120	32 bits	All bits
Display capture memory length register	DCMLR	R/W	FFF8C150	32 bits	All bits

Register Name	Abbreviation	R/W	Address	Access Size	Bits with Internal Update Function
Color palette register 000	CP1_000R	R/W	FFF81000	32 bits	All bits
			—		
Color palette register 255	CP1_255R	R/W	FFF813FC	32 bits	All bits
External synchronization control register	ESCR	R/W	FFF90000	32 bits	None
Output signal timing adjustment register	OTAR	R/W	FFF90004	32 bits	These bits are updated by the display reset (DRES) bit in the display unit system control register (DSYSR).

23.2 Register Descriptions

Legend:

Initial value: Value of a register after a reset has been issued

—: Undefined value

R/W: Read and write enabled. A write value can be read.

R/WC0: Read and write enabled. The bit is initialized when 0 is written to; writing 1 to this bit is ignored.

R: Read only. The write value should always be 0.

—/WB: Write only. The read value is undefined.

Plane n: Planes 1 to 8

23.2.1 Display Control Registers

(1) Display Unit System Control Register (DSYSR)

Address: FFF80000

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			ILTS									DSEC				IUPD
Initial value:	—	—	0	—	—	—	—	—	—	—	—	0	—	—	—	0
R/W:	R	R	R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							DRES	DEN	TVM		SCM					
Initial value:	—	—	—	—	—	—	1	0	1	0	0	0	—	—	—	—
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31, 30	—	—	R	—	Reserved
29	ILTS	0	R/W	Yes	Input Pad Latch Timing Select To enable bit 29, set the DEFE bit in DEFR to 1. This bit is fixed to 0 by default. 0: Signals from input pins are latched on rising edges of DCLKIN (initial value) 1: Signals from input pins are latched on falling edges of DCLKIN; this is outside the scope of the electrical characteristics specification.
28 to 21	—	—	R	—	Reserved
20	DSEC	0	R/W	Yes	Display Data Endian Change For details of data swap, see section 23.3.7, Endian Conversion. 0: Display data in memory is not byte-data/word-data swapped 1: Display data in memory is byte-data/word-data swapped
19 to 17	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
16	IUPD	0	R/W	Yes	Internal Updating Disable When DRES = 1, internal update is performed regardless of this bit. For details of internal update, see (2) Internal Update in section 23.1.4, Register Configuration. 0: Internal update is performed upon each vertical sync signal (VSYNC) assertion 1: By setting this bit to 1, internal updates can be prohibited. When this bit is set to 0, register update is performed upon the next vertical sync signal (VSYNC).
15 to 10	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
9	DRES	1	R/W	None	Display Reset
8	DEN	0	R/W	Yes	Display Enable 00: Starts display synchronization operation. In the case of a register not yet set, unexpected operation may occur; hence DRES should be set to 0 after setting all the registers in the display unit (DU). When DEN = 0, the display data is the value set in DOOR. 01: Starts display synchronization operation. In the case of a register not yet set, unexpected operation may occur; hence DRES and DEN should be set to 0 and 1 respectively after setting all the registers in the display unit (DU). When DEN = 1, the display data is the value stored in memory from the next frame. 10: Halts display and synchronization operation. Halts display operation and synchronization operation. Except for the following bits in DSSR, register settings are held. For these settings, operation is as follows. 1. All display data output is 0. 2. The following bits in DSSR are cleared to 0. — TV sync signal error flag (TVR) — Frame flag (FRM) — Vertical blanking flag (VBK) — Raster interrupt flag (RINT) — Horizontal blanking flag (HBK) 3. The $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, and ODDF pins are input pins. However, when the ODPM bit in DSMR is 1, the ODDF pin output is clamped. 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
7, 6	TVM	10	R/W	None	TV Synchronization Mode 00: Master mode HSYNC, VSYNC, and CSYNC are output. 01: Synchronization method switching mode When switching from TV sync mode to master mode, or from master mode to TV sync mode, is necessary, the switching should pass through this mode. In this mode, operation of the display system is forcibly halted, and DISP outputs a low level signal. Clock signal supply to the DCLK pin can also be halted (input disabled) (within the LSI the level is fixed high). EXHSYNC, VSYNC, and ODDF are input. 10: TV synchronization mode (initial value) EXHSYNC, VSYNC, and ODDF are input. However, when the ODPM bit in DSMR is 1, the ODDF pin is output. 11: Setting prohibited
5, 4	SCM	00	R/W	None	Scan Mode 00: Non-interlaced mode 01: Setting prohibited 10: Interlaced sync mode 11: Interlaced sync and video mode
3 to 0	—	—	R	—	Reserved

(2) Display Mode Register (DSMR)

Address: FFF80004

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			ODPM2	VSPM	ODPM	DIPM		CSPM					DIL	VSL	HSL	DDIS
Initial value:	—	—	0	0	0	0	0	0	—	—	—	—	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDEL	CDEM		CDED				ODEV	CSY							
Initial value:	0	0	0	0	—	—	—	0	0	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31, 30	—	—	R	—	Reserved
29	ODPM2	0	R/W	Yes	See the description of the ODPM bit.
28	VSPM	0	R/W	Yes	VSYNC Pin Mode 0: VSYNC signal is output to the VSYNC pin (initial value) 1: CSYNC signal is output to the VSYNC pin
27	ODPM	0	R/W	Yes	ODDF Pin Mode When ODPM2 is 0: 0: ODDF signal is output to the ODDF pin (initial value) 1: CLAMP signal is output to the ODDF pin The ODDF pin is an output pin even when the TVM bit in DSYSR is set to TV sync mode. When ODPM2 is 1: 0: CSYNC signal is output to the ODDF pin 1: Setting prohibited (fixed to 0)
26, 25	DIPM	00	R/W	Yes	DISP Pin Mode 00: DISP signal is output to the DISP pin (initial value) 01: CSYNC signal is output to the DISP pin 10: Setting prohibited (fixed to 0) 11: DE signal is output to the DISP pin

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
24	CSPM	0	R/W	Yes	CSYNC Pin Mode 0: CSYNC signal is output to the HSYNC pin (initial value) 1: HSYNC signal is output to the HSYNC pin
23 to 20	—	—	R	—	Reserved
19	DIL	0	R/W	Yes	DISP Polarity Select 0: DISP signal at high level during display interval (initial value) 1: DISP signal polarity is inverted
18	VSL	0	R/W	Yes	VSYNC Polarity Select 0: VSYNC signal is low-active (initial value) 1: VSYNC signal polarity is inverted
17	HSL	0	R/W	Yes	HSYNC Polarity Select 0: HSYNC signal is low-active (initial value) 1: HSYNC signal polarity is inverted
16	DDIS	0	R/W	Yes	DISP Output Disable 0: DISP signal is output (initial value) 1: DISP signal is not output
15	CDEL	0	R/W	Yes	CDE Polarity Select 0: CDE signal is high when output display data and CDER match (initial value) 1: CDE signal polarity is inverted
14, 13	CDEM	00	R/W	Yes	CDE Output Mode 00: CDE signal is output without change (initial value) 01: CDE signal is output without change 10: Low level output outside of display interval 11: High level output outside of display interval
12	CDED	0	R/W	Yes	CDE Disable 0: CDE signal is output (initial value) 1: CDE signal output is prohibited
11 to 9	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
8	ODEV	0	R/W	Yes	ODD Even Select for ODDF Signal 0: In interlaced display of the same frame, when the ODDF pin is at low level the first half of the field is displayed. (initial value) 1: In interlaced display of the same frame, when the ODDF pin is at high level the first half of the field is displayed.
7, 6	CSY	00	R/W	Yes	CSYNC Mode 00: The waveform produced by taking the logical exclusive-OR of VSYNC and HSYNC is output as CSYNC. (initial value) 01: Setting prohibited 10: For the interval of three raster scans after the falling edge of SYNC an equivalent pulse is output, followed by serration pulse for three raster scans, then an equivalent pulse for three raster scans, and for the interval after this the HSYNC waveform is output as CSYNC. 11: 1/2 raster scan after the falling edge of VSYNC, an equivalent pulse is output for 2.5 raster scans, then serration pulse for 2.5 raster scans, then an equivalent pulse for 2.5 raster scans, and for the interval after this the HSYNC waveform is output as CSYNC.
5 to 0	—	—	R	—	Reserved

(3) Display Status Register (DSSR)

Address: FFF80008

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			VCFB					DFB9					DFB4	DFB3	DFB2	DFB1
Initial value:	—	—	1	1	—	—	0	0	—	—	—	—	0	0	0	0
R/W:	R	R	R	R	R	R	—	R	—	—	—	—	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TVR	FRM			VBK		RINT	HBK					ADC4	ADC3	ADC2	ADC1
Initial value:	0	0	—	—	0	—	0	0	—	—	—	—	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	—	—	—	—	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31, 30	—	—	R	—	Reserved
29, 28	VCFB	11	R	None	Video Capture Frame Buffer Flag 00: The address indicated by PnDSA0R in the plane that has been set for the video capture is being used as the display area start address 01: The address indicated by PnDSA1R in the plane that has been set for the video capture is being used as the display area start address 10: The address indicated by PnDSA2R in the plane that has been set for the video capture is being used as the display area start address 11: The video capture module is in the initial state (initial value)
27, 26	—	—	R	—	Reserved
25	—	0	—	—	Reserved
24	DFB9	0	R	None	Display Frame Buffer 9 Flag 0: The address indicated by AP1SA0R in alpha-ratio plane 1 is being used as the display area start address (initial value) 1: The address indicated by AP1SA1R in alpha-ratio plane 1 is being used as the display area start address

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
23 to 20	—	—	—	—	Reserved
19	DFB4	0	R	None	Display Frame Buffer 4 Flag 0: The address indicated by P4DSA0R in plane 4 is being used as the display area start address (initial value) 1: The address indicated by P4DSA1R in plane 4 is being used as the display area start address
18	DFB3	0	R	None	Display Frame Buffer 3 Flag 0: The address indicated by P3DSA0R in plane 3 is being used as the display area start address (initial value) 1: The address indicated by P3DSA1R in plane 3 is being used as the display area start address
17	DFB2	0	R	None	Display Frame Buffer 2 Flag 0: The address indicated by P2DSA0R in plane 2 is being used as the display area start address (initial value) 1: The address indicated by P2DSA1R in plane 2 is being used as the display area start address
16	DFB1	0	R	None	Display Frame Buffer 1 Flag 0: The address indicated by P1DSA0R in plane 1 is being used as the display area start address (initial value) 1: The address indicated by P1DSA1R in plane 1 is being used as the display area start address

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
15	TVR	0	R	None	TV Synchronization Error Flag 0: After using the DRES bit in DSYSR or the TVCL bit in DSRCR to clear the TVR, indicates that the rising edge of EXVSYNC is being detected each time within the vertical period determined by the setting of VCR (initial value). 1: Indicates that the rising edge of EXVSYNC was not detected within the vertical period determined by the setting of VCR in TV sync mode. The TVR bit holds its state until cleared to 0 by the DRES bit or the TVCL bit.
14	FRM	0	R	None	Frame Flag 0: After clearing the FRM bit using either the DRES bit in DSYSR or the FRCL bit in DSRCR, in non-interlaced mode indicates the interval to the next display end, and in interlaced sync mode or in interlaced sync & video mode indicates the interval to the display end of the next even field (initial value). 1: After clearing the FRM bit using either the DRES bit or the FRCL bit, indicates the interval until the next time the FRM bit is cleared, from the first even-field vertical blanking interval (frame units).
13, 12	—	—	R	—	Reserved
11	VBK	0	R	None	Vertical Blanking Flag 0: Indicates the interval to the next display end after clearing the VBK bit using either the DRES bit in DSYSR or the VBCL bit in DSRCR. (initial value) 1: Indicates the interval, after clearing the VBK bit using either the DRES bit or the VBCL bit, from the first vertical blanking interval until the VBK bit is again cleared. (field units)

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
10	—	—	R	—	Reserved
9	RINT	0	R	None	Raster Interrupt Flag 0: Indicates the interval from the start of the next display until raster scans set in RINTOFSR have elapsed, after clearing the RINT bit using either the DRES bit in DSYSR or the RICL bit in DSRCR. (initial value) 1: After clearing the RINT bit using either the DRES bit or the RICL bit, indicates the interval from the start of the next display after raster scans set in RINTOFSR have elapsed until the bit is again cleared.
8	HBK	0	R	None	Horizontal Blanking Flag 0: Indicates the interval, after clearing the HBK bit using the DRES bit in DSYSR or the HBCL bit in DSRCR, to the next interval of HSYNC assertion. (initial value) 1: Indicates the interval, after clearing the HBK bit using either the DRES bit or the HBCL bit, from the first horizontal blanking interval until the HBK bit is again cleared.
7 to 4	—	—	—	—	Reserved
3	ADC4	0	R	None	Auto Rendering Display Change Flag 4 0: The frame buffer in plane 4 has not been switched. (initial value) 1: The frame buffer in plane 4 has been switched. The bit retains its state until it is cleared.
2	ADC3	0	R	Yes	Auto Rendering Display Change Flag 3 0: The frame buffer in plane 3 has not been switched. (initial value) 1: The frame buffer in plane 3 has been switched. The bit retains its state until it is cleared
1	ADC2	0	R	None	Auto Rendering Display Change Flag 2 0: The frame buffer in plane 2 has not been switched. (initial value) 1: The frame buffer in plane 2 has been switched. The bit retains its state until it is cleared

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
0	ADC1	0	R	None	Auto Rendering Display Change Flag 1 0: The frame buffer in plane 1 has not been switched. (initial value) 1: The frame buffer in plane 1 has been switched. The bit retains its state until it is cleared

Explanation related to the ADC bits

These bits can be used to display moving pictures in auto rendering mode. In this case, the following requirements apply.

(1) TRAP interrupts must be checked.

(2) After that, VBK interrupts must be checked.

(3) Flags must be checked twice when rendering starts after a VBK interrupt has been generated.

By using these flag bits (ADC4 to ADC1), rendering can be initiated by checking the flag bits once and issuing of an ADC interrupt.

(4) Display Unit Status Register Clear Register (DSRCR)

Address: FFF8000C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TVCL	FRCL			VBCL		RICL	HBCL					ADCL4	ADCL3	ADCL2	ADCL1
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	—/WB	—/WB	R	R	—/WB	R	—/WB	—/WB	—	—	—	—	—/WB	—/WB	—/WB	—/WB

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 16	—	—	R	—	Reserved
15	TVCL	—	—/WB	None	TV Synchronization Signal Error Flag Clear 0: The TVR flag in DSSR is not changed. 1: The TVR flag in DSSR is cleared to 0.
14	FRCL	—	—/WB	None	Frame Flag Clear 0: The FRM flag in DSSR is not changed. 1: The FRM flag in DSSR is cleared to 0.
13, 12	—	—	R	—	Reserved
11	VBCL	—	—/WB	None	Vertical Blanking Flag Clear 0: The VBK flag in DSSR is not changed. 1: The VBK flag in DSSR is cleared to 0.
10	—	—	R	—	Reserved
9	RICL	—	—/WB	None	Raster Interrupt Flag Clear 0: The RINT flag in DSSR is not changed. 1: The RINT flag in DSSR is cleared to 0.
8	HBCL	—	—/WB	None	HBK Flag Clear 0: The HBK flag in DSSR is not changed. 1: The HBK flag in DSSR is cleared to 0.
7 to 4	—	—	—	None	Reserved
3	ADCL4	—	—/WB	None	Auto Rendering Display Change Flag Clear 4 0: ADC flag 4 in DSSR is not changed. 1: ADC flag 4 in DSSR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
2	ADCL3	—	—/WB	None	Auto Rendering Display Change Flag Clear 3 0: ADC flag 3 in DSSR is not changed. 1: ADC flag 3 in DSSR is cleared to 0.
1	ADCL2	—	—/WB	None	Auto Rendering Display Change Flag Clear 2 0: ADC flag 2 in DSSR is not changed. 1: ADC flag 2 in DSSR is cleared to 0.
0	ADCL1	—	—/WB	None	Auto Rendering Display Change Flag Clear 1 0: ADC flag 1 in DSSR is not changed. 1: ADC flag 1 in DSSR is cleared to 0.

(5) Display Unit Interrupt Enable Register (DIER)

Address: FFF80010

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TVE	FRE			VBE		RIE	HBE					ADCE4	ADCE3	ADCE2	ADCE1
Initial value:	0	0	—	—	0	—	0	0	—	—	—	—	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R	R/W	R/W	—	—	—	—	R/W	R/W	R/W	R/W

The display unit interrupt enable register (DIER) is a register which enables interrupts to the CPU the causes of which are internal states of the display unit (DU) reflected in DSSR. When bits are set in this register, if bits in the same bit positions in DSSR are set, an interrupt is issued to the CPU.

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 16	—	—	R	—	Reserved
15	TVE	0	R/W	None	TV Synchronous Signal Error Flag Interrupt Enable 0: Disables interrupt by the TVR flag in DSSR 1: Enables interrupt by the TVR flag in DSSR
14	FRE	0	R/W	None	Frame Flag Interrupt Enable 0: Disables interrupt by the FRM flag in DSSR 1: Enables interrupt by the FRM flag in DSSR
13, 12	—	—	R	—	Reserved
11	VBE	0	R/W	None	Vertical Blanking Flag Interrupt Enable 0: Disables interrupt by the VBK flag in DSSR 1: Enables interrupt by the VBK flag in DSSR
10	—	—	R	—	Reserved
9	RIE	0	R/W	None	Raster Interrupt Flag Interrupt Enable 0: Disables interrupt by the RINT flag in DSSR 1: Enables interrupt by the RINT flag in DSSR

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
8	HBE	0	R/W	None	HBK Flag Interrupt Enable 0: Disables interrupt by the HBK flag in DSSR 1: Enables interrupt by the HBK flag in DSSR
7 to 4	—	—	—	—	Reserved
3	ADCE4	0	R/W	None	Auto Rendering Display Change Flag 4 Interrupt Enable 0: Disables interrupt by the ADC flag 4 in DSSR 1: Enables interrupt by the ADC flag 4 in DSSR
2	ADCE3	0	R/W	None	Auto Rendering Display Change Flag 3 Interrupt Enable 0: Disables interrupt by the ADC flag 3 in DSSR 1: Enables interrupt by the ADC flag 3 in DSSR
1	ADCE2	0	R/W	None	Auto Rendering Display Change Flag 2 Interrupt Enable 0: Disables interrupt by the ADC flag 2 in DSSR 1: Enables interrupt by the ADC flag 2 in DSSR
0	ADCE1	0	R/W	None	Auto Rendering Display Change Flag 1 Interrupt Enable 0: Disables interrupt by the ADC flag 1 in DSSR 1: Enables interrupt by the ADC flag 1 in DSSR

The following are conditions, based on DSSR and this register, for issuing an interrupt to the CPU from the display unit (DU).

Conditions for issuing an interrupt = a + b + c + d + e + h + i + j + k

- a = TVR · TVE
- b = FRM · FRE
- c = VBK · VBE
- d = RINT · RIE
- e = HBK · HBE
- h = ADC4 · ADCE4
- i = ADC3 · ADCE3
- j = ADC2 · ADCE2
- k = ADC1 · ADCE1

(6) Color Palette Control Register (CPCR)

Address: FFF80014

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																CP1CE
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 17	—	—	R	—	Reserved
16	CP1CE	0	R/W	Yes	Color Palette Change Enable 0: Switching of the color palette is not performed. 1: Switching of the color palette is performed. Switching is performed when bit 9 (DRES) in DSYSR is changed from 1 to 0, or with the timing of an internal update. This bit can only be set to 1; an operation to set the bit to 0 is invalid. After switching of the color palette, the bit is cleared to 0. When setting to 1 and clearing occur simultaneously, clearing to 0 takes priority.
15 to 0	—	—	R	—	Reserved

(7) Display Plane Priority Register (DPPR)

Address: FFF80018

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													DPE5	DPS5		
Initial value:	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0
R/W:	—	—	—	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPE4	DPS4			DPE3	DPS3			DPE2	DPS2			DPE1	DPS1		
Initial value:	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The display plane priority register (DPPR) defines the order for combining planes and turns the display on and off.

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31	—	0	—	—	Reserved
30 to 28	—	111	—	—	
27	—	0	—	—	
26 to 24	—	110	—	—	
23	—	0	—	—	
22 to 20	—	101	—	—	
19	DPE5	0	R/W	Yes	Display Plane Priority 5 Enable
18 to 16	DPS5	100	R/W	Yes	Display Plane Priority 5 Select 1000: Selects and displays plane 1 in priority 5 1001: Selects and displays plane 2 in priority 5 1010: Selects and displays plane 3 in priority 5 1011: Selects and displays plane 4 in priority 5 0---: Priority 5 is not displayed
15	DPE4	0	R/W	Yes	Display Plane Priority 4 Enable
14 to 12	DPS4	011	R/W	Yes	Display Plane Priority 4 Select 1000: Selects and displays plane 1 in priority 4 1001: Selects and displays plane 2 in priority 4 1010: Selects and displays plane 3 in priority 4 1011: Selects and displays plane 4 in priority 4 0---: Priority 4 is not displayed

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
11	DPE3	0	R/W	Yes	Display Plane Priority 3 Enable
10 to 8	DPS3	010	R/W	Yes	Display Plane Priority 3 Select 1000: Selects and displays plane 1 in priority 3 1001: Selects and displays plane 2 in priority 3 1010: Selects and displays plane 3 in priority 3 1011: Selects and displays plane 4 in priority 3 0---: Priority 3 is not displayed
7	DPE2	0	R/W	Yes	Display Plane Priority 2 Enable
6 to 4	DPS2	001	R/W	Yes	Display Plane Priority 2 Select 1000: Selects and displays plane 1 in priority 2 1001: Selects and displays plane 2 in priority 2 1010: Selects and displays plane 3 in priority 2 1011: Selects and displays plane 4 in priority 2 0---: Priority 2 is not displayed
3	DPE1	0	R/W	Yes	Display Plane Priority 1 Enable
2 to 0	DPS1	000	R/W	Yes	Display Plane Priority 1 Select 1000: Selects and displays plane 1 in priority 1 1001: Selects and displays plane 2 in priority 1 1010: Selects and displays plane 3 in priority 1 1011: Selects and displays plane 4 in priority 1 0---: Priority 1 is not displayed

(8) Display Unit Extensional Function Enable Register (DEFR)

Address: FFF80020

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CODE															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB	—/WB

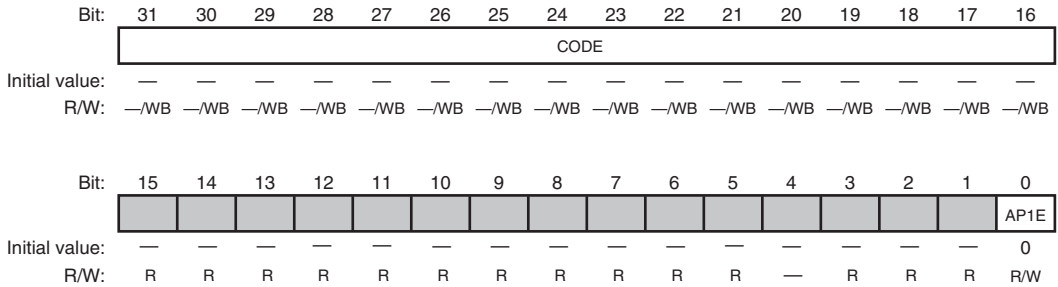
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				EXSL	EXVL						EXUP	VCUP				DEFE
Initial value:	—	—	—	0	0	—	—	—	—	—	0	0	—	—	—	0
R/W:	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 16	CODE	—	—/WB	None	Register Available Code To make DEFR accessible, set bits 31 to 16 to 0x7773 when writing to it.
15 to 13	—	—	R	—	Reserved
12	EXSL	0	R/W	Yes	External Sync Signal Select 0: The external sync signal (EXVSYNC or EXHSYNX) from the corresponding pin is directly latched by the frequency-divided clock signal. 1: The external sync signal (EXVSYNC or EXHSYNX) from the corresponding pin is acquired by the pre-division clock signal and then latched by the frequency-divided clock signal.
11	EXVL	0	R/W	Yes	External Vsync Latch Select 0: The external VSYNC signal (EXVSYNC) from the corresponding pin is latched on each edge of the clock signal. 1: The external VSYNC signal (EXVSYNC) from the corresponding pin is latched on rising edges of the external HSYNC signal (EXHSYNC).
10 to 6	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
5	EXUP	0	R/W	Yes	External Updating Mode 0: The EXUP bit is internally updated. 1: The EXUP bit is externally updated. The setting of this bit takes priority over that of the IUPD bit in DSYSR.
4	VCUP	0	R/W	Yes	Vertical Cycle Register Update Timing Select 0: VCR is internally updated on falling edges of VSYNC. 1: VCR is internally updated on rising edges of VSYNC; this stops the VSYNC signal generating distortion when VCR is changed.
3 to 1	—	—	R	—	Reserved
0	DEFE	0	R/W	Yes	Display Unit Extensional Function Enable 0: Extensional functions are disabled. 1: Extensional functions are enabled. That is, the following settings become effective: • Bits 31 to 30 in the display status register (DSSR) • Bits 26 to 24 in the plane n mode register (PnMR) • Bits 31 to 29 of the display area start address registers • Bits 24 and 5 in the external synchronization control register (ESCR) • Bit 10 in the plane n blending ratio register (PnALPHAR)

(9) Display Alpha Ratio Plane Control Register (DAPCR)

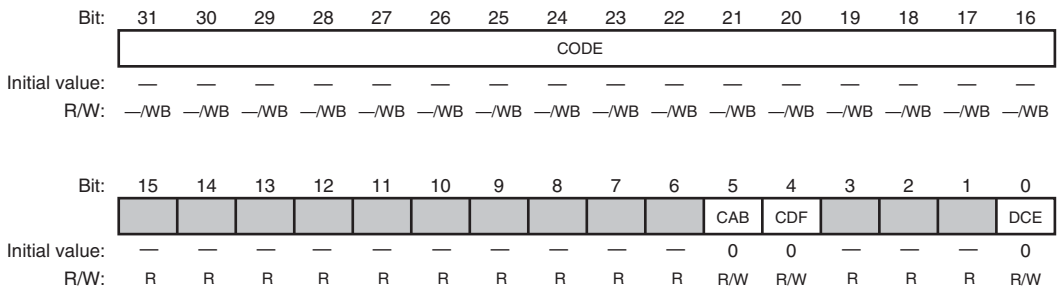
Address: FFF80024



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 16	CODE	—	—/WB	None	Register Available Code To make DAPCR accessible, set bits 31 to 16 to 0x7773 when writing to it.
15 to 5	—	—	R	—	Reserved
4	—	—	—	—	Reserved
3 to 1	—	—	R	—	Reserved
0	AP1E	0	R/W	Yes	Alpha Ratio Plane 1 Enable 0: Alpha-ratio plane 1 is not available. 1: Alpha-ratio plane 1 is available.

(10) Display Capture Control Register (DCPCR)

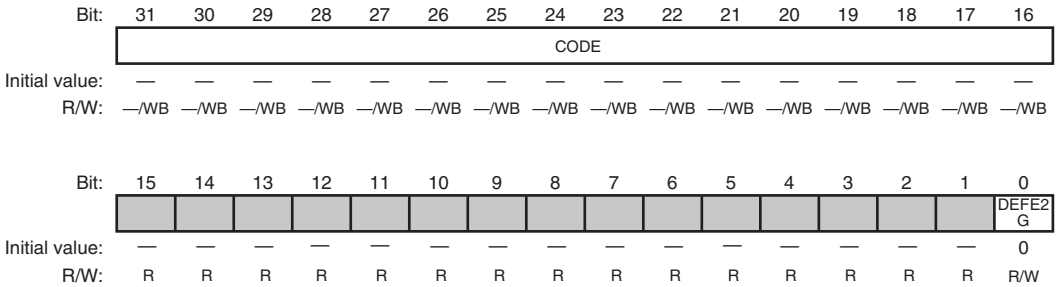
Address: FFF80028



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 16	CODE	—	—/WB	None	Register Available Code To make DCPCR accessible, set bits 31 to 16 to 0x7773 when writing to it.
15 to 6	—	—	R	—	Reserved
5	CAB	0	R/W	Yes	Display Capture A Bit Function Select To enable the operation of bit 5, set the DEFE2G bit in DEFR2 to 1. Bit 5 is fixed to 0 by default. 0: When the display capture data format is ARGB1555, the value of bit A is 0. 1: When the display capture data format is ARGB1555, the value of bit A is 1.
4	CDF	0	R/W	Yes	Display Capture Data Format To enable the operation of bit 4, set the DEFE2G bit in DEFR2 to 1. Bit 4 is fixed to 0 by default. 0: The display capture data is RGB565. 1: The display capture data is ARGB1555. The value of bit A is indicated by bit 5 in this register.
3 to 1	—	—	R	—	Reserved
0	DCE	0	R/W	Yes	Display Capture Enable 0: The display data is not captured. 1: When the DRES and DEN bits in DSYSR are 01, this setting starts the capture of display data. After this bit has been set to 1, capturing starts from the next frame.

(11) Display Unit Extensional Function Enable Register 2 (DEFR2)

Address: FFF80034



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 16	CODE	—	—/WB	None	Register Available Code To make DEFR2 accessible, set bits 31 to 16 to 0x7775 when writing to it.
15 to 1	—	—	R	—	Reserved
0	DEFE2G	0	R/W	Yes	Display Unit Extensional Function Enable SH-Navi2G 0: Extensional functions are disabled. 1: Extensional functions are enabled. That is, the following settings become effective: • Bits 13 and 12 in the plane n blending ratio register (PnALPHAR) • Bits 5 and 4 in the display capture control register (DCPCR)

23.2.2 Display Timing Generation Registers

(1) Horizontal Display Start Register (HDSR)

Address: FFF80040

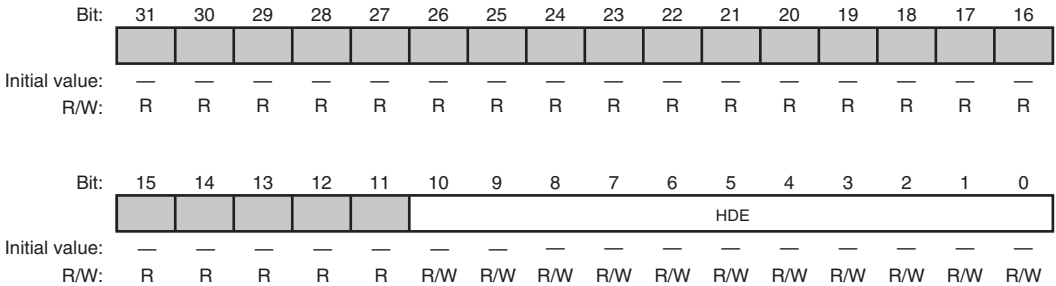
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HDS								
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 9	—	—	R	—	Reserved
8 to 0	HDS	—	R/W	Yes	Horizontal Display Start The horizontal display start position is set in dot clock units. The value is retained during a reset.

(2) Horizontal Display End Register (HDER)

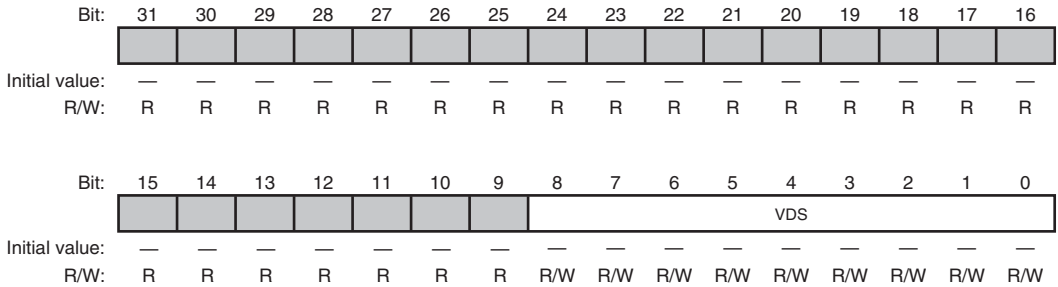
Address: FFF80044



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 11	—	—	R	—	Reserved
10 to 0	HDE	—	R/W	Yes	Horizontal Display End The horizontal display end position is set in dot clock units. The value is retained during a reset.

(3) Vertical Display Start Register (VDSR)

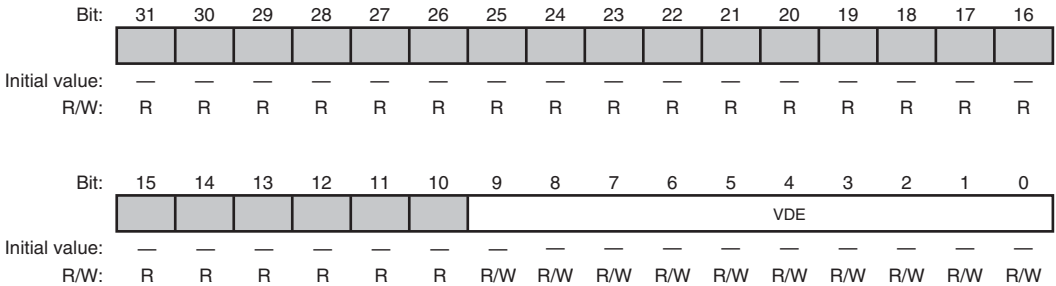
Address: FFF80048



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 9	—	—	R	—	Reserved
8 to 0	VDS	—	R/W	Yes	Vertical Display Start The vertical display start position is set in raster line units. The value is retained during a reset.

(4) Vertical Display End Register (VDER)

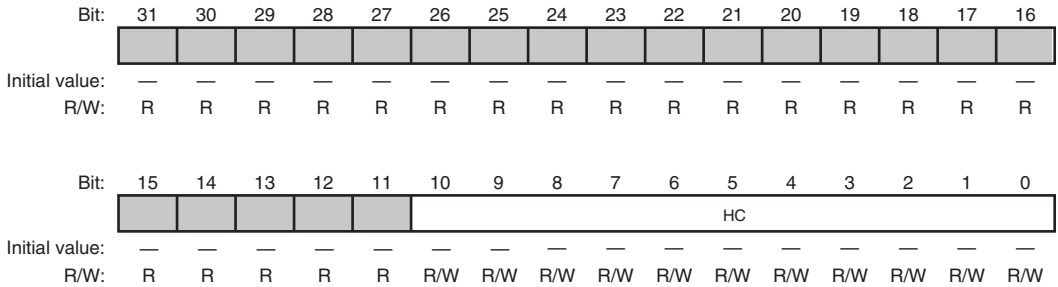
Address: FFF8004C



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 10	—	—	R	—	Reserved
9 to 0	VDE	—	R/W	Yes	Vertical Display End The vertical display end position is set in raster line units. The value is retained during a reset.

(5) Horizontal Cycle Register (HCR)

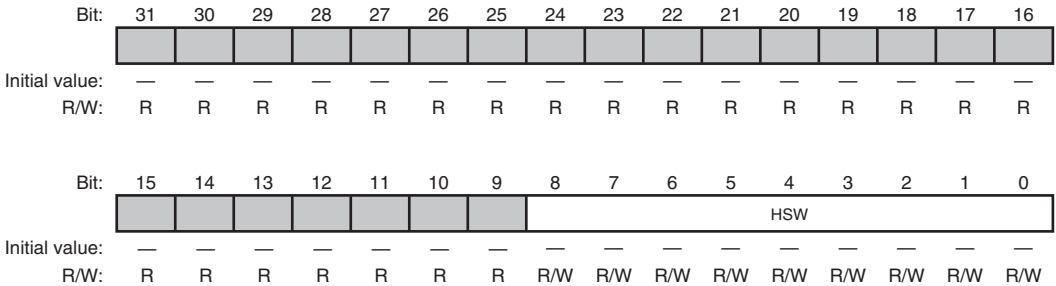
Address: FFF80050



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 11	—	—	R	—	Reserved
10 to 0	HC	—	R/W	Yes	Horizontal Cycle One horizontal scan period, including the horizontal blanking interval, is set in dot clock units. In TV sync mode, set this register so that the HSYNC period determined by this register is the same as or greater than the EXHSYNC period. The value is retained during a reset.

(6) Horizontal Sync Width Register (HSWR)

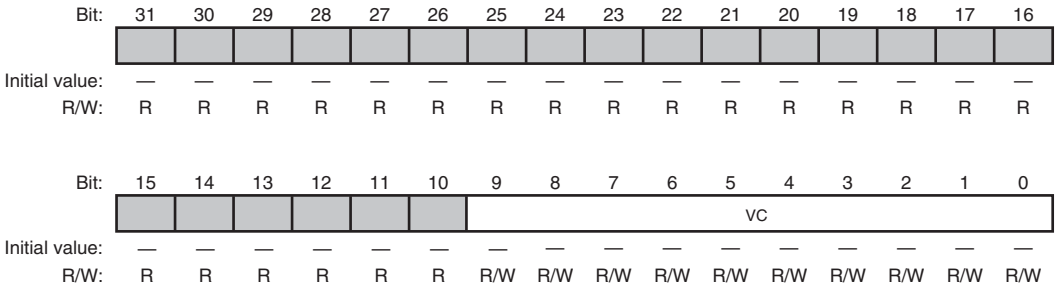
Address: FFF80054



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 9	—	—	R	—	Reserved
8 to 0	HSW	—	R/W	Yes	Horizontal Sync Width The low-level pulse width of the horizontal sync signal is set in dot clock units. The value is retained during a reset.

(7) Vertical Cycle Register (VCR)

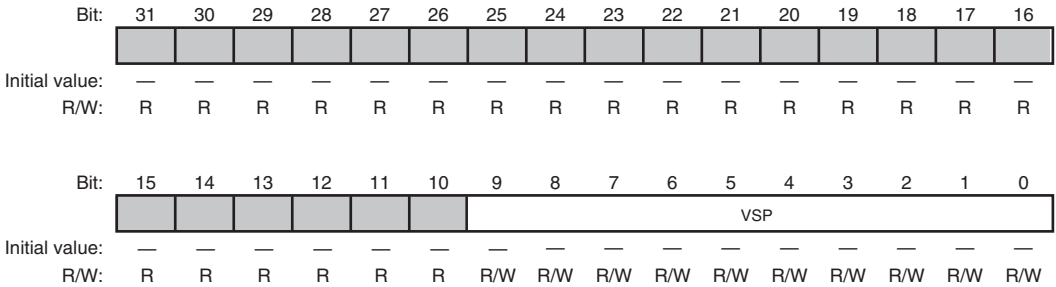
Address: FFF80058



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 10	—	—	R	—	Reserved
9 to 0	VC	—	R/W	Yes	Vertical Cycle The vertical scan interval, including the vertical blanking interval, is set in raster line units. In TV sync mode, the EXVSYNC rising-edge detection interval time is set. If not detected within the interval, the result is reflected in the TVR flag. The value is retained during a reset.

(8) Vertical Sync Point Register (VSPR)

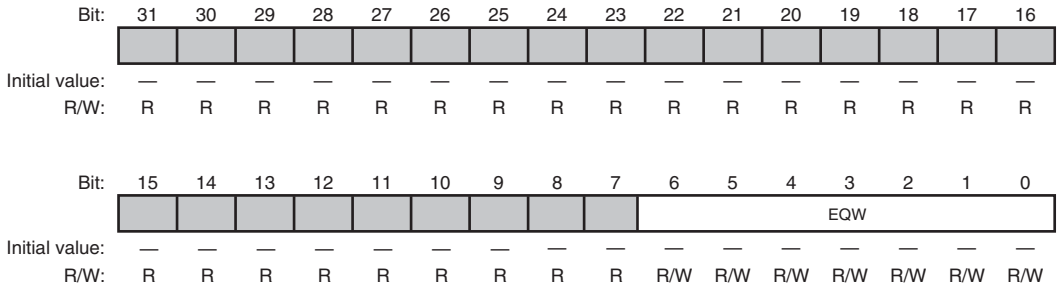
Address: FFF8005C



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 10	—	—	R	—	Reserved
9 to 0	VSP	—	R/W	Yes	Vertical Sync Point The start position of the vertical sync signal is set in raster line units. In TV sync mode, set this register so that the VSYNC falling edge set position in this register is the same as or later than the EXVSYNC falling edge. The value is retained during a reset.

(9) Equal Pulse Width Register (EQWR)

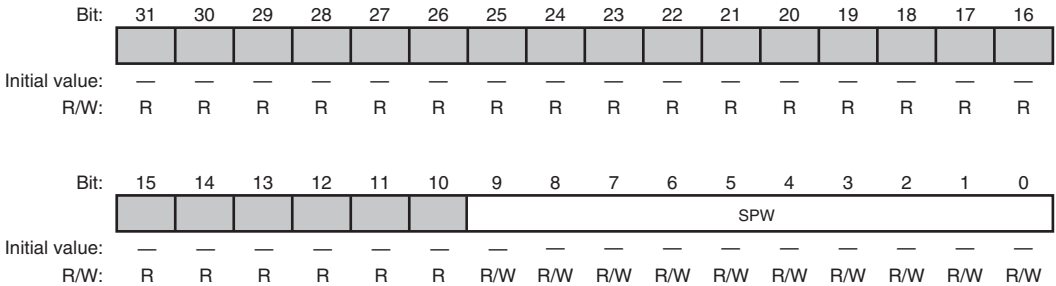
Address: FFF80060



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 7	—	—	R	—	Reserved
6 to 0	EQW	—	R/W	Yes	Equal Pulse Width The low-level pulse width of a pulse equivalent to the CSYNC signal is set in dot clock units. To enable this setting, set bit 7 of the CSYNC mode (CSY) in DSMR to 1. The value is retained during a reset.

(10) Serration Width Register (SPWR)

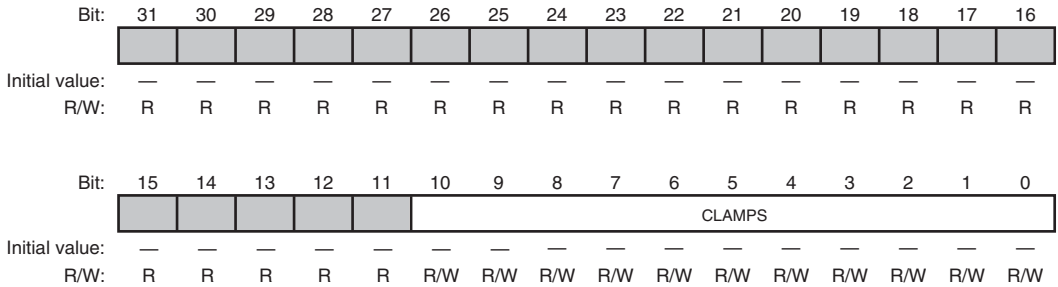
Address: FFF80064



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 10	—	—	R	—	Reserved
9 to 0	SPW	—	R/W	Yes	Serration Width The low-level pulse width of the serration pulse for the CSYNC signal is set in dot clock units. The value set should be smaller than 1/2 the HC bits. To enable this setting, bit 7 of the CSYNC mode (CSY) in DSMR to 1. The value is retained during a reset.

(11) CLAMP Signal Start Register (CLAMPSR)

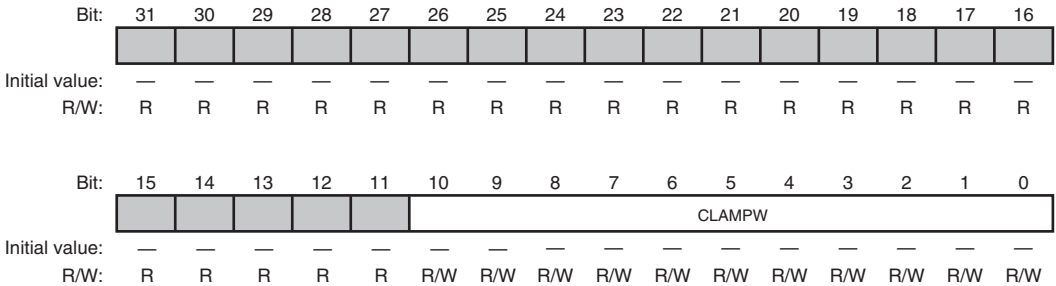
Address: FFF80070



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 11	—	—	R	—	Reserved
10 to 0	CLAMPS	—	R/W	Yes	Clamp Signal Start The CLAMP signal rising edge position is set in dot clock units relative to the falling edge of the HSYNC signal. The CLAMP signal rises after (setting value + 1) cycles of the falling edge of the HSYNC signal. Hence the CLAMP signal cannot be made to rise in the same cycle as the falling edge of the HSYNC signal. The value is retained during a reset.

(12) CLAMP Signal Width Register (CLAMPWR)

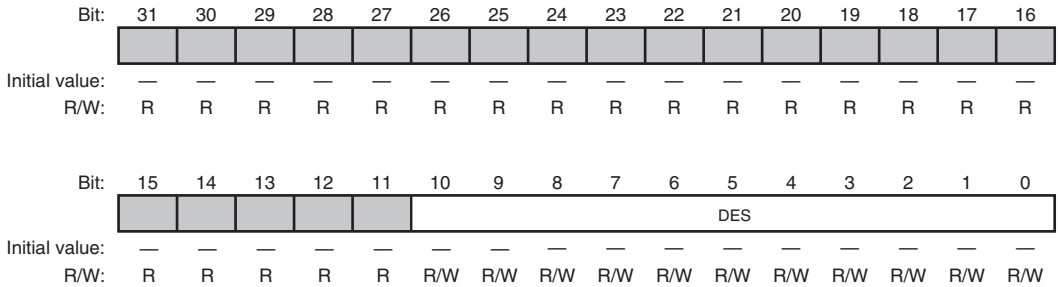
Address: FFF80074



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 11	—	—	R	—	Reserved
10 to 0	CLAMPW	—	R/W	Yes	Clamp Signal Width The high-level width of the CLAMP signal is set in dot clock units. If the HSYNC signal falls while the CLAMP signal is at high level, the CLAMP signal also falls. The value is retained during a reset.

(13) DE Signal Start Register (DESR)

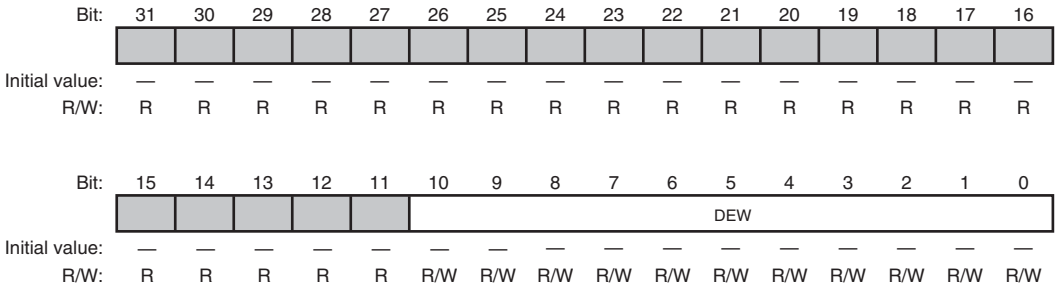
Address: FFF80078



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 11	—	—	R	—	Reserved
10 to 0	DES	—	R/W	Yes	DE Signal Start The DE signal rising edge position is set in dot clock units relative to the falling edge of the HSYNC signal. The DE signal rises after (setting value + 1) cycles of the falling edge of the HSYNC signal. Hence the DE signal cannot be made to rise in the same cycle as the falling edge of the HSYNC signal. During the vertical blanking interval the level is fixed at low level. The value is retained during a reset.

(14) DE Signal Width Register (DEWR)

Address: FFF8007C



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 11	—	—	R	—	Reserved
10 to 0	DEW	—	R/W	Yes	DE Signal Width The high-level width of the DE signal is set in dot clock units. If the HSYNC signal falls while the DE signal is at high level, the DE signal also falls. The value is retained during a reset.

23.2.3 Display Attribute Registers

(1) Color Palette Transparent Color Register (CP1TR)

Address: FFF80080

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CP1IF	CP1IE	CP1ID	CP1IC	CP1IB	CP1IA	CP1I9	CP1I8	CP1I7	CP1I6	CP1I5	CP1I4	CP1I3	CP1I2	CP1I1	CP1I0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

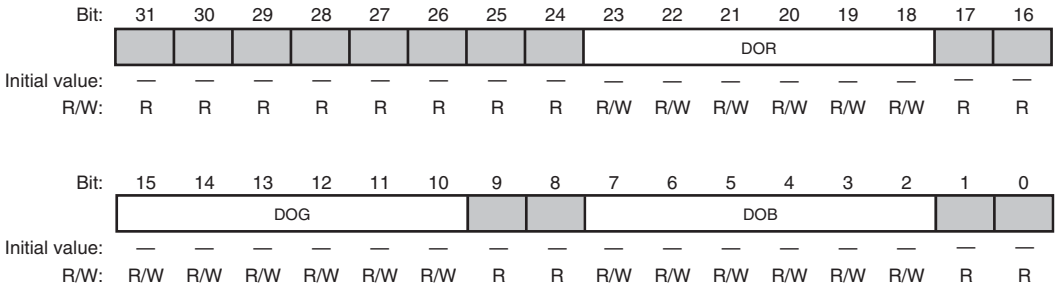
Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 16	—	—	R	—	Reserved
15	CP1IF	0	R/W	Yes	Color Palette Index F 0: Transparency is not set for the color with index F in the color palette. 1: Transparency is set for the color with index F in the color palette.
14	CP1IE	0	R/W	Yes	Color Palette Index E 0: Transparency is not set for the color with index E in the color palette. 1: Transparency is set for the color with index E in the color palette.
13	CP1ID	0	R/W	Yes	Color Palette Index D 0: Transparency is not set for the color with index D in the color palette. 1: Transparency is set for the color with index D in the color palette.
12	CP1IC	0	R/W	Yes	Color Palette Index C 0: Transparency is not set for the color with index C in the color palette. 1: Transparency is set for the color with index C in the color palette.

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
11	CP1IB	0	R/W	Yes	Color Palette Index B 0: Transparency is not set for the color with index B in the color palette. 1: Transparency is set for the color with index B in the color palette.
10	CP1IA	0	R/W	Yes	Color Palette Index A 0: Transparency is not set for the color with index A in the color palette. 1: Transparency is set for the color with index A in the color palette.
9	CP1I9	0	R/W	Yes	Color Palette Index 9 0: Transparency is not set for the color with index 9 in the color palette. 1: Transparency is set for the color with index 9 in the color palette.
8	CP1I8	0	R/W	Yes	Color Palette Index 8 0: Transparency is not set for the color with index 8 in the color palette. 1: Transparency is set for the color with index 8 in the color palette.
7	CP1I7	0	R/W	Yes	Color Palette Index 7 0: Transparency is not set for the color with index 7 in the color palette. 1: Transparency is set for the color with index 7 in the color palette.
6	CP1I6	0	R/W	Yes	Color Palette Index 6 0: Transparency is not set for the color with index 6 in the color palette. 1: Transparency is set for the color with index 6 in the color palette.
5	CP1I5	0	R/W	Yes	Color Palette Index 5 0: Transparency is not set for the color with index 5 in the color palette. 1: Transparency is set for the color with index 5 in the color palette.

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
4	CP1I4	0	R/W	Yes	Color Palette Index 4 0: Transparency is not set for the color with index 4 in the color palette. 1: Transparency is set for the color with index 4 in the color palette.
3	CP1I3	0	R/W	Yes	Color Palette Index 3 0: Transparency is not set for the color with index 3 in the color palette. 1: Transparency is set for the color with index 3 in the color palette.
2	CP1I2	0	R/W	Yes	Color Palette Index 2 0: Transparency is not set for the color with index 2 in the color palette. 1: Transparency is set for the color with index 2 in the color palette.
1	CP1I1	0	R/W	Yes	Color Palette Index 1 0: Transparency is not set for the color with index 1 in the color palette. 1: Transparency is set for the color with index 1 in the color palette.
0	CP1I0	0	R/W	Yes	Color Palette Index 0 0: Transparency is not set for the color with index 0 in the color palette. 1: Transparency is set for the color with index 0 in the color palette.

(2) Display-Off Mode Output Register (DOOR)

Address: FFF80090



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 24	—	—	R	—	Reserved
23 to 18	DOR	—	R/W	Yes	Display Off Mode Output Red These bits indicate red-color data to be output when the display is off (i.e. the DRES and DEN bits in DSYSR are 00). The value is retained during a reset.
17, 16	—	—	R	—	Reserved
15 to 10	DOG	—	R/W	Yes	Display Off Mode Output Green These bits indicate green-color data to be output when the display is off (i.e. the DRES and DEN bits in DSYSR are 00). The value is retained during a reset.
9, 8	—	—	R	—	Reserved
7 to 2	DOB	—	R/W	Yes	Display Off Mode Output Blue These bits indicate blue-color data to be output when the display is off (i.e. the DRES and DEN bits in DSYSR are 00). The value is retained during a reset.
1, 0	—	—	R	—	Reserved

(3) Color Detection Register (CDER)

Address: FFF80094

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									CDR							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

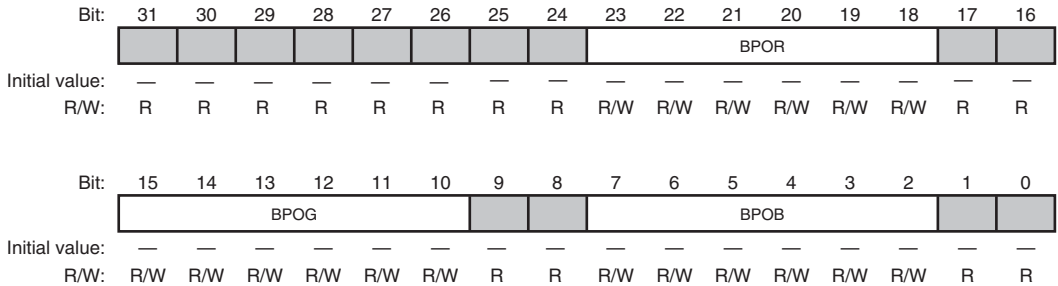
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDG								CDB							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 24	—	—	R	—	Reserved
23 to 18	CDR	—	R/W	Yes	Color Detection Red These bits indicate the red-color data for color detection. The value is retained during a reset.
17, 16	—	—	R	—	Reserved
15 to 10	CDG	—	R/W	Yes	Color Detection Green These bits indicate the green-color data for color detection. The value is retained during a reset.
9, 8	—	—	R	—	Reserved
7 to 2	CDB	—	R/W	Yes	Color Detection Blue These bits indicate the blue-color data for color detection. The value is retained during a reset.
1, 0	—	—	R	—	Reserved

Note: When the output data match the settings of this register, a high-level signal is output from the CDE pin. For information on the output color data format, please refer to section 23.3.6, Output Data Format and Display Capture Data Format.

(4) Background Plane Output Register (BPOR)

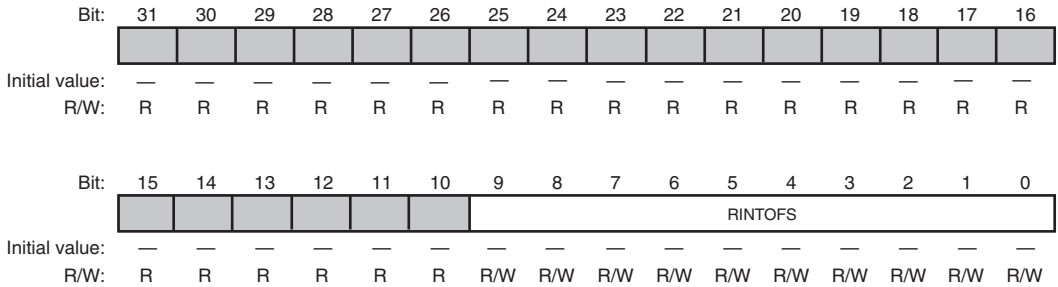
Address: FFF80098



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 24	—	—	R	—	Reserved
23 to 18	BPOR	—	R/W	Yes	Background Plane Output Red These bits indicate red-color data to be output when there is no plane-data for display (due to the size of the display, transparent colors, etc.). The value is retained during a reset.
17, 16	—	—	R	—	Reserved
15 to 10	BPOG	—	R/W	Yes	Background Plane Output Green These bits indicate green-color data to be output when there is no plane-data for display (due to the size of the display, transparent colors, etc.). The value is retained during a reset.
9, 8	—	—	R	—	Reserved
7 to 2	BPOB	—	R/W	Yes	Background Plane Output Blue These bits indicate blue-color data to be output when there is no plane-data for display (due to the size of the display, transparent colors, etc.). The value is retained during a reset.
1, 0	—	—	R	—	Reserved

(5) Raster Interrupt Offset Register (RINTOFSR)

Address: FFF8009C



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 10	—	—	R	—	Reserved
9 to 0	RINTOFS	—	R/W	Yes	Raster Interrupt Offset These bits indicate the raster offset value (H value) relative to the number of raster lines set in VDSR. If the offset is n, the RINT bit in DSSR is set to 1 on the falling edge of HSYNC at the end of the horizontal display interval for the (VDS + n)th raster line. The value is retained during a reset.

23.2.4 Display Plane Registers (Alpha Plane Registers)

In descriptions of registers that are common to planes 1 to 4, the planes are generically referred to as plane n. The meanings of characters n and # are given below.

n: 1, 2, 3, or 4
#: Replaces n in addresses. For example, address FFF80#00 for the plane 3 mode register corresponds to FFF80300.

Descriptions of the registers for the alpha-ratio plane are also given in this section, because a given register for the alpha-ratio plane has almost the same functionality as the corresponding register for the display planes.

Register names:
The names of corresponding registers for the alpha-ratio plane start with the string "Alpha".

Abbreviation:
Abbreviations of the names of corresponding registers for the alpha-ratio plane start with "A".

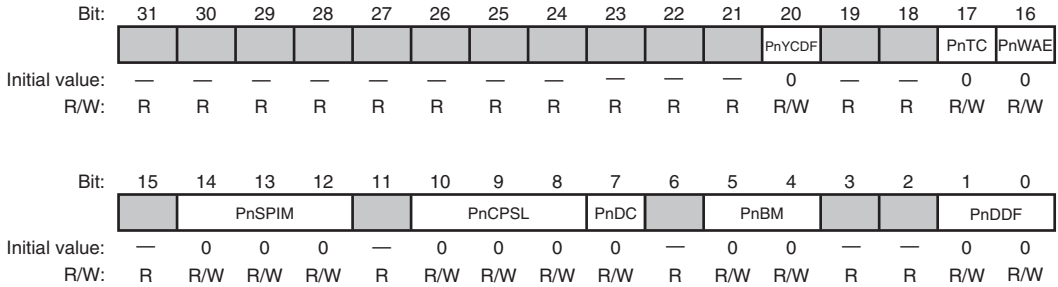
Address:
The addresses of registers for the alpha-ratio plane are given to the right of the address information for the corresponding display-plane registers (# = 1).

The plane n memory width register (PnMWR) for the alpha-ratio plane, for example, is the alpha plane n memory width register (APnMWR).

The alpha plane n mode register (APnMR) is described separately because its functionality differs from that of a plane n mode register (PnMR).

(1) Plane n Mode Register (PnMR)

Address: FFF80#00



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 21	—	—	R	—	Reserved
20	PnYCDF	0	R/W	Yes	Plane n YC Data Format 0: Sets the order of YC data to UYVY format. 1: Sets the order of YC data to YUYV format.
19, 18	—	—	R	—	Reserved
17	PnTC	0	R/W	Yes	Plane n Transparent Color 0: When an 8 bits/pixel display has been selected, the transparent color setting in PnTC1R is enabled 1: When an 8 bits/pixel display has been selected, the transparent color setting in CP1TR is enabled
16	PnWAE	0	R/W	Yes	Plane n Wrap Around Enable 0: Wrapping-around is not performed for plane n 1: Wrapping-around is performed for plane n
15	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
14 to 12	PnSPIM	000	R/W	Yes	Plane n Super Impose Mode 000: Transparent color processing is performed for plane n. When plane n is in the transparent color, the lower plane is displayed. 001: Blending of plane n and the lower plane is performed. When plane n is the transparent color blending is not performed, and the lower plane is displayed. 010: An EOR operation is performed on plane n and the lower plane. When plane n is the transparent color the EOR operation is not performed, and the lower plane is displayed. 011: Setting prohibited 100: Transparent color processing is not performed for plane n. Plane n is displayed. 101: Blending of plane n and the lower plane is performed. The transparent color specification for plane n is ignored, and blending is performed between all the pixels of plane n and the lower plane. 110: An EOR operation is performed on plane n and the lower plane. The transparent color specification for plane n is ignored, and EOR operation is performed on all the pixels of plane n and the lower plane. 111: Setting prohibited
11	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
10 to 8	PnCPSL	000	R/W	Yes	Plane n Color Palette Select These bits indicate whether the color palette is to be used when the value of the PnDDF bits is 00 (i.e. the plane n display data format is 8 bits/pixel). 000: Use the color palette 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7	PnDC	0	R/W	Yes	Plane n Display Area Change 0: In manual display change mode, switching of the frame buffer is not performed. 1: In manual display change mode, switching of the frame buffer is performed. When the PnDC bit is 0, bit setting is possible. Switching is performed in frame units. After frame buffer switching (after vertical blanking detection), this bit is cleared to 0.
6	—	—	R	—	Reserved
5, 4	PnBM	00	R/W	Yes	Plane n Buffer Mode 00: Manual display change mode 01: Auto rendering mode 10: Auto display change mode (blinking mode) 11: Video capture mode In manual display change mode, auto rendering mode, or auto display change mode (blinking mode), double-buffering control is performed using addresses 0 and 1, respectively indicated by the PnDSA0 and PnDSA1 bits in PnDSA0R and PnDSA1R. In video capture mode, triple-buffering control is performed using addresses 0 to 2 indicated by the VCFB bits in DSSR.

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
3, 2	—	—	R	—	Reserved
1, 0	PnDDF	00	R/W	Yes	Plane n Display Data Format 00: 8 bits/pixel mode 01: 16 bits/pixel mode 10: ARGB mode 11: YC mode (4:2:2 YUV is converted to 8:8:8 RGB)

(2) Plane n Memory Width Register (PnMWR)

Address: FFF80#04 (APnMWR: FFF8A#04)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				PnMWX												
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 13	—	—	R	—	Reserved
12 to 4	PnMWX	—	R/W	Yes	Plane n Memory Width X The plane n memory width should be set in the range 16 pixels to 4096 pixels, in 16-pixel units. The value is retained during a reset.
3 to 0	—	—	R	—	Reserved

(3) Plane n Blending Ratio Register (PnALPHAR)

Address: FFF80#08 (no equivalent for the alpha-ratio plane)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			PnABIT			PnBRSL			PnALPHA							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 14	—	—	R	—	Reserved
13, 12	PnABIT	—	R/W	Yes	Plane n A Bit Function Select In the initial state, these bits are fixed to 0. To enable these bits, set the DEFE2G bit in DEFR2 to 1. 00: In ARGB mode (i.e. when the value of the PnDDF bits in PnMR is 10), blending is performed when the A bit is 1. 01: In ARGB mode (i.e. when the value of the PnDDF bits in PnMR is 10), blending is performed when the A bit is 0. 1-: In ARGB mode (i.e. when the value of the PnDDF bits in PnMR is 10), blending is performed regardless of the value of the A bit. The value is retained during a reset.
11		—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
10 to 8	PnBRSL	—	R/W	Yes	Plane n Blending Ratio Select This bit is valid when the value of the PnSPIM bits in PnMR allows blending. In the initial state, bit 10 is fixed to 0. To enable bit 10, set the DEFE bit in DEFR to 1. -00: The value of bits 7 to 0 in this register is taken to be the blending ratio. -01: Setting prohibited -10: Bits 31 to 24 of the color palette register specified by the PnCPSL bits in PnMR are taken to be the blending ratio. Note: This setting is only effective when the display data format specified by the PnDDF bits in PnMR is 8 bits/pixel. For formats other than 8 bits/pixel, the value of bits 7 to 0 in this register is taken to be the blending ratio. 011: The display data for the plane specified by bits 2 to 0 in this register is taken to be the blending ratio. Bits 2 to 0 = 000: Display data for plane 1 is the blending ratio Bits 2 to 0 = 001: Display data for plane 2 is the blending ratio Bits 2 to 0 = 010: Display data for plane 3 is the blending ratio Bits 2 to 0 = 011: Display data for plane 4 is the blending ratio Notes: 1. When the register's own plane is specified, the value of bits 7 to 0 in this register is taken to be the blending ratio. 2. The specified plane should satisfy the following conditions. If the conditions are not satisfied, the blending ratio is undefined. - The display should be turned on by using DPPR. - The display data format should be set to 8 bits/pixel. - The display size should be greater than or equal to the size of the plane for this register. - Ensure that display positions (X and Y) are the same as those in the plane for this register. 111: The display data for the alpha-ratio plane specified by bits 2 to 0 in this register is taken to be the blending ratio. Bits 2 to 0 = 000: Display data for alpha-ratio plane 1 is the blending ratio Bits 2 to 0 = Other: Value of bits 7 to 0 in this register is the blending ratio Take the following steps before setting PnBRSL = 111. 1. Set bit 0 or 4 in DAPCR to 1. 2. Set the alpha plane registers. The value is retained during a reset.

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
7 to 0	PnALPHA	—	R/W	Yes	Plane n Blending Ratio These bits indicate the alpha value (α) which is the blending ratio for plane n. Blending result = (α × plane n + (H'100 − α) × lower plane)/ H'100 Note: Blending result, α, plane n, and lower plane in the above formula are all 8-bit data. The value is retained during a reset.

(4) Plane n Display Size X Register (PnDSXR)

Address: FFF80#10 (APnDSXR: FFF8A#10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PnDSX										
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 11	—	—	R	—	Reserved
10 to 0	PnDSX	—	R/W	Yes	Plane n Display Size X The horizontal-direction display size of plane n should be set in dot clock units. Note: When the YC mode has been selected by the PnDDF bits in PnMR, this value should be set to an even number. The value is retained during a reset.

(5) Plane n Display Size Y Register (PnDSYR)

Address: FFF80#14 (APnDSYR: FFF8A#14)

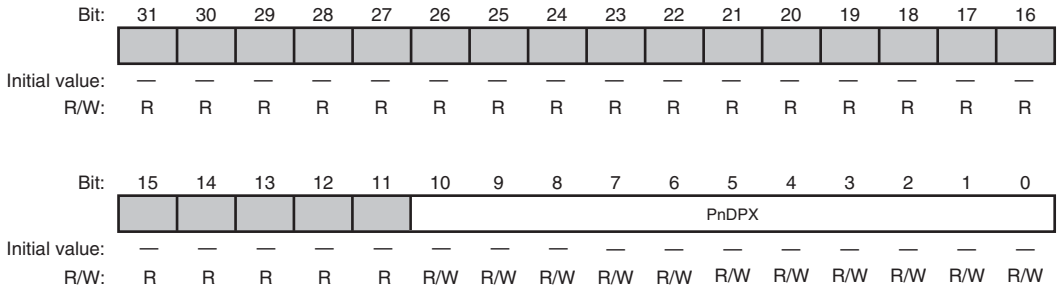
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							PnDSY									
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 10	—	—	R	—	Reserved
9 to 0	PnDSY	—	R/W	Yes	Plane n Display Size Y The vertical-direction display size of plane n should be set in raster line units. The value is retained during a reset.

(6) Plane n Display Position X Register (PnDPXR)

Address: FFF80#18 (APnDPXR: FFF8A#18)



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 11	—	—	R	—	Reserved
10 to 0	PnDPX	—	R/W	Yes	Plane n Display Position X The horizontal start position on the display monitor of plane n should be set in dot clock units, taking as the origin the upper-left corner of the display monitor. The value is retained during a reset.

(7) Plane n Display Position Y Register (PnDPYR)

Address: FFF80#1C (APnDPYR: FFF8A#1C)

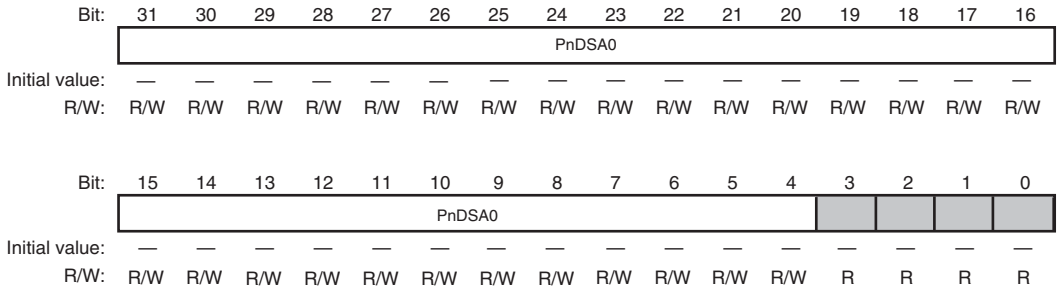
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							PnDPY									
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 10	—	—	R	—	Reserved
9 to 0	PnDPY	—	R/W	Yes	Plane n Display Position Y The vertical start position on the display monitor of plane n should be set in raster line units, taking as the origin the upper-left corner of the display monitor. The value is retained during a reset.

(8) Plane n Display Area Start Address 0 Register (PnDSA0R)

Address: FFF80#20 (APnDSA0R: FFF8A#20)



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 4	PnDSA0	—	R/W	Yes	Plane n Display Area Start Address 0 To enable bits 31 to 29, the DEFE bit in DEFR must be set to 1. In the initial state, these bits are not enabled and are fixed to 0. When the buffer mode for plane n is manual display, auto rendering, auto display change, or video capture, the area indicated by this register is used as frame buffer 0. Note: When bits 31 to 29 are disabled in 32-bit address mode, out of the lower-order 29 bits in a 32-bit memory location to be specified, specify a 25-bit address (A28 to A4) in bits 28 to 4. The value is retained during a reset.
3 to 0	—	—	R	—	Reserved

(9) Plane n Display Area Start Address 1 Register (PnDSA1R)

Address: FFF80#24 (APnDSA1R: FFF8A#24)

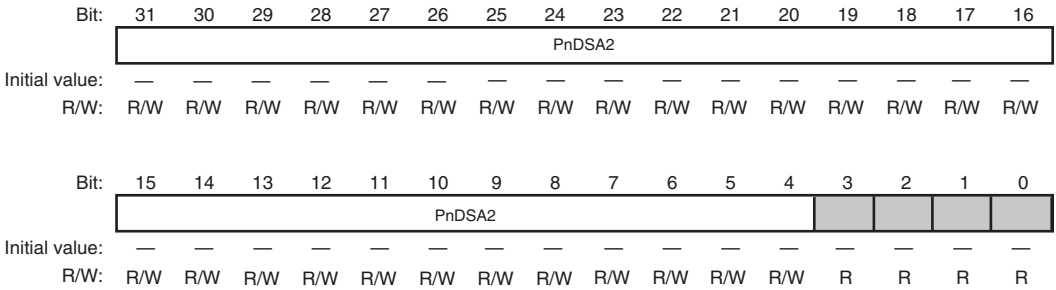
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PnDSA1															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PnDSA1															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 4	PnDSA1	—	R/W	Yes	Plane n Display Area Start Address 1 To enable bits 31 to 29, the DEFE bit in DEFR must be set to 1. In the initial state, these bits are not enabled and are fixed to 0. When the buffer mode for plane n is manual display, auto rendering, auto display change, or video capture, the area indicated by this register is used as frame buffer 1. Note: When bits 31 to 29 are disabled in 32-bit address mode, out of the lower-order 29 bits in a 32-bit memory location to be specified, specify a 25-bit address (A28 to A4) in bits 28 to 4. The value is retained during a reset.
3 to 0	—	—	R	—	Reserved

(10) Plane n Display Area Start Address 2 Register (PnDSA2R)

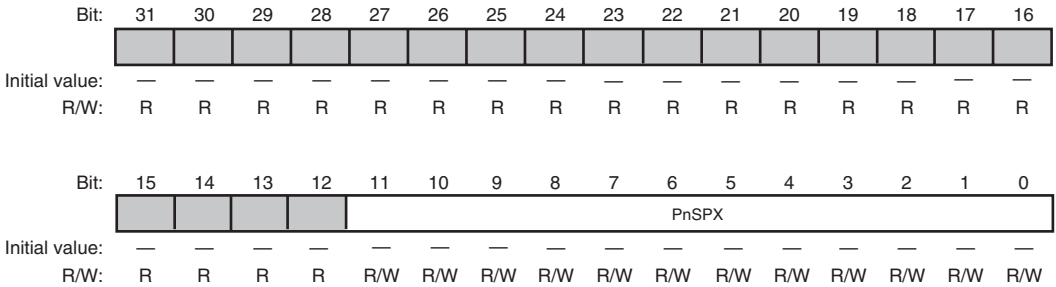
Address: FFF80#28 (APnDSA2R: FFF8A#28)



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 4	PnDSA2	—	R/W	Yes	Plane n Display Area Start Address 2 To enable bits 31 to 29, the DEFE bit in DEFR must be set to 1. In the initial state, these bits are not enabled and are fixed to 0. When the buffer mode for plane n is video capture, the area indicated by this register is used as frame buffer 2. Note: When bits 31 to 29 are disabled in 32-bit address mode, out of the lower-order 29 bits in a 32-bit memory location to be specified, specify a 25-bit address (A28 to A4) in bits 28 to 4. The value is retained during a reset.
3 to 0	—	—	R	—	Reserved

(11) Plane n Start Position X Register (PnSPXR)

Address: FFF80#30 (APnSPXR: FFF8A#30)



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 12	—	—	R	—	Reserved
11 to 0	PnSPX	—	R/W	Yes	Plane n Start Position X Specify the distance in the X direction to the position where plane n starts in memory. Notes: 1. When the YC mode has been selected by the PnDDF bits in PnMR, the value should be even. 2. Setting of a value greater than twice the value of the PnMWX bits is prohibited. The value is retained during a reset.

(12) Plane n Start Position Y Register (PnSPYR)

Address: FFF80#34 (APnSPYR: FFF8A#34)

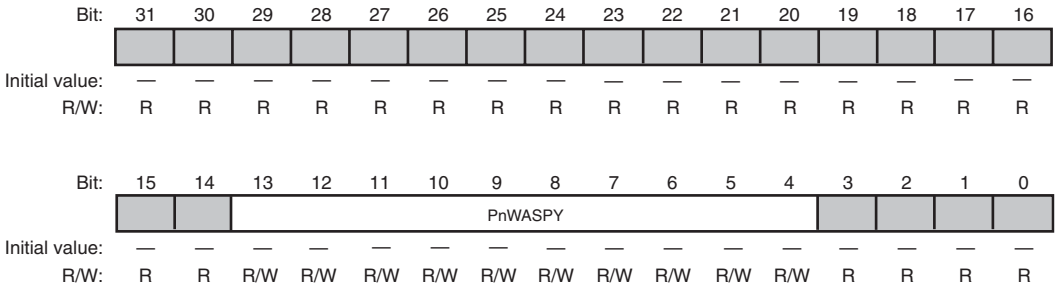
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PnSPY															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 16	—	—	R	—	Reserved
15 to 0	PnSPY	—	R/W	Yes	Plane n Start Position Y Specify the distance in the Y direction to the position where plane n starts in memory. Note: Setting of a value greater than twice {the value of the PnWASPY bits + the value of the PnWAMWY bits} is prohibited. The value is retained during a reset.

(13) Plane n Wrap-Around Start Position Register (PnWASPR)

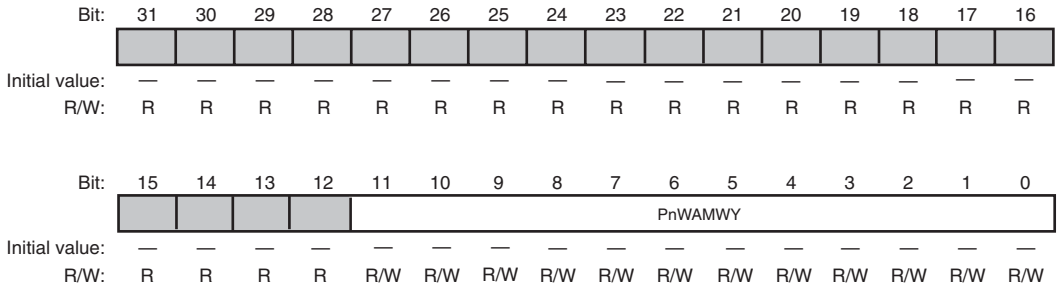
Address: FFF80#38 (APnWASPR: FFF8A#38)



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 14	—	—	R	—	Reserved
13 to 4	PnWASPY	—	R/W	Yes	Plane n Wrap Around Start Position Y The Y direction start position of one wrap-around area should be set with reference to the address specified in PnDSA0R and PnDSA1R. The position where wrapping-around is to start can be set in 16-pixel units (bits 3 to 0: Fixed to 0). The value is retained during a reset.
3 to 0	—	—	R	—	Reserved

(14) Plane n Wrap-Around Memory Width Register (PnWAMWR)

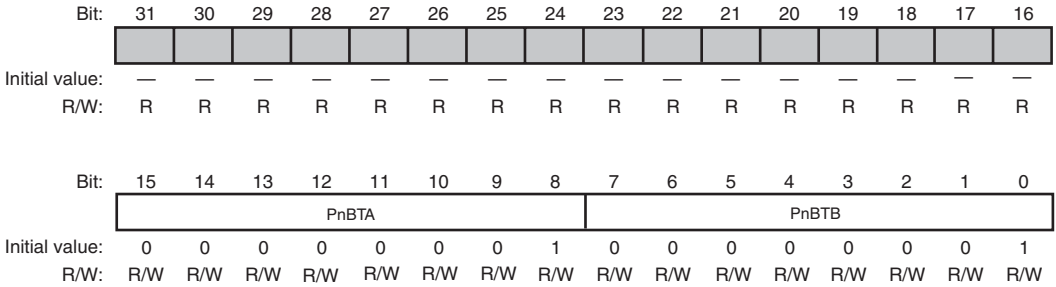
Address: FFF80#3C (APnWAMWR: FFF8A#3C)



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 12	—	—	R	—	Reserved
11 to 0	PnWAMWY	—	R/W	Yes	Plane n Wrap Around Memory Width Y The memory width for wrap-around in the Y-direction should be set to a number corresponding to a value in the range from 240 to 4095 raster lines. The value is retained during a reset.

(15) Plane n Blinking Time Register (PnBTR)

Address: FFF80#40 (APnBTR: FFF8A#40)

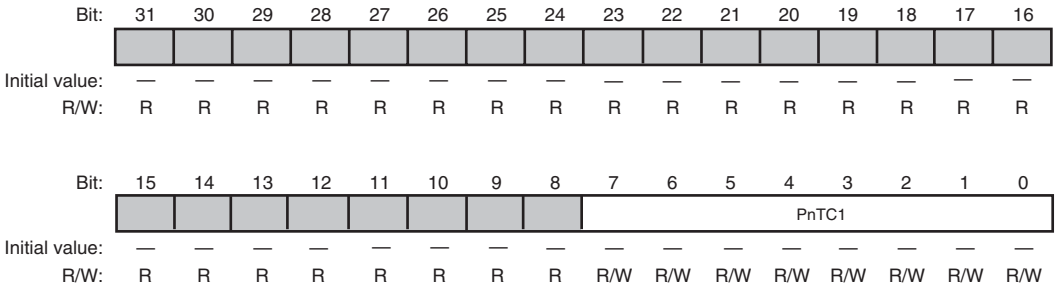


Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 16	—	—	R	—	Reserved
15 to 8	PnBTA	00000001	R/W	Yes	Plane n Blinking Time A
7 to 0	PnBTB	00000001	R/W	Yes	Plane n Blinking Time B

Note: When the PnBM bits in PnMR are set to select the auto display change mode (blinking mode), specify, as numbers of fields, the times over which PnDSA0 and PnDSA1 are to be displayed. Blinking operation employs the settings in PnDSA0 and PnDSA1.
Setting this register to 1 (the value should be other than 0) leads to switching between the buffers at the addresses indicated by PnDSA0 and PnDSA1 on each field unit.

(16) Plane n Transparent Color 1 Register (PnTC1R)

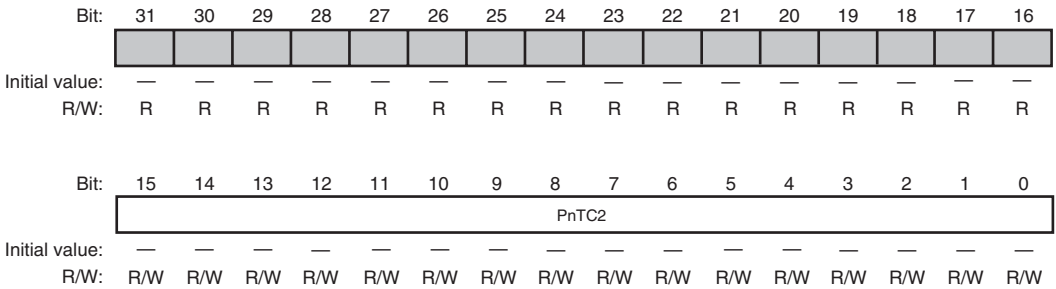
Address: FFF80#44 (no equivalent for the alpha-ratio plane)



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 8	—	—	R	—	Reserved
7 to 0	PnTC1	—	R/W	Yes	Plane n Transparent Color 1 for 8 Bits This setting is for a transparent color for plane n in the 8 bits/pixel data format. To enable the transparent color setting in this register, the PnTC bit in PnMR must be set to 0. The value is retained during a reset.

(17) Plane n Transparent Color 2 Register (PnTC2R)

Address: FFF80#48 (no equivalent for the alpha-ratio plane)



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 16	—	—	R	—	Reserved
15 to 0	PnTC2	—	R/W	Yes	Plane n Transparent Color 2 for 16 Bits This setting is for a transparent color for plane n in the 16 bits/pixel or ARGB data format. In the case of ARGB, comparison is with bits 14 to 0 of this register, i.e. bit 15 is ignored. The value is retained during a reset.

(18) Plane n Memory Length Register (PnMLR)

Address: FFF80#50 (APnMLR: FFF8A#50)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																PnMLY
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PnMLY															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 17	—	—	R	—	Reserved
16 to 0	PnMLY	0	R/W	Yes	Plane n Memory Length Y These bits indicate the length in memory (memory area in the Y-direction) of plane n. When the actual display is beyond this area, the data selected in BPOR will be displayed. When the value is 0 (initial value), the area is handled as an infinite area. Thus the data selected in BPOR will never be displayed.

(19) Alpha Plane n Mode Register (APnMR)

Address: FFF8A#00

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																PnWAE
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									PnDC		PnBM					
Initial value:	—	—	—	—	—	—	—	—	0	—	0	0	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 17	—	—	R	—	Reserved
16	PnWAE	0	R/W	Yes	Plane n Wrap Around Enable 0: Wrapping-around is not performed for plane n 1: Wrapping-around is performed for plane n
15 to 8	—	—	R	—	Reserved
7	PnDC	0	R/W	Yes	Plane n Display Area Change 0: In manual display change mode, switching of the frame buffer is not performed. 1: In manual display change mode, switching of the frame buffer is performed. When the PnDC bit is 0, bit setting is possible. Switching is performed in frame units. After frame buffer switching (after vertical blanking detection), this bit is cleared to 0.
6	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
5, 4	PnBM	00	R/W	Yes	Plane n Buffer Mode 00: Manual display change mode 01: Auto rendering mode 10: Auto display change mode (blinking mode) 11: Video capture mode In manual display change mode, auto rendering mode, or auto display change mode (blinking mode), double-buffering control is performed using addresses 0 and 1, respectively indicated by the PnDSA0 and PnDSA1 bits in PnDSA0R and PnDSA1R. In video capture mode, triple-buffering control is performed using addresses 0 to 2 indicated by the VCFB bits in DSSR.
3 to 0	—	—	R	—	Reserved

Notes: Bit 20 (PnYCDF) is absent because the display data format is fixed to 8 bits/pixel.
Bit 17 (PnTC) is absent because the alpha ratios cannot be treated as a transparent color.
Bits 14 to 12 (PnSPIM) are absent because superpositioning is not available.
Bits 10 to 8 (PnCPSL) are absent because the use of a color palette is not possible.
Bits 1 and 0 (PnDDF) are absent because the display data format is fixed to 8 bits/pixel.

23.2.5 Display Capture Registers

(1) Display Capture Memory Width Register (DCMWR)

Address: FFF8C104

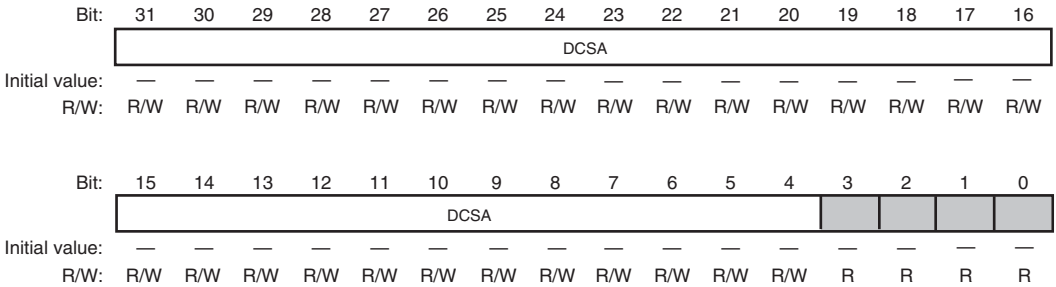
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				DCMWX												
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 13	—	—	R	—	Reserved
12 to 4	DCMWX	—	R/W	Yes	Display Capture Memory Width X Select the memory width for display capture to a value between 16 and 4096 pixels, in 16-pixel units. When the width of captured data exceeds this size, the excess data will not be captured. The value is retained during a reset.
3 to 0	—	—	R	—	Reserved

(2) Display Capture Area Start Address Register (DCSAR)

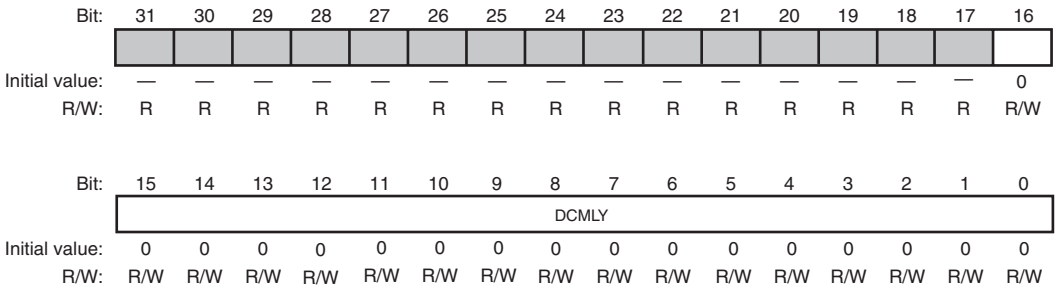
Address: FFF8C120



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 4	DCSA	—	R/W	Yes	Display Capture Area Start Address To enable bits 31 to 29, the DEFE bit in DEFR must be set to 1. In the initial state, these bits are not enabled and are fixed to 0. Note: When bits 31 to 29 are disabled in 32-bit address mode, out of the lower-order 29 bits in a 32-bit memory location to be specified, specify a 25-bit address (A28 to A4) in bits 28 to 4. The value is retained during a reset.
3 to 0	—	—	R	—	Reserved

(3) Display Capture Memory Length Register (DCMLR)

Address: FFF8C150



Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 17	—	—	R	—	Reserved
16 to 0	DCMLY	—	R/W	Yes	Display Capture Memory Length Y Select the memory length (Y-direction memory area) for display capture. When the length of captured data exceeds this size, the excess data will not be captured.

23.2.6 Color Palette Registers

(1) Color Palette 1 Register 000 to 255 (CP1_000R to CP1_255R)

Addresses: FFF81000 to FFF813FC

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CP1_000A								CP1_000R							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CP1_000G								CP1_000B							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

•
•
•

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CP1_255A								CP1_255R							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CP1_255G								CP1_255B							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 24	CP1_000A to CP1_255A	—	R/W	Yes	Color Palette 1_000 to 255 Blending Ratio When the PnBRSL bits in PnALPHAR are 10, the value is the alpha value, which is the blending ratio. The value is retained during a reset.
23 to 18	CP1_000R to CP1_255R	—	R/W	Yes	Color Palette 1_000 to 255 Red Red-color data of color palette 1. The value is retained during a reset.
17, 16	—	—	R	—	Reserved
15 to 10	CP1_000G to CP1_255G	—	R/W	Yes	Color Palette 1_000 to 255 Green Green-color data of color palette 1. The value is retained during a reset.
9, 8	—	—	R	—	Reserved
7 to 2	CP1_000B to CP1_255B	—	R/W	Yes	Color Palette 1_000 to 255 Blue Blue-color data of color palette 1. The value is retained during a reset.
1, 0	—	—	R	—	Reserved

Note: CP1_000R to CP1_255R form a group of 256 registers in which six bits are set for each of the R, G, and B components of a color, setting up the color palette of 256 colors from among the 260,000 possible colors for display. The values are valid for 8-bit/pixel data display.
After the CP1CE bit in CPCR has been set to 1, settings for CP1_000R to CP1_255R become effective on the next falling edge of VSYNC (timing of an internal update) or when the display is reset. When you update the color palette, rewrite the entire palette. Reading of the color palette by the CPU must proceed before the CP1CE bit has been set to 1. The color palette is accessible in longword units.

23.2.7 External Synchronization Control Registers

(1) External Synchronization Control Register (ESCR)

Address: FFF90000

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												DCLK SEL				DCLK DIS
Initial value:	—	—	—	—	—	—	—	—	—	—	—	0	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SYNCSEL				FRQSEL					
Initial value:	—	—	—	—	—	—	0	0	—	—	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31 to 21	—	—	R	—	Reserved
20	DCLKSEL	0	R/W	None	DCLKIN Select 0: The input dot clock source is the DCLKIN pin 1: The input dot clock source is at 166.5 MHz. If this setting is made, ensure that the frequency of the input dot clock is divided by two or a greater value (so that the frequency after division is 83.3 MHz or lower).
19 to 17	—	—	R	—	Reserved
16	DCLKDIS	0	R/W	None	DCLKOUT Disable 0: DCLKOUT is output. 1: DCLKOUT is not output. DCLKOUT is fixed to low level.
15 to 10	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
9, 8	SYNCSEL	00	R/W	None	SYNC Select 00: Phases are not synchronized 01: Phases are not synchronized 10: Phases are synchronized by using the EXVSYNC signal 11: Phases are synchronized by using the EXHSYNC signal
7, 6	—	—	R	—	Reserved
5 to 0	FRQSEL	000000	R/W	None	Frequency Select To enable bit 5, the DEFE bit in DEFR must be set to 1. In the initial state, bit 5 is fixed to 0. 000000: Frequency division of the input dot clock (clock for division) is not performed. 000001: Division by 2 of the input dot clock (clock for division) 000010: Division by 3 of the input dot clock (clock for division) 000011: Division by 4 of the input dot clock (clock for division) 000100: Division by 5 of the input dot clock (clock for division) 000101: Division by 6 of the input dot clock (clock for division) 000110: Division by 7 of the input dot clock (clock for division) 000111: Division by 8 of the input dot clock (clock for division) 001000: Division by 9 of the input dot clock (clock for division) 001001: Division by 10 of the input dot clock (clock for division) 001010: Division by 11 of the input dot clock (clock for division) 001011: Division by 12 of the input dot clock (clock for division) 001100: Division by 13 of the input dot clock (clock for division) 001101: Division by 14 of the input dot clock (clock for division) 001110: Division by 15 of the input dot clock (clock for division) 001111: Division by 16 of the input dot clock (clock for division) 010000: Division by 17 of the input dot clock (clock for division) 010001: Division by 18 of the input dot clock (clock for division) 010010: Division by 19 of the input dot clock (clock for division) 010011: Division by 20 of the input dot clock (clock for division) 010100: Division by 21 of the input dot clock (clock for division) 010101: Division by 22 of the input dot clock (clock for division) 010110: Division by 23 of the input dot clock (clock for division) 010111: Division by 24 of the input dot clock (clock for division) 011000: Division by 25 of the input dot clock (clock for division) 011001: Division by 26 of the input dot clock (clock for division) 011010: Division by 27 of the input dot clock (clock for division) 011011: Division by 28 of the input dot clock (clock for division)

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
5 to 0 (cont.)	FRQSEL	000000	R/W	None	011100: Division by 29 of the input dot clock (clock for division) 011101: Division by 30 of the input dot clock (clock for division) 011110: Division by 31 of the input dot clock (clock for division) 011111: Division by 32 of the input dot clock (clock for division) 100000: Division by 33 of the input dot clock (clock for division) 100001: Division by 34 of the input dot clock (clock for division) 100010: Division by 35 of the input dot clock (clock for division) 100011: Division by 36 of the input dot clock (clock for division) 100100: Division by 37 of the input dot clock (clock for division) 100101: Division by 38 of the input dot clock (clock for division) 100110: Division by 39 of the input dot clock (clock for division) 100111: Division by 40 of the input dot clock (clock for division) 101000: Division by 41 of the input dot clock (clock for division) 101001: Division by 42 of the input dot clock (clock for division) 101010: Division by 43 of the input dot clock (clock for division) 101011: Division by 44 of the input dot clock (clock for division) 101100: Division by 45 of the input dot clock (clock for division) 101101: Division by 46 of the input dot clock (clock for division) 101110: Division by 47 of the input dot clock (clock for division) 101111: Division by 48 of the input dot clock (clock for division) 110000: Division by 49 of the input dot clock (clock for division) 110001: Division by 50 of the input dot clock (clock for division) 110010: Division by 51 of the input dot clock (clock for division) 110011: Division by 52 of the input dot clock (clock for division) 110100: Division by 53 of the input dot clock (clock for division) 110101: Division by 54 of the input dot clock (clock for division) 110110: Division by 55 of the input dot clock (clock for division) 110111: Division by 56 of the input dot clock (clock for division) 111000: Division by 57 of the input dot clock (clock for division) 111001: Division by 58 of the input dot clock (clock for division) 111010: Division by 59 of the input dot clock (clock for division) 111011: Division by 60 of the input dot clock (clock for division) 111100: Division by 61 of the input dot clock (clock for division) 111101: Division by 62 of the input dot clock (clock for division) 111110: Division by 63 of the input dot clock (clock for division) 111111: Division by 64 of the input dot clock (clock for division)

(2) Output Signal Timing Adjustment Register (OTAR)

Address: FFF90004

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DEA				CLAMPA				DRGBA						
Initial value:	—	0	0	0	—	0	0	0	—	0	0	0	—	—	—	—
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						CDEA				DISPA				SYNCA		
Initial value:	—	—	—	—	—	0	0	0	—	0	0	0	—	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
31	—	—	R	—	Reserved
30 to 28	DEA	000	R/W	None	DE Output Timing Adjustment 000: Adjustment of output timing is not performed. The DE signal is output at the rising edge of the dot clock, with the reference timing. 001: The DE signal is output at the rising edge, delayed one dot clock cycle relative to the reference timing. 010: The DE signal is output at the rising edge, delayed two dot clock cycles relative to the reference timing. 011: The DE signal is output at the rising edge, delayed three dot clock cycles relative to the reference timing. 100: The DE signal is output at the falling edge, preceding the reference timing by 1/2 dot clock cycle. 101: The DE signal is output at the falling edge, delayed 1/2 dot clock cycle relative to the reference timing. 110: The DE signal is output at the falling edge, delayed (1+1/2) dot clock cycles relative to the reference timing. 111: The DE signal is output at the falling edge, delayed (2+1/2) dot clock cycles relative to the reference timing.

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
27	—	—	R	—	Reserved
26 to 24	CLAMPA	000	R/W	None	CLAMP Output Timing Adjustment 000: Adjustment of output timing is not performed. The CLAMP signal is output at the rising edge of the dot clock, with the reference timing. 001: The CLAMP signal is output at the rising edge, delayed one dot clock cycle relative to the reference timing. 010: The CLAMP signal is output at the rising edge, delayed two dot clock cycles relative to the reference timing. 011: The CLAMP signal is output at the rising edge, delayed three dot clock cycles relative to the reference timing. 100: The CLAMP signal is output at the falling edge, preceding the reference timing by 1/2 dot clock cycle. 101: The CLAMP signal is output at the falling edge, delayed 1/2 dot clock cycle relative to the reference timing. 110: The CLAMP signal is output at the falling edge, delayed (1+1/2) dot clock cycles relative to the reference timing. 111: The CLAMP signal is output at the falling edge, delayed (2+1/2) dot clock cycles relative to the reference timing.
23	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
22 to 20	DRGBA	000	R/W	None	Digital RGB Output Timing Adjustment 000: Adjustment of output timing is not performed. The digital RGB signal is output at the rising edge of the dot clock, with the reference timing. 001: The digital RGB signal is output at the rising edge, delayed one dot clock cycle relative to the reference timing. 010: The digital RGB signal is output at the rising edge, delayed two dot clock cycles relative to the reference timing. 011: The digital RGB signal is output at the rising edge, delayed three dot clock cycles relative to the reference timing. 100: The digital RGB signal is output at the falling edge, preceding the reference timing by 1/2 dot clock cycle. 101: The digital RGB signal is output at the falling edge, delayed 1/2 dot clock cycle relative to the reference timing. 110: The digital RGB signal is output at the falling edge, delayed (1+1/2) dot clock cycles relative to the reference timing. 111: The digital RGB signal is output at the falling edge, delayed (2+1/2) dot clock cycles relative to the reference timing.
19 to 11	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
10 to 8	CDEA	000	R/W	None	CDE Output Timing Adjustment 000: Adjustment of output timing is not performed. The CDE signal is output at the rising edge of the dot clock, with the reference timing. 001: The CDE signal is output at the rising edge, delayed one dot clock cycle relative to the reference timing. 010: The CDE signal is output at the rising edge, delayed two dot clock cycles relative to the reference timing. 011: The CDE signal is output at the rising edge, delayed three dot clock cycles relative to the reference timing. 100: The CDE signal is output at the falling edge, preceding the reference timing by 1/2 dot clock cycle. 101: The CDE signal is output at the falling edge, delayed 1/2 dot clock cycle relative to the reference timing. 110: The CDE signal is output at the falling edge, delayed (1+1/2) dot clock cycles relative to the reference timing. 111: The CDE signal is output at the falling edge, delayed (2+1/2) dot clock cycles relative to the reference timing.
7	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
6 to 4	DISPA	000	R/W	None	DISP Output Timing Adjustment 000: Adjustment of output timing is not performed. The DISP signal is output at the rising edge of the dot clock, with the reference timing. 001: The DISP signal is output at the rising edge, delayed one dot clock cycle relative to the reference timing. 010: The DISP signal is output at the rising edge, delayed two dot clock cycles relative to the reference timing. 011: The DISP signal is output at the rising edge, delayed three dot clock cycles relative to the reference timing. 100: The DISP signal is output at the falling edge, preceding the reference timing by 1/2 dot clock cycle. 101: The DISP signal is output at the falling edge, delayed 1/2 dot clock cycle relative to the reference timing. 110: The DISP signal is output at the falling edge, delayed (1+1/2) dot clock cycles relative to the reference timing. 111: The DISP signal is output at the falling edge, delayed (2+1/2) dot clock cycles relative to the reference timing.
3	—	—	R	—	Reserved

Bit	Bit Name	Initial Value	R/W	Internal Update	Description
2 to 0	SYNCA	000	R/W	None	SYNC* Output Timing Adjustment 000: Adjustment of output timing is not performed. The SYNC* signal is output at the rising edge of the dot clock, with the reference timing. 001: The SYNC* signal is output at the rising edge, delayed one dot clock cycle relative to the reference timing. 010: The SYNC* signal is output at the rising edge, delayed two dot clock cycles relative to the reference timing. 011: The SYNC* signal is output at the rising edge, delayed three dot clock cycles relative to the reference timing. 100: The SYNC* signal is output at the falling edge, preceding the reference timing by 1/2 dot clock cycle. 101: The SYNC* signal is output at the falling edge, delayed 1/2 dot clock cycle relative to the reference timing. 110: The SYNC* signal is output at the falling edge, delayed (1+1/2) dot clock cycles relative to the reference timing. 111: The SYNC* signal is output at the falling edge, delayed (2+1/2) dot clock cycles relative to the reference timing. Note: * HSYNC, VSYNC, CSYNC, and ODDF signals

Note: When signals are to be output on the falling edge, the electrical characteristics do not apply.

23.3 Display Function

23.3.1 Configuration of Output Screen

The display unit (DU) executes window displays with up to a maximum of four window layers. Each of these windows is called a "plane", and the order of stacking of the planes can be set arbitrarily. For each plane, display can be turned on and off, and the display data format (8 bits/pixel, 16 bits/pixel, ARGB, YC), blending functions, and other settings can be changed independently. In addition to the window display plane, a plane that is dedicated to the α value can be synthesized (a maximum of one layer).

Each plane has a double-buffer configuration (triple buffer only for a video capture plane), so that smooth display is possible.

Note: In cases of high-resolution display, the unified memory traffic volume may be considerable depending on the number of combined planes and display size, and constraints may arise owing to the traffic volume; but there are no constraints on display functions.

Table 23.3 Display Functions of Planes

		Display Data Format								
	Display On/Off	8 bits/ pixel	16 bits/ pixel	ARGB	YC	Superpositioning	Blink-ing	Size	Scroll -ing	Wrap-around
Plane 1	O	O*[1]	O	O	O*[2]	α blending/ transparent color/ EOR operation	O	X, Y arbitrary	O	O
Plane 2	O	O*[1]	O	O	O*[2]	α blending/ transparent color/ EOR operation	O	X, Y arbitrary	O	O
Plane 3	O	O*[1]	O	O	O*[2]	α blending/ transparent color/ EOR operation	O	X, Y arbitrary	O	O
Plane 4	O	O*[1]	O	O	O*[2]	α blending/ transparent color/ EOR operation	O	X, Y arbitrary	O	O
α plane 1	O	O	×	×	×	×	O	X, Y arbitrary	O	O
Back-ground color*[3]	×	×	×	×	×	×	×	×	×	×

Notes: 1. Any among the color the palette 1, 2, 3, 4 is selected.
2. YC→RGB conversion can be performed only for the YUV data of the uppermost plane.
3. The data format for background color is RGB:6·6·6.

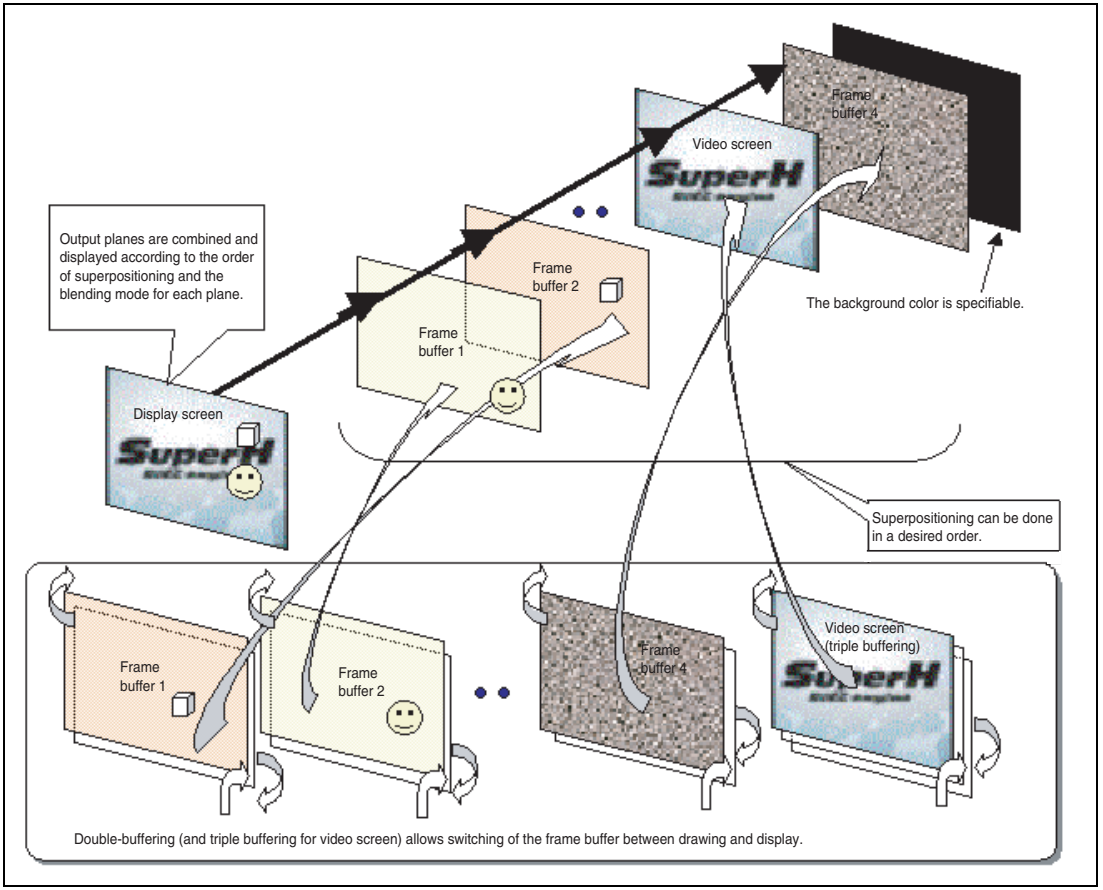


Figure 23.2 Block Diagram of Plane Configuration and Superpositioning

23.3.2 Display On/Off

All plane display can be turned on and off using the DEN bit in DSYSR. When the DEN bit is 0, the display data set in DOOR is displayed.

Display is turned on and off for planes 1 to 4 using DPPR. Under the following display conditions, display data set in BPOR is displayed.

1. When display of all planes 1 to 4 is turned off
2. In an area with no plane for display, due to the display size and display position
3. When the pixels in a plane for display are all a transparent color

Table 23.4 Display On/Off of Plane 1 to 4

Display Plane	Display Plane Priority Register (DPPR)
Plane 1	Plane 1 is selected in one among priority positions 1 to 8, and the corresponding enable bit is set to 1
Plane 2	Plane 2 is selected in one among priority positions 1 to 8, and the corresponding enable bit is set to 1
Plane 3	Plane 3 is selected in one among priority positions 1 to 8, and the corresponding enable bit is set to 1
Plane 4	Plane 4 is selected in one among priority positions 1 to 8, and the corresponding enable bit is set to 1

Note: Even if display on is set using DPPR, under the following conditions, the setting is handled as display off, and the corresponding plane is not displayed.

- Planes for which the value set in PnDPXR is greater than the screen size (horizontal display end register (HDE) - horizontal display start register (HDS))
- Planes for which the value set in PnDPYR is greater than the screen size (vertical display end register (VDE) - vertical display start register (VDS))
- Planes for which the value set in PnDSXR is 0
- Planes for which the value set in PnDSYR is 0
- Planes for which the value set in PnMWR is 0
- Planes for which the value set in PnSPXR is equal to or greater than twice the value set in PnMWR

23.3.3 Plane Parameter

For each plane, a display area start position, memory width, display start position, and display size are set using registers.

The followings are the schematic diagram of start positions and sizes related to planes and the registers used for setting start positions and sizes.

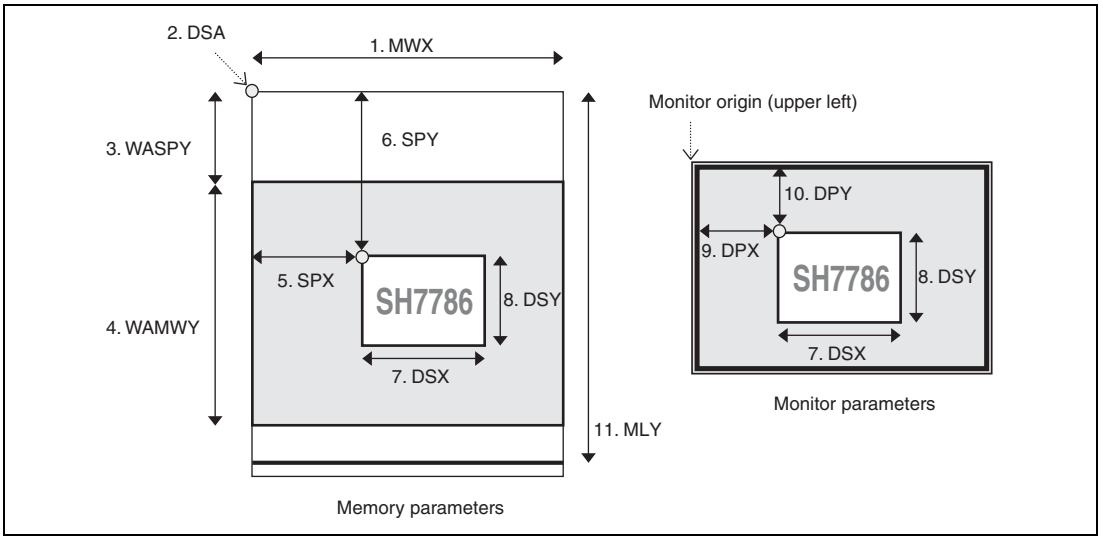


Figure 23.3 Parameters

Table 23.5 Memory Parameter/ Monitor Parameter Setting Registers

No.	Name Used in the Figure	Register	Description
1	MWX (Plane memory width)	PnMWR	The plane X-direction memory width is set between 16 and 4096 pixels, in 16 pixel units.
2	DSA (Display area start address)	PnDSA0R to PnDSA2R	The start address in memory area is set for plane n.
3	WASPY (Plane n wrap-around start position)	PnWASPR	The Y direction start position of the plane n wrap-around area is set in pixel units, with the address set by DSA as reference.
4	WAMWY (Wrap-around memory width)	PnWAMWR	The wrap-around Y-direction memory width is set arbitrarily in the range 240 to 4095 lines.
5	SPX (Start position X)	PnSPXR	The distance in the X direction to the display start position is set in pixel units, taking the address set by DSA as the origin.
6	SPY (Start position Y)	PnSPYR	The distance in the Y direction to the display start position is set in raster line units, taking the address set by DSA as the origin.
7	DSX (Display size X)	PnDSXR	The X-direction display size of plane n is set in pixel units.
8	DSY (Display size Y)	PnDSYR	The Y-direction display size of plane n is set in raster units.
9	DPX (Display position X)	PnDPXR	The X-direction distance to the display position is set in pixel units, taking the upper-left corner of monitor as the origin.
10	DPY (Display position Y)	PnDPYR	The Y-direction distance to the display position is set in raster units, taking the upper-left corner of monitor as the origin.
11	MLY (Memory length Y)	PnMLR	The Y-direction memory area of plane n is set in raster units.

23.3.4 Memory Allocation

A display start address for the display screen, drawing screen 1, and drawing screen 2 used for video display can be set individually for each plane. The start addresses for the memory areas used are set in each of the display area start address registers.

In the display unit (DU), when the display plane is video captured, the display area start addresses 0, 1, and 2 for each plane are used to perform triple-buffering control and display the plane. When the display plane is not video captured, the display area start addresses 0 and 1 for each plane are used to perform double-buffering control and display the plane.

The buffering control that is the same as for the display plane can be performed for the alpha-ratio plane. In display capture, only a single area can be set to store the data.

Below is a list of display area start address registers used for each of the planes.

Table 23.6 Memory Allocation Registers

Display Screen	Register Name	
Plane 1	Plane 1 display area start address register 0	P1DSA0
	Plane 1 display area start address register 1	P1DSA1
	Plane 1 display area start address register 2	P1DSA2
Plane 2	Plane 2 display area start address register 0	P2DSA0
	Plane 2 display area start address register 1	P2DSA1
	Plane 2 display area start address register 2	P2DSA2
Plane 3	Plane 3 display area start address register 0	P3DSA0
	Plane 3 display area start address register 1	P3DSA1
	Plane 3 display area start address register 2	P3DSA2
Plane 4	Plane 4 display area start address register 0	P4DSA0
	Plane 4 display area start address register 1	P4DSA1
	Plane 4 display area start address register 2	P4DSA2
α plane 1	Alpha plane 1 display area start address register 0	AP1DSA0
	Alpha plane 1 display area start address register 1	AP1DSA1
	Alpha plane 1 display area start address register 2	AP1DSA2
Display capture	Display capture area start address register	DCSAR

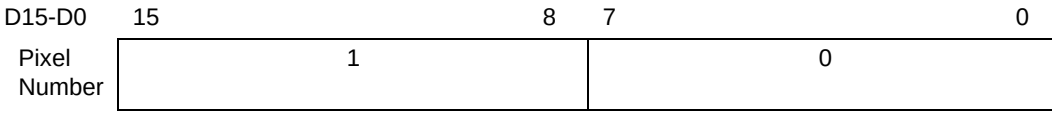
23.3.5 Display Data Format

The following format is used for color data used in display. A data configuration is shown in which data is allocated to the unified memory in little endian.

- 8 bits/pixel

  A color palette index is used. The color palette is used to convert and display image data into RGB data with 6 bits for each RGB color.

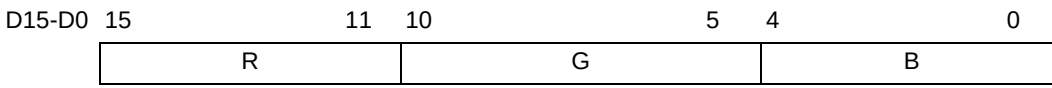
8 bits/pixel data (index color)



- 16 bits/pixel: RGB

  The RGB levels are represented using 5 bits for R, 6 bits for G, and 5 bits for B.

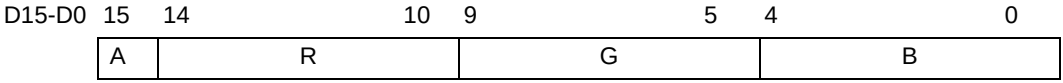
16 bits/pixel data (RGB data) format



- 16 bits/pixel: ARGB

  The ARGB levels are represented using A:1, R:5, G:5, B:5 bits. In addition to the RGB values, an alpha ratio is set. Blending control using the A value is valid when the PnSPIM bit in PnMR is set to perform blending. When the PnABIT bits of (PnALPHAR)/PnALPHA are 00, α blending is performed when A = 1. When the PnABIT bits are 01, α blending is performed when A = 0. When the PnABIT bits are 10 or 11, α blending is performed regardless of A value. When the PnSPIM bit is not set to perform blending, blending is not performed regardless of A value.

16 bits/pixel data (ARGB data) format

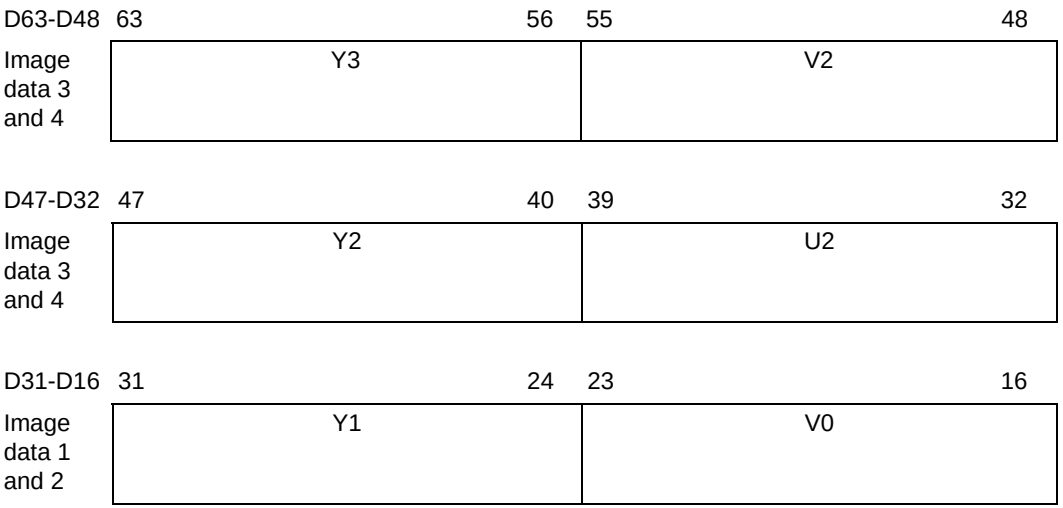


- YC

  Image data has the format YC (YCbCr) = 4:2:2. A calculation circuit is used to convert each of the 8 bits of the RGB colors of image data.

  The YC data order corresponds to the UYVY format and YUYV format. The UYVY format and YUYV format can be selected using the PnYCDF bits in PnMR.

(a) UYVY format



D15-D0	15	8	7	0
Image data 1 and 2	Y0		U0	

(b) YUYV format

D63-D48	63	56	55	48
Image data 3 and 4	V2		Y3	

D47-D32	47	40	39	32
Image data 3 and 4	U2		Y2	

D31-D16	31	24	23	16
Image data 1 and 2	V0		Y1	

D15-D0	15	8	7	0
Image data 1 and 2	U0		Y0	

23.3.6 Output Data Format and Display Capture Data Format

When outputting digital RGB data from the display unit (DU), the display data format is expanded into the RGB888 format before output. When data is output to the pin, the upper 6 bits from among 8 bits that have been expanded are output. The upper 5 bits each from the red and blue pin, and 6 bits from the green pin are stored as display capture data. The format at the time of output is as indicated in the following table.

Table 23.7 Output Data Format

	Red							Green							Blue						
RGB Pin Output Data	**5**	**4**	**3**	**2**	**1**	**0**			**5**	**4**	**3**	**2**	**1**	**0**			**5**	**4**	**3**	**2**	**1**	**0**	
8 bits/pixel	R (6 bits)						0	0	G (6 bits)						0	0	B (6 bits)						0	0
16 bits/pixel	R (5 bits)					0	0	0	G (6 bits)						0	0	B (5 bits)					0	0	0
ARGB	R (5 bits)					0	0	0	G (5 bits)					0	0	0	B (5 bits)					0	0	0
YC->RGB	R (6 bits)						0	0	G (6 bits)						0	0	B (6 bits)						0	0
Display capture	R (5 bits)								G (6 bits)								B (5 bits)							

23.3.7 Endian Conversion

The display unit (DU) can perform big-endian/little-endian conversion according to the setting of the DSEC bit in DSYSR.

The internal data format in the display unit (DU) is fixed at little-endian; by setting the DSEC bit in DSYSR to 1, display data arranged in big-endian format in memory is converted into little-endian format and read.

The units for endian conversion (byte/word) is determined by the setting of the PnDDF bit in PnMR.

Table 23.8 Endian Conversion

PnMR/PnDDF	Data Format	Units for Endian Conversion
00	8 bits/pixel	Byte
01	16 bits/pixel (RGB data)	Word
10	ARGB mode	Word
11	YC mode	Byte

Endian conversion in each of the units indicated below is shown in figure 23.4.

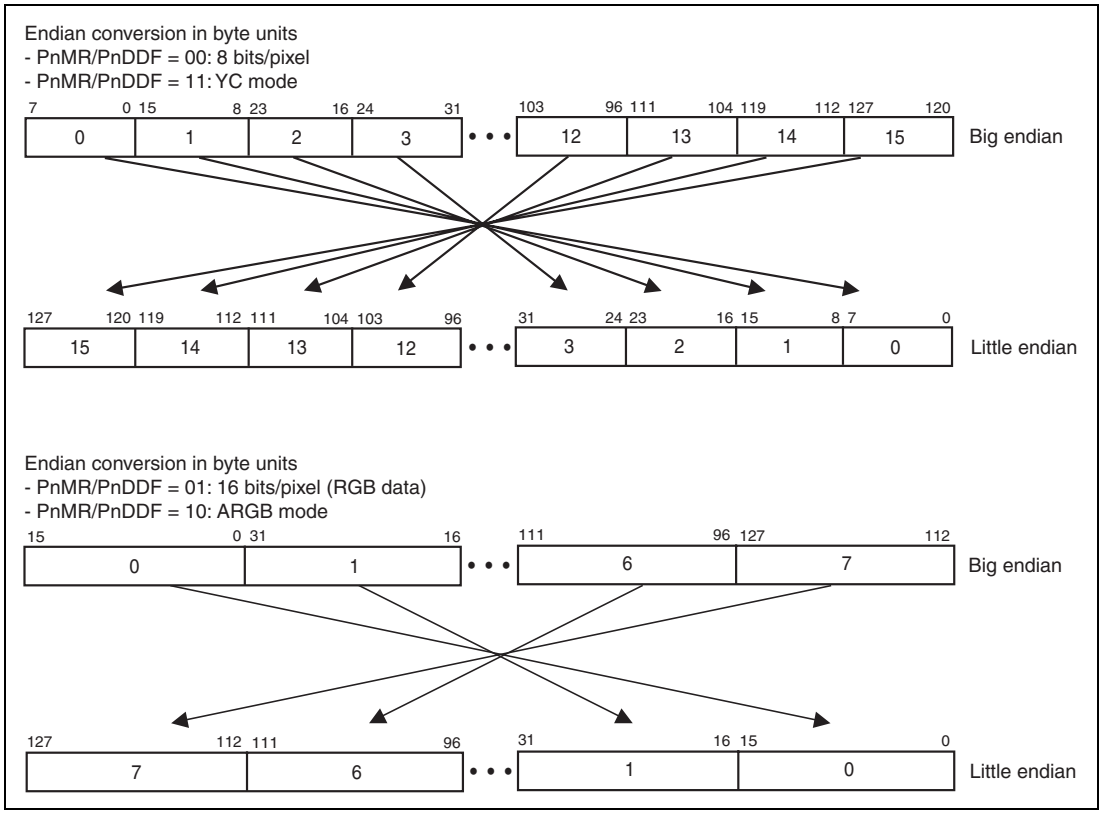


Figure 23.4 Endian Conversion

23.3.8 Color Palette

8 bits/pixel data employs color palettes.

The color palette can be set using the PnCPSL bits in PnMR. Each of the color palettes consists of two alternate buffers; one serves as a display buffer, and the other is for CPU access. After setting each color palette, by setting the color palette switching enable bits CP1CE in CPCR to 1, the color palette thus set becomes valid at the next VSYNC falling edge (internal update timing), or upon display reset (when the DRES bit in DSYSR is changed from 1 to 0).

Notes on Use of Color Palettes:

1. Because palettes consist of alternate buffers, complete overwriting is necessary upon a color palette update. However, when the details of color palette updates are being managed, there is no problem with overwriting only the relevant part.
2. Upon completion of color palette settings, the switching enable bit must always be set to 1.
3. When reading a color palette from the CPU, reading should be performed before setting the switching enabled bit to 1.

Procedure for Setting a Color Palette:

- Procedure for switching from the initial state
  The initial state (after power-on reset) is the display reset state.
  A. Set the display control register.
  B. Set the color palette.
  C. After setting the color palette, set the color palette switching enable bit to 1.
  D. Cancel the display reset.
- Procedure for switching from display state
  In the display state, the DRES bit and DEN bit in the DSYSR are 0 and 1 respectively.
  A. Confirm that the color palette switching enable bit is 0.
  B. Set the color palette.
  C. After setting the color palette, set the color palette switching enable bit to 1.

23.3.9 Superpositioning of Planes

For each plane, three types of combined superpositioning are possible: α blending, transparent colors, and EOR operations. By setting the PnSPIM bits in PnMR, the superpositioned display type can be selected.

However, α blending and EOR operation cannot be performed simultaneously on the same plane.

Table 23.9 Superpositioning

PnSPIM	Superpositioning
000	Transparency processing is performed for the specified plane. When the specified plane is a transparent color, the lower plane is displayed. (Initial value)
001	Blending of the specified plane with the lower plane is performed. When the specified plane is a transparent color, blending is not performed and the lower plane is displayed.
010	EOR operation of the specified plane and the lower plane is performed. When the specified plane is a transparent color, EOR operation is not performed and the lower plane is displayed.
011	Setting prohibited (The lower plane is displayed.)
100	Transparency processing is not performed for the specified plane. The specified plane is displayed.
101	Blending of the specified plane with the lower plane is performed. Transparent color specification for the specified plane is ignored, and blending of all the pixels in the specified plane with the lower plane is performed.
110	EOR operation of the specified plane and the lower plane is performed. Transparent color specification for the specified plane is ignored, and EOR operation of all the pixels in the specified plane and the lower plane is performed.
111	Setting prohibited (The lower plane is displayed.)

After the display format has been expanded to R:G:B = 8:8:8, α blending or EOR operation is performed. The complementary format of each display data format is shown in the next table.

Table 23.10 RGB888 Bit Configuration in Each Display Data Format

Data Format	R (8 bits)	G (8 bits)	B (8 bits)
8 bits/pixel	R (6 bits) 0 0	G (6 bits) 0 0	B (6 bits) 0 0
16 bits/pixel	R (5 bits) 0 0 0	G (6 bits) 0 0	B (5 bits) 0 0 0
ARGB	R (5 bits) 0 0 0	G (5 bits) 0 0 0	B (5 bits) 0 0 0
YC→RGB	R (8 bits)	G (8 bits)	B (8 bits)

α blending and EOR operation are performed in the sequence of the lower plane to the upper plane. The block diagram is shown below.

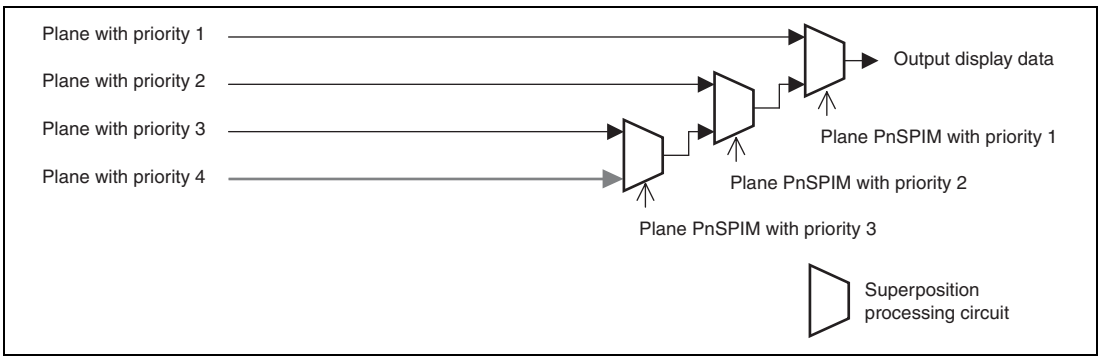


Figure 23.5 Plane Processing Sequence in α Blending and EOR Operation

When the format of display data for α blending or EOR operation is 8 bits/pixel, after selection in advance of the color palette to be used, the α blending or EOR operation on/off should be specified. At this time, when both planes for α blending or for EOR operation have the same color palette selected (color palette contention), only the specified plane is displayed, with no α blending or EOR operation performed. When display of all lower planes is turned off, the specified plane is displayed. That is, blending or EOR operation of the specified plane with the display data specified in BPOR is not performed.

α Blending: In α blending, blending processing is performed according to the alpha (α) ratio set by the PnALPHA bits in PnALPHAR, the alpha (α) ratio set by the blending ratio bits in the color palette, or the alpha (α) ratio of the display data of the display plane, or the data of the plane that is dedicated to the α value.

$$\text{Blending result} = (\alpha \times \text{specified plane} + (\text{H'100}-\alpha) \times \text{lower plane})/\text{H'100}$$

In the above formula, the blending result, α, the specified plane, and the lower plane are all given as 8-bit data.

When 0 is set as the alpha ratio, only the lower plane is displayed. When the PnDDF bit in PnMR is set to ARGB, and moreover the PnSPIM bit in PnMR is set to perform blending, α blending is performed according to the A value of the output ARGB data format and by the a value specified by (PnALPHAR)/PnALPHA.

When the PnABIT bits of (PnALPHAR)/PnALPHA are 00, α blending is performed when A value is 1. When the PnABIT bits are 01, α blending is performed when A value is 1. When PnABIT bits are 10 or 11, α blending is performed regardless of A value.

Transparent Colors: For each plane, transparent color processing can be performed between the specified plane and the lower plane by setting PnSPIM bit in PnMR to 0. However, transparent color processing cannot be performed in YC mode.

- In 8 bits/pixel mode

  When the PnTC bit in PnMR is 0 (initial value), transparent color processing is performed according to the setting in the plane n transparent color 1 register (PnTC1R). When the PnTC bit in PnMR is 1, a color that is specified by CPT1R can be set as a transparent color. Only the indexes H'0x00 to H'0x0F can be specified as transparent colors; H'0x 10 to H'0x FF cannot be specified as transparent colors.

  The color palette transparent color register can be selected using the PnCPSL bits in PnMR.
- In 16 bits/pixel mode and ARGB mode

  Transparent color processing is performed according to PnTC2R, regardless of the setting of the PnTC bit in PnMR.

  In the case of ARGB, bits 14 to 0 of PnTC2R are compared, and bit 15 is ignored.

The above is summarized in table 23.11, which indicates the transparent color specification registers which are valid when the PnTC bit is 0 and 1.

Table 23.11 Transparent Color Specification Registers

Data Format	Transparent Color Specification Bit	Color Palette Select Bit	Transparent Color Specification Register
—	(PnMR) /PnTC	(PnMR) /PnCPSL	—
8 bits/pixel	0	—	PnTC1R
	1	000	CP1TR
16 bits/pixel	—	—	PnTC2R
ARGB	—	—	PnTC2R

EOR Operation: EOR operation of the specified plane with the lower plane is performed.

YC Data Contention: The display unit (DU) has only one YC-RGB conversion circuit internally, and so YC-RGB conversion cannot be performed simultaneously for two or more planes. When there are pixels requiring YC-RGB conversion on two or more planes simultaneously, the pixels on the uppermost plane are YC-RGB converted, and the lower plane is not displayed.

Figure 23.6 describes YC-RGB conversion when the data for three planes is in YC format.

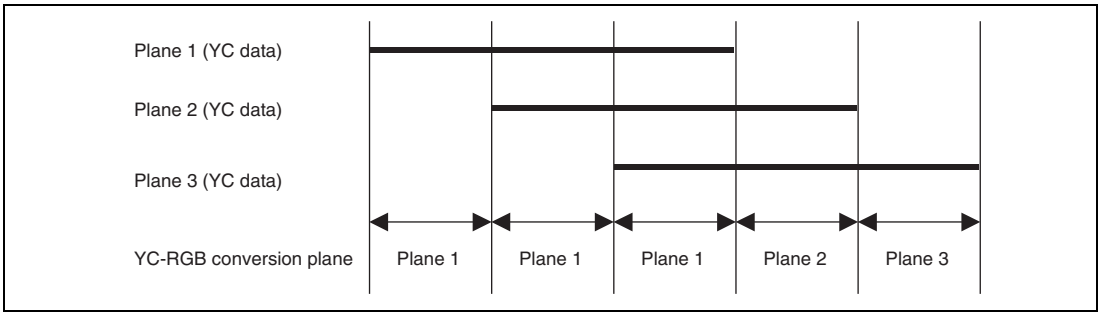


Figure 23.6 YC Data Contention

Plane Priority Order: The display priority order for planes is set using DPPR; if one plane is set in two or more places in the priority order, the place with highest priority is selected.

For example, if the setting in DPPR is H'00CBD888, then the results of the priority order and display on/off settings are as follows.

Plane with priority 1	Plane 1
Plane with priority 2	No corresponding plane
Plane with priority 3	No corresponding plane
Plane with priority 4	Plane 4
Display off planes	Plane 2 and plane 3

23.3.10 Blinking

For each plane, blinking operation can be performed by using the display area start addresses 0 and 1.

Usually, double-buffering control is performed for each plane according to the setting of the PnBM bit in PnMR. However, blinking is performed with the period specified by the PnBTA and PnBTB bits in PnBTR by setting the PnBM bits in PnMR to 10 (auto display change mode (blinking mode)). When the blinking period is set to 1, the display area start addresses 0 and 1 can be switched for every VSYNC; the same function as the auto display change mode can be achieved.

Note: Set a value other than 0 to the PnBTA and PnBTB bits in PnBTR.

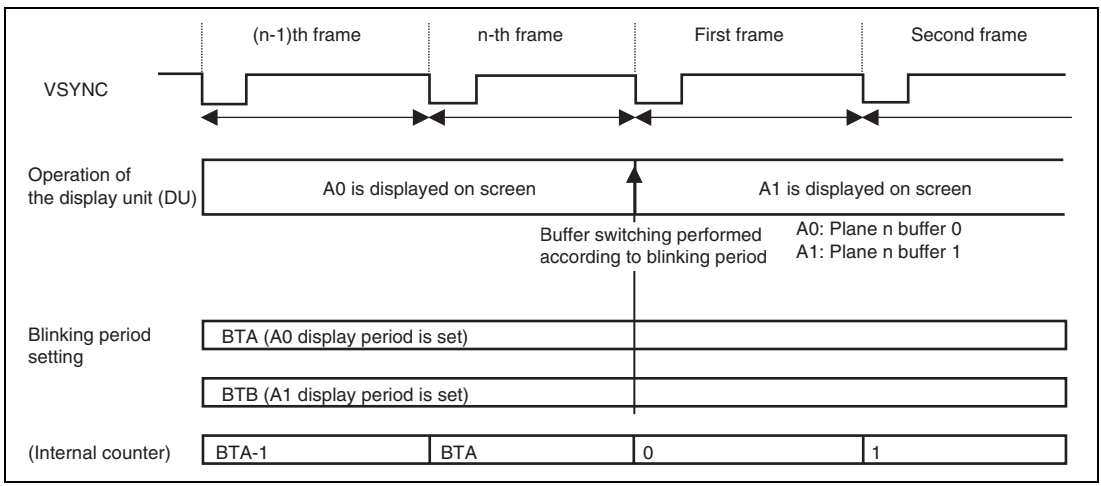


Figure 23.7 Blinking

23.3.11 Scroll Display

By setting display area and display screen sizes and start positions independently for each plane, smooth scroll processing can be performed independently for each plane.

The display can be scrolled by cyclically setting the plane n display start position X, Y values (coordinates specified by PnSPXR and PnSPYR), taking as the origin the start address in memory specified by PnDSA0R to PnDSA2R for each plane.

Figure 23.8 summarizes display scrolling. The display is scrolled by setting the display start position from A to B.

Note: Display sizes and other area settings for each plane should be set such that there is no data display outside the memory configuration area.

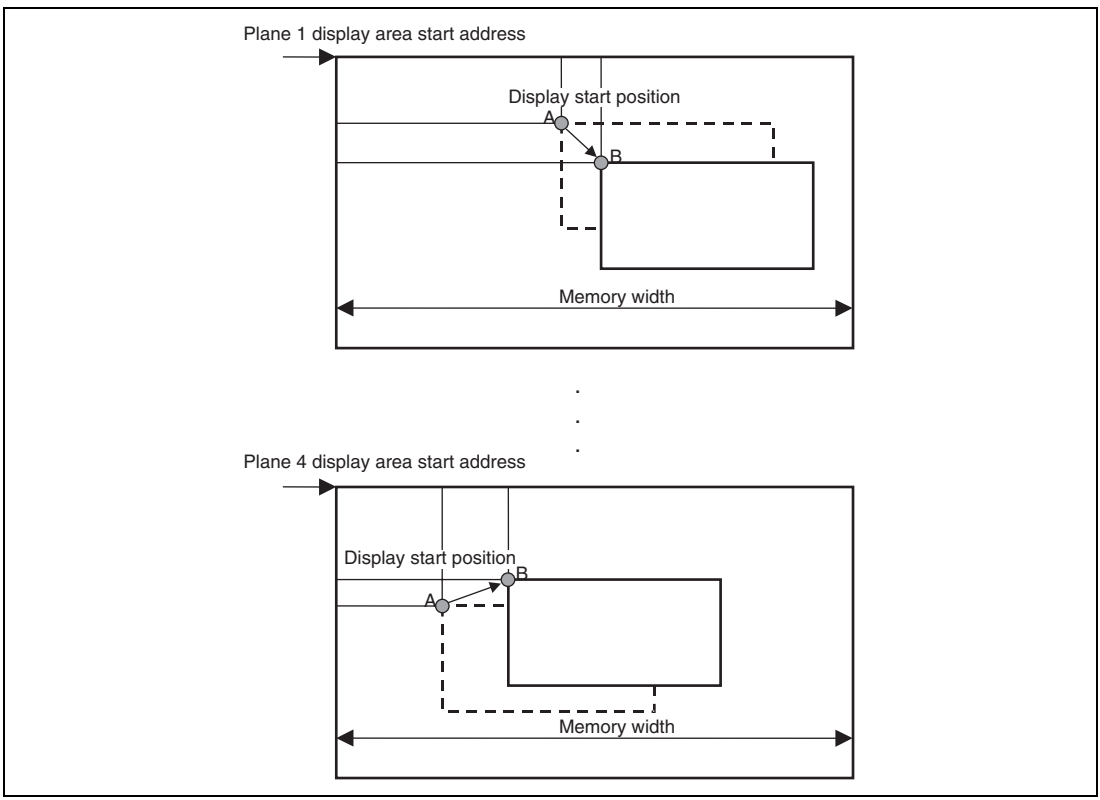


Figure 23.8 Schematic Diagram of Scroll Display

23.3.12 Wrap-Around Display

In addition to display scrolling, wrap-around display, which can be used in spherical scrolling, is possible for each plane. When enabling wrap-around display, the PnWAE bit in PnMR is set. As a result of changing the plane n display start position X, Y (the plane n start position X set in PnSPXR and the plane n start position Y set in PnSPYR) in order to scroll the display, even when plane n overflows the wrap-around area, the wrap-around area is seen as a spherical surface in wrap-around display, as in figure 23.9, and the part overflowing is complemented and displayed. The method used to specify the wrap-around area is described below.

1. The start address of the memory used for plane n is specified in PnDSA0R to PnDSA2R.
2. With the beginning of the specified memory as origin, the upper-left coordinates of the wrap-around area are specified in PnWASPR. The X-direction width of the wrap-around area is the memory width set in PnMWR.
3. The Y-direction width of the wrap-around area is set in PnWAMWR.

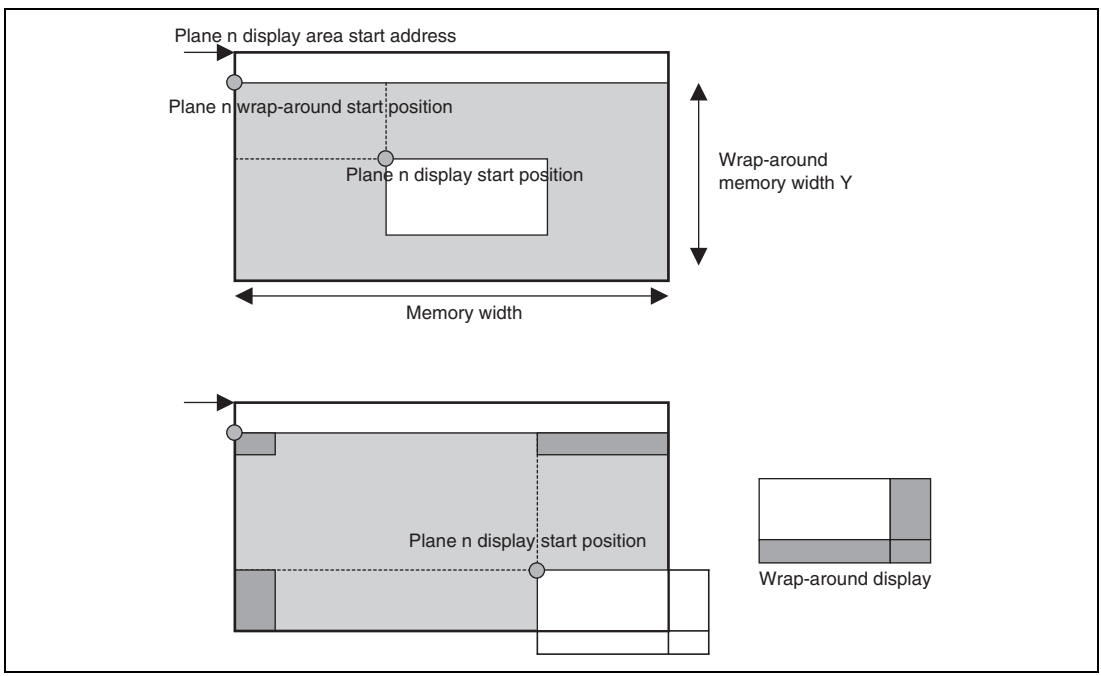


Figure 23.9 Schematic Diagram of Wrap-Around Display

Note: When wrap-around display is disabled (when the PnWAE bit in PnMR is 0), the part overflowing the wrap-around area becomes the color specified by BPOR, and superposition processing using this color is performed.

23.3.13 Upper-Left Overflow Display

For each plane, a display start position in memory (PnSPXR, PnSPYR) and display size (PnDSXR, PnDSYR) can be set arbitrarily, so that by combining and using these registers, areas overflowing the upper-left relative to the monitor origin (upper-left corner) can be displayed without overwriting display data in memory.

For a picture of size (DSX, DSY) and with start position (SPX, SPY), by setting the size to (DSX-ΔX, DSY-ΔY) and the start position to (SPX+ΔX, SPY+ΔY), the ΔX part overflowing on the left side and the ΔY part overflowing on top can be displayed. At this time, the display position (PnDPXR, PnDPYR) is fixed at 0.

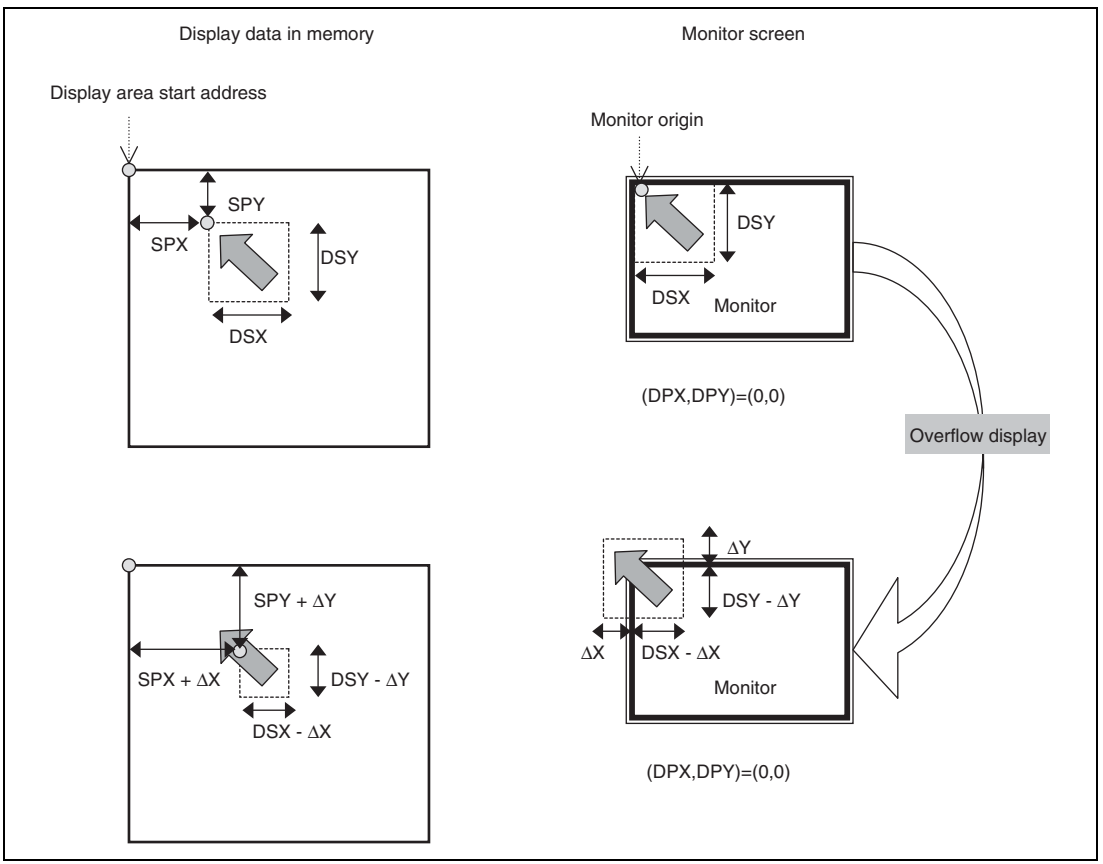


Figure 23.10 Schematic Diagram of Upper-Left Overflow Display

23.3.14 Double-Buffering Control

The double-buffering control of the display unit (DU) includes four types of functions, which are a auto rendering mode that does not switch the display until drawing is completed, a manual display change mode in which display or drawing switching is all controlled by software, an auto display change mode that realize blinking, and a video capture mode that is based on a frame ID of the video input (VIN) module.

In the case of auto rendering mode and manual display change mode, the display change is performed in frame units for non-interlaced and interlaced sync display, and in field units for interlaced sync & video display. In the case of auto display change mode, all switching is performed in field units. For video capture mode, all switching is performed in frame units.

Auto Rendering Mode: In auto rendering mode, display is not switched until drawing is completed. Even if drawing is not completed within a single frame period, drawing operation continues as it is.

Manual Display Change Mode: In manual display change mode, display frame switching and start of drawing are controlled by software. Display switching can either be performed by software using the PnDC bit in PnMR, or by setting the buffer 0 or buffer 1 start address in PnDSA0R and PnDSA1R indicated by the DFBn bit in DSSR.

Start of drawing is controlled by the rendering start bit. The control timing is determined by a VBK or TRA interrupt. When making a transition from this mode to another mode, the PnDC bit should always be set to 1 first.

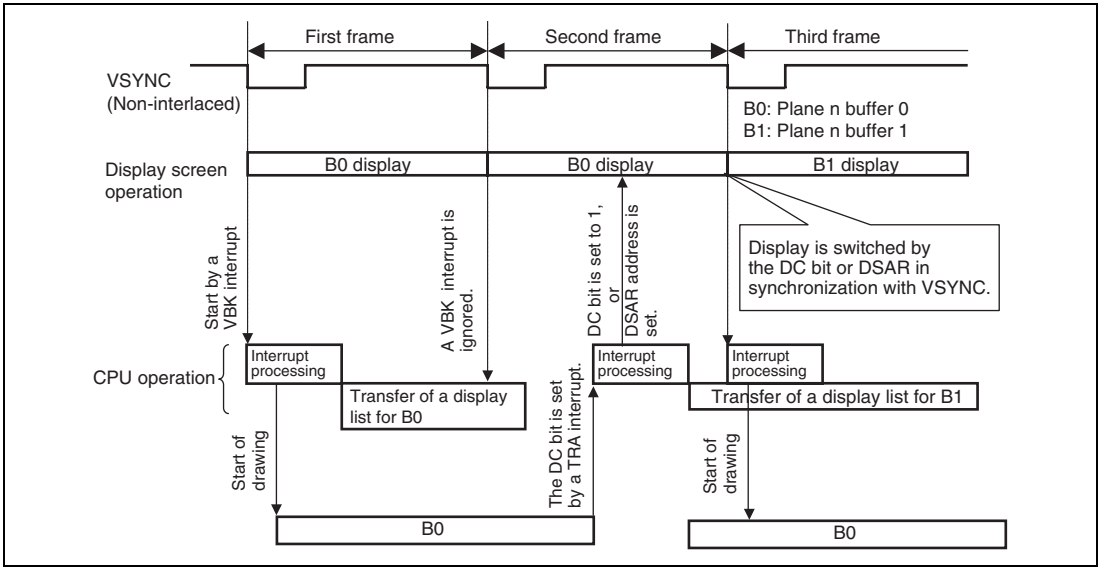


Figure 23.11 Manual Display Change Mode

Auto Display Change Mode: For information on the auto display change mode, refer to section 23.3.10, Blinking.

Video Capture Mode: In video capture mode, a display frame buffer is switched in frame units according to a frame ID, indicating the latest capture frame, output from the video input module.

23.3.15 Sync Mode

In order to facilitate synchronization with external equipment, in addition to master mode, a TV synchronization function is provided. Selection of master mode and TV sync mode is performed using the TVM bit in DSYSR. In master mode (internal sync mode), the position of the falling edge of the vertical sync signal (VSYNC) set by VSPR is detected. In TV sync mode (external sync mode), the position of the falling edge of the EXVYNC signal is detected. The results are then reflected in the FRM and VBK bits of DSSR.

Master Mode (Internal Sync Mode): By setting the period and pulse width of the horizontal and vertical sync signals (HSYNC, VSYNC) in the display timing generation registers, the corresponding waveforms are output. Also, display data is output in sync with these signals.

In interlaced sync mode and interlaced sync & video mode, a signal is output to the ODDF pin indicating odd/even fields.

TV Sync Mode (External Sync Mode): In TV sync mode, display data is output in sync with a horizontal sync signal and vertical sync signal (EXHSYNC, EXVSYNC) input from a TV, video, or other external sync signal generation circuit. Display data is output with reference to the falling edge of the EXHSYNC signal and the rising edge of the EXVSYNC signal.

The horizontal sync signal, vertical sync signal, and clock signal from the external sync signal generation circuit are input to the EXHSYNC, EXVSYNC, and DCLKIN pin, respectively. CSYNC is at high level. In interlaced sync mode and interlaced sync & video mode, a signal should be input to the ODDF pin indicating odd/even fields. In non-interlaced mode, the input to the ODDF pin should be fixed at low level or at high level.

When operating the unit in TV sync mode also, values must be set in HCR, HSWR, VCR, and VSPR (display timing generation registers).

When the EXVSYNC signal is input, either before or after completion of display of the display size portion set in the display unit (DU), the display unit (DU) performs vertical display completion operation and transitions to control for the next screen. When the EXVSYNC signal is not input, the unit continues to wait for the EXVSYNC signal while remaining in the vertical blanking interval (auto-control is not performed). Similarly, when the EXHSYNC signal is input the display unit (DU) performs horizontal display completion operation and transitions to control for the next raster line; but if the EXHSYNC signal is not input, the unit continues to wait for the EXHSYNC signal while remaining in the horizontal blanking interval (auto-control is not performed).

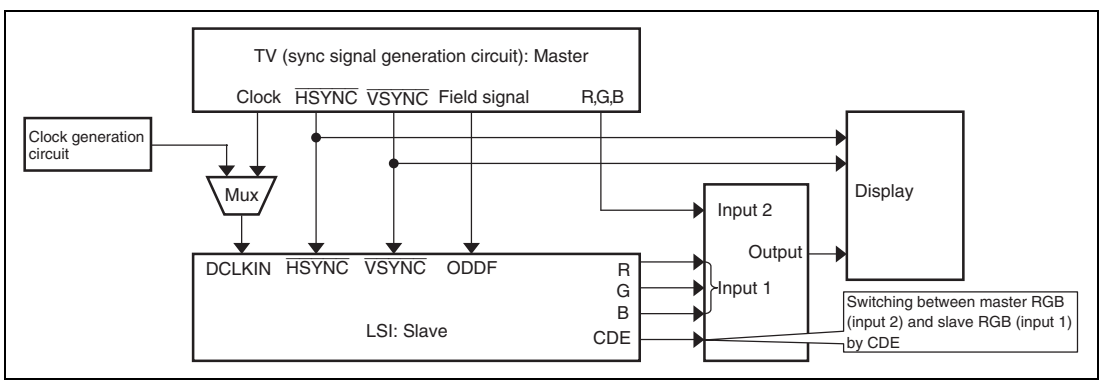


Figure 23.12 Signal Flow in TV Sync Mode

Sync Method Switching Mode: When switching from master mode into TV sync mode, or from TV sync mode into master mode, when necessary this mode should be switched into first. Even if a transition to this mode is not made first, switching of the synchronization method is possible.

In this mode, input/output pins connected to the display unit (DU) are for input, and so collision of pin signals can be avoided. Also, in this mode the internal dot clock is stopped, so that disorder in the input dot clock has no effect on display operation.

23.3.16 Alpha-Ratio Plane

α blending can be performed by using the four planes (display planes) that can display the current display data with the alpha ratio, plus a plane for the alpha ratio (alpha-ratio plane). Data in the alpha ratio plane are limited to the alpha ratio; they cannot be composed as display data.

To enable the alpha-ratio plane, set bit 0 and bit 4 of DAPCR to 1. PnALPHAR is used to select the alpha-ratio plane. In PnALPHAR, set bits 10 to 8 to 111 and bits 2 to 0 to 000 or 001.

23.3.17 Display Capture

In display capture, display data (RGB-666) that have been composed for output on the relevant pins are converted to RGB-565 or ARGB-1555 data, stored in a buffer having the same configuration as the read buffer, and then stored in the area specified by DCSAR via the SuperHyway. To capture the display data, set bit 0 of DCPCR to 1. Capturing starts from the next frame after this setting has been made. The specifications for display capturing are as follows:

Data formats: RGB-565; the lowest-order bits of the R and B bits of the RGB-666 data for output from the pins are discarded.

ARGB-1555; the lowest-order bits of the R, G, and B bits of the RGB-666 data for output from the pins are discarded; the A value is specified by a register setting.

Capture area start address: Only a single address can be specified.

Capture size X: The same as the monitor size (horizontal display end position (HDE) – horizontal display start position (HDS)

Capture size Y: The same as the monitor size (vertical display end position (VDE) – vertical display start position (VDS)

Memory width X: Specified by a register. Writing to the buffer ends when the memory width is exceeded.

Memory length Y: Specified by a register. Storage of data in the memory ends when the memory length (in lines) is exceeded.

23.4 Display Control

23.4.1 Display Timing Generation

In the display unit (DU), display timing is generated for the horizontal direction and vertical direction of the display screen. Display timing is set by using display timing generation registers in section 23.2.2. Figure 23.13 shows the display timing in non-interlaced mode. Here, the display screen is defined in terms of the variables of table 23.12.

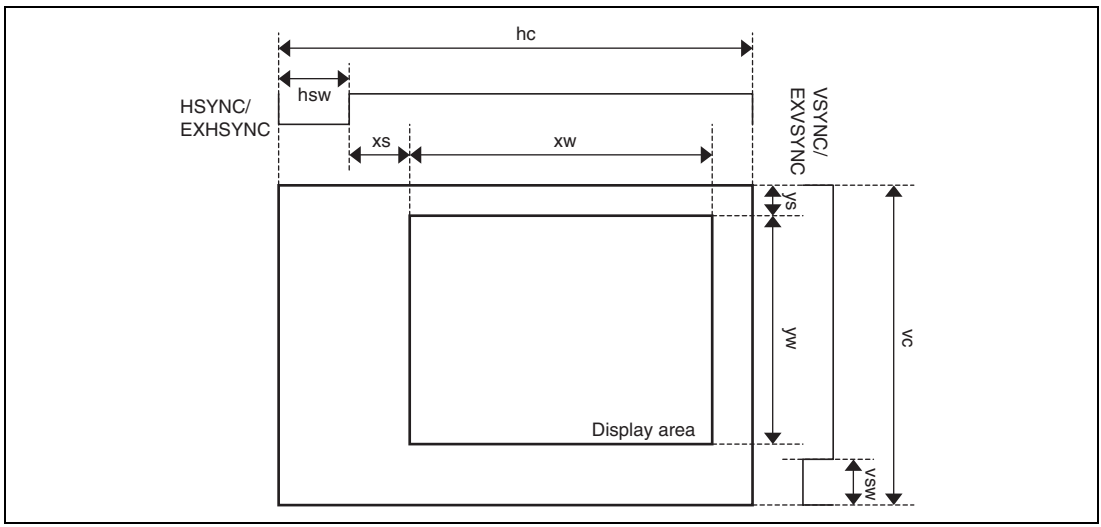


Figure 23.13 Display Timing Generation for Horizontal Direction and Vertical Direction of Display Screen

Table 23.12 Variables Defined in Display Screen

Variables	Contents	Units
hc*[1]	Horizontal scan period	Dot clock
hsw	Horizontal sync pulse width	Dot clock
xs	From rise of HSYNC to display start position in the horizontal direction of the display screen	Dot clock
xw	Display width per 1 raster of display screen	Dot clock
vc*[2]	Vertical scan period	Raster line
vsw	Vertical sync pulse width	Raster line
ys	From rise of VSYNC to display start position in the vertical direction of the display screen	Raster line
yw	Vertical display period of display screen	Raster line

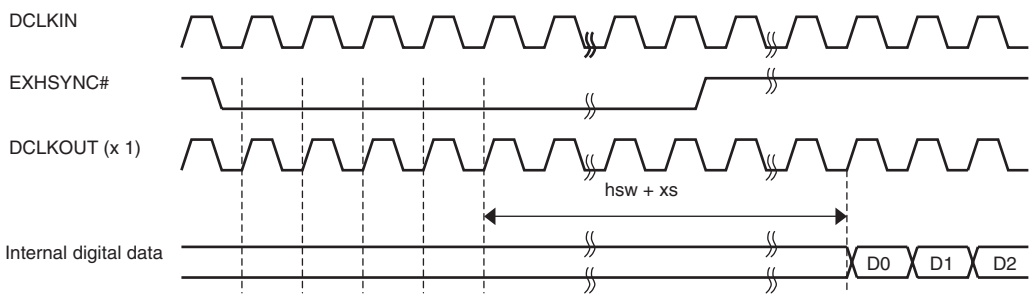
Notes: 1. Should be set such that hsw + xs + xw < hc + 18 (decimal)
2. Should be set such that vsw + ys + yw < vc

The display timing generation register settings are different depending on the scan method and synchronization method. Hence the value of the display timing generation registers should be set after performing calculations like those indicated in table 23.13.

Table 23.13 Correspondence Table of Settings of Display Timing Generation Registers

		Synchronization Method	
Register Name	**Bit Name**	**Master Mode**	**TV Sync Mode**
Horizontal display start register (HDSR)	HDS	hsw + xs - 19	hsw + xs – 25 *2
Horizontal display end register (HDER)	HDE	hsw + xs – 19 + xw	hsw + xs – 25 + xw *2
Vertical display start register (VDSR)	VDS	ys – 2 *3	ys – 2 *3
Vertical display end register (VDER)	VDE	ys – 2 + yw	ys – 2 + yw
Horizontal synch width register (HSWR)	HSW	hsw - 1	hsw - 1
Horizontal cycle register (HCR)	HC	hc - 1	hc - 1
Vertical synch point register (VSPR)	VSP	vc – vsw - 1	vc – vsw - 1
Vertical cycle register (VCR)	VC	vc - 1	vc - 1

Notes: 1. In all scan modes, VDS, VDE, VSP, VC settings are in single-field units.
2. The values of HDS and HDE are from the fourth rising edge of DCLKOUT after detection of the falling edge of EXHSYNC through the rising edge of DCLKOUT.



3. VDS should be set to 1 or greater.
4. HC should be set so as to satisfy HC > HDE.

23.4.2 CSYNC

In master mode, a CSYNC (composite sync) signal is output. EQWR is used to set the low-level pulse width of the CSYNC equivalent pulse. SPWR is used to set the low-level pulse width of the CSYNC serration pulse.

The CSYNC waveform is selected using the CSY (CSYNC mode) bit in DSMR.

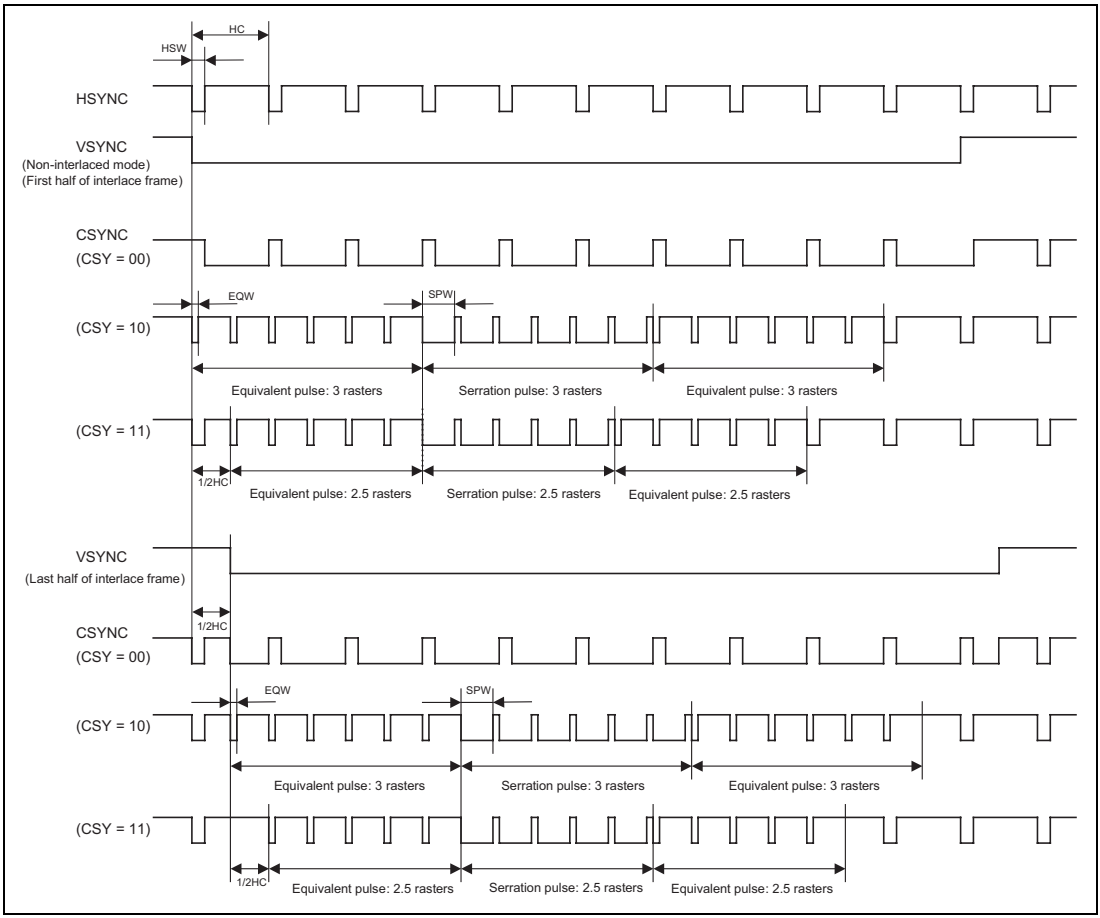


Figure 23.14 CSYNC Timing Chart

23.4.3 Scan Method

The scan method can be selected from among non-interlaced mode, interlaced sync mode, and interlaced sync & video mode. The mode is selected using the SCM bit in DSYSR.

- Non-interlaced mode

  In this scan method, one frame consists of a single field.
- Interlaced sync mode

  In this scan method, one frame consists of two fields. The two fields are an even field and an odd field, displaying the same data.
- Interlaced sync & video mode

  In this scan method, one frame consists of two fields. The two fields are an even field and an odd field, displaying different data.

The ODEV bit in DSMR is used to set the display order of fields in interlaced sync mode and in interlaced sync & video mode. When the ODEV bit is 0, the display order for one frame is odd field, then even field; when the ODEV bit is 1, the order for one frame is even field, then odd field.

In master mode, high level is output from the ODDF pin during even field display, and low level is output during odd field display. In TV sync mode, high level is input to the ODDF pin to cause display of the even field, and low level is input to cause display of the odd field.

Note: When non-interlaced mode is selected in TV sync mode, the ODDF pin should be fixed at low level or high level.

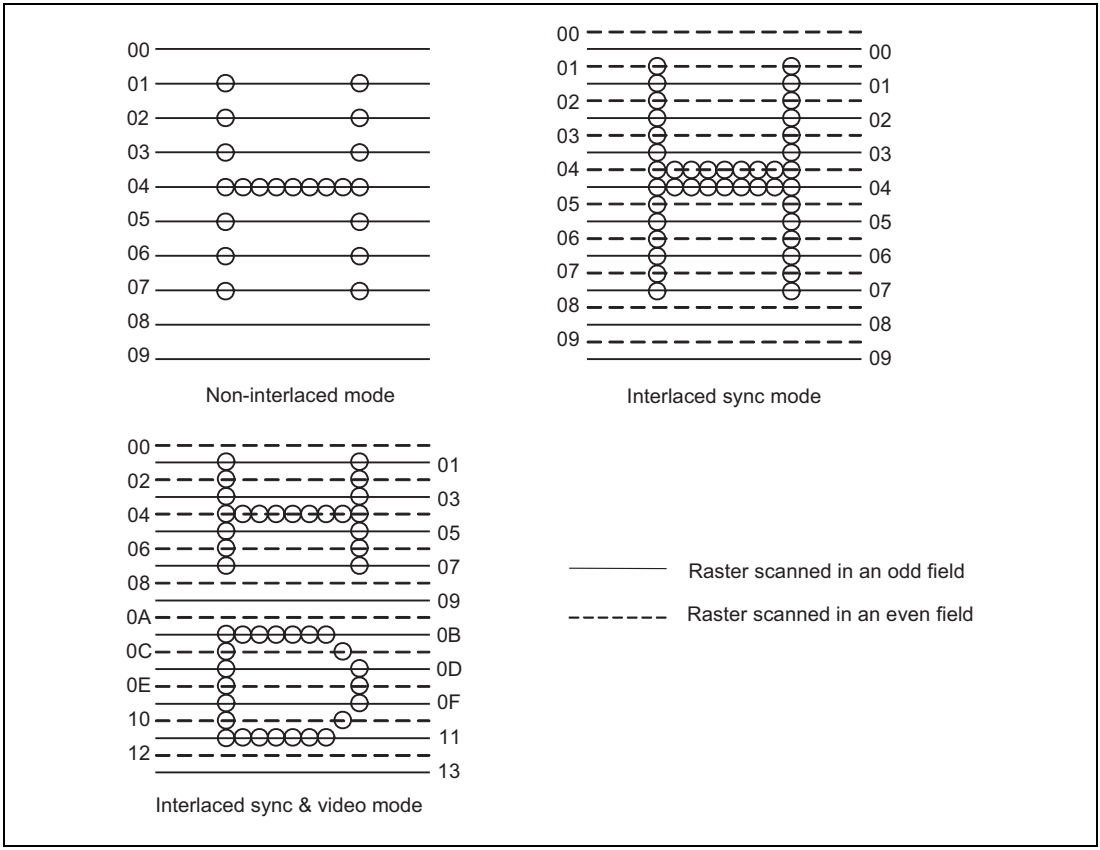


Figure 23.15 Example of Display in Each Scan Mode

- Example of vertical scan period

  Non-interlaced mode: 1/60 second/field, 1/30 second/field

  Interlaced sync mode: 1/30 second/frame

  Interlaced sync & video mode: 1/30 second/frame

- Display in non-interlaced method

  In this method, all lines are displayed at once without providing intervals between input video signals.

  This input method is for monitors capable of high-resolution display.

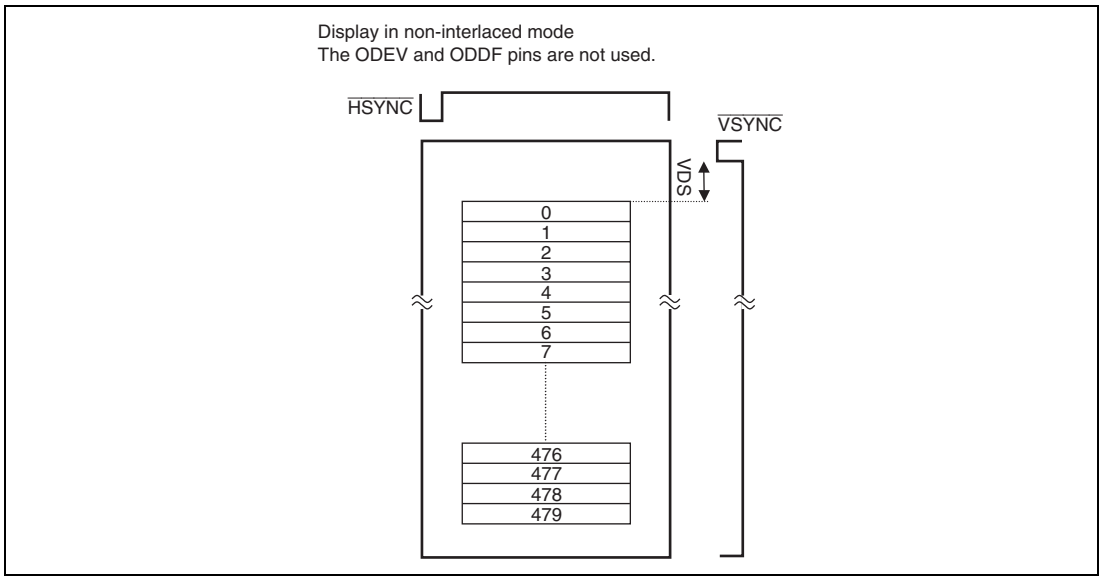


Figure 23.16 Display in Non-Interlaced Method

- Display in interlaced method

  At every scan period VC of the input video signal, even lines and odd lines are switched and displayed in alternation, and a single screen (one frame) is combined and displayed (with the afterimage of the preceding VC) with a period of 2VC. This is a basic TV input method.

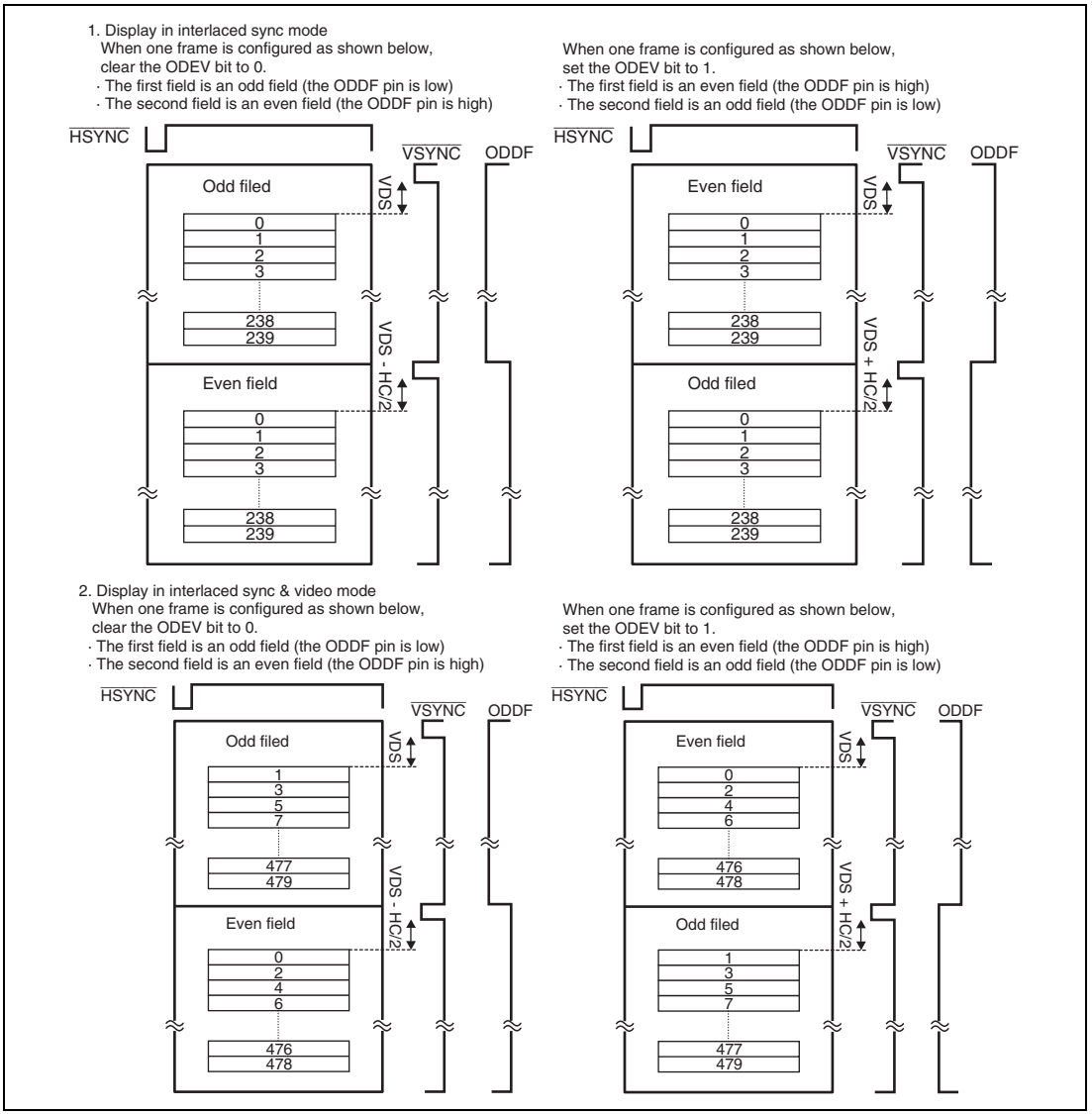


Figure 23.17 Display in Interlaced Method

23.4.4 Color Detection

When output display data matches a color set in CDER, high level is output from the CDE pin. The CDEM bit in DSMR can be used to fix the level outside display intervals. Also, the CDEL bit in DSMR can be used to select the polarity of the output level.

Table 23.14 Output Level of the CDE Pin

		The CDE pin in display intervals		The CDE pin outside display intervals	
		Result of comparison of output display data and color detection register		Value of the color detection register*	
	CDEM	Same	Different	0	Other than 0
0	00	High level	Low level	High level	Low level
0	01	High level	Low level	High level	Low level
0	10	High level	Low level	Low level	Low level
0	11	High level	Low level	High level	High level
1	00	Low level	High level	Low level	High level
1	01	Low level	High level	Low level	High level
1	10	Low level	High level	High level	High level
1	11	Low level	High level	Low level	Low level

Note: * Output display data is 0 outside display intervals.

23.4.5 External Sync Control

Dot Clock Sync Control

In TV-sync mode, the display unit (DU) is capable of using an externally input dot clock (clock signal for frequency multiplication: DCLKIN) to generate a dot clock (output dot clock: DCLKOUT) that is in accord with external synchronizing signals (EXHSYNC, EXVSYNC). Supply the externally input dot clock (multiplication clock: DCLKIN) and set the following parameters.

Table 23.15 External Sync Control Parameters

Variable	Function
ESCR/SYNCSEL	Selects the sync signal (EXHSYNC or EXVSYNC) to use in phase matching of the dot clock.
ESCR/FRQSEL	Selects the dot-clock division ratio.

1. Use the SYNCSEL bits of ESCR to set the sync timing of the dot clock (output dot clock: DCLKOUT) generated from the internal dot clock.
2. Use the FRQSEL bits of ESCR to set the division ratio for generation of the internal dot clock. The following figure shows the internal dot clock timing where the input dot clock has been synchronized with EXHSYNC and then divided by four.

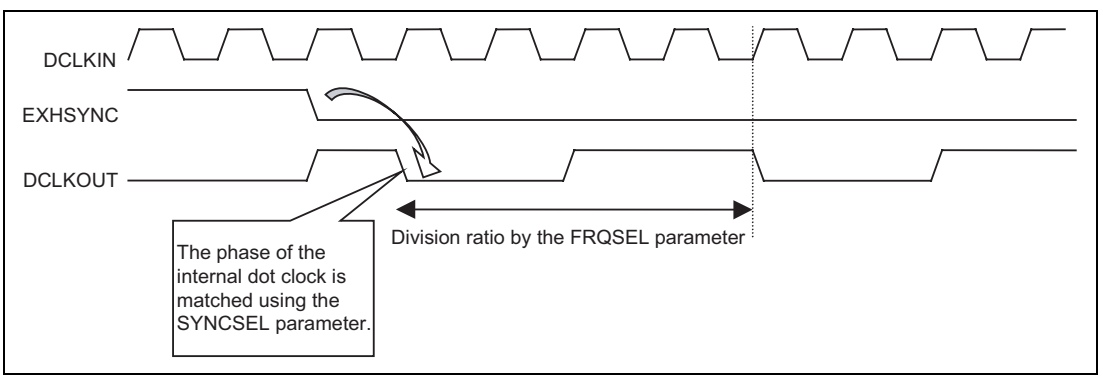


Figure 23.18 DCLKOUT Timing Chart where DCLKIN Synchronized with EXHSYNC is Divided by Four

23.4.6 Output Signal Timing Adjustment

The display unit (DU) enables selection of output timing, with respect to the output dot clock, of the various output signals (the four sync signals HSYNC, VSYNC, CSYNC, ODDF, as well as DISP, CDE, CLAMP, DE, digital RGB signals). Timing is selected by setting OTAR.

Table 23.16 Output Signal Timing Setting Parameters

Variable	Description
SYNCA	Sets output timing of the HSYNC, VSYNC, CSYNC, ODDF signal
DISPA	Sets output timing of the DISP signal
CDEA	Sets output timing of the CDE signal
DRGBA	Sets output timing of digital RGB signal
CLAMPA	Sets output timing of the CLAMP signal
DEA	Sets output timing of the DE signal

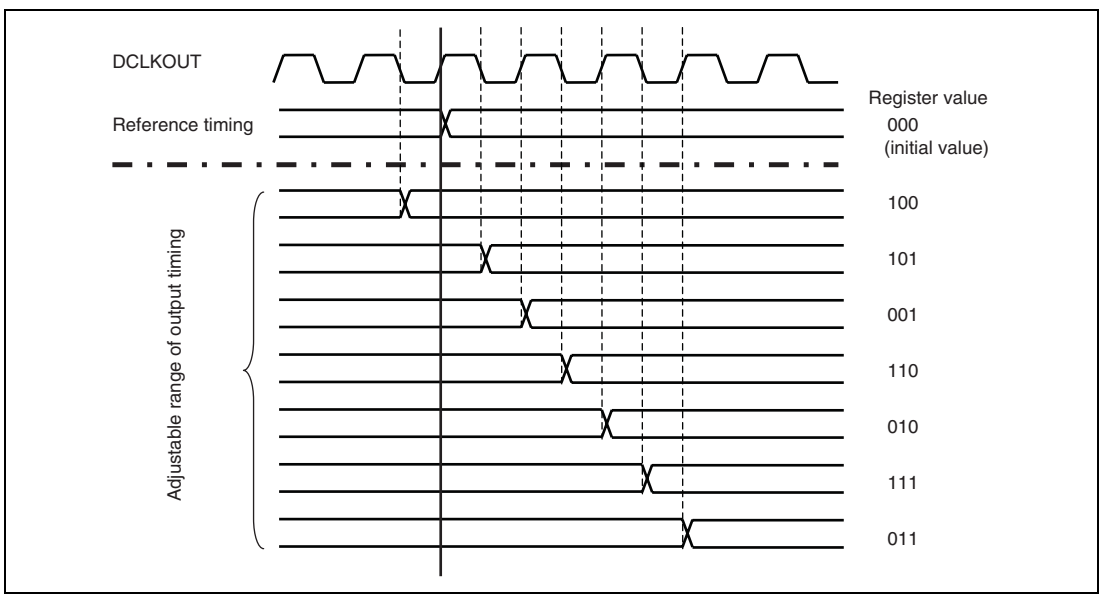


Figure 23.19 Adjustable Range of Output Timing

23.5 Note on Usage

23.5.1 Module Standby Mode

Module Standby mode, in which supply of the clock signal to the display unit (DU) is stopped, is supported.

Even when the display unit (DU) enters the standby mode, the register values are retained. Do not access the display unit (DU) during periods in standby mode.

23.5.2 Transition to Module Standby Mode

1. Turn off the display by setting both the DEN and DRES bits in DSYSR to 0.
2. Test the VBK bit in DSSR to confirm the next VBK flag (because the display is turned off with the timing of VBK).
3. Stop the clock.

23.5.3 Release from Module Standby Mode and Restarting Display

1. Start the clock.
2. Make settings to turn the display on by setting the DEN and DRES bits in DSYSR to 1 and 0 respectively.

23.5.4 Acquisition of External Sync Signal

The following three ways of acquiring the external SYNC signal are available.

(a) The SYNC signal is acquired on rising edges of DCLKIN (or the frequency-divided clock signal derived from this if division has been set up).
(b) The SYNC signal is acquired on falling edges of DCLKIN (or the frequency-divided clock signal derived from this if division has been set up).
(c) Assuming that division has been set up, the SYNC signal that is acquired by the pre-division clock signal is latched on edges of the frequency-divided clock signal.

The electrical characteristics (AC spec.) are only guaranteed for case (a) above. There is no guarantee of electrical characteristics (AC spec.) for cases (b) and (c).

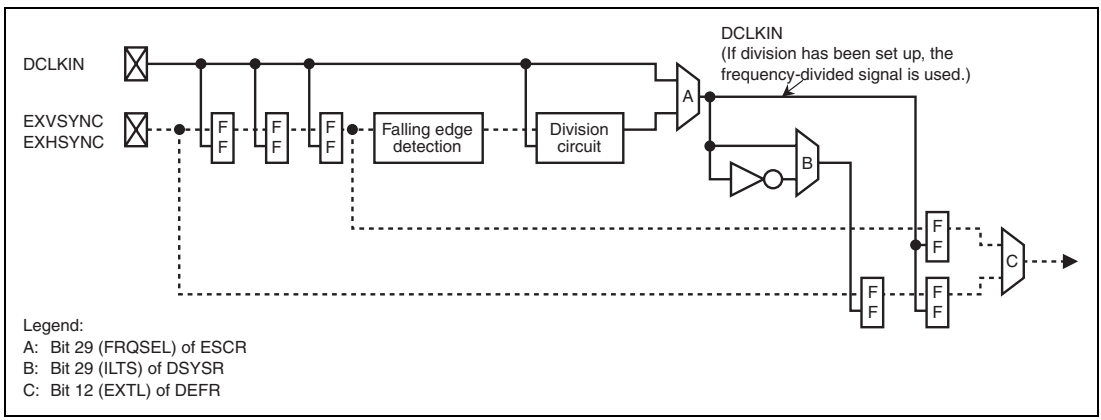


Figure 23.20 Diagram of Circuit that Generates the Display Timing from an External Sync Signal

23.5.5 Restrictions on Changing the Synchronization of the External SYNC Signal

When electrical characteristics (AC spec.) related to the acquisition of an external signal are not satisfied, ensure an interval of at least two cycles between changes of EXHSYNC or EXVSYNC, i.e. the external SYNC signals.

The frequency-divided dot clock is used as the basis of the cycle period when EXSL (bit 12) of DEFR is 0. The pre-division dot clock is used when EXSL is 1.

Section 24 Serial Communication Interface with FIFO (SCIF)

This LSI is equipped with a 6-channel serial communication interface with built-in FIFO buffers (Serial Communication Interface with FIFO: SCIF). The SCIF can perform both asynchronous and clocked synchronous serial communications.

64-stage FIFO buffers are provided for transmission and reception, enabling fast, efficient, and continuous communication.

Channel 0 has modem control functions ($\overline{\text{RTS}}$ and $\overline{\text{CTS}}$).

24.1 Features

The SCIF has the following features.

- Asynchronous serial communication mode

  Serial data communication is executed using an asynchronous system in which synchronization is achieved character by character. Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA).

  There is a choice of 8 serial data transfer formats.

  — Data length: 7 or 8 bits

  — Stop bit length: 1 or 2 bits

  — Parity: Even/odd/none

  — Receive error detection: Parity, framing, and overrun errors

  — Break detection: A break is detected when a framing error lasts for more than 1 frame length at Space 0 (low level). When a framing error occurs, a break can also be detected by reading the SCIF0_RXD to SCIF5_RXD pin levels directly from the serial port register (SCSPTR).

- Clocked synchronous serial communication mode

  Serial data communication is synchronized with a clock. Serial data communication can be carried out with other LSIs that have a synchronous communication function.

  There is a single serial data communication format.

  Data length: 8 bits

  Receive error detection: Overrun errors

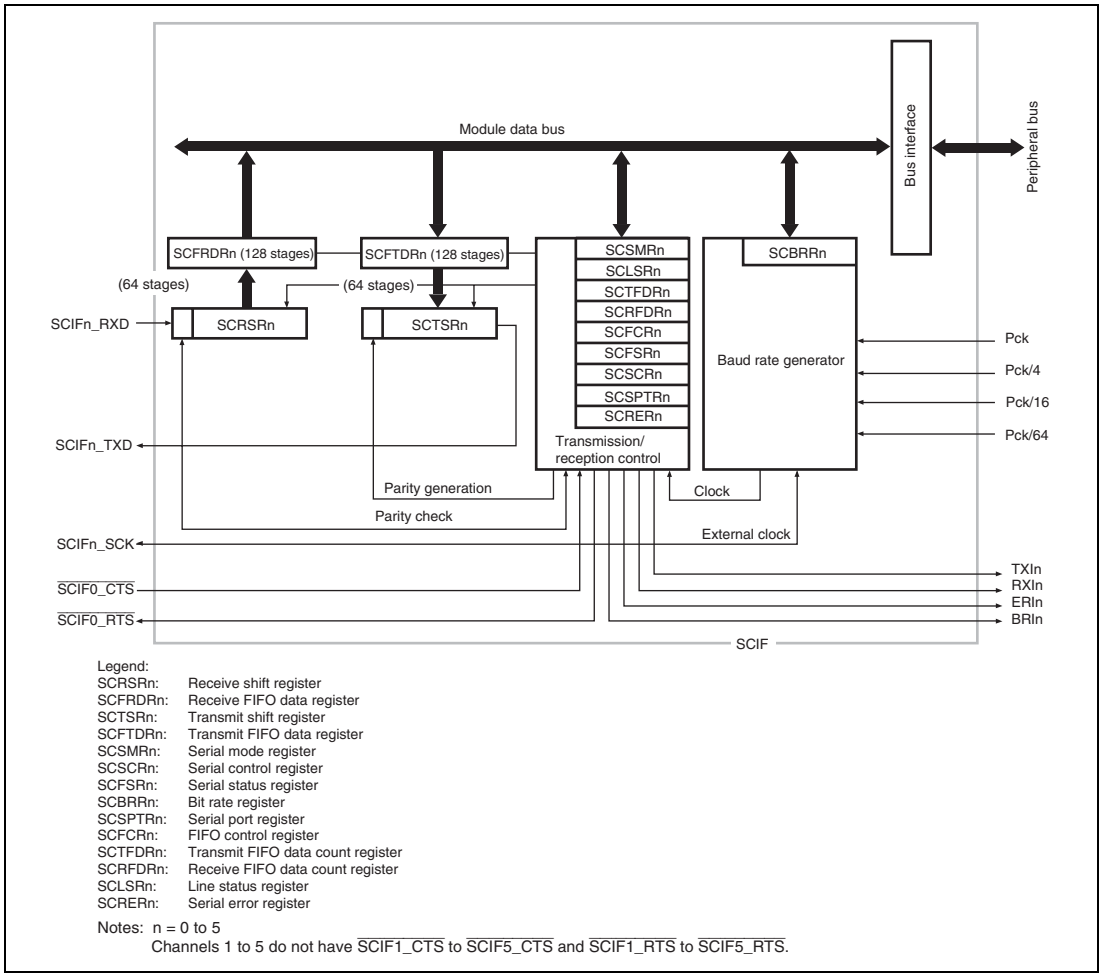
- Full-duplex communication capability

  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 continuous transmission and reception of serial data.
- LSB first for data transmission and reception
- On-chip baud rate generator allows any bit rate to be selected.
- Choice of clock source: internal clock from baud rate generator or external clock from SCIF0_SCK to SCIF5_SCK pins
- Four interrupt sources

  There are four interrupt sources – transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive error – that can issue requests independently.
- The DMA controller (DMAC) can be activated to execute a data transfer by issuing a DMA transfer request in the event of a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- When not in use, the SCIF can be stopped by halting its clock supply to reduce power consumption.
- In asynchronous mode, modem control functions ($\overline{\text{SCIF0\_RTS}}$ and $\overline{\text{SCIF0\_CTS}}$) are provided.(only in channel 0)
- The amount of data in the transmit/receive FIFO registers, and the number of receive errors in the receive data in the receive FIFO register, can be ascertained.
- In asynchronous mode, a timeout error (DR) can be detected during reception.

Figure 24.1 shows a block diagram of the SCIF. Figures 24.2 to 24.6 show block diagrams of the I/O ports in the SCIF. There are six channels in this LSI. In figures 24.2 to 24.6, the channel number is omitted.



Legend:

SCRSRn:	Receive shift register
SCFRDRn:	Receive FIFO data register
SCTSRn:	Transmit shift register
SCFTDRn:	Transmit FIFO data register
SCSMRn:	Serial mode register
SCSCRn:	Serial control register
SCFSRn:	Serial status register
SCBRRn:	Bit rate register
SCSPTRn:	Serial port register
SCFCRn:	FIFO control register
SCTFDRn:	Transmit FIFO data count register
SCRFDRn:	Receive FIFO data count register
SCLSRn:	Line status register
SCRERn:	Serial error register

Notes: n = 0 to 5
Channels 1 to 5 do not have $\overline{\text{SCIF1\_CTS}}$ to $\overline{\text{SCIF5\_CTS}}$ and $\overline{\text{SCIF1\_RTS}}$ to $\overline{\text{SCIF5\_RTS}}$.

Figure 24.1 Block Diagram of SCIF

Figures 24.2 to 24.6 show block diagrams of the I/O ports in SCIF.

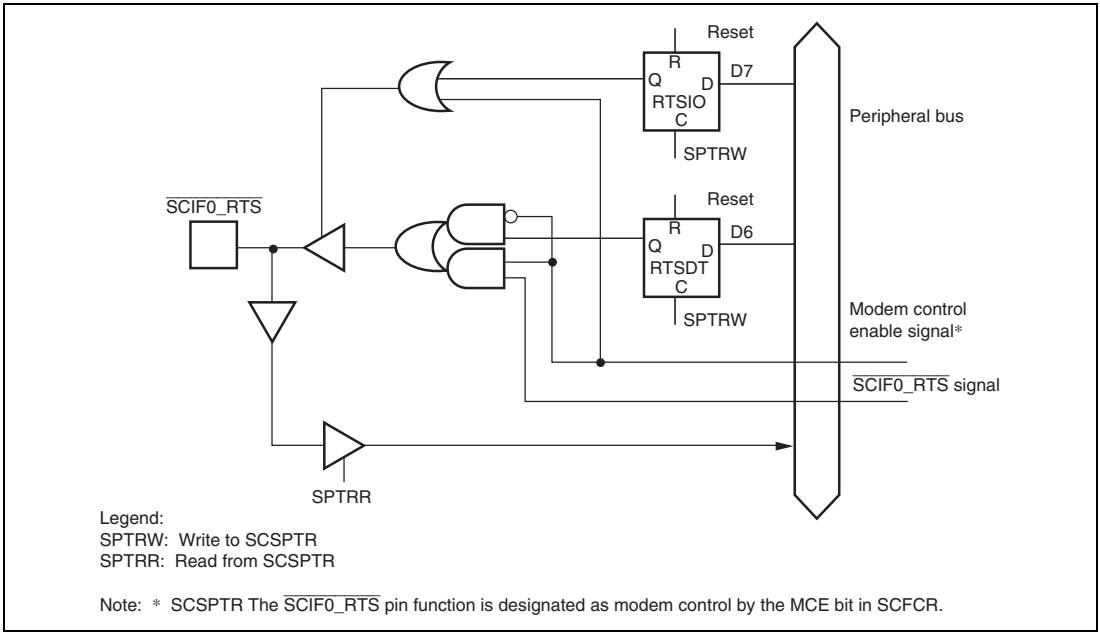


Figure 24.2 SCIF0_RTS Pin

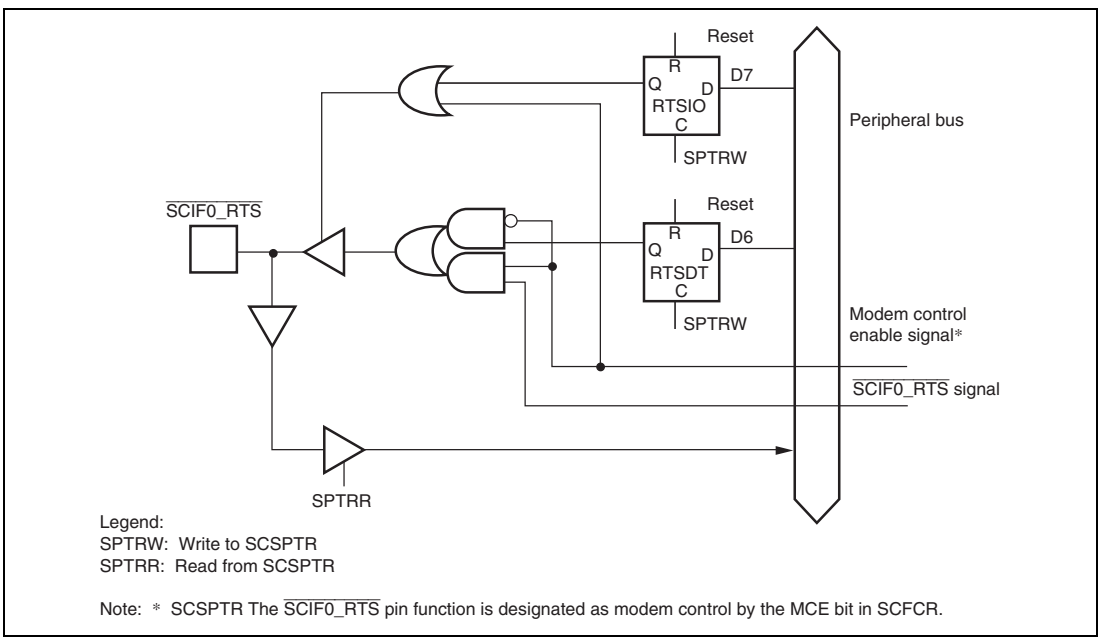


Figure 24.3 SCIF0_CTS Pin

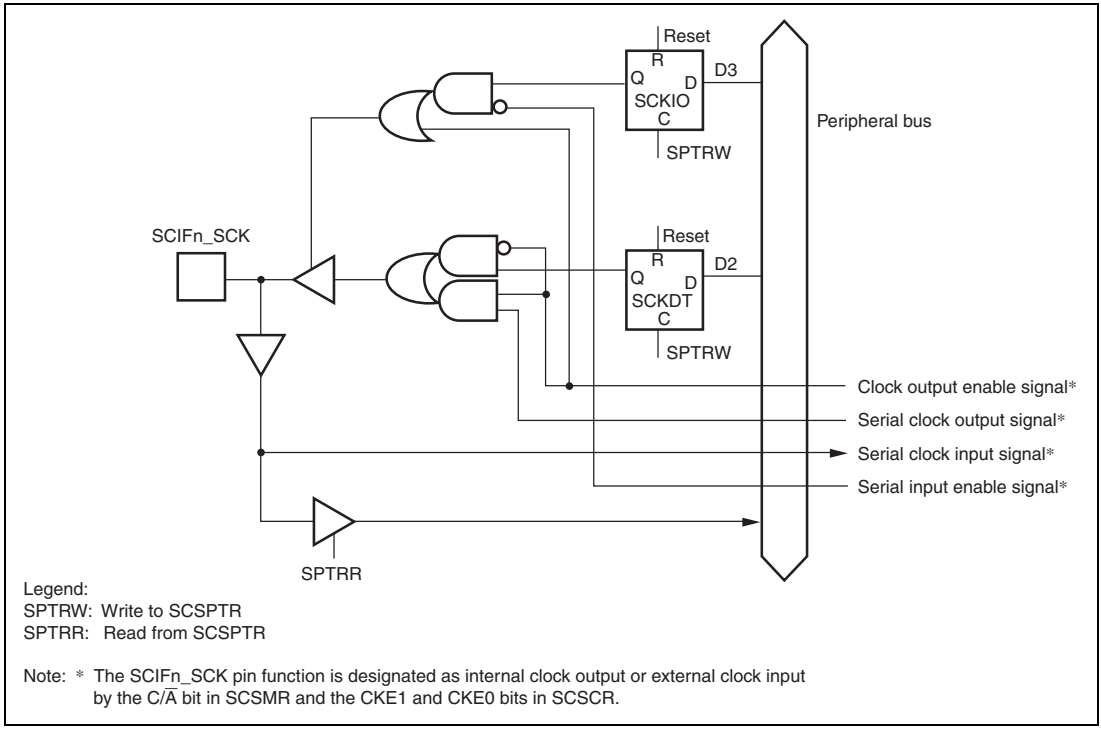


Figure 24.4 SCIFn_SCK Pin (n = 0 to 5)

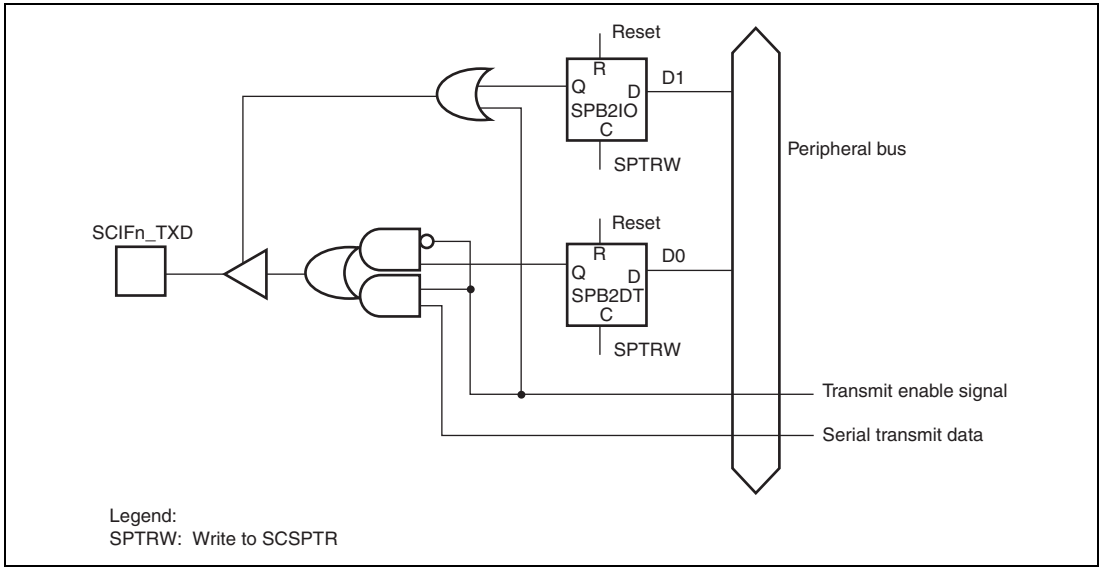


Figure 24.5 SCIFn_TXD Pin (n = 0 to 5)

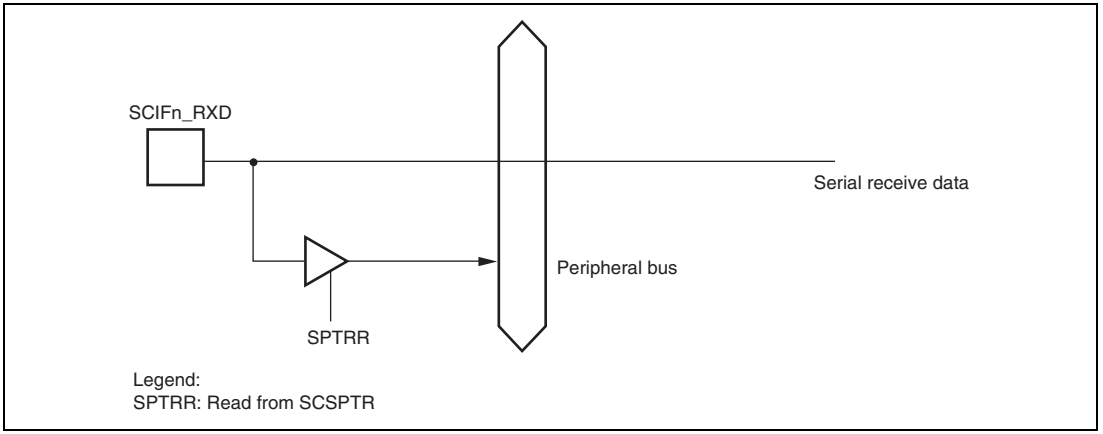


Figure 24.6 SCIFn_RXD Pin (n = 0 to 5)

24.2 Input/Output Pins

Table 24.1 shows the SCIF pin configuration. Since the pin functions are the same in each channel, the channel number is omitted in the description below. The modem control pins are available only in channel 0.

Table 24.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCIF0_SCK to SCIF5_SCK	I/O	Clock input/output
Receive data pin	SCIF0_RXD to SCIF5_RXD	Input	Receive data input
Transmit data pin	SCIF0_TXD to SCIF5_TXD	Output	Transmit data output
Modem control pin	$\overline{\text{SCIF0\_CTS}}$	I/O	Transmission enabled
Modem control pin	$\overline{\text{SCIF0\_RTS}}$	I/O	Transmission request

Note: These pins function as serial pins by performing SCIF operation settings with the C/$\overline{\text{A}}$ bit in SCSMR, the TE, RE, CKE1, and CKE0 bits in SCSCR, and the MCE bit in SCFCR. Break state transmission and detection can be set by SCSPTR of the SCIF.

24.3 Register Descriptions

The SCIF has the following registers. Since the register functions are the same in each channel, the channel number is omitted in the description below.

Table 24.2 Register Configuration (1)

Ch.	Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Serial mode register 0	SCSMR0	R/W	H'FFEA 0000	H'1FEA 0000	16	Pck
	Bit rate register 0	SCBRR0	R/W	H'FFEA 0004	H'1FEA 0004	8	Pck
	Serial control register 0	SCSCR0	R/W	H'FFEA 0008	H'1FEA 0008	16	Pck
	Transmit FIFO data register 0	SCFTDR0	W	H'FFEA 000C	H'1FEA 000C	8	Pck
	Serial status register 0	SCFSR0	R/W*[1]	H'FFEA 0010	H'1FEA 0010	16	Pck
	Receive FIFO data register 0	SCFRDR0	R	H'FFEA 0014	H'1FEA 0014	8	Pck
	FIFO control register 0	SCFCR0	R/W	H'FFEA 0018	H'1FEA 0018	16	Pck
	Transmit FIFO data count register 0	SCTFDR0	R	H'FFEA 001C	H'1FEA 001C	16	Pck
	Receive FIFO data count register 0	SCRFDR0	R	H'FFEA 0020	H'1FEA 0020	16	Pck
	Serial port register 0	SCSPTR0	R/W	H'FFEA 0024	H'1FEA 0024	16	Pck
	Line status register 0	SCLSR0	R/W*[2]	H'FFEA 0028	H'1FEA 0028	16	Pck
	Serial error register 0	SCRER0	R	H'FFEA 002C	H'1FEA 002C	16	Pck
1	Serial mode register 1	SCSMR1	R/W	H'FFEB 0000	H'1FEB 0000	16	Pck
	Bit rate register 1	SCBRR1	R/W	H'FFEB 0004	H'1FEB 0004	8	Pck
	Serial control register 1	SCSCR1	R/W	H'FFEB 0008	H'1FEB 0008	16	Pck
	Transmit FIFO data register 1	SCFTDR1	W	H'FFEB 000C	H'1FEB 000C	8	Pck
	Serial status register 1	SCFSR1	R/W*[1]	H'FFEB 0010	H'1FEB 0010	16	Pck
	Receive FIFO data register 1	SCFRDR1	R	H'FFEB 0014	H'1FEB 0014	8	Pck
	FIFO control register 1	SCFCR1	R/W	H'FFEB 0018	H'1FEB 0018	16	Pck
	Transmit FIFO data count register 1	SCTFDR1	R	H'FFEB 001C	H'1FEB 001C	16	Pck
	Receive FIFO data count register 1	SCRFDR1	R	H'FFEB 0020	H'1FEB 0020	16	Pck
	Serial port register 1	SCSPTR1	R/W	H'FFEB 0024	H'1FEB 0024	16	Pck
	Line status register 1	SCLSR1	R/W*[2]	H'FFEB 0028	H'1FEB 0028	16	Pck
	Serial error register 1	SCRER1	R	H'FFEB 002C	H'1FEB 002C	16	Pck

Ch.	Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Size	Sync Clock
2	Serial mode register 2	SCSMR2	R/W	H'FFEC 0000	H'1FEC 0000	16	Pck
	Bit rate register 2	SCBRR2	R/W	H'FFEC 0004	H'1FEC 0004	8	Pck
	Serial control register 2	SCSCR2	R/W	H'FFEC 0008	H'1FEC 0008	16	Pck
	Transmit FIFO data register 2	SCFTDR2	W	H'FFEC 000C	H'1FEC 000C	8	Pck
	Serial status register 2	SCFSR2	R/W*[1]	H'FFEC 0010	H'1FEC 0010	16	Pck
	Receive FIFO data register 2	SCFRDR2	R	H'FFEC 0014	H'1FEC 0014	8	Pck
	FIFO control register 2	SCFCR2	R/W	H'FFEC 0018	H'1FEC 0018	16	Pck
	Transmit FIFO data count register 2	SCTFDR2	R	H'FFEC 001C	H'1FEC 001C	16	Pck
	Receive FIFO data count register 2	SCRFDR2	R	H'FFEC 0020	H'1FEC 0020	16	Pck
	Serial port register 2	SCSPTR2	R/W	H'FFEC 0024	H'1FEC 0024	16	Pck
	Line status register 2	SCLSR2	R/W*[2]	H'FFEC 0028	H'1FEC 0028	16	Pck
	Serial error register 2	SCRER2	R	H'FFEC 002C	H'1FEC 002C	16	Pck
3	Serial mode register 3	SCSMR3	R/W	H'FFED 0000	H'1FED 0000	16	Pck
	Bit rate register 3	SCBRR3	R/W	H'FFED 0004	H'1FED 0004	8	Pck
	Serial control register 3	SCSCR3	R/W	H'FFED 0008	H'1FED 0008	16	Pck
	Transmit FIFO data register 3	SCFTDR3	W	H'FFED 000C	H'1FED 000C	8	Pck
	Serial status register 3	SCFSR3	R/W*[1]	H'FFED 0010	H'1FED 0010	16	Pck
	Receive FIFO data register 3	SCFRDR3	R	H'FFED 0014	H'1FED 0014	8	Pck
	FIFO control register 3	SCFCR3	R/W	H'FFED 0018	H'1FED 0018	16	Pck
	Transmit FIFO data count register 3	SCTFDR3	R	H'FFED 001C	H'1FED 001C	16	Pck
	Receive FIFO data count register 3	SCRFDR3	R	H'FFED 0020	H'1FED 0020	16	Pck
	Serial port register 3	SCSPTR3	R/W	H'FFED 0024	H'1FED 0024	16	Pck
	Line status register 3	SCLSR3	R/W*[2]	H'FFED 0028	H'1FED 0028	16	Pck
	Serial error register 3	SCRER3	R	H'FFED 002C	H'1FED 002C	16	Pck

Ch.	Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Size	Sync Clock
4	Serial mode register 4	SCSMR4	R/W	H'FFEE 0000	H'1FEE 0000	16	Pck
	Bit rate register 4	SCBRR4	R/W	H'FFEE 0004	H'1FEE 0004	8	Pck
	Serial control register 4	SCSCR4	R/W	H'FFEE 0008	H'1FEE 0008	16	Pck
	Transmit FIFO data register 4	SCFTDR4	W	H'FFEE 000C	H'1FEE 000C	8	Pck
	Serial status register 4	SCFSR4	R/W*[1]	H'FFEE 0010	H'1FEE 0010	16	Pck
	Receive FIFO data register 4	SCFRDR4	R	H'FFEE 0014	H'1FEE 0014	8	Pck
	FIFO control register 4	SCFCR4	R/W	H'FFEE 0018	H'1FEE 0018	16	Pck
	Transmit FIFO data count register 4	SCTFDR4	R	H'FFEE 001C	H'1FEE 001C	16	Pck
	Receive FIFO data count register 4	SCRFDR4	R	H'FFEE 0020	H'1FEE 0020	16	Pck
	Serial port register 4	SCSPTR4	R/W	H'FFEE 0024	H'1FEE 0024	16	Pck
	Line status register 4	SCLSR4	R/W*[2]	H'FFEE 0028	H'1FEE 0028	16	Pck
	Serial error register 4	SCRER4	R	H'FFEE 002C	H'1FEE 002C	16	Pck
5	Serial mode register 1	SCSMR5	R/W	H'FFEF 0000	H'1FEF 0000	16	Pck
	Bit rate register 5	SCBRR5	R/W	H'FFEF 0004	H'1FEF 0004	8	Pck
	Serial control register 5	SCSCR5	R/W	H'FFEF 0008	H'1FEF 0008	16	Pck
	Transmit FIFO data register 5	SCFTDR5	W	H'FFEF 000C	H'1FEF 000C	8	Pck
	Serial status register 5	SCFSR5	R/W*[1]	H'FFEF 0010	H'1FEF 0010	16	Pck
	Receive FIFO data register 5	SCFRDR5	R	H'FFEF 0014	H'1FEF 0014	8	Pck
	FIFO control register 5	SCFCR5	R/W	H'FFEF 0018	H'1FEF 0018	16	Pck
	Transmit FIFO data count register 5	SCTFDR5	R	H'FFEF 001C	H'1FEF 001C	16	Pck
	Receive FIFO data count register 5	SCRFDR5	R	H'FFEF 0020	H'1FEF 0020	16	Pck
	Serial port register 5	SCSPTR5	R/W	H'FFEF 0024	H'1FEF 0024	16	Pck
	Line status register 5	SCLSR5	R/W*[2]	H'FFEF 0028	H'1FEF 0028	16	Pck
	Serial error register 5	SCRER5	R	H'FFEF 002C	H'1FEF 002C	16	Pck

Table 24.2 Register Configuration (2)

Ch.	Register Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/ WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction	Module Standby
0	Serial mode register 0	SCSMR0	H'0000	Retained	Retained	Retained
	Bit rate register 0	SCBRR0	H'FF	Retained	Retained	Retained
	Serial control register 0	SCSCR0	H'0000	Retained	Retained	Retained
	Transmit FIFO data register 0	SCFTDR0	Undefined	Undefined	Retained	Retained
	Serial status register 0	SCFSR0	H'0060	Retained	Retained	Retained
	Receive FIFO data register 0	SCFRDR0	Undefined	Undefined	Retained	Retained
	FIFO control register 0	SCFCR0	H'0000	Retained	Retained	Retained
	Transmit FIFO data count register 0	SCTFDR0	H'0000	Retained	Retained	Retained
	Receive FIFO data count register 0	SCRFDR0	H'0000	Retained	Retained	Retained
	Serial port register 0	SCSPTR0	H'0000*[3]	Retained	Retained	Retained
	Line status register 0	SCLSR0	H'0000	Retained	Retained	Retained
	Serial error register 0	SCRER0	H'0000	Retained	Retained	Retained
1	Serial mode register 1	SCSMR1	H'0000	Retained	Retained	Retained
	Bit rate register 1	SCBRR1	H'FF	Retained	Retained	Retained
	Serial control register 1	SCSCR1	H'0000	Retained	Retained	Retained
	Transmit FIFO data register 1	SCFTDR1	Undefined	Undefined	Retained	Retained
	Serial status register 1	SCFSR1	H'0060	Retained	Retained	Retained
	Receive FIFO data register 1	SCFRDR1	Undefined	Undefined	Retained	Retained
	FIFO control register 1	SCFCR1	H'0000	Retained	Retained	Retained
	Transmit FIFO data count register 1	SCTFDR1	H'0000	Retained	Retained	Retained
	Receive FIFO data count register 1	SCRFDR1	H'0000	Retained	Retained	Retained
	Serial port register 1	SCSPTR1	H'0000*[4]	Retained	Retained	Retained
	Line status register 1	SCLSR1	H'0000	Retained	Retained	Retained
	Serial error register 1	SCRER1	H'0000	Retained	Retained	Retained

Ch.	Register Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/ WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction	Module Standby
2	Serial mode register 2	SCSMR2	H'0000	Retained	Retained	Retained
	Bit rate register 2	SCBRR2	H'FF	Retained	Retained	Retained
	Serial control register 2	SCSCR2	H'0000	Retained	Retained	Retained
	Transmit FIFO data register 2	SCFTDR2	Undefined	Undefined	Retained	Retained
	Serial status register 2	SCFSR2	H'0060	Retained	Retained	Retained
	Receive FIFO data register 2	SCFRDR2	Undefined	Undefined	Retained	Retained
	FIFO control register 2	SCFCR2	H'0000	Retained	Retained	Retained
	Transmit FIFO data count register 2	SCTFDR2	H'0000	Retained	Retained	Retained
	Receive FIFO data count register 2	SCRFDR2	H'0000	Retained	Retained	Retained
	Serial port register 2	SCSPTR2	H'0000*[4]	Retained	Retained	Retained
	Line status register 2	SCLSR2	H'0000	Retained	Retained	Retained
	Serial error register 2	SCRER2	H'0000	Retained	Retained	Retained
3	Serial mode register 3	SCSMR3	H'0000	Retained	Retained	Retained
	Bit rate register 3	SCBRR3	H'FF	Retained	Retained	Retained
	Serial control register 3	SCSCR3	H'0000	Retained	Retained	Retained
	Transmit FIFO data register 3	SCFTDR3	Undefined	Undefined	Retained	Retained
	Serial status register 3	SCFSR3	H'0060	Retained	Retained	Retained
	Receive FIFO data register 3	SCFRDR3	Undefined	Undefined	Retained	Retained
	FIFO control register 3	SCFCR3	H'0000	Retained	Retained	Retained
	Transmit FIFO data count register 3	SCTFDR3	H'0000	Retained	Retained	Retained
	Receive FIFO data count register 3	SCRFDR3	H'0000	Retained	Retained	Retained
	Serial port register 3	SCSPTR3	H'0000*[4]	Retained	Retained	Retained
	Line status register 3	SCLSR3	H'0000	Retained	Retained	Retained
	Serial error register 3	SCRER3	H'0000	Retained	Retained	Retained

Ch.	Register Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/ WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction	Module Standby
4	Serial mode register 4	SCSMR4	H'0000	Retained	Retained	Retained
	Bit rate register 4	SCBRR4	H'FF	Retained	Retained	Retained
	Serial control register 4	SCSCR4	H'0000	Retained	Retained	Retained
	Transmit FIFO data register 4	SCFTDR4	Undefined	Undefined	Retained	Retained
	Serial status register 4	SCFSR4	H'0060	Retained	Retained	Retained
	Receive FIFO data register 4	SCFRDR4	Undefined	Undefined	Retained	Retained
	FIFO control register 4	SCFCR4	H'0000	Retained	Retained	Retained
	Transmit FIFO data count register 4	SCTFDR4	H'0000	Retained	Retained	Retained
	Receive FIFO data count register 4	SCRFDR4	H'0000	Retained	Retained	Retained
	Serial port register 4	SCSPTR4	H'0000*4	Retained	Retained	Retained
	Line status register 4	SCLSR4	H'0000	Retained	Retained	Retained
	Serial error register 4	SCRER4	H'0000	Retained	Retained	Retained
5	Serial mode register 5	SCSMR5	H'0000	Retained	Retained	Retained
	Bit rate register 5	SCBRR5	H'FF	Retained	Retained	Retained
	Serial control register 5	SCSCR5	H'0000	Retained	Retained	Retained
	Transmit FIFO data register 5	SCFTDR5	Undefined	Undefined	Retained	Retained
	Serial status register 5	SCFSR5	H'0060	Retained	Retained	Retained
	Receive FIFO data register 5	SCFRDR5	Undefined	Undefined	Retained	Retained
	FIFO control register 5	SCFCR5	H'0000	Retained	Retained	Retained
	Transmit FIFO data count register 5	SCTFDR5	H'0000	Retained	Retained	Retained
	Receive FIFO data count register 5	SCRFDR5	H'0000	Retained	Retained	Retained
	Serial port register 5	SCSPTR5	H'0000*4	Retained	Retained	Retained
	Line status register 5	SCLSR5	H'0000	Retained	Retained	Retained
	Serial error register 5	SCRER5	H'0000	Retained	Retained	Retained

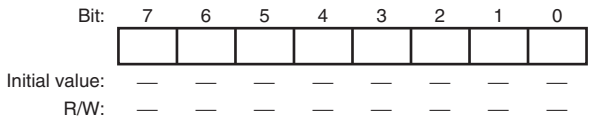
Notes: 1. Only 0 can be written to bits 7 to 4, 1, and 0 to clear the flags.
2. Only 0 can be written to bit 0 to clear the flags.
3. Bits 2 and 0 are undefined.
4. Bits 6, 4, 2, and 0 are undefined.

24.3.1 Receive Shift Register (SCRSR)

SCRSR is the register used to receive serial data.

The SCIF sets serial data input from the SCIF_RXD pin in SCRSR in the order received, starting with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to SCFRDR, automatically.

SCRSR cannot be directly read from and written to by the CPU.



Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—

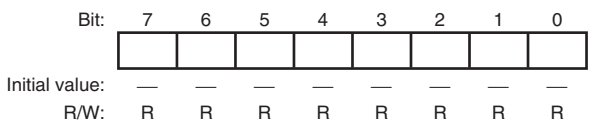
24.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is an 8-bit FIFO register of 64 stages that stores received serial data.

When the SCIF has received one byte of serial data, it transfers the received data from SCRSR to SCFRDR where it is stored, and completes the receive operation. SCRSR is then enabled for reception, and consecutive receive operations can be performed until SCFRDR is full.

SCFRDR is a read-only register, and cannot be written to by the CPU.

If a read is performed when there is no receive data in SCFRDR, an undefined value is returned. When SCFRDR is full of receive data, subsequent serial data is lost.



Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

24.3.3 Transmit Shift Register (SCTSR)

SCTSR is a register used to transmit serial data.

To perform serial data transmission, the SCIF first transfers transmit data from SCFTDR to SCTSR, then sends the data to the TXD pin starting with the LSB (bit 0). When transmission of one byte of serial data is completed, the next transmit data is automatically transferred from SCFTDR to SCTSR, and transmission started.

SCTSR cannot be directly read from and written to by the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—

24.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is an 8-bit FIFO register of 64 stages that stores data for serial transmission.

If SCTSR is empty when transmit data has been written to SCFTDR, the SCIF transfers the transmit data written in SCFTDR to SCTSR and starts serial transmission.

SCFTDR is a write-only register, and cannot be read from by the CPU. The next data cannot be written when SCFTDR is filled with 64 bytes of transmit data. Data written in this case is ignored.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W

24.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit register used to set the SCIF's serial communication format and select the baud rate generator clock source.

SCSMR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects asynchronous mode or clocked synchronous mode as the SCIF operating mode. 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length Selects 7 or 8 bits as the asynchronous mode data length. In clocked synchronous mode, the data length is fixed at 8 bits regardless of the CHR bit setting. When 7-bit data is selected, the MSB (bit 7) of SCFTDR is not transmitted. 0: 8-bit data 1: 7-bit data
5	PE	0	R/W	Parity Enable In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking is performed in reception. In clocked synchronous mode, parity bit addition and checking is disabled regardless of the PE bit setting. 0: Parity bit addition and checking disabled 1: Parity bit addition and checking enabled*[1]

Bit	Bit Name	Initial Value	R/W	Description
4	O/$\overline{E}$	0	R/W	Parity Mode Selects either even or odd parity for use in parity addition and checking. In asynchronous mode, the O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking. In clocked synchronous mode or when parity addition and checking is disabled in asynchronous mode, the O/$\overline{E}$ bit setting is invalid. 0: Even parity 1: Odd parity When even parity is set, parity bit addition is performed in transmission so that the total number of 1-bits in the transmit character plus the parity bit is even. In reception, a check is performed to see if the total number of 1-bits in the receive character plus the parity bit is even. When odd parity is set, parity bit addition is performed in transmission so that the total number of 1-bits in the transmit character plus the parity bit is odd. In reception, a check is performed to see if the total number of 1-bits in the receive character plus the parity bit is odd.
3	STOP	0	R/W	Stop Bit Length In asynchronous mode, selects 1 or 2 bits as the stop bit length. The stop bit setting is valid only in asynchronous mode. Since the stop bit is not added in clocked synchronous mode, the STOP bit setting is invalid. 0: 1 stop bit*[2] 1: 2 stop bits*[3] In reception, 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. If it is 0, it is treated as the start bit of the next transmit character.

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	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the on-chip baud rate generator. The clock source can be selected from Pck, Pck/4, Pck/16, and Pck/64, according to the CKS1 and CKS0 settings. For details on the relationship among clock sources, bit rate register settings, and baud rate, see section 24.3.8, Bit Rate Register n (SCBRR). 00: Pck clock 01: Pck/4 clock 10: Pck/16 clock 11: Pck/64 clock

Legend:

Pck: Peripheral Clock

Notes: 1. When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

2. In transmission, a single 1-bit (stop bit) is added to the end of a transmit character before it is sent.

3. In transmission, two 1-bits (stop bits) are added to the end of a transmit character before it is sent.

24.3.6 Serial Control Register (SCSCR)

SCSCR is a register used to enable/disable transmission/reception by SCIF, serial clock output, interrupt requests, and to select transmission/reception clock source for the SCIF.

SCSCR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables transmit-FIFO-data-empty interrupt (TXI) request generation when serial transmit data is transferred from SCFTDR to SCTSR, the number of data bytes in SCFTDR falls to or below the transmit trigger setting count, and the TDFE flag in SCFSR is set to 1. TXI interrupt requests can be released by the following methods: Either by reading 1 from the TDFE flag in SCFSR, writing transmit data exceeding the transmit trigger setting count to SCFTDR and then clearing the TDFE flag in SCFSR to 0, or by clearing the TIE bit to 0. 0: Transmit-FIFO-data-empty interrupt (TXI) request disabled 1: Transmit-FIFO-data-empty interrupt (TXI) request enabled

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable*[1] Enables or disables generation of a receive-data-full interrupt (RXI) request when the RDF flag or DR flag in SCFSR is set to 1, a receive-error interrupt (ERI) request when the ER flag in SCFSR is set to 1, and a break interrupt (BRI) request when the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1. 0: Receive-data-full interrupt (RXI) request, receive-error interrupt (ERI) request, and break interrupt (BRI) request disabled 1: Receive-data-full interrupt (RXI) request, receive-error interrupt (ERI) request, and break interrupt (BRI) request enabled
5	TE	0	R/W	Transmit Enable Enables or disables the start of serial transmission by the SCIF. Serial transmission is started when transmit data is written to SCFTDR while the TE bit is set to 1. 0: Transmission disabled 1: Transmission enabled*[2]
4	RE	0	R/W	Receive Enable Enables or disables the start of serial reception by the SCIF. Serial reception is started when a start bit or a synchronization clock input is detected in asynchronous mode or clocked synchronous mode, respectively, while the RE bit is set to 1. It should be noted that clearing the RE bit to 0 does not affect the ER, BRK, FER, PER, RDF, and DR flags in SCFSR, and ORER flag in SCLSR, which retain their states. Serial reception begins once the start bit is detected in these states. 0: Reception disabled 1: Reception enabled*[3]

Bit	Bit Name	Initial Value	R/W	Description
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables generation of receive-error interrupt (ERI) and break interrupt (BRI) requests. The REIE bit setting is valid only when the RIE bit is 0. Receive-error interrupt (ERI) and break interrupt (BRI) requests can be cleared by reading 1 from ER and BRK in SCFSR, or the ORER flag in SCLSR, then clearing the flag to 0, or by clearing the RIE and REIE bits to 0. When REIE is set to 1, ERI and BRI interrupt requests are generated even if RIE is cleared to 0. In DMA transfer, this setting is made if the interrupt controller is to be notified of ERI and BRI interrupt requests. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests disabled 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests enabled
2	—	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
1	CKE1	0	R/W	Clock Enable 1, 0
0	CKE0	0	R/W	These bits select the SCIF clock source and whether to enable or disable the clock output from the SCIF_SCK pin. The CKE1 and CKE0 bits are used together to specify whether the SCIF_SCK pin functions as a serial clock output pin or a serial clock input pin. Note however that the CKE0 bit setting is valid only when an internal clock is selected as the SCIF clock source (CKE1 = 0). When an external clock is selected (CKE1 = 1), the CKE0 bit setting is invalid. The CKE1 and CKE0 bit must be set before determining the SCIF's operating mode with SCSMR. • Asynchronous mode 00: Internal clock/SCIF_SCK pin functions as port according to the SCSPTR settings 01: Internal clock/SCIF_SCK pin functions as clock output*4 1x: External clock/SCIF_SCK pin functions as clock input*5 • Clocked synchronous mode 0x: Internal clock/SCIF_SCK pin functions as synchronization clock output 1x: External clock/SCIF_SCK pin functions as synchronization clock input

Legend:

x: Don't care

Notes: 1. An RXI interrupt request can be canceled by reading 1 from the RDF or DR flag in SCFSR, then clearing the flag to 0, or by clearing the RIE bit to 0. ERI and BRI interrupt requests can be canceled by reading 1 from ER and BRK in SCFSR, or ORER flag in SCFSR, then clearing the flag to 0, or by clearing the RIE and REIE bits to 0.

2. SCSMR and SCFCR settings must be made, the transmission format determined, and the transmit FIFO reset (the TFCL bit in SCFCR set to 1), before the TE bit is set to 1.
3. SCSMR and SCFCR settings must be made, the reception format determined, and the receive FIFO reset (the RFCL bit in SCFCR set to 1), before the RE bit is set to 1.
4. The output clock frequency is 16 times the bit rate.
5. The input clock frequency is 16 times the bit rate.

   (For the relation between the value set in SCBRR and the baud rate generator, see section 24.3.8, Bit Rate register n.)

24.3.7 Serial Status Register n (SCFSR)

SCFSR is a 16-bit register that consists of status flags that indicate the operating status of the SCIF.

SCFSR can be always read from or written to by the CPU. However, 1 cannot be written to the ER, TEND, TDFE, BRK, RDF, and DR flags. Also note that in order to clear these flags they must be read as 1 beforehand. The FER flag and PER flag are read-only flags and cannot be modified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R	R	R/W*	R/W*

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	ER	0	R/W*[1]	Receive Error Indicates that a framing error or parity error occurred during reception. The ER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0. When a receive error occurs, the receive data is still transferred to SCFRDR, and reception continues. The FER and PER bits in SCFSR can be used to determine whether there is a receive error in the readout data from SCFRDR. 0: No framing error or parity error occurred during reception [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ER after reading ER = 1 1: A framing error or parity error occurred during reception [Setting conditions] • When the SCIF checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0*[2] • When, in reception, the number of 1-bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SCSMR.
6	TEND	1	R/W*[1]	Transmit End Indicates that transmission has been ended without valid data in SCFTDR on transmission of the last bit of the transmit character. 0: Transmission is in progress [Clearing conditions] • When transmit data is written to SCFTDR, and 0 is written to TEND after reading TEND = 1 • When data is written to SCFTDR by the DMAC 1: Transmission has been ended [Setting conditions] • Power-on reset or manual reset • When the TE bit in SCSCR is 0 • When there is no transmit data in SCFTDR on transmission of the last bit of a 1-byte serial transmit character

Bit	Bit Name	Initial Value	R/W	Description
5	TDFE	1	R/W*[1]	Transmit FIFO Data Empty Indicates that data has been transferred from SCFTDR to SCTSR, the number of data bytes in SCFTDR has fallen to or below the transmit trigger data count set by the TTRG1 and TTRG0 bits in SCFCR, and new transmit data can be written to SCFTDR. 0: A number of transmit data bytes exceeding the transmit trigger setting count have been written to SCFTDR [Clearing conditions] • When transmit data exceeding the transmit trigger setting count is written to SCFTDR after reading TDFE = 1, and 0 is written to TDFE • When transmit data exceeding the transmit trigger setting count is written to SCFTDR by the DMAC 1: The number of transmit data bytes in SCFTDR does not exceed the transmit trigger setting count [Setting conditions] • Power-on reset or manual reset • When the number of SCFTDR transmit data bytes falls to or below the transmit trigger setting count as the result of a transmit operation*[3]
4	BRK	0	R/W*[1]	Break Detection Indicates that a receive data break signal has been detected. 0: A break signal has not been received [Clearing conditions] • Power-on reset or manual reset • When 0 is written to BRK after reading BRK = 1 1: A break signal has been received*[4] [Setting condition] • When data with a framing error is received, followed by the space 0 level (low level) for at least one frame length

Bit	Bit Name	Initial Value	R/W	Description
3	FER	0	R	Framing Error Display In asynchronous mode, indicates whether or not a framing error has been found in the data that is to be read from SCFRDR. 0: There is no framing error that is to be read next from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no framing error in the data that is to be read next from SCFRDR 1: There is a framing error that is to be read from SCFRDR [Setting condition] • When there is a framing error in the data that is to be read next from SCFRDR
2	PER	0	R	Parity Error Display In asynchronous mode, indicates whether or not a parity error has been found in the data that is to be read next from SCFRDR. 0: There is no parity error that is to be read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no parity error in the data that is to be read next from SCFRDR 1: There is a parity error in the receive data that is to be read next from SCFRDR [Setting condition] • When there is a parity error in the data that is to be read next from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/W*[1]	Receive FIFO Data Full Indicates that the received data has been transferred from SCRSR to SCFRDR, and the number of receive data bytes in SCFRDR is equal to or greater than the receive trigger count set by the RTRG1 and RTRG0 bits in SCFCR. 0: The number of receive data bytes in SCFRDR is less than the receive trigger setting count. [Clearing conditions] • Power-on reset or manual reset • When SCFRDR is read until the number of receive data bytes in SCFRDR falls below the receive trigger setting count after reading RDF = 1, and 0 is written to RDF • When SCFRDR is read by the DMAC until the number of receive data bytes in SCFRDR falls below the receive trigger setting count 1: The number of receive data bytes in SCFRDR is equal to or greater than the receive trigger setting count [Setting condition] • When SCFRDR contains at least the receive trigger setting count of receive data bytes*[5]

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/W*[1]	Receive Data Ready In asynchronous mode, indicates that there are fewer than the receive trigger setting count of data bytes in SCFRDR, and no further data has arrived for at least 15 etu after the stop bit of the last data received. This is not set when using clocked synchronous mode. 0: Reception is in progress or has ended normally and there is no receive data left in SCFRDR [Clearing conditions] • Power-on reset or manual reset • When all the receive data in SCFRDR has been read after reading DR = 1, and 0 is written to DR • When all the receive data in SCFRDR has been read by the DMAC 1: No further receive data has arrived [Setting condition] • When SCFRDR contains fewer than the receive trigger setting count of receive data bytes, and no further data has arrived for at least 15 etu after the stop bit of the last data received*[6]

Legend:

etu: Elementary time unit (time for transfer of 1 bit)

Notes: 1. Only 0 can be written to clear the flag.

2. In 2-stop bit mode, only the first stop bit is checked for a value of 1; the second stop bit is not checked.
3. As SCFTDR is a 64-byte FIFO register, the maximum number of bytes that can be written when TDFE = 1 is 64 − (transmit trigger setting count). Data written in excess of this is ignored. The upper bits of SCTFDR indicate the number of data bytes transmitted to SCFTDR.
4. When a break is detected, the receive data (H'00) following detection is not transferred to SCFRDR. When the break ends and the receive signal returns to mark 1, receive data transfer is resumed.
5. SCFRDR is a 64-byte FIFO register. When RDF = 1, at least the receive trigger setting count of data bytes can be read. If all the data in SCFRDR is read and another read is performed, the data value is undefined. The number of receive data bytes in SCFRDR is indicated by SCRFDR.
6. Equivalent to 1.5 frames with an 8-bit, 1-stop-bit format

24.3.8 Bit Rate Register n (SCBRR)

SCBRR is an 8-bit register that sets the serial transmission/reception bit rate in accordance with the baud rate generator operating clock selected by the CKS1 and CKS0 bits in SCSMR.

SCBRR can always be read from and written to by the CPU.

The SCBRR setting is found from the following equation.

Asynchronous mode:

$$N = \frac{Pck}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clocked synchronous mode:

$$N = \frac{Pck}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bit/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
Pck: Peripheral module operating frequency (MHz)
n: 0, 1, 2, 3
(See table 24.3 for the relation between n and the baud rate generator input clock.)

Table 24.3 SCSMR Settings

		SCSMR Setting	
n	**Baud Rate Generator Input Clock**	**CKS1**	**CKS0**
0	Pck	0	0
1	Pck/4	0	1
2	Pck/16	1	0
3	Pck/64	1	1

The bit rate error in asynchronous mode is found from the following equation:

$$Error\ (\%) = \left\{ \frac{Pck \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

24.3.9 FIFO Control Register n (SCFCR)

SCFCR is a register that performs data count resetting and trigger data number setting for transmit and receive FIFO registers, and also contains a loopback test enable bit.

SCFCR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RSTRG2*1	RSTRG1*1	RSTRG0*1	RTRG1	RTRG0	TTRG1	TTRG0	MCE*1	TFCL	RFCL	LOOP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	RSTRG2*1	0	R/W	$\overline{\text{SCIF\_RTS}}$ Output Active Trigger
9	RSTRG1*1	0	R/W	The $\overline{\text{SCIF\_RTS}}$ signal becomes high when the number of receive data stored in SCFRDR exceeds the trigger setting count shown below.
8	RSTRG0*1	0	R/W	000:63 001:1 010:8 011:16 100:32 101:48 110:54 111:60
7	RTRG1	0	R/W	Receive FIFO Data Count Trigger
6	RTRG0	0	R/W	These bits are used to set the number of receive data bytes that sets the RDF flag in SCFSR. The RDF flag is set when the number of receive data bytes in SCFRDR is equal to or greater than the trigger setting count shown below. 00:1 01:16 10:32 11:48

Bit	Bit Name	Initial Value	R/W	Description
5	TTRG1	0	R/W	Transmit FIFO Data Count Trigger
4	TTRG0	0	R/W	These bits are used to set the number of remaining transmit data bytes that sets the TDFE flag in SCFSR. The TDFE flag is set when the number of transmit data bytes in SCFTDR is equal to or less than the trigger setting count shown below. 00: 32 (32)*[2] 01:16 (48) 10: 2 (62) 11: 0 (64)
3	MCE*[1]	0	R/W	Modem Control Enable Enables the $\overline{\text{SCIF\_CTS}}$ and $\overline{\text{SCIF\_RTS}}$ modem control signals. Always set the MCE bit to 0 in clocked synchronous mode. 0: Modem signals disabled*[3] 1: Modem signals enabled
2	TFCL	0	R/W	Transmit FIFO Data Count Register Clear Clears the transmit data count in the transmit FIFO data count register to 0. 0: The FIFO data count not cleared*[4] 1: The FIFO data count cleared to 0
1	RFCL	0	R/W	Receive FIFO Data Count Register Clear Clears the receive data count in the receive FIFO data count register to 0. 0: The FIFO data count not cleared*[4] 1: The FIFO data count cleared
0	LOOP	0	R/W	Loopback Test Internally connects the transmit output pin (SCIF_TXD) and receive input pin (SCIF_RXD), and the $\overline{\text{SCIF\_RTS}}$ pin and $\overline{\text{SCIF\_CTS}}$ pin, enabling loopback testing. 0: Loopback test disabled 1: Loopback test enabled

Notes: 1. Only channel 0. Reserved bit in channels 1 to 5.
2. Figures in parentheses are the number of empty bytes in SCFTDR when the flag is set.
3. $\overline{\text{SCIF\_CTS}}$ is fixed at active-0 regardless of the input value, and $\overline{\text{SCIF\_RTS}}$ output is also fixed at 0.
4. A reset operation is performed in the event of a power-on reset or manual reset.

24.3.10 Transmit FIFO Data Count Register n (SCTFDR)

SCTFDR is a 16-bit register that indicates the number of transmit data bytes stored in SCFTDR.

SCTFDR can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	T6	T5	T4	T3	T2	T1	T0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	T6 to T0	All 0	R	These bits show the number of untransmitted data bytes in SCFTDR. A value of H'00 indicates that there is no transmit data, and a value of H'40 indicates that SCFTDR is full of the maximum number (64 bytes) of transmit data.

24.3.11 Receive FIFO Data Count Register n (SCRFDR)

SCRFDR is a 16-bit register that indicates the number of receive data bytes stored in SCFRDR.

SCRFDR can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	R6	R5	R4	R3	R2	R1	R0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	R6 to R0	All 0	R	These bits show the number of receive data bytes in SCFRDR. A value of H'00 indicates that there is no receive data, and a value of H'40 indicates that SCFRDR is full of receive data.

24.3.12 Serial Port Register n (SCSPTR)

SCSPTR is a 16-bit readable/writable register that controls input/output and data for the port pins multiplexed with the serial communication interface (SCIF) pins at all times. Input data can be read from the RXD pin, and output data written to the TXD pin, and breaks in serial transmission/reception controlled, by means of bits 1 and 0.

All SCSPTR bits except bits 6, 4, 2, and 0 are initialized to 0 by a power-on reset or manual reset; the value of bits 6, 4, 2, and 0 is undefined. SCSPTR is not initialized in the module standby state.

Note that when reading data via a serial port pin in the SCIF, the peripheral clock value from 2 cycles before is read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RTS IO*	RTS DT*	CTS IO*	CTS DT*	SCK IO	SCK DT	SPB2 IO	SPB2 DT
Initial value:	0	0	0	0	0	0	0	0	0	—	0	—	0	—	0	—
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	RTSIO*	0	R/W	Serial Port $\overline{\text{SCIF\_RTS}}$ Port Input/Output Specifies the serial port $\overline{\text{SCIF\_RTS}}$ pin input/output condition. When actually setting the $\overline{\text{SCIF\_RTS}}$ pin as a port output pin to output the value set by the RTSDT bit, the MCE bit in SCFCR should be cleared to 0. 0: RTSDT bit value is not output to $\overline{\text{SCIF\_RTS}}$ pin 1: RTSDT bit value is output to $\overline{\text{SCIF\_RTS}}$ pin
6	RTSDT*	—	R/W	Serial Port $\overline{\text{SCIF\_RTS}}$ Port Data Specifies the serial port $\overline{\text{SCIF\_RTS}}$ pin input/output data. Input or output is specified by the RTSIO bit. In output mode, the RTSDT bit value is output to the $\overline{\text{SCIF\_RTS}}$ pin. The $\overline{\text{SCIF\_RTS}}$ pin value is read from the RTSDT bit regardless of the value of the RTSIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

Bit	Bit Name	Initial Value	R/W	Description
5	CTSIO*	0	R/W	Serial Port $\overline{\text{SCIF\_CTS}}$ Port Input/Output Specifies the serial port $\overline{\text{SCIF\_CTS}}$ pin input/output condition. When actually setting the $\overline{\text{SCIF\_CTS}}$ pin as a port output pin to output the value set by the CTSDT bit, the MCE bit in SCFCR should be cleared to 0. 0: CTSDT bit value is not output to $\overline{\text{SCIF\_CTS}}$ pin 1: CTSDT bit value is output to $\overline{\text{SCIF\_CTS}}$ pin
4	CTSDT*	—	R/W	Serial Port $\overline{\text{SCIF\_CTS}}$ Port Data Specifies the serial port $\overline{\text{SCIF\_CTS}}$ pin input/output data. Input or output is specified by the CTSIO bit. In output mode, the CTSDT bit value is output to the $\overline{\text{SCIF\_CTS}}$ pin. The $\overline{\text{SCIF\_CTS}}$ pin is read from the CTSDT bit regardless of the value of the CTSIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level
3	SCKIO	0	R/W	Serial Port Clock Port Input/Output Specifies the serial port SCIF_SCK pin input/output condition. When actually setting the SCIF_SCK pin as a port output pin to output the value set by the SCKDT bit, the CKE1 and CKE0 bits in SCSCR should be cleared to 0. 0: SCKDT bit value is not output to SCIF_SCK pin 1: SCKDT bit value is output to SCIF_SCK pin
2	SCKDT	—	R/W	Serial Port Clock Port Data Specifies the serial port SCIF_SCK pin input/output data. Input or output is specified by the SCKIO bit. In output mode, the SCKDT bit value is output to the SCIF_SCK pin. The SCIF_SCK pin value is read from the SCKDT bit regardless of the value of the SCKIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

Bit	Bit Name	Initial Value	R/W	Description
1	SPB2IO	0	R/W	Serial Port Break Input/Output This bit specifies serial port SCIF_TXD pin output condition. When actually setting the SCIF_TXD pin as a port output pin to output the value set by the SPB2DT bit, the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value is not output to the SCIF_TXD pin 1: SPB2DT bit value is output to the SCIF_TXD pin
0	SPB2DT	—	R/W	Serial Port Break Data Specifies the serial port SCIF_RXD pin input data and SCIF_TXD pin output data. The SCIF_TXD pin output condition is specified by the SPB2IO bit. When the SCIF_TXD pin is designated as an output, the value of the SPB2DT bit is output to the SCIF_TXD pin. The SCIF_RXD pin value is read from the SPB2DT bit regardless of the value of the SPB2IO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

Note: * Only channel 0. Reserved bit in channels 1 to 5.

24.3.13 Line Status Register n (SCLSR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ORER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ORER	0	R/W*[1]	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. 0: Reception in progress, or reception has ended normally*[2] [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ORER after reading ORER = 1, 1: An overrun error occurred during reception*[3] [Setting condition] • When the next serial reception is completed while SCFRDR receives 64-byte data (SCFRDR is full)

Notes: 1. Only 0 can be written to clear the flag.
2. The ORER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0.
3. The receive data prior to the overrun error is retained in SCFRDR, and the data received subsequently is lost. Serial reception cannot be continued while the ORER flag is set to 1.

24.3.14 Serial Error Register n (SCRER)

SCRER is a 16-bit register that indicates the number of receive errors in the data in SCFRDR. SCRER can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PER5	PER4	PER3	PER2	PER1	PER0	—	—	FER5	FER4	FER3	FER2	FER1	FER0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PER5	0	R	Number of Parity Errors
12	PER4	0	R	These bits indicate the number of data bytes in which a parity error occurred in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits PER5 to PER0 is the number of data bytes in which a parity error occurred. If all 64 bytes of receive data in SCFRDR have parity errors, the value indicated by bits PER5 to PER0 is 0.
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	Number of Framing Errors
4	FER4	0	R	These bits indicate the number of data bytes in which a framing error occurred in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits FER5 to FER0 is the number of data bytes in which a framing error occurred. If all 64 bytes of receive data in SCFRDR have framing errors, the value indicated by bits FER5 to FER0 is 0.
3	FER3	0	R	
2	FER2	0	R	
1	FER1	0	R	
0	FER0	0	R	

24.4 Operation

24.4.1 Overview

The SCIF can perform serial communication in asynchronous mode, in which synchronization is achieved character by character and in clocked synchronous mode, in which synchronization is achieved with clock pulses. For details on asynchronous mode, see section 24.4.2, Operation in Asynchronous Mode.

64-stage FIFO buffers are provided for both transmission and reception, reducing the CPU overhead, and enabling fast and continuous communication to be performed.

$\overline{\text{SCIF\_RTS}}$ and $\overline{\text{SCIF\_CTS}}$ signals are also provided as modem control signals (only in channel 0).

The serial transfer format is selected using SCSMR as shown in table 24.4. The SCIF clock source is determined by the combination of the C/$\overline{\text{A}}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR, as shown in table 24.4.

(1) Asynchronous Mode:

- Data length: Choice of 7 or 8 bits
- LSB first for data transmission and reception
- Choice of parity addition and addition of 1 or 2 stop bits (the combination of these parameters determines the transfer format and character length)
- Detection of framing errors, parity errors, receive-FIFO-data-full state, overrun errors, receive-data-ready state, and breaks, during reception
- Indication of the number of data bytes stored in the transmit and receive FIFO registers
- Choice of peripheral clock (Pck) or SCIF_SCK pin input as SCIF clock source

  When peripheral clock is selected: The SCIF operates on the baud rate generator clock and can output a clock with frequency of 16 times the bit rate.

  When SCIF_SCK pin input is selected: A clock with frequency of 16 times the bit rate must be input (the on-chip baud rate generator is not used).

Clocked Synchronous Mode

- Data length: Fixed at 8 bits
- LSB first for data transmission and reception
- Detection of overrun errors during reception
- Choice of peripheral clock (Pck) or SCIF_SCK pin input as SCIF clock source

  When peripheral clock (Pck) is selected: The SCIF operates on the baud rate generator clock and a serial clock is output to external devices.

  When SCIF_SCK pin input is selected: The on-chip baud rate generator is not used and the SCIF operates on the input serial clock.

Table 24.4 SCSMR Settings for Serial Transfer Format Selection

SCSMR Settings					SCIF Transfer Format		
Bit 7: C/$\overline{A}$	**Bit 6: CHR**	**Bit 5: PE**	**Bit 3: STOP**	**Mode**	**Data Length**	**Parity Bit**	**Stop Bit Length**
0	0	0	0	Asynchronous mode	8-bit data	No	1 bit
			1				2 bits
		1	0			Yes	1 bit
			1				2 bits
	1	0	0		7-bit data	No	1 bit
			1				2 bits
		1	0			Yes	1 bit
			1				2 bits
1	x	x	x	Clocked synchronous mode	8-bit data	No	No

Note: x: Don't care

Table 24.5 SCSMR and SCSCR Settings for SCIF Clock Source Selection

SCSMR	SCSCR Settings				
Bit 7: C/$\overline{A}$	**Bit 1: CKE1**	**Bit 0: CKE0**	**Mode**	**Clock Source**	**SCIF_SCK Pin Function**
0	0	0	Asynchronous mode	Internal	SCIF does not use SCIF_SCK pin
		1			Outputs clock with frequency of 16 times the bit rate
	1	0		External	Inputs clock with frequency of 16 times the bit rate
		1			
1	0	0	Clocked synchronous mode	Internal	Outputs synchronization clock
		1			
	1	0		External	Inputs synchronization clock
		1			

24.4.2 Operation in Asynchronous Mode

In asynchronous mode, a character that consists of data with a start bit indicating the start of communication and a stop bit indicating the end of communication is transmitted or received. In this mode, serial communication is performed with synchronization achieved character by character.

Inside the SCIF, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and receiver have a 64-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

Figure 24.7 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCIF monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication.

One character in serial communication consists of a start bit (low level), followed by transmit/receive data (LSB first; from the lowest bit), a parity bit (high or low level), and finally stop bits (high level).

In reception in asynchronous mode, the SCIF synchronizes with the fall of the start bit. Receive data can be latched at the middle of each bit because the SCIF samples data at the eighth clock with frequency of 16 times the bit rate.

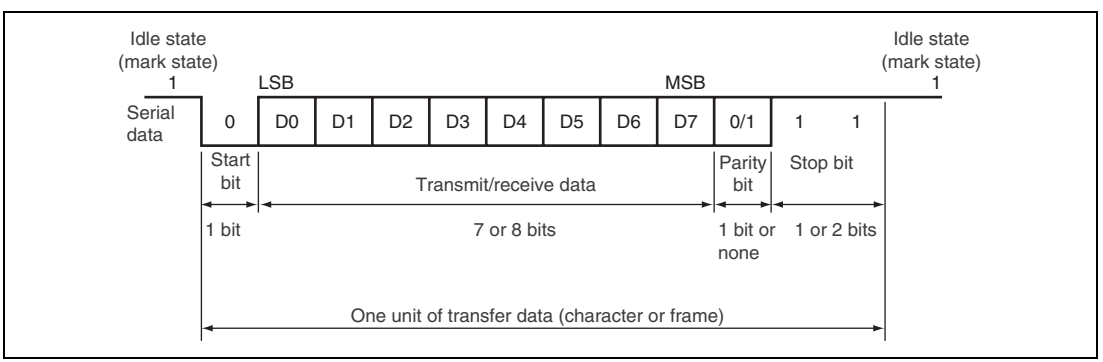


Figure 24.7 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, and Two Stop Bits)

(1) Data Transfer Format

Table 24.6 shows the data transfer formats that can be used. Any of 8 transfer formats can be selected according to the SCSMR settings.

Table 24.6 Serial Transfer Formats (Asynchronous Mode)

SCSMR Settings			Serial Transfer Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	S	8-bit data								STOP		
0	0	1	S	8-bit data								STOP	STOP	
0	1	0	S	8-bit data								P	STOP	
0	1	1	S	8-bit data								P	STOP	STOP
1	0	0	S	7-bit data							STOP			
1	0	1	S	7-bit data							STOP	STOP		
1	1	0	S	7-bit data							P	STOP		
1	1	1	S	7-bit data							P	STOP	STOP	

Legend:
S: Start bit
STOP: Stop bit
P: Parity bit

(2) Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCIF_SCK pin can be selected as the SCIF's serial clock, according to the settings of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details on SCIF clock source selection, see table 24.5.

When an external clock is input to the SCIF_SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCIF is operated on an internal clock, a clock with frequency of 16 times the bit rate is output from the SCIF_SCK pin.

(3) SCIF Initialization (Asynchronous Mode)

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 operating mode or transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure.

1. When the TE bit is cleared to 0, SCTSR is initialized. Note that clearing the TE and RE bits to 0 does not change the contents of SCFSR, SCFTDR, or SCFRDR.
2. The TE bit should be cleared to 0 after all transmit data has been sent and the TEND flag in SCFSR has been set. The TE bit can also be cleared to 0 during transmission, but the data being transmitted goes to the mark state after the clearance. Before setting TE again to start transmission, the TFRST bit in SCFCR should first be set to 1 to reset SCFTDR.
3. When an external clock is used, the clock should not be stopped during operation, including initialization, since operation will be unreliable in this case.

Figure 24.8 shows a sample SCIF initialization flowchart.

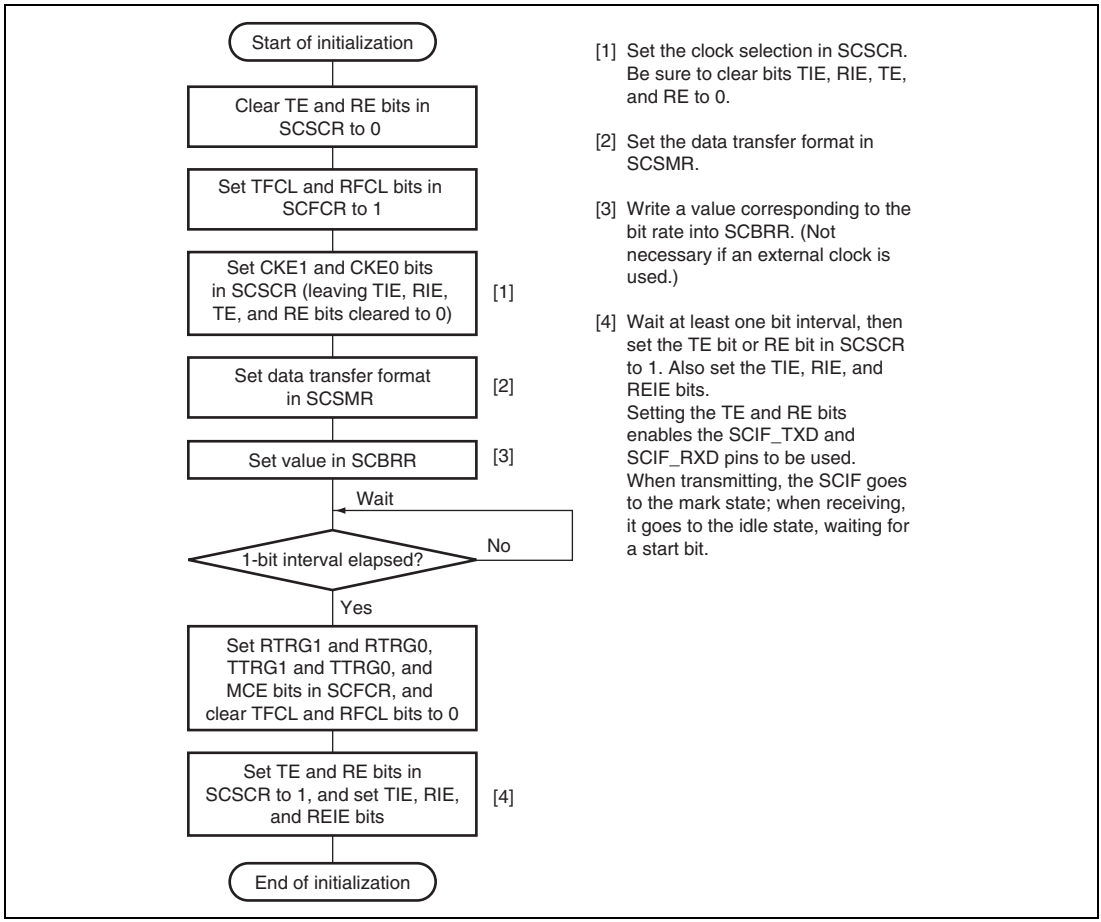


Figure 24.8 Sample SCIF Initialization Flowchart

(4) Serial Data Transmission (Asynchronous Mode)

Figure 24.9 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

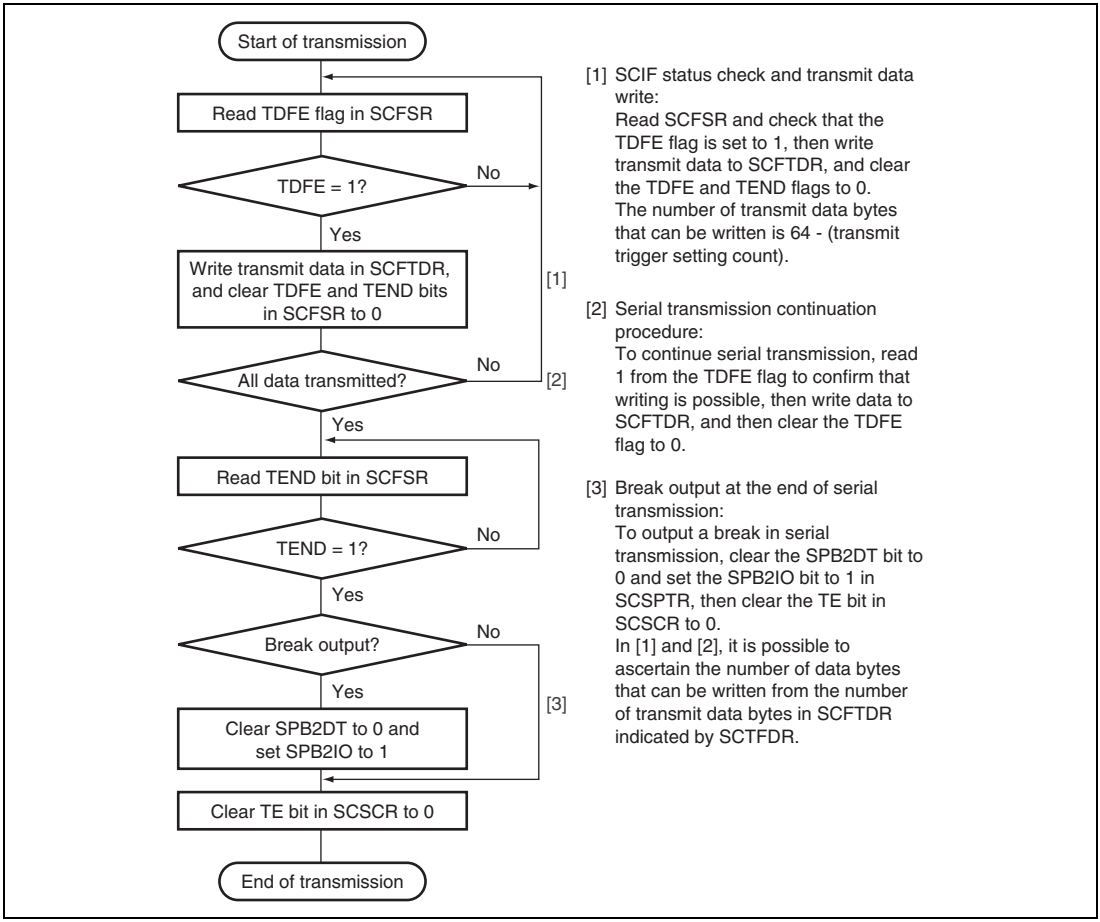


Figure 24.9 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as follows.

1. When data is written to SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmission. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least 64 – (transmit trigger setting count).
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 to or below the transmit trigger count set in SCFCR, the TDFE flag is set. If the TIE bit in SCSCR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the SCIF_TXD pin in the following order.

   (a) Start bit: One 0-bit is output.

   (b) Transmit data: 8-bit or 7-bit data is output in LSB-first order.

   (c) Parity bit: One parity bit (even or odd parity) is output. A format in which a parity bit is not output can also be selected.

   (d) Stop bit(s): One or two 1-bits (stop bits) are output.

   (e) Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If there is no transmit data after the stop bit is sent, the TEND flag in SCFSR is set to 1, and then the SCIF goes to the mark state in which 1 is output from the SCIF_TXD pin.

Figure 24.10 shows an example of the operation for transmission in asynchronous mode.

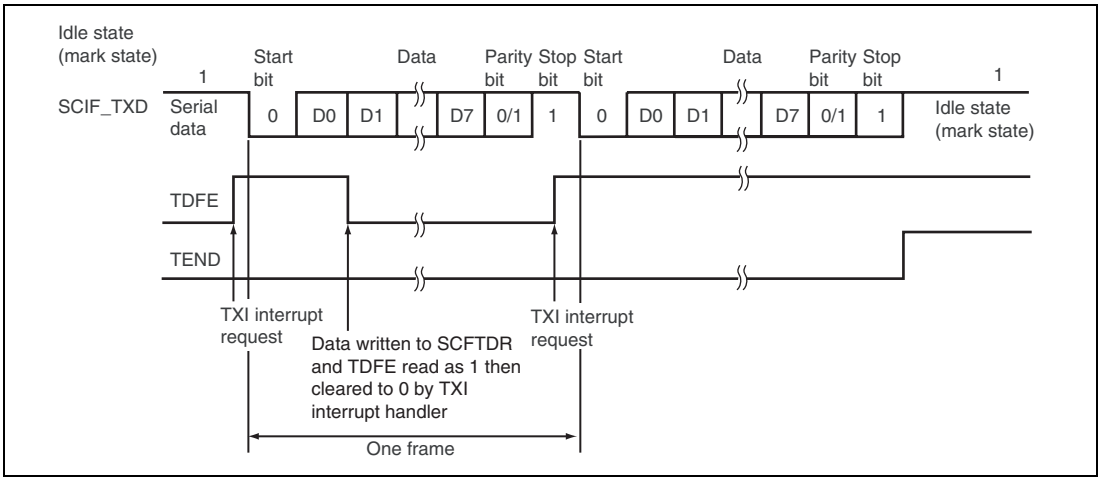


Figure 24.10 Example of SCIF Transmission Operation (Example with 8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the $\overline{\text{SCIF\_CTS}}$ input value. When $\overline{\text{SCIF\_CTS}}$ is set to 1 during transmission, the SCIF goes to the mark state after transmission of one frame. When $\overline{\text{SCIF\_CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

Figure 24.11 shows an example of the operation when modem control is used.

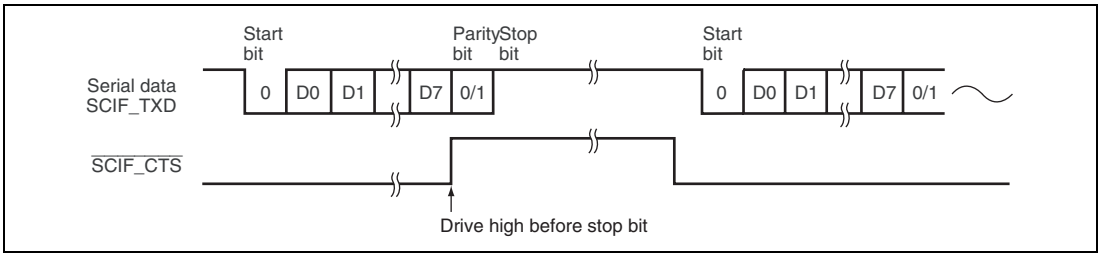


Figure 24.11 Example of Operation Using Modem Control ($\overline{\text{SCIF\_CTS}}$) (Only in Channel 0)

(5) Serial Data Reception (Asynchronous Mode)

Figure 24.12 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

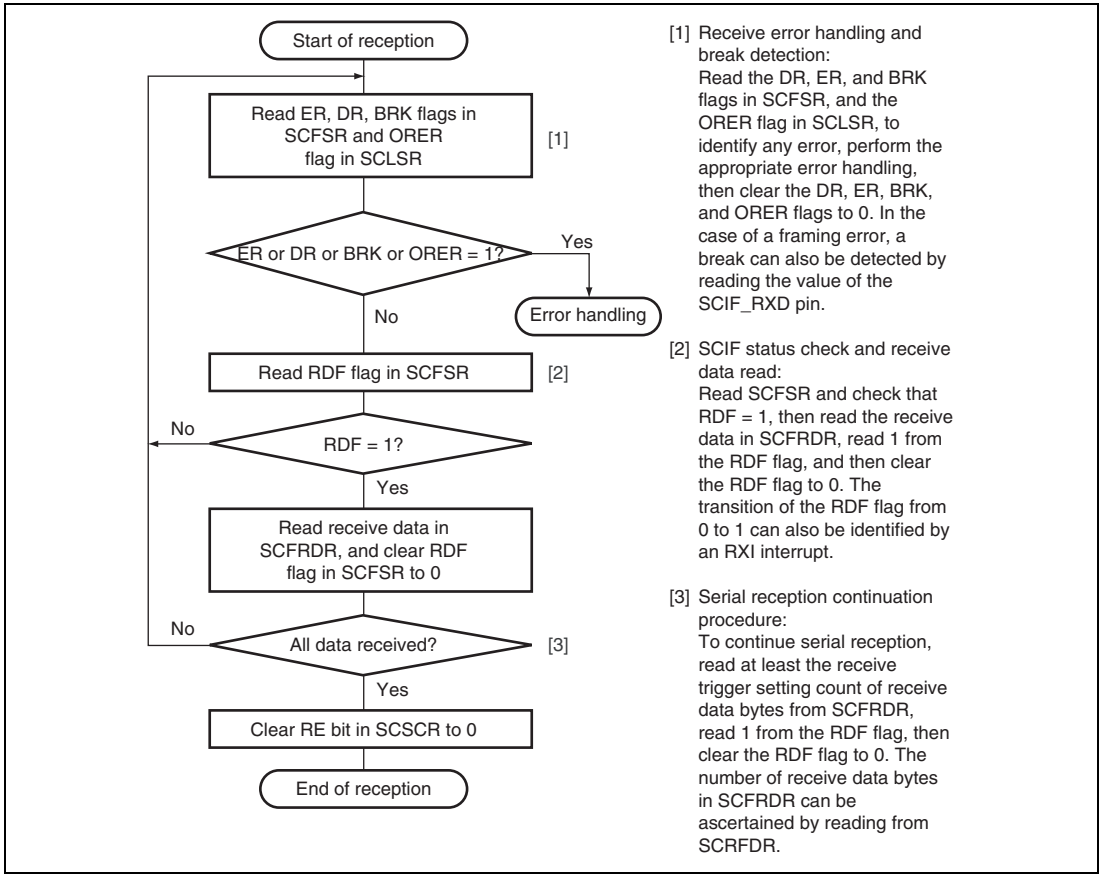
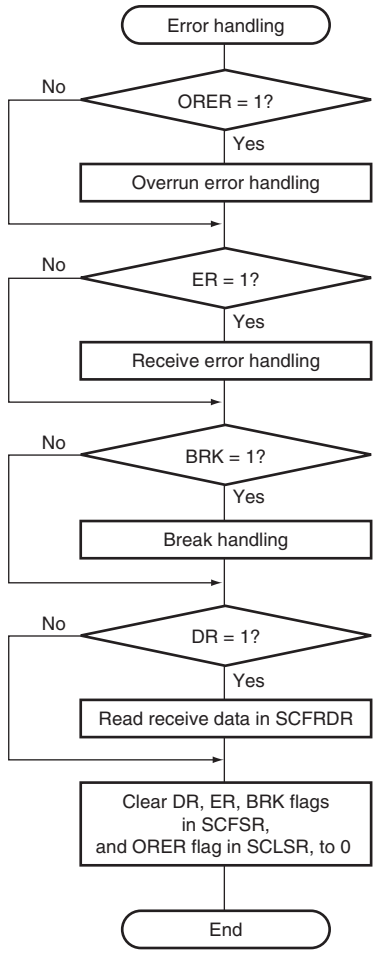


Figure 24.12 Sample Serial Reception Flowchart (1)



[1] Whether a framing error or parity error has occurred in the receive data that is to be read from SCFRDR can be ascertained from the FER and PER bits in SCFSR.

[2] When a break signal is received, receive data (H'00) is not transferred to SCFRDR.
However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.
When a break handling is completed and a receive signal returns to 1, the receive data transfer resumes.

Figure 24.12 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as follows.

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 SCRSR to SCFRDR.*

   (c) Overrun error check: The SCIF checks that the ORER flag is 0, indicating that no overrun error has occurred.*

   (d) Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.*

   If (b), (c), and (d) checks are passed, the receive data is stored in SCFRDR.

   Note: * Reception continues even when a parity error or framing error occurs.
4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 24.13 shows an example of the operation for reception in asynchronous mode.

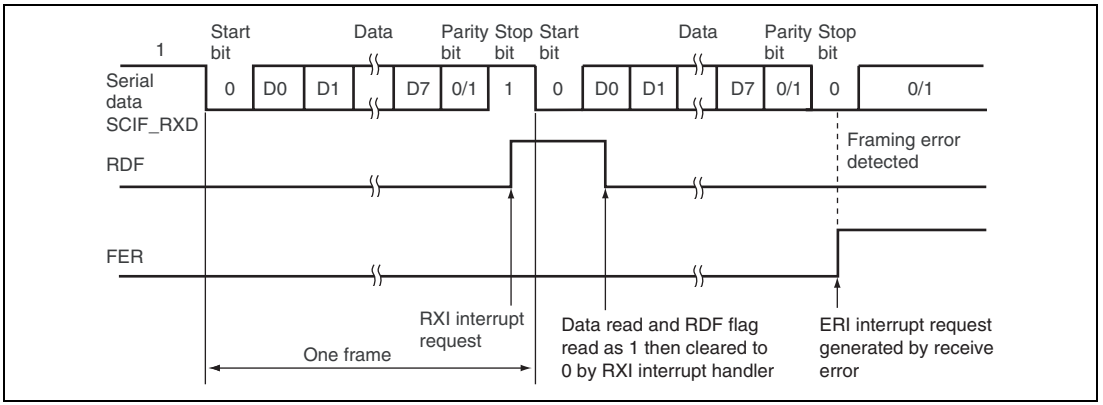


Figure 24.13 Example of SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{SCIF\_RTS}}$ signal is output when SCFRDR is empty. When $\overline{\text{SCIF\_RTS}}$ is 0, reception is possible. When $\overline{\text{SCIF\_RTS}}$ is 1, this indicates that SCFRDR contains bytes of data equal to or more than the $\overline{\text{SCIF\_RTS}}$ output active trigger count. The $\overline{\text{SCIF\_RTS}}$ output active trigger value is specified by bits 10 to 8 in SCFCR. For details, see section 24.3.9, FIFO control register n (SCFCR). In addition, $\overline{\text{SCIF\_RTS}}$ is also 1 when the RE bit in SCSCR is cleared to 0.

Figure 24.14 shows an example of the operation when modem control is used.

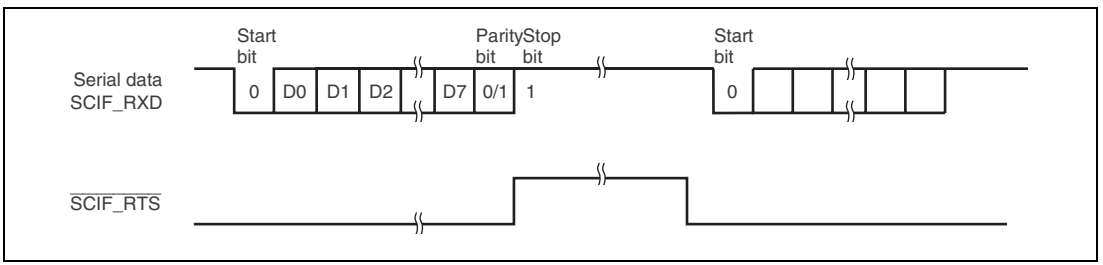


Figure 24.14 Example of Operation Using Modem Control ($\overline{\text{SCIF\_RTS}}$) (Only in Channels 1 and 2)

24.4.3 Operation in Clocked Synchronous Mode

Clocked synchronous mode, in which data is transmitted or received in synchronization with clock pulses, is suitable for fast serial communication.

Since the transmitter and receiver are independent units in the SCIF, full-duplex communication can be performed by sharing the clock. Both the transmitter and receiver have a 64-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transmission or reception.

Figure 24.15 shows the general format for clocked synchronous communication.

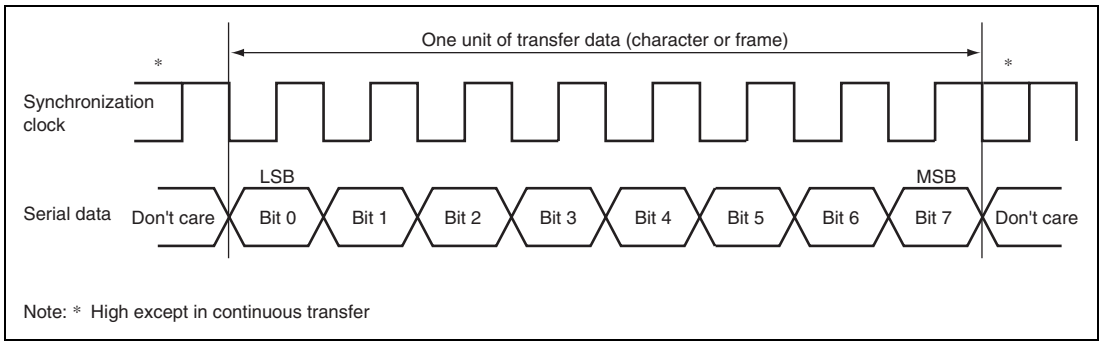


Figure 24.15 Data Format in Clocked Synchronous Communication

In clocked synchronous serial communication, data on the communication line is output from one fall of the synchronization clock to the next fall. Data is guaranteed to be accurate at the start of the synchronization 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 last data.

In clocked synchronous mode, the SCIF receives data in synchronization with the rise of the synchronization clock.

(1) Data Transfer Format

A fixed 8-bit data format is used. No parity bit can be added.

(2) Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCIF_SCK pin can be selected as the SCIF's serial clock,

according to the settings of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details on SCIF clock source selection, see table 24.5.

When the SCIF is operated on an internal clock, the synchronization clock is output from the SCIF_SCK pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed, the clock is fixed high. When an internal clock is selected in a receive operation only, as long as the RE bit in SCSCR is set to 1, clock pulses are output until the number of receive data bytes in the receive FIFO data register reaches the receive trigger count.

(3) SCIF Initialization (Clocked Synchronous Mode)

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 changing the operating mode or transfer format, etc., 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, SCTSR is initialized. Note that clearing the RE bit to 0 does not initialize the RDF, PER, FER, or ORER flag state or change the contents of SCFRDR.

Figure 24.16 shows a sample SCIF initialization flowchart.

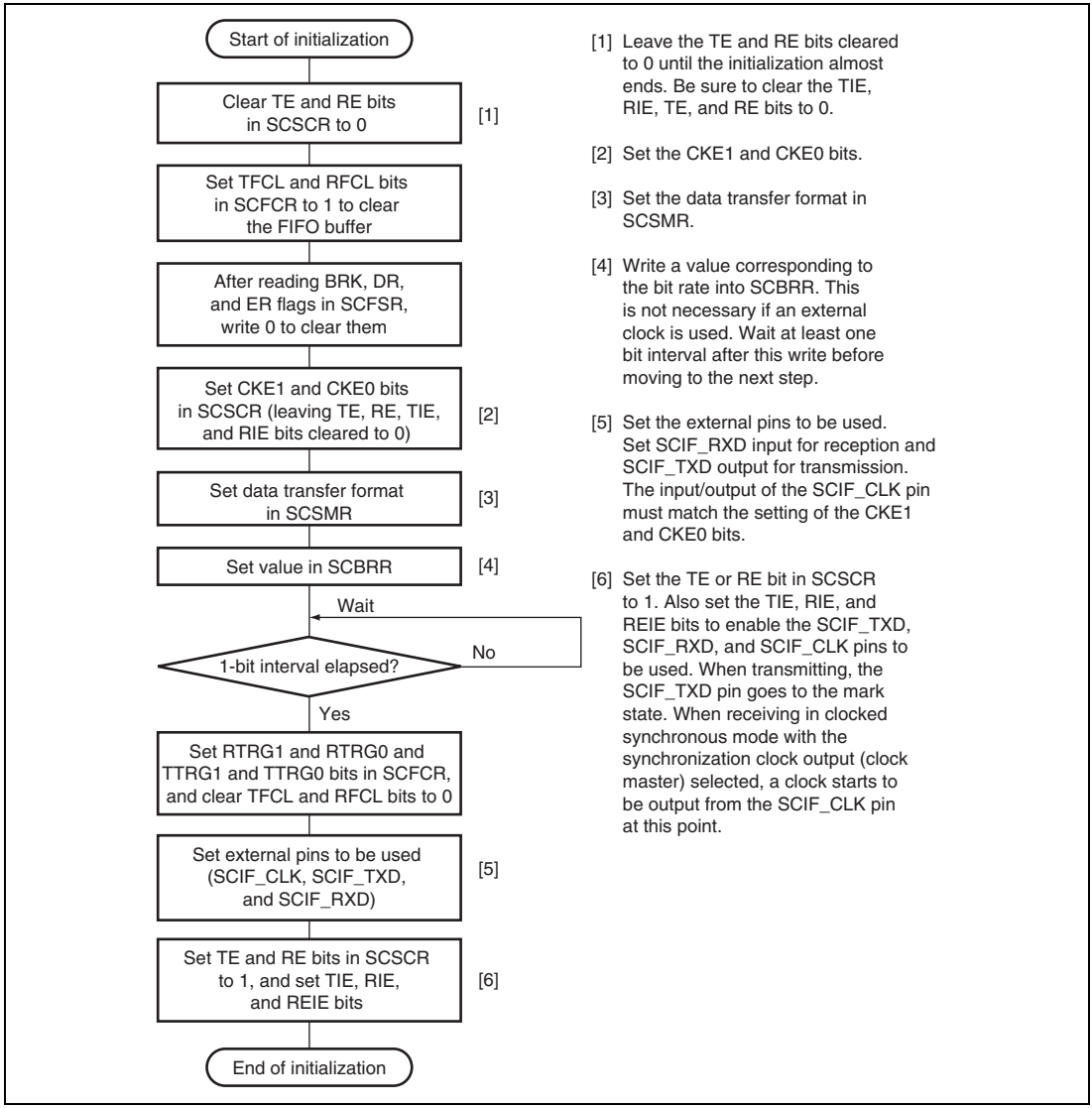
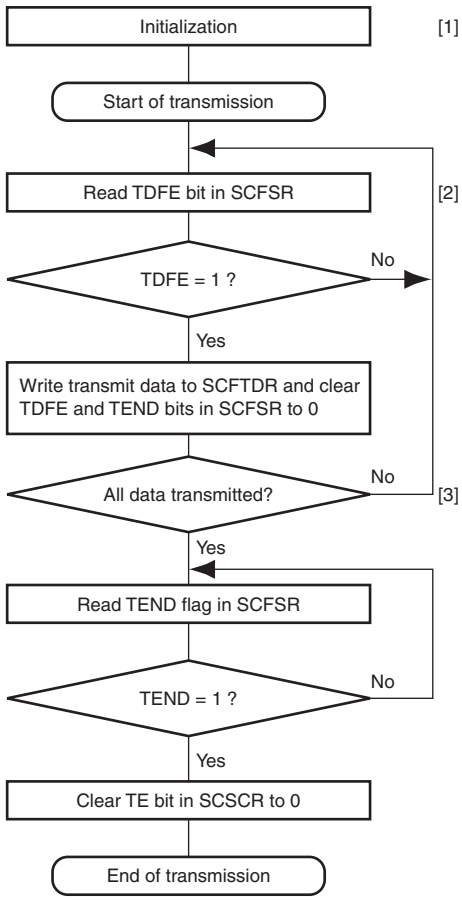


Figure 24.16 Sample SCIF Initialization Flowchart

(4) Serial Data Transmission (Clocked Synchronous Mode)

Figure 24.17 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.



[1] SCIF initialization:
See Sample SCIF Initialization Flowchart in figure 21.16.

[2] SCIF status check and transmit data write:
Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and clear the TDFE flag to 0. The transition of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE flag to 0.

Figure 24.17 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as follows.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least 64 (transmit trigger setting count).
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 to or below the transmit trigger count set in SCFCR, the TDFE flag is set. If the TIE bit in SCSCR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCIF outputs eight synchronization clock pulses for each data.

   When the external clock is selected, data is output in synchronization with the input clock.

   The serial transmit data is sent from the SCIF_TXD pin in LSB-first order.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the last bit. If data is present, the data is transferred from SCFTDR to SCTSR, and then serial transmission of the next frame is started. If there is no transmit data, the TEND flag in SCFSR is set to 1 after the last bit is sent, and the transmit data pin (SCIF_TXD pin) retains the output state of the last bit.
4. After serial transmission ends, the SCIF_SCK pin is fixed high when the CKE1 bit in SCSCR is 0.

Figure 24.18 shows an example of the SCIF transmission operation.

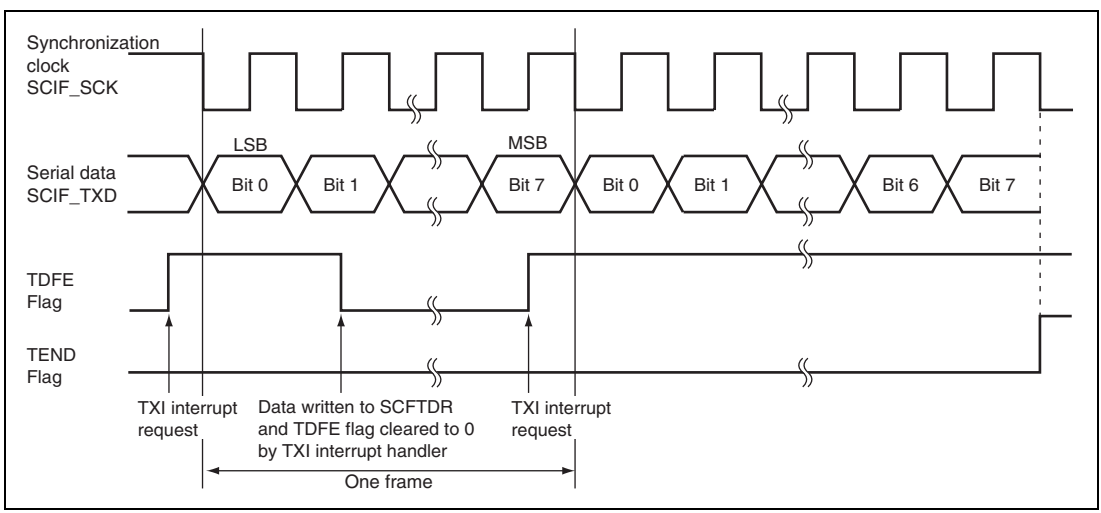


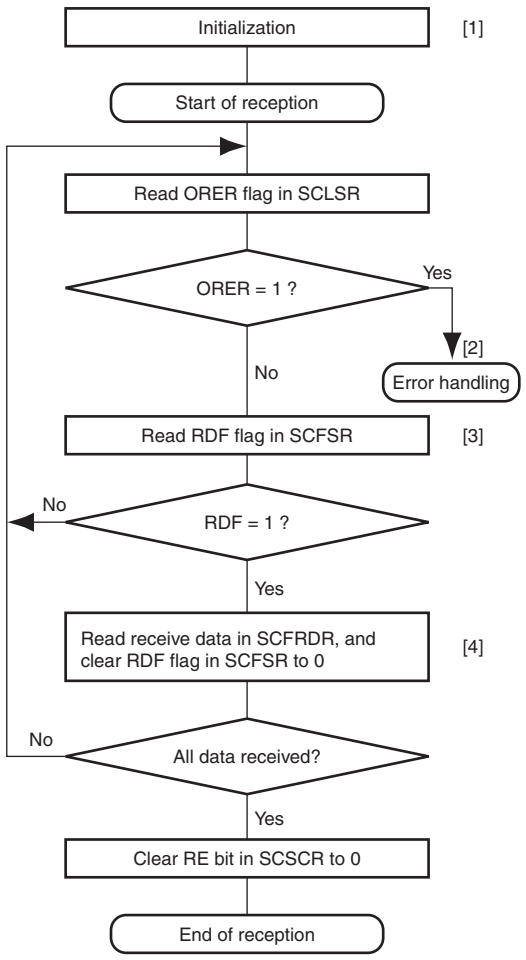
Figure 24.18 Example of SCIF Transmission Operation

(5) Serial Data Reception (Clocked Synchronous Mode)

Figure 24.19 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

When switching the operating mode from asynchronous mode to clocked synchronous mode without initializing the SCIF, make sure that the ORER, PER7 to PER0, and FER7 to FER0 flags are cleared to 0.



[1] SCIF initialization:
See Sample SCIF Initialization Flowchart in figure 21.16.

[2] Receive error handling:
Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0.
Transmission/reception cannot be resumed while the ORER flag is set to 1.

[3] SCIF status check and receive data read:
Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[4] Serial reception continuation procedure:
To continue serial reception, read at east the receive trigger setting count of receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading SCRFDR.

Figure 24.19 Sample Serial Reception Flowchart (1)

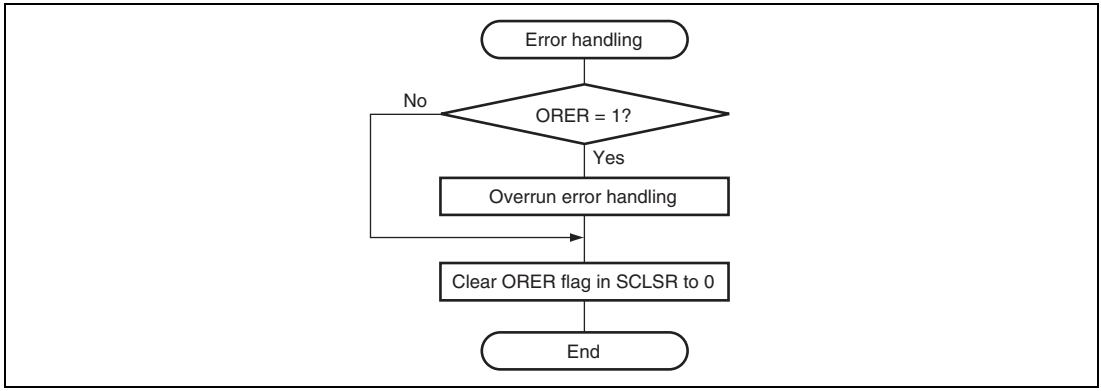


Figure 24.19 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as follows.

1. The SCIF is initialized internally in synchronization with the input or output of the synchronization clock.
2. The receive data is stored in SCRSR in LSB-to-MSB order.

   After receiving the data, the SCIF checks whether the receive data can be transferred from SCRSR to SCFRDR. If this check is passed, the receive data is stored in SCFRDR. If an overrun error is detected in the error check, reception cannot continue.
3. If the RIE bit in SCSCR is set to 1 when the RDF flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated.

   If the RIE bit in SCSCR is set to 1 when the ORER flag changes to 1, a break interrupt (BRI) request is generated.

Figure 24.20 shows an example of the SCIF reception operation.

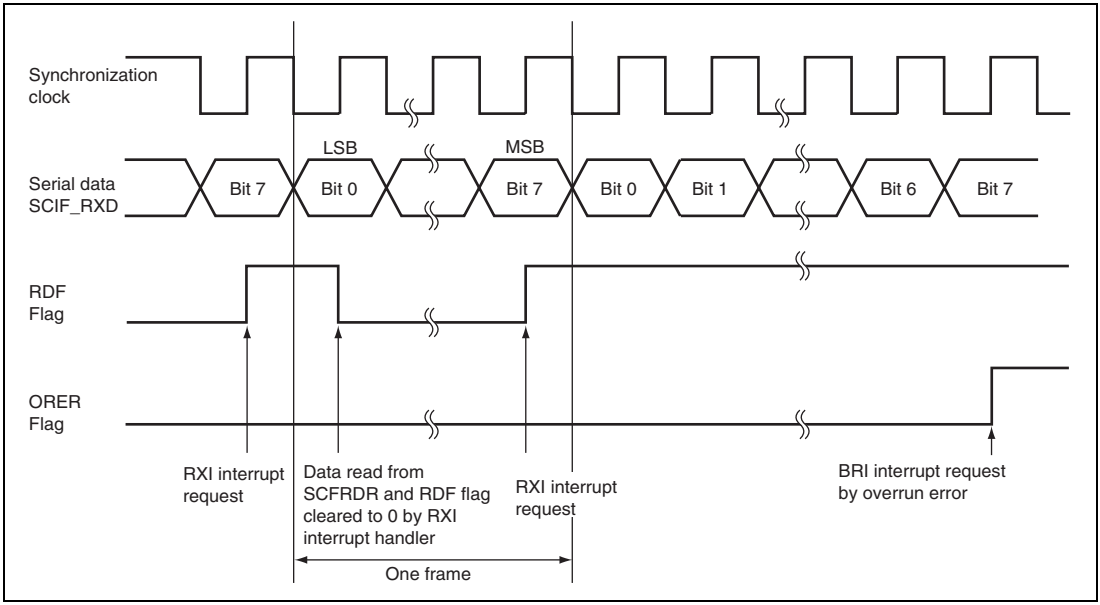
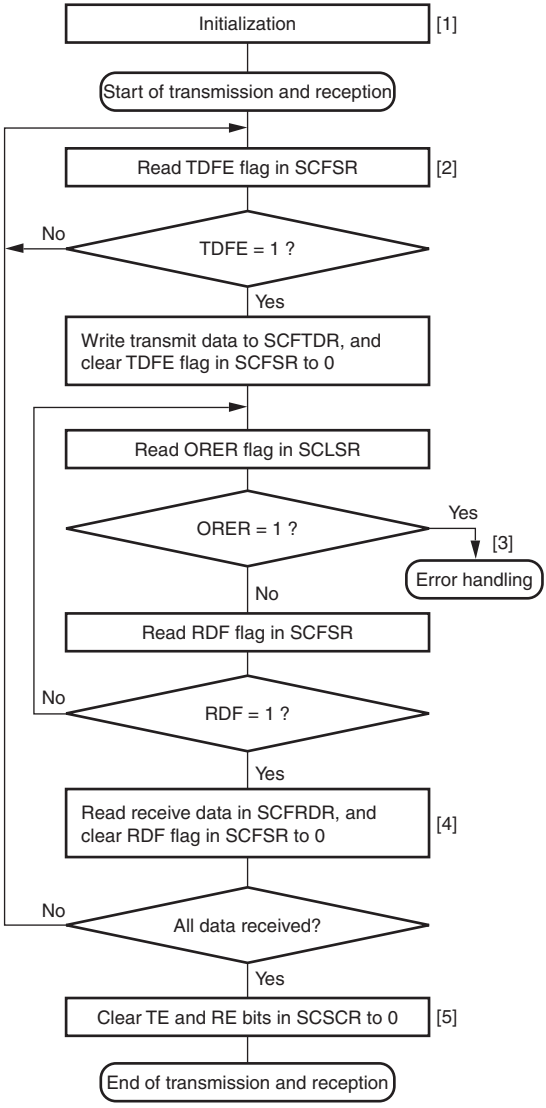


Figure 24.20 Example of SCIF Reception Operation

(6) Transmitting and Receiving Serial Data Simultaneously (Clocked Synchronous Mode)

Figure 24.21 shows a sample flowchart for transmitting and receiving data simultaneously.

Use the following procedure for the simultaneous serial transmission/reception of serial data, after enabling the SCIF transmission/reception.



[1] SCIF initialization:
See Sample SCIF Initialization Flowchart in figure 21.16.

[2] SCIF status check and transmit data write:
Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and clear the TDFE flag to 0. The transition of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error handling:
Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Reception cannot be resumed while the ORER flag is set to 1.

[4] SCIF status check and receive data read:
Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission and reception continuation procedure:
To continue serial transmission and reception, read RDF flag and SCFRDR, and clear the RDF flag to 0 before receiving the MSB in the current frame. Similarly, read 1 from the TDFE flag to confirm that writing is possible before transmitting the MSB in the current frame. Then write data to SCFTDR and clear the TDFE flag to 0.

Note: When switching from a transmit operation or receive operation to simultaneous transmission and reception operations, clear the TE and RE bits to 0, and then set them simultaneously to 1.

Figure 24.21 Sample Flowchart for Transmitting/Receiving Serial Data

24.5 SCIF Interrupt Sources and the DMAC

The SCIF has four interrupt sources for each channel: transmit-FIFO-data-empty interrupt (TXI) request, receive-error interrupt (ERI) request, receive-FIFO-data-full interrupt (RXI) request, and break interrupt (BRI) request.

Table 24.7 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, and REIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

If the TDFE flag in SCFSR is set to 1 when a TXI interrupt is enabled by the TIE bit, a TXI interrupt request and a transmit-FIFO-data-empty request for DMA transfer are generated. If the TDFE flag is set to 1 when a TXI interrupt is disabled by the TIE bit, only a transmit-FIFO-data-empty request for DMA transfer is generated. A transmit-FIFO-data-empty request can activate the DMAC to perform data transfer.

If the RDF or DR flag in SCFSR is set to 1 when an RXI interrupt is enabled by the RIE bit, an RXI interrupt request and a receive-FIFO-data-full request for DMA transfer are generated. If the RDF or DR flag is set to 1 when an RXI interrupt is disabled by the RIE bit, only a receive-FIFO-data-full request for DMA transfer is generated. A receive-FIFO-data-full request can activate the DMAC to perform data transfer. Note that generation of an RXI interrupt request or a receive-FIFO-data-full request by setting the DR flag to 1 occurs only in asynchronous mode.

When the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1, a BRI interrupt request is generated.

If transmission/reception is carried out using the DMAC, set and enable the DMAC before making the SCIF setting. Also make settings to inhibit output of RXI and TXI interrupt requests to the interrupt controller. If output of interrupt requests is enabled, these interrupt requests to the interrupt controller can be cleared by the DMAC regardless of the interrupt handler.

By setting the REIE bit to 1 while the RIE bit is cleared to 0 in SCSCR, it is possible to output ERI interrupt requests, but not RXI interrupt requests.

Table 24.7 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error flag (ER)	Not possible	High
RXI	Interrupt initiated by receive FIFO data full flag (RDF) or receive data ready flag (DR)*	Possible	
BRI	Interrupt initiated by break flag (BRK) or overrun error flag (ORER)	Not possible	
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	Low

Note: * An RXI interrupt by setting of the DR flag is available only in asynchronous mode.

24.6 Power-Down Mode

When using following Power-Down mode or changing clock frequency, it is necessary to stop operations of receiving or transmitting before using them.

(1) Sleep mode

The operation of the SCIF is continued.

(2) Light Sleep mode

The SCIF continues operation, but the operation of receiving/transmitting using the DMAC is not guaranteed. In such cases, following methods are needed before enter light sleep mode.

i. Check all reading/writing operations are finished.

Finish of all transfer operations is confirmed by checking the TEND flag of the SCFSR register that indicates all data transmitting are done is set, or checking all receiving data are read. In addition, clear the TE and RE bit in the SCSCR register to 0 to prevent beginning the SCIF serial transfer operation by read from the SCFRDR (receive FIFO data register) or write to the SCFTDR (transmit FIFO data register).

ii. Disable all DMA requests from the SCIF.

(3) Module standby mode

In this mode, clock supplying to the SCIF is stop. Following methods are needed before enter module standby mode.

i. Check all reading/writing operations are finished.

The procedure is same as the case of light sleep mode.

ii. Disable all DMA requests from the SCIF.

iii. Set the corresponding bit in the MSTPCR (standby control register) to 1.

(4) Changing clock frequency

If the operating frequency of the SCIF is changed while the SCIF is operating, correct operation is not guaranteed. Before changing the operating frequency of the SCIF while transmitting or receiving, following procedure is needed.

i. Check all reading/writing operations are finished.

The processing is same as the case of deep sleep mode.

ii. Disable all DMA and interrupt requests from the SCIF.

24.7 Usage Notes

Note the following when using the SCIF.

(1) SCFTDR Writing and the TDFE Flag

The TDFE flag in SCFSR is set when the number of transmit data bytes written in SCFTDR has fallen to or below the transmit trigger count set by bits TTRG1 and TTRG0 in 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 equal to or less than the transmit trigger count, the TDFE flag will be set to 1 again, even after being read as 1 and cleared to 0. Therefore, the TDFE flag should be cleared when SCFTDR contains more than the transmit trigger count of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from SCTFDR.

(2) SCFRDR Reading and the RDF Flag

The RDF flag in SCFSR is set when the number of receive data bytes in SCFRDR has become equal to or greater than the receive trigger count set by bits RTRG1 and RTRG0 in SCFCR. After RDF is set, receive data equivalent to the trigger count can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes read in SCFRDR is equal to or greater than the trigger count, the RDF flag will be set to 1 again even if it is cleared to 0. After the receive data is read, clear the RDF flag readout to 0 in order to reduce the number of data bytes in SCFRDR to less than the trigger count.

The number of receive data bytes in SCFRDR can be found from SCRFDR.

(3) Break Detection and Processing

If a framing error (FER) is detected, break signals can also be detected by reading the SCIF_RXD pin value directly. In the break state the input values from the SCIF_RXD consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set.

Although the SCIF stops transferring receive data to SCFRDR after receiving a break, the receive operation continues.

(4) Sending a Break Signal

The input/output condition and level of the SCIF_TXD pin are determined by bits SPB2IO and SPB2DT in SCSPTR. This feature can be used to send a break signal.

After the serial transmitter is initialized and until the TE bit is set to 1 (enabling transmission), the SCIF_TXD pin function is not selected and the value of the SPB2DT bit substitutes for the mark state. The SPB2IO and SPB2DT bits should therefore be set to 1 (designating output and high level) in the beginning.

To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating low level), and then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized, regardless of the current transmission state, and 0 is output from the SCIF_TXD pin.

(5) Receive Data Sampling Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCIF operates on a base clock with frequency of 16 times the bit rate.

In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse.

Figure 24.22 shows the timing.

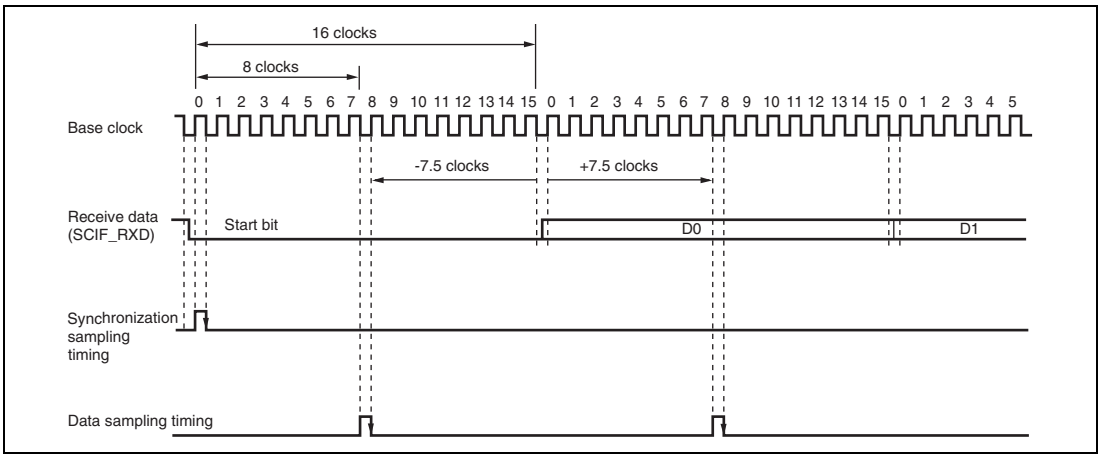


Figure 24.22 Receive Data Sampling Timing in Asynchronous Mode

Thus, the reception margin in asynchronous mode is given by formula (1).

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{| D - 0.5 |}{N} (1 + F) \right| \times 100\ \% \quad \ldots\ldots\ldots\ldots\ (1)$$

M: Receive margin (%)

N: Ratio of bit rate to clock (N = 16)

D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute value of clock rate deviation

From equation (1), if F = 0 and D = 0.5, the reception margin is 46.875%, as given by formula (2).

When D = 0.5 and F = 0:

$$M = (0.5 - 1 / (2 \times 16)) \times 100\% = 46.875\% \quad \ldots\ldots\ldots\ldots\ (2)$$

However, this is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Section 25 Serial Peripheral Interface (HSPI)

This LSI incorporates one channel of the Serial Protocol Interface (HSPI).

25.1 Features

The HSPI has the following features.

- Operating mode: Master mode or Slave mode.
- The transmit and receive sections within the module are double buffered to allow duplex communication.
- A flexible peripheral clock (Pck) division strategy allows a wide range of bit rates to be supported.
- The programmable clock control logic allows setting for two different transmit protocols and accommodates transmit and receive functions on either edge of the serial clock.
- Error detection logic is provided for warning of the receive buffer overflow.
- The HSPI has a facility to generate the chip select signal to slave modules when configured as a master either automatically as part of the data transfer process, or under the manual control of the host processor.
- Both the transmit data and receive data can be DMA transferred independently via the two DMA channels.

Figure 25.1 is a block diagram of the HSPI.

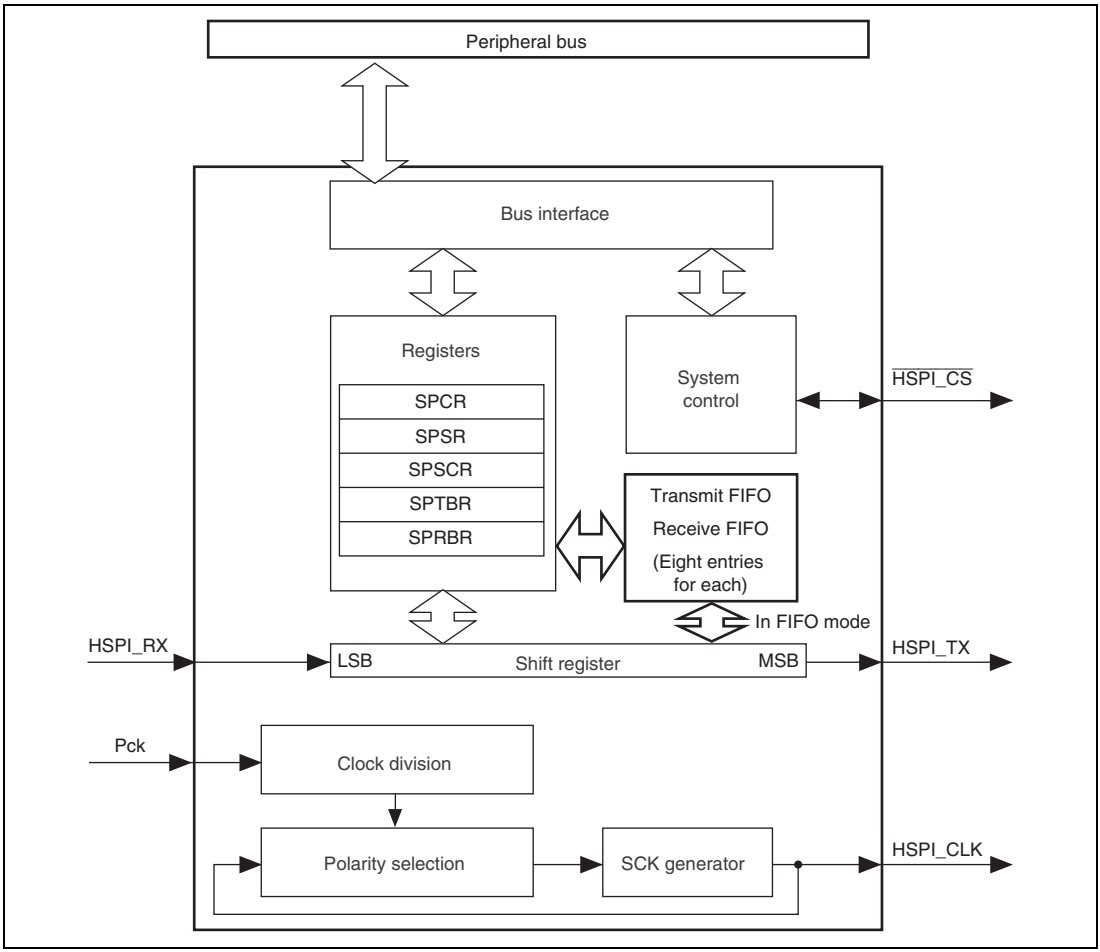


Figure 25.1 Block Diagram of HSPI

25.2 Input/Output Pins

The input/output pins of the HSPI are shown in table 25.1.

Table 25.1 Pin Configuration

Pin Name	Abbrev.	I/O	Description
Serial bit clock pin	HSPI_CLK	I/O	Clock input/output
Transmit data pin	HSPI_TX	Output	Transmit data output
Receive data pin	HSPI_RX	Input	Receive data input
Chip select pin	HSPI_CS	I/O	Chip select

25.3 Register Descriptions

Table 25.2 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Control register	SPCR	R/W	H'FFF4 0000	H'1FF4 0000	32	Pck
Status register	SPSR	R*	H'FFF4 0004	H'1FF4 0004	32	Pck
System control register	SPSCR	R/W	H'FFF4 0008	H'1FF4 0008	32	Pck
Transmit buffer register	SPTBR	R/W	H'FFF4 000C	H'1FF4 000C	32	Pck
Receive buffer register	SPRBR	R	H'FFF4 0010	H'1FF4 0010	32	Pck

Note: For bits 4 and 3, only 0s can be written to clear the flags.

Table 25.3 Register Configuration (2)

Register Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiple Exception	Sleep/Light Sleep by SLEEP Instruction	Module Standby	Software Reset
Control register	SPCR	H'0000 0000	Retained	Retained	Retained	Retained
Status register	SPSR	H'xxxx x120*[1]	Retained	Retained	Retained	b'xxxx x1xx*[2]
System control register	SPSCR	H'0000 0040	Retained	Retained	Retained	Retained
Transmit buffer register	SPTBR	H'0000 0000	Retained	Retained	Retained	Retained
Receive buffer register	SPRBR	H'0000 0000	Retained	Retained	Retained	Retained

Notes: 1. "x" represents an undefined value.
2. "x" represents an undefined value. Bits 9, 6, 4, and 3 are retained. The other bits are initialized except those of which the initial values are undefined.

25.3.1 Control Register (SPCR)

SPCR is a 32-bit readable/writable register that controls the transfer data of shift timing and specifies the clock polarity and frequency.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FBS	CLKP	IDIV	CLKC4	CLKC3	CLKC2	CLKC1	CLKC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
7	FBS	0	R/W	First Bit Start Controls the timing relationship between each bit of transferred data and the serial clock. 0: The first bit transmitted from the HSPI module is set up such that it can be sampled by the receiving device at the first edge of HSPI_CLK specified by the register after the $\overline{\text{HSPI\_CS}}$ pin goes low. Similarly the first received bit is sampled at the first edge of HSPI_CLK after the $\overline{\text{HSPI\_CS}}$ pin goes low. 1: The first bit transmitted from the HSPI module is set up such that it can be sampled by the receiving device at the second edge of HSPI_CLK after the $\overline{\text{HSPI\_CS}}$ pin goes low. Similarly the first received bit is sampled at the second edge of HSPI_CLK specified by the register after the $\overline{\text{HSPI\_CS}}$ pin goes low.
6	CLKP	0	R/W	Serial Clock Polarity 0: HSPI_CLK signal is not inverted and so is low when inactive. 1: HSPI_CLK signal is inverted and so is high when inactive.

Bit	Bit Name	Initial Value	R/W	Description
5	IDIV	0	R/W	Initial Clock Division Ratio 0: The peripheral clock (Pck) is divided by a factor of 4 initially to create an intermediate frequency, which is further divided to create the serial clock for master mode. 1: The peripheral clock (Pck) is divided by a factor of 32 initially to create an intermediate frequency, which is further divided to create the serial clock for master mode.
4 to 0	CLKC4 to CLKC0	All 0	R/W	Clock Division Count These bits determine the frequency dividing ratio that is used to obtain the serial clock from the intermediate clock. 00000: 1 intermediate frequency cycle. Serial clock frequency = Intermediate frequency/2. 00001: 2 Intermediate frequency cycles. Serial clock frequency = Intermediate frequency/4. 00010: 3 intermediate frequency cycles. Serial clock frequency = Intermediate frequency/6. 11111: 32 intermediate frequency cycles. Serial clock frequency = Intermediate frequency/64.

The serial clock frequency can be computed using the following formula:

$$\text{Serial clock frequency} = \frac{\text{Peripheral clock frequency}}{(\text{Initial division ratio} \times (\text{Clock division count} + 1) \times 2)}$$

When the HSPI is configured as a slave, the IDIV and CLKC bits are ignored and the HSPI synchronizes to the externally supplied serial clock. The maximum value of the external serial clock that the module can operate with is Pck/8.

If any of the FBS, CLKP, IDIV or CLKC bit values are changed, the HSPI will undergo a software reset.

When IPIV or CLKC is specified or changed, the internal serial clock generation counter is reset. In this case, data transmit/receive should be performed after the time length of at least one serial clock cycle (depends on IDIV or CLKC specified) has elapsed.

25.3.2 Status Register (SPSR)

SPSR is a 32-bit readable/writable register. Through the status flags in SPSR, it can be confirmed whether or not the system is correctly operating. If the ROIE bit in SPSCR is set to 1, an interrupt request is generated due to the occurrence of the receive buffer overrun error or the warning of the receive buffer overrun error. When the TFIE bit in SPSCR is set to 1, an interrupt request is generated by the transmit complete status flag. If the appropriate enable bit in SPSCR is set to 1, an interrupt request is generated due to the receive FIFO halfway, receive FIFO full, transmit FIFO empty, or transmit FIFO halfway flag. If the RNIE bit in SPSCR is set to 1, an interrupt request is generated when the receive FIFO is not empty.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TXFU	TXHA	TXEM	RXFU	RXHA	RXEM	RXOO	RXOW	RXFL	TXFN	TXFL
Initial value:	—	—	—	—	—	0	0	1	0	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W*	R/W*	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	Undefined	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
10	TXFU	0	R	Transmit FIFO Full Flag This status flag is enabled only in FIFO mode. The flag is set to 1 when the transmit FIFO is full of bytes for transmission and cannot accept any more. It is cleared to 0 when data is transmitted from the transmit FIFO to the HSPI bus.
9	TXHA	0	R	Transmit FIFO Halfway Flag This status flag is enabled only in FIFO mode. The flag is set to 1 when the transmit FIFO reaches the halfway point, that is, it has four bytes of data and free space for four bytes of data. It is cleared to 0 when more data is written to the transmit FIFO. It remains set to 1 until cleared to 0 even if data stored in the FIFO becomes less than four bytes (halfway point). If TXHA = 1 and THIE = 1, an interrupt is generated.

Bit	Bit Name	Initial Value	R/W	Description
8	TXEM	1	R	Transmit FIFO Empty Flag This status flag is enabled only in FIFO mode. The flag is set to 1 when the transmit FIFO is empty of data to transmit. It is cleared to 0 when data is written to the transmit FIFO. If TXEM = 1 and TEIE = 1, an interrupt is generated.
7	RXFU	0	R	Receive FIFO Full Flag This status flag is enabled only in FIFO mode. The flag is set to 1 when the receive FIFO is full of received bytes and cannot accept any more. It is cleared to 0 when data is read out of the receive FIFO. If RXFU = 1 and RFIE = 1, an interrupt is generated.
6	RXHA	0	R	Receive FIFO Halfway Flag This status flag is enabled only in FIFO mode. The flag is set to 1 when the receive FIFO reaches the halfway point, that is, it has four bytes of data and free space for four bytes of data. This flag is cleared to 0 when the receive data is read from receive FIFO. If RXHA = 1 and RHIE = 1, an interrupt is generated.
5	RXEM	1	R	Receive FIFO Empty Flag This status flag is enabled only in FIFO mode. The flag is set to 1 when the receive FIFO is empty of received data. It is cleared to 0 when data is written to the receive FIFO. If RXEM = 0 and RNIE = 1, an interrupt is generated.
4	RXOO	0	R/W*	Receive Buffer Overrun Occurred Flag This status flag is set to 1 when new data has been received but the previous received data has not been read from SPRBR. The previously received data will not be overwritten by the newly received data. The RXOO flag remains set to 1 until writing 0 to this bit position. If RXOO = 1 and ROIE = 1, an interrupt is generated.

Bit	Bit Name	Initial Value	R/W	Description
3	RXOW	0	R/W*	Receive Buffer Overrun Warning Flag This status flag is set to 1 when a new serial data transfer has started before the previous received data is read from SPRBR. The RXOW remains set to 1 until writing 0 to this bit position. If RXOW= 1 and ROIE = 1, an interrupt is generated.
2	RXFL	0	R	Receive Buffer Full Status Flag This status flag indicates that new data is available in the SPRBR and has not yet been read. It is set to 1 when the shift register contents are loaded into the SPRBR in the end of a serial bus transfer. This bit is cleared to 0 by reading SPRBR.
1	TXFN	0	R	Transmit Complete Status Flag This status flag indicates that the last transmission has been completed. It is set to 1 when SPTBR is ready to accept data from the peripheral bus. This bit is cleared to 0 by writing data to SPTBR. If TXFN = 1 and TFIE = 1, an interrupt is generated.
0	TXFL	0	R	Transmit Buffer Full Status Flag This status flag indicates the SPTBR stores data that has not been transmitted. It is set to 1 when SPTBR is written with data from the peripheral bus. This bit is cleared to 0 when SPTBR is ready to accept data from the peripheral bus.

Note: * These bits are readable/writable bits. Writing 0 initializes these bits to the initial values of respective bits, while writing 1 is ignored.

25.3.3 System Control Register (SPSCR)

SPSCR is a 32-bit readable/writable register that enables or disables interrupts or FIFO mode, selects either LSB first or MSB first in transmitting/receiving data, and master or slave mode.

If any of the FFEN, LMSB, CSA, or MASL bit values are changed, the module will undergo a software reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TEIE	THIE	RNIE	RHIE	RFIE	FFEN	LMSB	CSV	CSA	TFIE	ROIE	RXDE	TXDE	MASL
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
13	TEIE	0	R/W	Transmit FIFO Empty Interrupt Enable 0: Transmit FIFO empty interrupt disabled 1: Transmit FIFO empty interrupt enabled
12	THIE	0	R/W	Transmit FIFO Halfway Interrupt Enable 0: Transmit FIFO halfway interrupt disabled 1: Transmit FIFO halfway interrupt enabled
11	RNIE	0	R/W	Receive FIFO Not Empty Interrupt Enable 0: Receive FIFO not empty interrupt disabled 1: Receive FIFO not empty interrupt enabled
10	RHIE	0	R/W	Receive FIFO Halfway Interrupt Enable 0: Receive FIFO halfway interrupt disabled 1: Receive FIFO halfway interrupt enabled
9	RFIE	0	R/W	Receive FIFO Full Interrupt Enable 0: Receive FIFO full interrupt disabled 1: Receive FIFO full interrupt enabled

Bit	Bit Name	Initial Value	R/W	Description
8	FFEN	0	R/W	FIFO Mode Enable Enables or disables the FIFO mode. When FIFO mode is enabled, two 8-entry FIFOs are made available, one for transmit data and one for receive data. These FIFOs are read and written via SPTBR and SPRBR, respectively. When FIFO mode is disabled, the SPTBR and SPRBR are used directly so new data must be written to SPTBR and read from SPRBR for each and every transfer through the HSPI bus. FIFO mode must be disabled if DMA requests are also to be used to service SPTBR and SPRBR. 0: FIFO mode disabled 1: FIFO mode enabled
7	LMSB	0	R/W	LSB/MSB First Control 0: Data is transmitted and received most significant bit (MSB) first. 1: Data is transmitted and received least significant bit (LSB) first.
6	CSV	1	R/W	Chip Select Value Controls the value output as the chip select signal when the HSPI is a master and manual generation of the chip select signal has been selected. 0: Chip select output is low. 1: Chip select output is high.
5	CSA	0	R/W	Automatic/Manual Chip Select 0: Chip select output is automatically generated during data transfer. 1: Chip select output is manually controlled, with its value being determined by the CSV bit.
4	TFIE	0	R/W	Transmit Complete Interrupt Enable 0: Transmit complete interrupt disabled 1: Transmit complete interrupt enabled
3	ROIE	0	R/W	Receive Overrun Occurred/Warning Interrupt Enable 0: Receive overrun occurred/warning interrupt disabled 1: Receive overrun occurred/warning interrupt enabled

Bit	Bit Name	Initial Value	R/W	Description
2	RXDE	0	R/W	Receive DMA Enable 0: Receive DMA transfer request disabled 1: Receive DMA transfer request enabled
1	TXDE	0	R/W	Transmit DMA Enable 0: Transmit DMA transfer request disabled 1: Transmit DMA transfer request enabled
0	MASL	0	R/W	Master/Slave Select Bit 0: HSPI module configured as a slave 1: HSPI module configured as a master

25.3.4 Transmit Buffer Register (SPTBR)

SPTBR is a 32-bit readable/writable register that stores data to be transmitted.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TD							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
7 to 0	TD	All 0	R/W	Transmit Data Data written to this register is transferred to the shift register for transmission. When reading these bits, the data stored in the transmit buffer is always read.

25.3.5 Receive Buffer Register (SPRBR)

SPRBR is a 32-bit read-only register that stores received data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RD							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
7 to 0	RD	All 0	R	Receive Data Data is transferred from the shift register to these bits every time one byte of data is received if the previously received data has been read.

25.4 Operation

25.4.1 Operation Overview with FIFO Mode Disabled

Figure 25.2 shows the flow of a transmit/receive operation procedure.

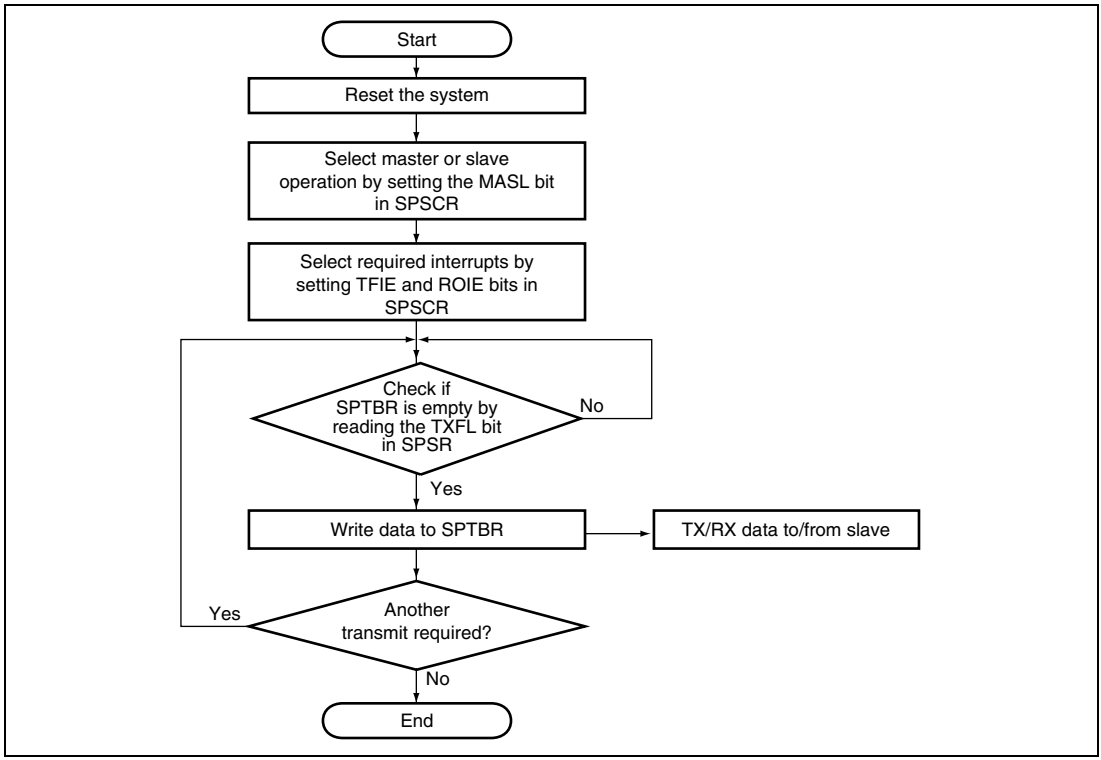


Figure 25.2 Operational Flowchart

Depending on the settings of SPCR, the master transmits data to the slave on either the falling or rising edge of HSPI_CLK and samples data from the slave at the opposite edge. The data transfer between the master and slave is completed when the transmit complete status flag (TXFN) in SPSR is set to 1. This flag should be used to identify when an HSPI transfer event (byte transmitted and byte received) has occurred, even in the case where the HSPI module is used to receive data only (null data being transmitted). By default data is transmitted MSB first, but LSB first is also possible depending on how the LMSB bit in SPSCR is set.

During the transmit operation the slave responds by sending data to the master synchronized with the HSPI_CLK from the master transmitted. Data from the slave is sampled and transferred to the shift register in the module and on completion of the transmit operation, is transferred to SPRBR.

The $\overline{\text{HSPI\_CS}}$ pin should be used to select the HSPI module and prepare it to receive data from an external master when the HSPI is configured as a slave. When the FBS bit in SPCR is 0, the $\overline{\text{HSPI\_CS}}$ pin must be driven high between successive bytes (the $\overline{\text{HSPI\_CS}}$ pin must be driven high after a byte transfer). When FBS = 1, the $\overline{\text{HSPI\_CS}}$ pin can stay low for several byte transmissions. In this case, if the system is configured such that FBS is always 1, the $\overline{\text{HSPI\_CS}}$ line can be fixed at ground (if the HSPI will only be used as a slave).

25.4.2 Operation with FIFO Mode Enabled

In order to reduce the interrupt overhead on the CPU, FIFO mode has been provided. When FIFO mode is enabled, up to eight bytes can be written in advance for transmission and up to eight bytes can be received before the receive FIFO needs to be read. To transfer the specified amount of data between the HSPI module and an external device, use the following procedure:

1. Set up the module for the required HSPI transfer characteristics (master/slave, clock polarity etc.) and enable FIFO mode.
2. Write bytes into the transmit FIFO via SPTBR. If more than eight bytes are to be transmitted, enable the transmit FIFO halfway interrupt to keep track of the FIFO level as data is transmitted.
3. Respond to the transmit FIFO halfway interrupt when it occurs by writing more data to the transmit FIFO and reading data from the receive FIFO via SPRBR.
4. When all of the transmit data has been written into the transmit FIFO, disable the transmit FIFO halfway interrupt and read the contents of the receive FIFO until it is empty. Enable the receive FIFO not empty interrupt to keep track of when the final bytes of the transfer are received.
5. Respond to the receive FIFO not empty interrupt until all the expected data has been received.
6. Disable the module until it is required again.

In some applications, an undefined amount of data will be received from an external HSPI device. If this is the case, use the following procedure:

1. Set up the module for the required HSPI transfer characteristics (master/slave, clock polarity, etc.) and enable FIFO mode.
2. Fill the transmit FIFO with the data to transmit. Enable the receive FIFO not empty interrupt.
3. Respond to the receive FIFO not empty interrupt and read data from the receive FIFO until it is empty. Write more data to the transmit FIFO if required.
4. Disable the module when the transfer is to stop.

25.4.3 Timing Diagrams

The following diagrams explain the timing relationship of all shift and sample processes in the HSPI. Figure 25.3 shows the conditions when FBS = 0, figure 25.4 shows the conditions when FBS = 0 (continuous transfer), figure 25.5 shows the conditions when FBS = 1, and figure 25.6 shows the conditions when FBS = 1 (continuous transfer). It can be seen that if CLKP in SPCR is 0, transmit data is shifted at the falling edge of HSPI_CLK and receive data is sampled at the rising edge of HSPI_CLK. The opposite is true when CLKP = 1.

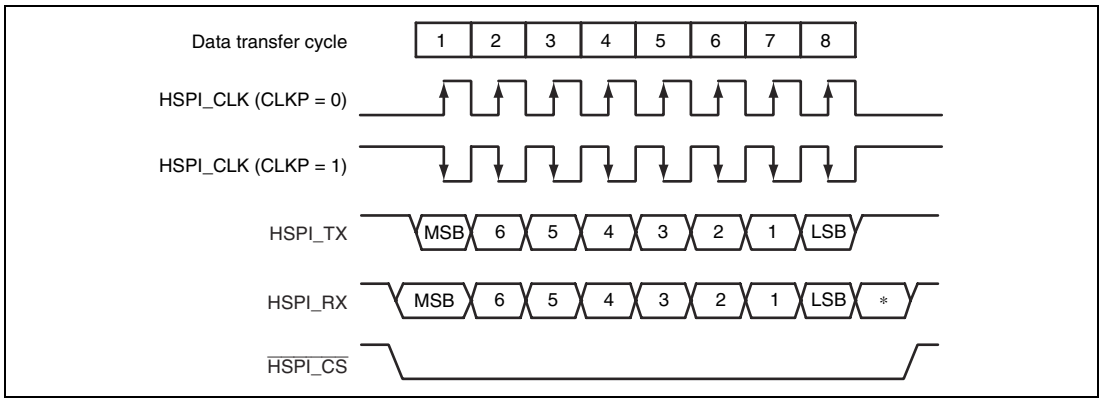


Figure 25.3 Timing Conditions when FBS = 0

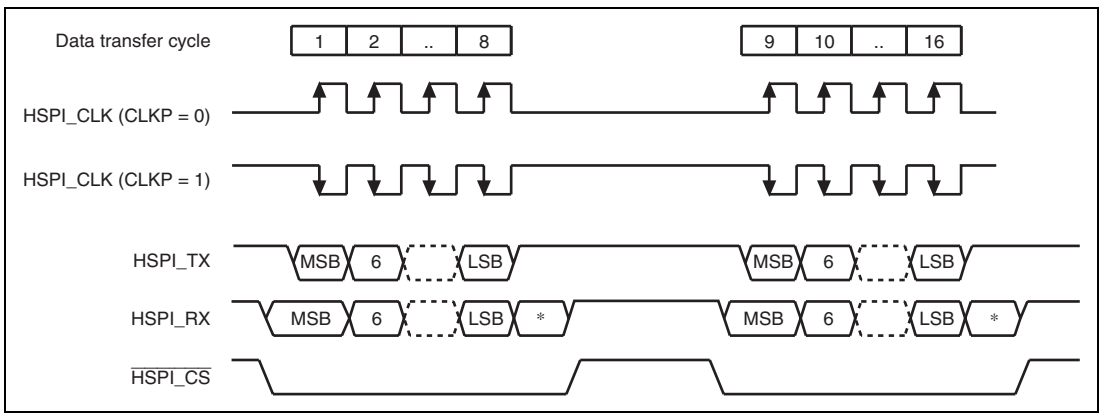


Figure 25.4 Timing Conditions when FBS = 0 (Continuous Transfer)

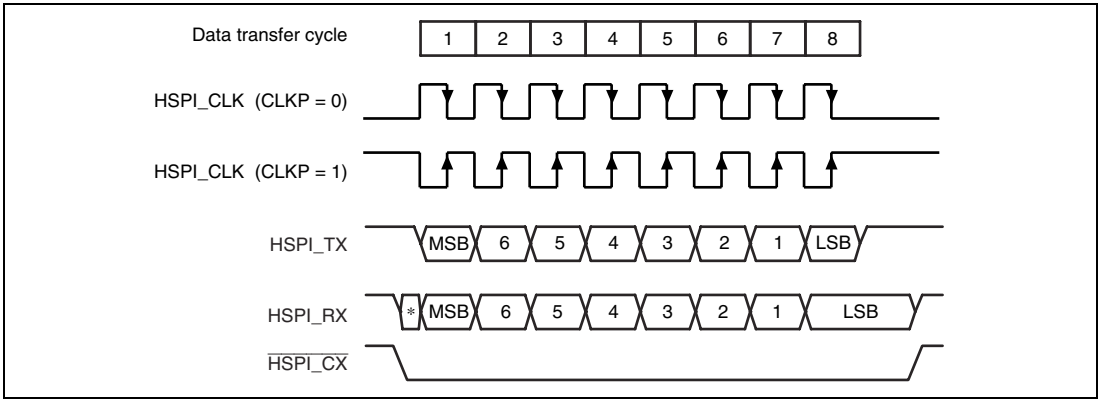


Figure 25.5 Timing Conditions when FBS = 1

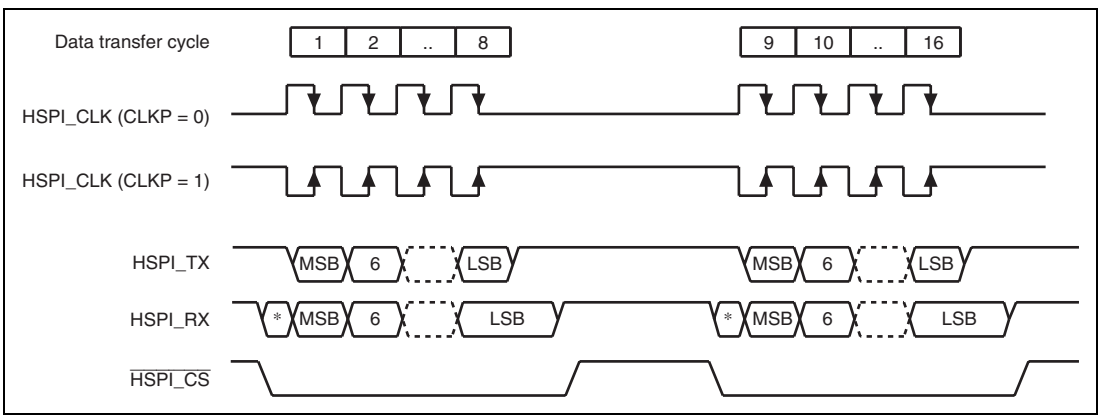


Figure 25.6 Timing Conditions when FBS = 1 (Continuous Transfer)

The asterisk (*) in the figures shows 0 or 1.

25.4.4 HSPI Software Reset

The software reset makes modules predefined value and initializes receiving/transmitting FIFO pointers. If any of the control bits, except for SPCR and the interrupt and chip select value bits of SPSCR, are changed, then the HSPI software reset is generated. Each interrupt enable bit should be cleared before executing the HSPI software reset. The data transmission after the HSPI software reset should conform to transmitting and receiving protocol of HSPI and be performed from the beginning; otherwise, correct operation is not guaranteed.

When asserting the $\overline{\text{HSPI\_CS}}$ except when the master device is transferring data with the HSPI in slave mode, set CSA again after a software reset. This prevents the HSPI from receiving erroneous data.

25.4.5 Clock Polarity and Transmit Control

SPCR also allows the user to define the shift timing for transmit data and polarity. The FBS bit in SPCR allows selection between two different transfer formats. When CSA of SPSR is 0, the MSB or LSB is valid at the falling edge of $\overline{\text{HSPI\_CS}}$. The CLKP bit in SPCR allows for control of the polarity select block, shown in figure 25.1, which selects the edge of HSPI_CLK on which data is shifted and sampled in the master and slave.

25.4.6 Transmit and Receive Routines

The master and slave can be considered linked together as a circular shift register synchronized with HSPI_CLK. The transmit byte from the master is replaced with the receive byte from the slave in eight HSPI_CLK cycles. Both the transmit and receive functions are double buffered to allow for continuous reads and writes. When FIFO mode is enabled, 8-entry FIFOs are available for both transmit and receive data.

25.4.7 Flags and Interrupt Timing

The interrupt timing when the flags of the status register (SPSR) and the system control register (SPSCR) are set is shown in figure 25.7.

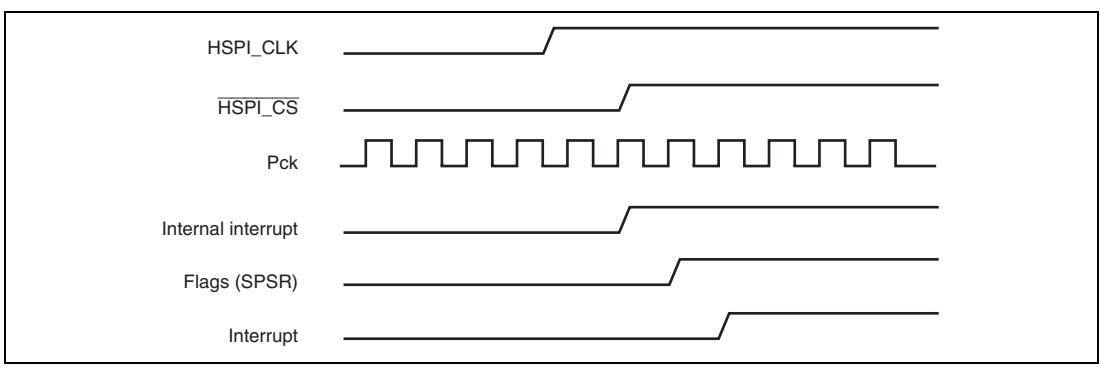


Figure 25.7 Flags and Interrupt Timing

If an interrupt cause (receive FIFO halfway, etc.) occurs, it is reflected to the status register (SPSR) in synchronization with Pck, and an interrupt occurs.

25.5 Power-Down Mode

Power-Down mode has following modes or functions.

(1) Sleep mode

The operation of the HSPI is continued.

(2) Light Sleep mode

The HSPI continues operation, but the operation of receiving/transmitting using the DMAC is not guaranteed. In such cases, following methods are needed before enter light sleep mode.

i. Check all reading/writing operations are finished.

   They are necessary to the transmit buffer (or FIFO) is empty and a content of the receive buffer (or FIFO) is all read out and becomes 0. So in FIFO mode, it is necessary to wait until the TXEM bit in the SPSR register is set to 1. And it is also necessary to read out the SPRBR (receive buffer register) until the RXEM bit in the SPSR register is set to 1.

   In non-FIFO mode, it is necessary to wait until the TXFL bit in the SPSR register is set to 0. And it is also necessary to read out the SPRBR (receive buffer register) until the RXFL bit in the SPSR register is set to 0.

ii. Disable all DMA requests from the HSPI.

(3) Module standby mode

If the HSPI enters module standby mode while operating, correct operation is not guaranteed because the clock supply (Pck) for the HSPI is stopped. Following procedure is needed before enter module standby mode.

i. Check all reading/writing operations are finished.

   The processing is same as the case of light sleep mode.

ii. Prohibit all DMA and interrupt requests in the HSPI. The FIFO mode also should be disabled.

iii. Set the MSTP2 bit in the MSTPCR0 (standby control register 0) to 1.

(4) Changing clock frequency

When clock frequency (Pck) is changed while the HSPI is operating, the operation is continued. But in master mode, there is a possibility that the communication with the external device is disconnected because the clock output from the HSPI is changed. In the slave mode, there is a possibility that the HSPI underflow or overflow occurs. So before changing the clock frequency while the HSPI is operating, following procedure is needed.

i. Check all reading/writing operations are finished.
   The procedure is same as the case of light sleep mode.

ii. Disable all DMA and interrupt requests from the HSPI. Also the FIFO mode should be disabled.

Section 26 NAND Flash Memory Controller (FLCTL)

The NAND flash memory controller (FLCTL) provides interfaces with an external NAND-type flash memory. And it has functions about ECC generating and error detection as a measure of read error that is specific to flash memory.

26.1 Features

(1) NAND-Type Flash Memory Interface

- Interface that can be connected to NAND-type flash memory
- Read or write in sector* units (512 + 16 bytes), and ECC processing.
- Read or write in byte units

Note: * In the data sheet of NAND-type flash memory, an access unit of some products is defined to be 2048 + 64 bytes as a page. In this document, a sector always refers to the access unit of 512 + 16 bytes.

(2) Access Modes

The FLCTL has two selectable access modes.

- Command access mode:
  Performs an access by specifying a command, address, and data size to be issued from the FLCTL to the flash memory. This mode can read from, write to or erase, without ECC processing.
- Sector access mode:
  Reads or writes in physical sector units by specifying a physical sector. Then ECC generation and checking are performed. By specifying the number of sectors, the continuous physical sectors can be read from or written to.

(3) Sectors and Control Codes

- A sector is comprised of 512-byte data and 16-byte control code. Control code includes 8-byte ECC.
- The byte location that a data is inserted in can be specified in 4-byte units.
- User information can be written in control codes other than ECC.

(4) ECC

- An 8-byte ECC code is generated and error is checked for one sector (512 + 16 bytes).
  (However, the number of bytes covered by ECC in 16-byte control code depends on settings.)

- The error-correcting capability is up to arbitrary 3 bytes in one sector.
- When writing, a data and control codes ahead of the ECC are covered by ECC generation. A control code behind the ECC is not covered by ECC.
- When reading, a data and control codes ahead of the ECC are covered by ECC generation. An ECC of a control code in FIFO is not an ECC itself that read from flash memory but replaced to the result of checking by the ECC circuit.
- FLCTL doesn't correct errors. Software processing is necessary.

(5) Data Error

- When a program error or erase error occurs, the error is reflected on the error source flags. Interrupts for each source can be specified.
- When read error occurs, ECC in the control code becomes other than zero.
- When an ECC error is detected, correct the error, specify a substitute sector as needed, and copy contents in the block to the sector.

(6) Data Transfer FIFO

- On-chip 224-byte FLDTFIFO for data transfer of flash memory
- On-chip 32-byte FLECFIFO for data transfer of a control code
- Flag bit for detection of overrun or underrun during access from the CPU or DMA

(7) DMA Transfer

- By individually specifying the transfer destinations of data and control code of flash memory to the DMA controller, data and control code can be transferred to different areas.

(8) Access Size

- Registers include 32-bit registers and an 8-bit register. Read from or write to the register with the specified access size.
- The access size of FIFO is 32 bits (4 bytes). In reading, set the byte number to a multiple of four. In writing, set the byte number to a multiple of four in writing.

(9) Access Time

- The operating frequency of the FLCTL pins can be specified by the FCKSEL bit and the QTSEL bit in FLCMNCR, regardless of the operating frequency of the peripheral bus.
- The operating clock, FCLK, on the pins for the NAND-type flash memory is used by dividing the operating clock of the peripheral bus (a peripheral clock Pck).

- In NAND-type flash memory, the FRE and $\overline{\text{FWE}}$ pins operate with the FCLK specified by FLCMNCR. To ensure the setup time, this operating frequencies should not exceed the maximum operating frequency of memory to be connected.

Figure 26.1 shows a block diagram of the FLCTL.

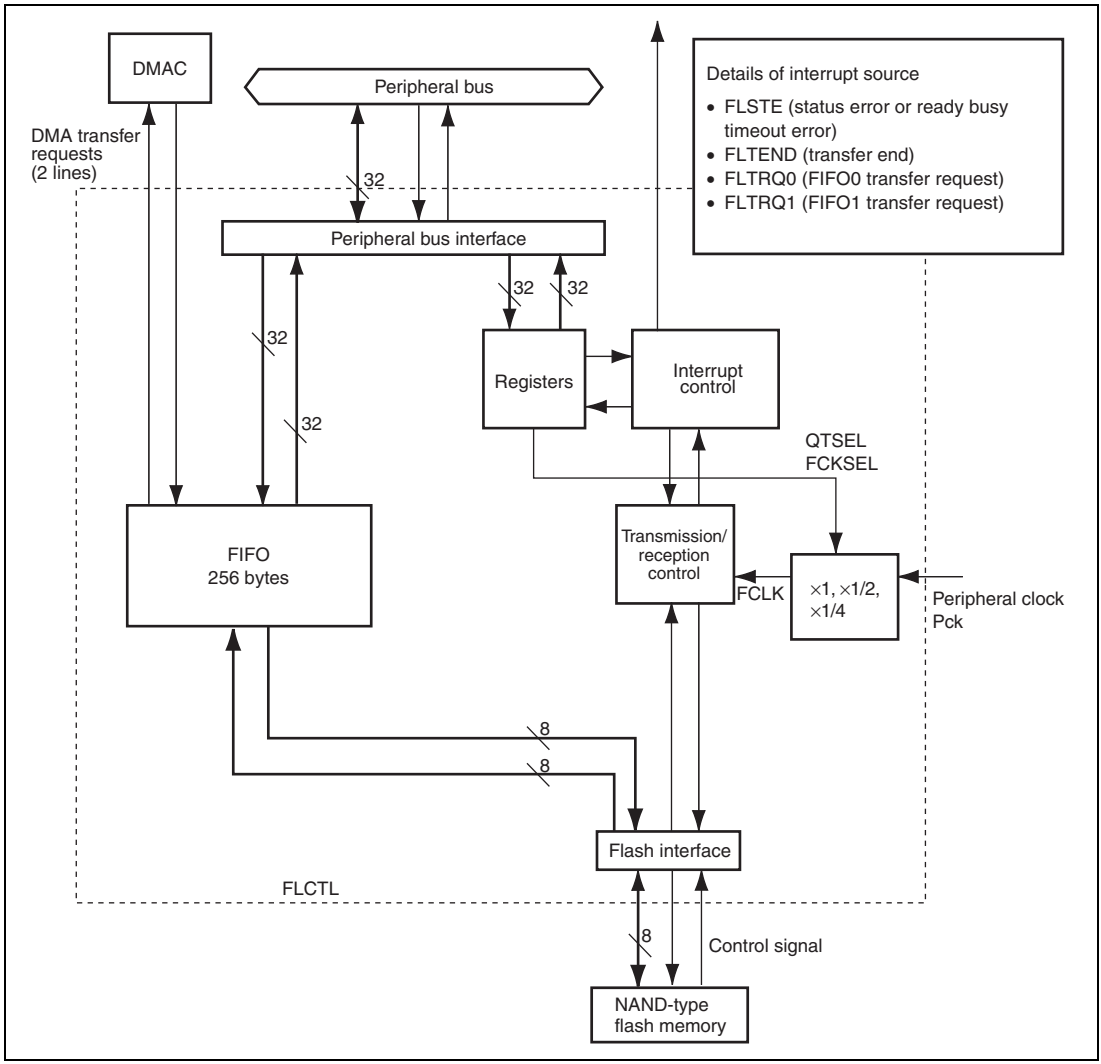


Figure 26.1 Block Diagram of FLCTL

26.2 Input/Output Pins

Table 26.1 shows the pin configuration of the FLCTL.

Table 26.1 Pin Configuration

			Corresponding Flash Memory Pin	
Pin Name	**Function**	**I/O**	**NAND Type**	**Description**
FCE	Chip enable	Output	CE	Enables flash memory connected to this LSI. Multiplexed with SCIF0_CTS.
FD7 to FD0	Data I/O	I/O	I/O7 to I/O0	I/O pins for command, address, and data. Multiplexed with D7 to D0.
FCLE	Command latch enable	Output	CLE	Command Latch Enable (CLE) Asserted when a command is output. Multiplexed with DACK0.
FALE	Address latch enable	Output	ALE	Address Latch Enable (ALE) Asserted when an address is output. Negated when data is input or output. Multiplexed with DACK1/BACK.
FRE	Read enable	Output	RE	Read Enable (RE) Reads data at the falling edge of RE. Multiplexed with RD/FRAME.
FWE	Write enable	Output	WE	Write Enable Flash memory latches a command, address, and data at the rising edge of WE. Multiplexed with WE1.
FRB	Ready/busy	Input	R/B	Ready/Busy Indicates ready state at high level. Indicates busy state at low level. Multiplexed with MODE14/SCIF5_SCK.
—*	—	—	WP	Write Protect/Reset Prevents accidental erasure or programming when power is turned on or off, at low level.

Pin Name	Function	I/O	Corresponding Flash Memory Pin: NAND Type	Description
$\overline{\text{FSE}}$	Spare area enable	Output	$\overline{\text{SE}}$	Spare Area Enable Enables access to spare area. This pin must be fixed low in sector access mode. Multiplexed with MODE8/SCIF4_TXD/DRAK0/SSI3_SCK.

Note: * Not supported by this LSI.

26.3 Register Descriptions

Table 26.2 shows the register configuration of FLCTL. Table 26.3 shows the register states in each processing mode.

Table 26.2 Register Configuration of FLCTL

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync Clock
Common control register	FLCMNCR	R/W	H'FF81 0000	H'1F81 0000	32	Pck
Command control register	FLCMDCR	R/W	H'FF81 0004	H'1F81 0004	32	Pck
Command code register	FLCMCDR	R/W	H'FF81 0008	H'1F81 0008	32	Pck
Address register	FLADR	R/W	H'FF81 000C	H'1F81 000C	32	Pck
Data register	FLDATAR	R/W	H'FF81 0010	H'1F81 0010	32	Pck
Data counter register	FLDTCNTR	R/W	H'FF81 0014	H'1F81 0014	32	Pck
Interrupt DMA control register	FLINTDMACR	R/W	H'FF81 0018	H'1F81 0018	32	Pck
Ready busy timeout setting register	FLBSYTMR	R/W	H'FF81 001C	H'1F81 001C	32	Pck
Ready busy timeout counter	FLBSYCNT	R	H'FF81 0020	H'1F81 0020	32	Pck
Data FIFO register	FLDTFIFO	R/W	H'FF81 0024	H'1F81 0024	32	Pck
Control code FIFO register	FLECFIFO	R/W	H'FF81 0028	H'1F81 0028	32	Pck
Transfer control register	FLTRCR	R/W	H'FF81 002C	H'1F81 002C	8	Pck
Address register 2	FLADR2	R/W	H'FF81 003C	H'1F81 003C	32	Pck
Data alignment register	FLALGCR	R/W	H'FF81 0200	H'1F81 0200	32	Pck
Local bus conflict control register	FLABTCTL	R/W	H'FF81 0300	H'1F81 0300	32	Pck
Local bus area select register	FLCSLR	R/W	H'FF80 2100	H'FF80 2100	32	Pck

Table 26.3 Register States in Each Processing Mode

Register Abbreviation	Power-On Reset	Manual Reset	Sleep/Light Sleep	Module Standby
FLCMNCR	H'0000 0000	Initialized	Retained	Retained
FLCMDCR	H'0000 0000	Initialized	Retained	Retained
FLCMCDR	H'0000 0000	Initialized	Retained	Retained
FLADR	H'0000 0000	Initialized	Retained	Retained
FLDATAR	H'0000 0000	Initialized	Retained	Retained
FLDTCNTR	H'0000 0000	Initialized	Retained	Retained
FLINTDMACR	H'0000 0000	Initialized	Retained	Retained
FLBSYTMR	H'0000 0000	Initialized	Retained	Retained
FLBSYCNT	H'0000 0000	Initialized	Retained	Retained
FLDTFIFO	Undefined	Undefined	Retained	Retained
FLECFIFO	Undefined	Undefined	Retained	Retained
FLTRCR	H'00	Initialized	Retained	Retained
FLADR2	H'0000 0000	Initialized	Retained	Retained
FLALGCR	H'0000 0000	Initialized	Retained	Retained
FLABTCTL	H'0000 0000	Initialized	Retained	Retained
FLCSLR	H'0000 0000	Initialized	Retained	Retained

26.3.1 Common Control Register (FLCMNCR)

FLCMNCR is a 32-bit readable/writable register that specifies the type (NAND) of flash memory, access mode, and $\overline{FCE}$ pin output.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SNAND	QT SEL	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCKSEL	—	ECCPOS[1:0]		ACM[1:0]		NANDWF	—	—	—	—	—	CE0	—	—	TYPESEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R	R/W

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	SNAND	0	R/W	Large Capacity NAND Flash Memory Select This bit is used to specify the NAND flash memory that a page consists of 2048 + 64 bytes. 0: Selects the flash memory that a page consists of 512 + 16 bytes 1: Selects the flash memory that a page consists of 2048 + 64 bytes
17	QTSEL	0	R/W	Quarter Flash Clock Select 0: Uses the FCLK selected by the FCKSEL bit 1: Divides the operating clock of the FLCTL (a peripheral clock) by four and uses it as the FCLK when FCKSEL = 0 Note: When FCKSEL = 1, setting this bit to 1 is prohibited.
16	—	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
15	FCKSEL	0	R/W	Flash Clock Select 0: Divides the operating clock of the FLCTL (a peripheral clock) by two and uses it as the FCLK 1: Uses the operating clock of the FLCTL (a peripheral clock) as the FCLK
14	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
13, 12	ECCPOS [1:0]]	00	R/W	Position of ECC Inclusion Specification [1:0] Specify inclusion position of ECC in a control code. 00: Allocate ECC to bytes of [0:7] 01: Allocate ECC to bytes of [4:11] 10: Allocate ECC to bytes of [8:15] 11: Setting prohibited
11, 10	ACM[1:0]	00	R/W	Access Mode Specification [1:0] Specify access mode. 00: Command access mode 01: Sector access mode 10: Setting prohibited 11: Setting prohibited
9	NANDWF	0	R/W	NAND Wait Insertion Operation 0: Reserved (Setting prohibited) 1: A wait cycle is inserted
8 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CE0	0	R/W	Chip Enable 0 0: Disabled (Outputs high level to the $\overline{\text{FCE}}$ pin) 1: Enabled (Outputs low level to the $\overline{\text{FCE}}$ pin)
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TYPESEL	0	R/W	Memory Select 0: Reserved (Setting prohibited) 1: NAND-type flash memory is selected Note: Set TYPESEL to 1 to use FLCTL.

26.3.2 Command Control Register (FLCMDCR)

FLCMDCR is a 32-bit readable/writable register that issues a command in command access mode, specifies address issue, and specifies destination of data to be input or output. In sector access mode, FLCMDCR specifies the number of sector transfers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR CNT2	SCTCNT[19:16]				ADRMD	CDSRC	DOSR	—	—	SELRW	DOADR	ADRCNT[1:0]		DOCMD2	DOCMD1
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCTCNT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	ADRCNT2	0	R/W	Address Issue Byte Number Specification Specifies the number of bytes issued in the address stage. 0: Issues address as many as the bytes specified in ADRCNT1 and ADRCNT 1: Issues 5-byte address Note: Set ADRCNT1 and ADRCNT0 to 0.
30 to 27	SCTCNT [19:16]	All 0	R/W	Selector Transfer Count Specification [19:16] These bits are extended bits of bits SCTCNT[15:0]. When bits SCTCNT[19:16] and bits SCTCNT[15:0] are put together, these bits operate as a 20-bit counter of bits SCTCNT[19:0].
26	ADRMD	0	R/W	Sector Access Address Specification This bit is invalid in command access mode. This bit is valid only in sector access mode. 0: The value of the address register is handled as a physical sector number. Always use this value in sector access. 1: The value of the address register is output as the address of flash memory. Note: Clear this bit to 0 in continuous sector access.

Bit	Bit Name	Initial Value	R/W	Description
25	CDSRC	0	R/W	Data Buffer Specification Specifies the data buffer to be read from or written to in the data stage* in command access mode. 0: Specifies FLDATAR as the data buffer. 1: Specifies FLDTFIFO as the data buffer.
24	DOSR	0	R/W	Status Read Check Specifies whether the status read is performed after the second command has been issued in command access mode. 0: Performs no status read 1: Performs status read
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	SELRW	0	R/W	Data Read/Write Specification Specifies whether the direction is read or write in data stage. 0: Read 1: Write
20	DOADR	0	R/W	Address Stage Execution Specification Specifies whether the address stage* is executed in command access mode. 0: Performs no address stage 1: Performs address stage
19, 18	ADRCNT [1:0]	00	R/W	Address Issue Byte Count Specification Specify the number of bytes for the address data to be issued in address stage*. 00: Issue 1-byte address 01: Issue 2-byte address 10: Issue 3-byte address 11: Issue 4-byte address
17	DOCMD2	0	R/W	Second Command Stage* Execution Specification Specifies whether the second command stage* is executed in command access mode. 0: Does not execute the second command stage 1: Executes the second command stage

Bit	Bit Name	Initial Value	R/W	Description
16	DOCMD1	0	R/W	First Command Stage* Execution Specification Specifies whether the first command stage* is executed in command access mode. 0: Does not execute the first command stage 1: Executes the first command stage
15 to 0	SCTCNT [15:0]	H'0000	R/W	Sector Transfer Count Specification Specify the number of sectors to be read continuously in sector access mode. These bits are counted down for each sector transfer end, and stop when they reach 0. When accessing one sector, set SCTCNT to 1.

Note: * For command stage, address stage, and data stage, see figure 26.2.

26.3.3 Command Code Register (FLCMCDR)

FLCMCDR is a 32-bit readable/writable register that specifies a command to be issued in command access or sector access.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMD[15:8]								CMD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 8	CMD[15:8]	H'00	R/W	Specify a command code to be issued in the second command stage.
7 to 0	CMD[7:0]	H'00	R/W	Specify a command code to be issued in the first command stage.

26.3.4 Address Register (FLADR)

FLADR is a 32-bit readable/writable register that specifies an address to be output in command access mode. In sector access mode, a physical sector number specified in the physical sector address bits is converted into an address to be output.

- Command access mode

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR[31:24]								ADR[23:16]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR[15:8]								ADR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ADR[31:24]	H'00	R/W	Fourth Address Data Specify the fourth data to be output to flash memory as an address in command access mode.
23 to 16	ADR[23:16]	H'00	R/W	Third Address Data Specify the third data to be output to flash memory as an address in command access mode.
15 to 8	ADR[15:8]	H'00	R/W	Second Address Data Specify the second data to be output to flash memory as an address in command access mode.
7 to 0	ADR[7:0]	H'00	R/W	First Address Data Specify the first data to be output to flash memory as an address in command access mode.

- Sector access mode

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ADR[25:16]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are undefined depending on the operation mode of the FLCTL.
25 to 0	ADR[25:0]	All 0	R/W	Physical Sector Address Specify a physical sector number to be accessed in sector access mode. The physical sector number is converted into an address and is output to flash memory. When the ADRCNT2 bit in FLCMDCR is 1, ADR25 to ADR0 are valid. When the ADRCNT2 bit in FLCMDCR is 0, ADR17 to ADR0 are valid.

26.3.5 Address Register 2 (FLADR2)

FLADR2 is a 32-bit readable/writable register that is valid when the ADRCNT2 bit in FLCMDCR is 1. This register specifies the value to be output as an address in command mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ADR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	ADR[7:0]	All 0	R/W	Fifth Address Data Specify the fifth data to be output to flash memory as an address in command access mode.

26.3.6 Data Counter Register (FLDTCNTR)

FLDTCNTR is a 32-bit readable/writable register that specifies the number of bytes to be read or written in command access mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECFLW[7:0]								DTFLW[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DTCNT[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ECFLW[7:0]	H'00	R	FLECFIFO Access Count Specify the number of longwords (4 bytes) in FLECFIFO to be read or written. These bit can be used when the CPU reads from or writes to FLECFIFO. In reading from FLECFIFO, these bits specify the number of longwords of the data that can be read from FLECFIFO. In writing to FLECFIFO, these bits specify the number of longwords of empty area that can be written to FLECFIFO.
23 to 16	DTFLW[7:0]	H'00	R	FLDTFIFO Access Count Specify the number of longwords (4 bytes) in FLDTFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLDTFIFO. In reading from FLDTFIFO, these bits specify the number of longwords of the data that can be read from FLDTFIFO. In writing to FLDTFIFO, these bits specify the number of longwords of empty area that can be written in FLDTFIFO.
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	DTCNT[11:0]	H'000	R/W	Data Count Specification Specify the number of bytes of data to be read or written in command access mode (Up to 2048 + 64 bytes can be specified.)

26.3.7 Data Register (FLDATAR)

FLDATAR is a 32-bit readable/writable register. It stores data to be input or output used when the CDSRC bit in FLCMDCR is cleared to 0 in command access mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DT[31:24]								DT[23:16]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT[15:8]								DT[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DT[31:24]	H'00	R/W	Fourth Data Specify the 4th data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data
23 to 16	DT[23:16]	H'00	R/W	Third Data Specify the 3rd data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data
15 to 8	DT[15:8]	H'00	R/W	Second Data Specify the 2nd data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data
7 to 0	DT[7:0]	H'00	R/W	First Data Specify the 1st data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data

26.3.8 Interrupt DMA Control Register (FLINTDMACR)

FLINTDMACR is a 32-bit readable/writable register that enables or disables DMA transfer requests or interrupts. A transfer request from the FLCTL to the DMAC is issued after each access mode has started.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ECER INTE	—	—	FIFOTRG[1:0]		AC1 CLR	AC0 CLR	DREQ1 EN	DREQ0 EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	EC ERB	STE RB	BTO ERB	TRR EQF1	TRR EQF0	STER INTE	RBER INTE	TE INTE	TR INTE1	TR INTE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	ECERINTE	0	R/W	ECC Error Interrupt Enable 0: Disables the interrupt when ECC error occurs 1: Enables the interrupt when ECC error occurs
23, 22	—	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
21, 20	FIFOTRG [1:0]	00	R/W	FIFO Trigger Setting Change the condition for the FIFO transfer request. In reading flash memory 00: Issue an interrupt to the CPU or a DMA transfer request when 4-byte data is written to FLDTFIFO 01: Issue an interrupt to the CPU when 16-byte data is written to FLDTFIFO (do not set DMA transfer) 10: Issue an interrupt to the CPU when 128-byte data is written to FLDTFIFO (do not set DMA transfer) 11: Setting prohibited In writing flash memory 00: Issue an interrupt to the CPU or DMA transfer request when FLDTFIFO has 4 bytes or more of empty area (do not set DMA transfer) 01: Issue an interrupt to the CPU when FLDTFIFO has 16 bytes or more of empty area (do not set DMA transfer) 10: Issue an interrupt to the CPU when FLDTFIFO has 128 bytes or more of empty area (do not set DMA transfer) 11: Setting prohibited
19	AC1CLR	0	R/W	FLECFIFO Clear Clears the address counter of FLECFIFO. FIFO needs to be cleared when changing access direction (read/write). 0: Retains the address counter value of FLECFIFO. In flash-memory access, clear this bit to 0. 1: Clears the address counter of FLECFIFO. After clearing the counter, clear this bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
18	AC0CLR	0	R/W	FLDTFIFO Clear Clears the address counter of FLDTFIFO. FIFO needs to be cleared when changing access direction (read/write). 0: Retains the address counter value of FLDTFIFO. In flash-memory access, clear this bit to 0. 1: Clears the address counter of FLDTFIFO. After clearing the counter, clear this bit to 0
17	DREQ1EN	0	R/W	FLECFIFODMA Request Enable Enables or disables the DMA transfer request issued from FLECFIFO. 0: Disables the issue of DMA transfer request from FLECFIFO 1: Enables the issue of DMA transfer request from FLECFIFO
16	DREQ0EN	0	R/W	FLDTFIFODMA Request Enable Enables or disables the DMA transfer request issued from FLDTFIFO. 0: Disables the issue of DMA transfer request from FLDTFIFO 1: Enables the issue of DMA transfer request from FLDTFIFO
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	ECERB	0	R/W	ECC Error Indicates the result of ECC error detection. This bit is set to 1 if an ECC error that is in a data read from flash memory by the sector access mode is detected. Since this bit is a flag bit, 1 cannot be written to this bit. Only 0 can be written to clear this flag. 0: Indicates that no ECC error occurs (All ECCs are 0) 1: Indicates that an ECC error occurs.

Bit	Bit Name	Initial Value	R/W	Description
8	STERB	0	R/W	Status Error Indicates the result of status read. This bit is set to 1 if the specific bit in bits STAT7 to STAT0 in FLBSYCNT is set to 1 in status read. Since this bit is a flag bit, 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no status error occurs (the specific bit in bits STAT7 to STAT0 in FLBSYCNT is 0.) 1: Indicates that a status error occurs For details on the specific bit, see section 26.4.5, Status Read.
7	BTOERB	0	R/W	Timeout Error This bit is set to 1 if a timeout error occurs (bits RBTIMCNT20 to RBTIMCNT0 in FLBSYCNT are set to 0 after they are decremented). Since this bit is a flag bit, 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no timeout error occurs 1: Indicates that a timeout error occurs
6	TRREQF1	0	R/W	FLECFIFO Transfer Request Flag Indicates that a transfer request is issued from FLECFIFO. Since this bit is a flag bit, 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no transfer request is issued from FLECFIFO 1: Indicates that a transfer request is issued from FLECFIFO
5	TRREQF0	0	R/W	FLDTFIFO Transfer Request Flag Indicates that a transfer request is issued from FLDTFIFO. Since this bit is a flag bit, 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no transfer request is issued from FLDTFIFO 1: Indicates that a transfer request is issued from FLDTFIFO

Bit	Bit Name	Initial Value	R/W	Description
4	STERINTE	0	R/W	Interrupt Enable at Status Error Enables or disables an interrupt request to the CPU when a status error has occurred. 0: Disables the interrupt request to the CPU by a status error 1: Enables the interrupt request to the CPU by a status error
3	BTOINTE	0	RW	Interrupt Enable at Timeout Error Enables or disables an interrupt request to the CPU when a timeout error has occurred. 0: Disables the interrupt request to the CPU by a timeout error 1: Enables the interrupt request to the CPU by a timeout error
2	TEINTE	0	R/W	Transfer End Interrupt Enable Enables or disables an interrupt request to the CPU when a transfer has been ended (TREND bit in FLTRCR). 0: Disables an interrupt to the CPU at the end of a transfer 1: Enables an interrupt to the CPU at the end of a transfer
1	TRINTE1	0	R/W	FLECFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the CPU by a transfer request from FLECFIFO. 0: Disables an interrupt request to the CPU by a transfer request from FLECFIFO. 1: Enables an interrupt request to the CPU by a transfer request from FLECFIFO. When the DMA transfer is enabled, this bit should be cleared to 0.
0	TRINTE0	0	R/W	FLDTFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the CPU by a transfer request from FLDTFIFO. 0: Disables an interrupt request to the CPU by a transfer request from FLDTFIFO 1: Enables an interrupt request to the CPU by a transfer request from FLDTFIFO When the DMA transfer is enabled, this bit should be cleared to 0.

26.3.9 Ready Busy Timeout Setting Register (FLBSYTMR)

FLBSYTMR is a 32-bit readable/writable register that specifies the timeout time when the FRB pin is busy.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	RBTMOUT[20:16]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBTMOUT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 0	RBTMOUT[20:0]	H'00000	R/W	Ready Busy Timeout Specify timeout time in the busy state. Set the timeout time in the busy state (with the clock cycles of a peripheral clock). When these bits are set to 0, timeout does not occur.

26.3.10 Ready Busy Timeout Counter (FLBSYCNT)

FLBSYCNT is a 32-bit read-only register.

The status of flash memory read by the status read is stored in the bits STAT7 to STAT0.

The timeout time set in bits RBTMOUT20 to RBTMOUT0 in FLBSYTMR is copied to bits RBTIMCNT20 to RBTIMCNT0 and counting down is started when the FRB pin enters the busy state. When values in bits RBTIMCNT20 to RBTIMCNT0 are decremented to 0, 1 the BTOERB bit in FLINTDMACR is set to 1, and the occurrence of a timeout error is notified. In this case, an FLSTE interrupt can be issued if an interrupt is enabled by the RBERINTE bit in FLINTDMACR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STAT[7:0]								—	—	—	RBTIMCNT[20:16]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBTIMCNT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	STAT[7:0]	H'00	R	Indicate the value obtained by the status read from flash memory
23 to 21	—	All 0	R	Reserved These bits are always read as 0.
20 to 0	RBTIMCNT[20:0]	H'00000	R	Ready Busy Timeout Counter When the FRB pin enters the busy state, the values of bits RBTMOUT20 to RBTMOUT0 in FLBSYTMR are copied to these bits. These bits are counted down while the FRB pin is busy. A timeout error occurs when these bits are decremented to 0.

26.3.11 Data FIFO Register (FLDTFIFO)

FLDTFIFO is used to read from or write to the data FIFO area.

The read and write directions specified by the SELRW bit in FLCMDCR must match the read or write access directions specified in this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTFO[31:24]								DTFO[23:16]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTFO[15:8]								DTFO[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DTFO[31:24]	—	R/W	First Data Specify the first data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data
23 to 16	DTFO[23:16]	—	R/W	Second Data Specify the second data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data
15 to 8	DTFO[15:8]	—	R/W	Third Data Specify the third data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data
7 to 0	DTFO[7:0]	—	R/W	Fourth Data Specify the fourth data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data

26.3.12 Control Code FIFO Register (FLECFIFO)

FLECFIFO is used to read from or write to the control code FIFO area.

The read and write directions specified by the SELRW bit in FLCMDCR must match the read and write access directions specified in this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECFO[31:24]								ECFO[23:16]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECFO[15:8]								ECFO[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ECFO[31:24]	Undefined	R/W	First Data Specify the first data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data
23 to 16	ECFO[23:16]	Undefined	R/W	Second Data Specify the second data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data
15 to 8	ECFO[15:8]	Undefined	R/W	Third Data Specify the third data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data
7 to 0	ECFO[7:0]	Undefined	R/W	Fourth Data Specify the fourth data to be input or output via the FD7 to FD0 pins. In writing: Specify write data In reading: Store read data

26.3.13 Transfer Control Register (FLTRCR)

Setting the TRSTRT bit to 1 starts access to flash memory. The completion of the access can be checked by the TREND bit.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TREND	TRSTRT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TREND	0	R/W	Processing End Flag Indicates that the processing performed in the specified access mode has been completed. The write value should always be 0.
0	TRSTRT	0	R/W	Transfer Start When the TREND bit is 0, processing in the access mode specified by the access mode specification bits ACM[1:0] is started by setting the TRSTRT bit from 0 to 1. 0: Stops transfer 1: Starts transfer

26.3.14 Data Alignment Register (FLALGCR)

FLALGCR is a 32-bit readable/writable register that specifies the data alignment mode at access to flash memory.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWAP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SWAP	0	R/W	Data Alignment Mode 0: Disable SWAP 1: Enable SWAP

26.3.15 Local Bus Conflict Control Register (FLABTCTL)

FLABTCTL is a 32-bit readable/writable register that controls the conflict mode with LBSC at access to the local bus.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CODE															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W	—/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	abt_mode	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	CODE	H'0000	—/W	When writing to the abt_mode, this code should be set to H'1234. The read value is always 0.
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1 to 0	abt_mode	00	R/W	Local Bus Conflict Control Mode 00: Not purge the right of local bus control from beginning the transfer (writing 1 to the TRSTRT bit of FLTRCR) to finish the transfer (the TREND bit of FLTRCR is set to 1) 01: Purge the right of local bus control in the state of the FLCTL operates (TRSTRT=1 and TREND=0) according to settings of the internal FIFO. 10,11: Setting prohibited

26.3.16 Local Bus Area Select Register (FLCSLR)

FLCSLR is a 32-bit readable/writable register that specifies the access area of the local bus.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	fcs_sel		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	fcs_sel	000	R/W	Local Bus Area Select 000: Reserved (Setting prohibited) 001: Local area 1 010: Local area 2 011: Local area 3 100: Local area 4 101: Local area 5 110: Local area 6 111: Setting prohibited Access without FLCTL to a local area set by this register is prohibited.

26.4 Operation

26.4.1 Operating Modes

Two operating modes are supported.

- Command access mode
- Sector access mode

26.4.2 Command Access Mode

Command access mode is a mode that accesses flash memory by specifying a command to be issued to flash memory, address, data, read or write direction, and number of times for the registers. In this mode, DMA transfer of input or output data can be performed by using FLDTFIFO.

(1) NAND-Type Flash Memory Access (512 + 16 Bytes)

Figure 26.2 shows an example of reading operation for NAND-type flash memory. In this example, the first command is set to H'00, address data length is set to 3 bytes, and the number of read bytes is set to 6 bytes in the data counter.

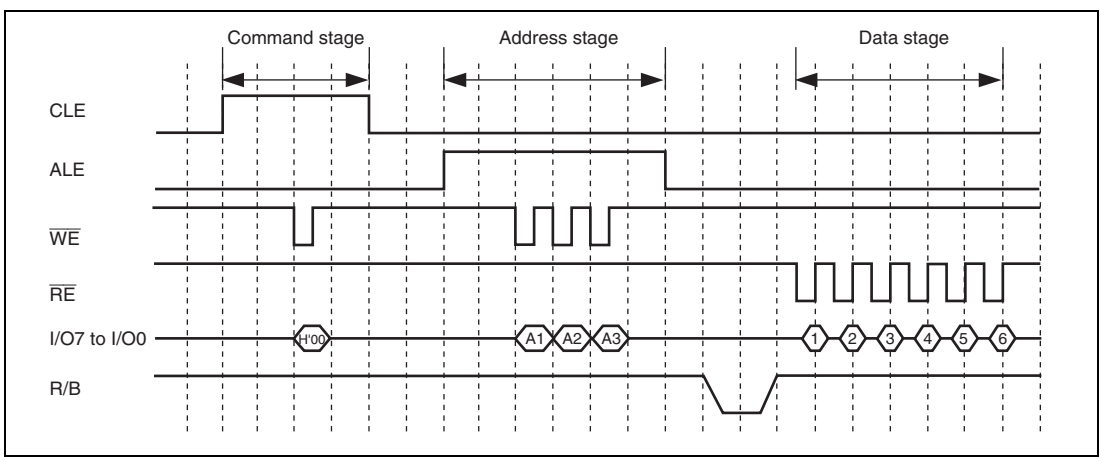


Figure 26.2 Reading Operation Timing for NAND-Type Flash Memory

Figures 26.3 and 26.4 show examples of writing operation for NAND-type flash memory (512 + 16 bytes).

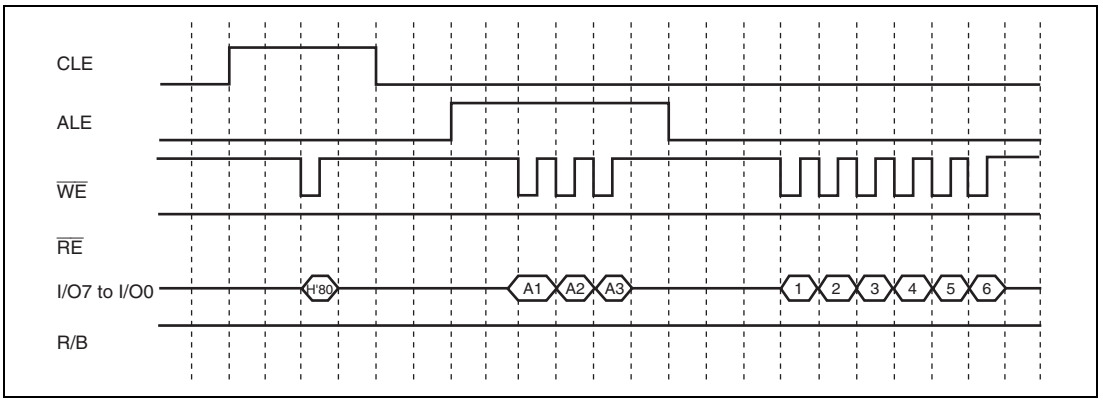


Figure 26.3 Writing Operation Timing for NAND-Type Flash Memory

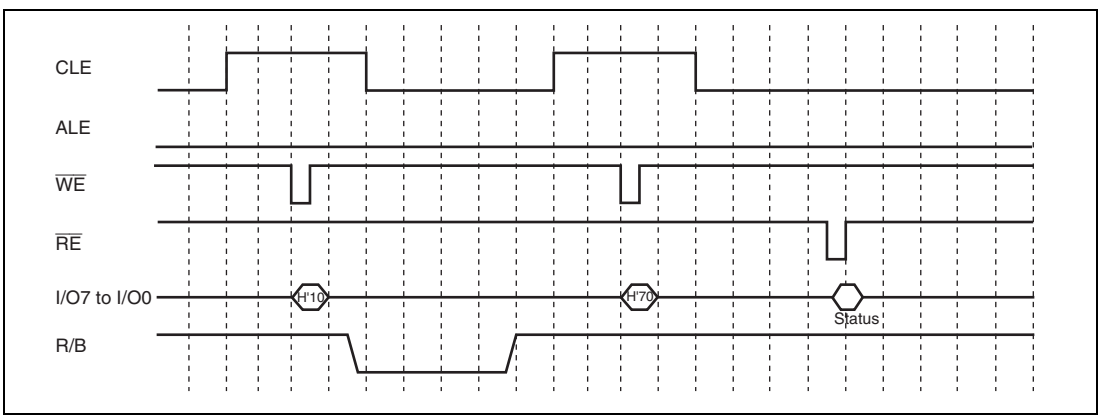


Figure 26.4 Status Read Operation Timing for NAND-Type Flash Memory

(2) NAND-Type Flash Memory Access (2048 + 64 Bytes)

Figure 26.5 shows an example of reading operation for NAND-type flash memory. In this example, the first command is set to H'00, the second command is set to H'30, address data length is set to 4 bytes, and the number of read bytes is set to 4 bytes in the data counter.

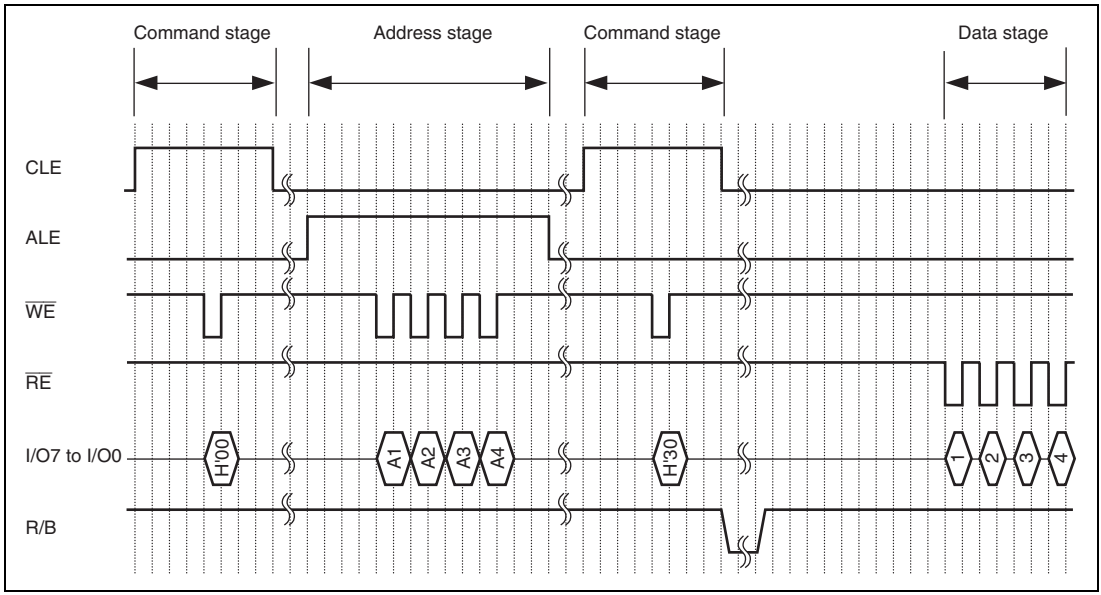


Figure 26.5 Reading Operation Timing for NAND-Type Flash Memory

Figures 26.6 and 26.7 show examples of writing operation for NAND-type flash memory (2048 + 64 bytes).

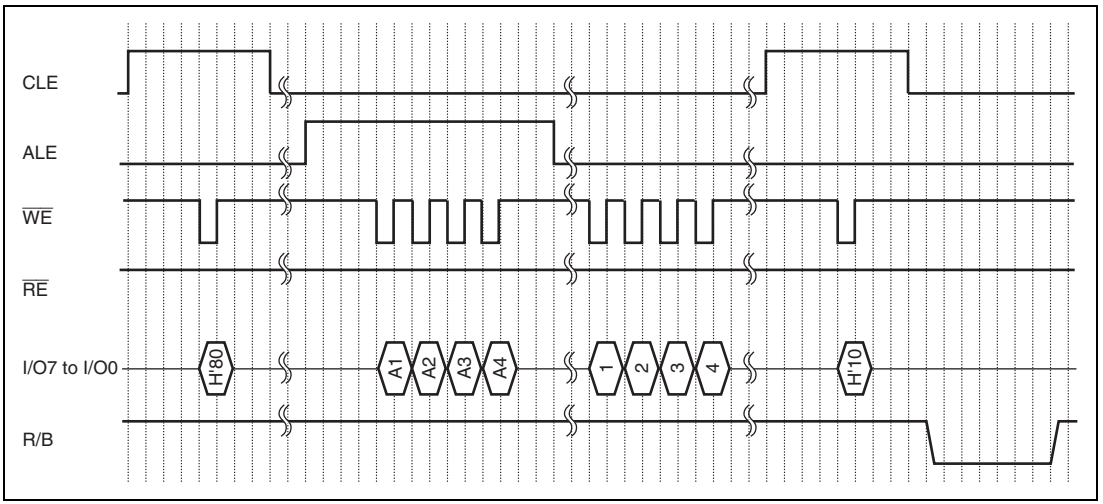


Figure 26.6 Writing Operation Timing for NAND-Type Flash Memory

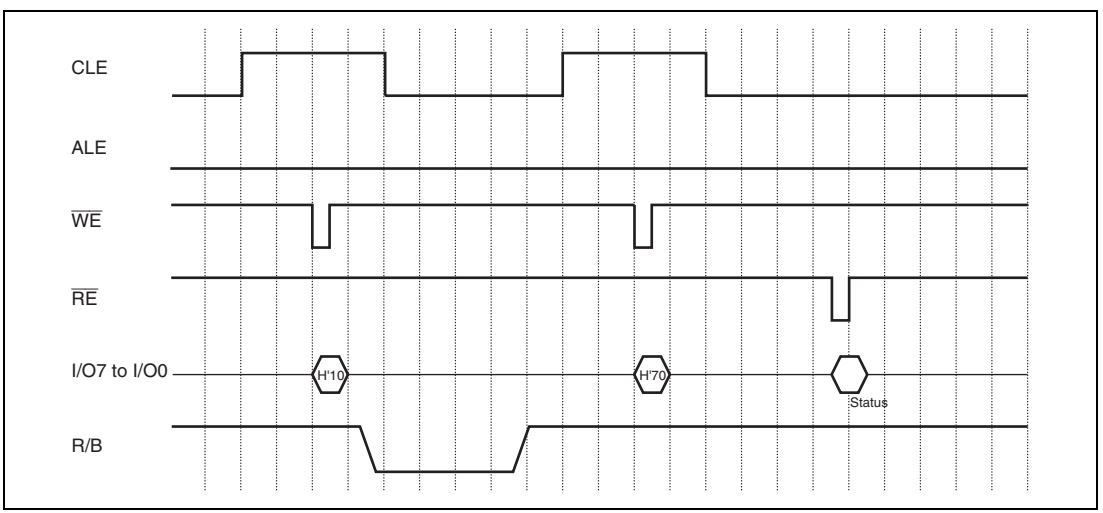


Figure 26.7 Status Read Operation Timing for NAND-Type Flash Memory

26.4.3 Sector Access Mode

In sector access mode, flash memory can be read or written the data in sector units by specifying a physical sector number that will be accessed. ECC addition and detecting are processed at writing and reading respectively.

Since 512-byte data is stored in FLDTFIFO and 16-byte control code is stored in FLECFIFO, DMA transfer can be performed by setting the DREQ1EN and DREQ0EN bits in FLINTDMACR.

Figure 26.8 shows the relationship of DMA transfer between sectors in flash memory (data and control code) and memory in the address space.

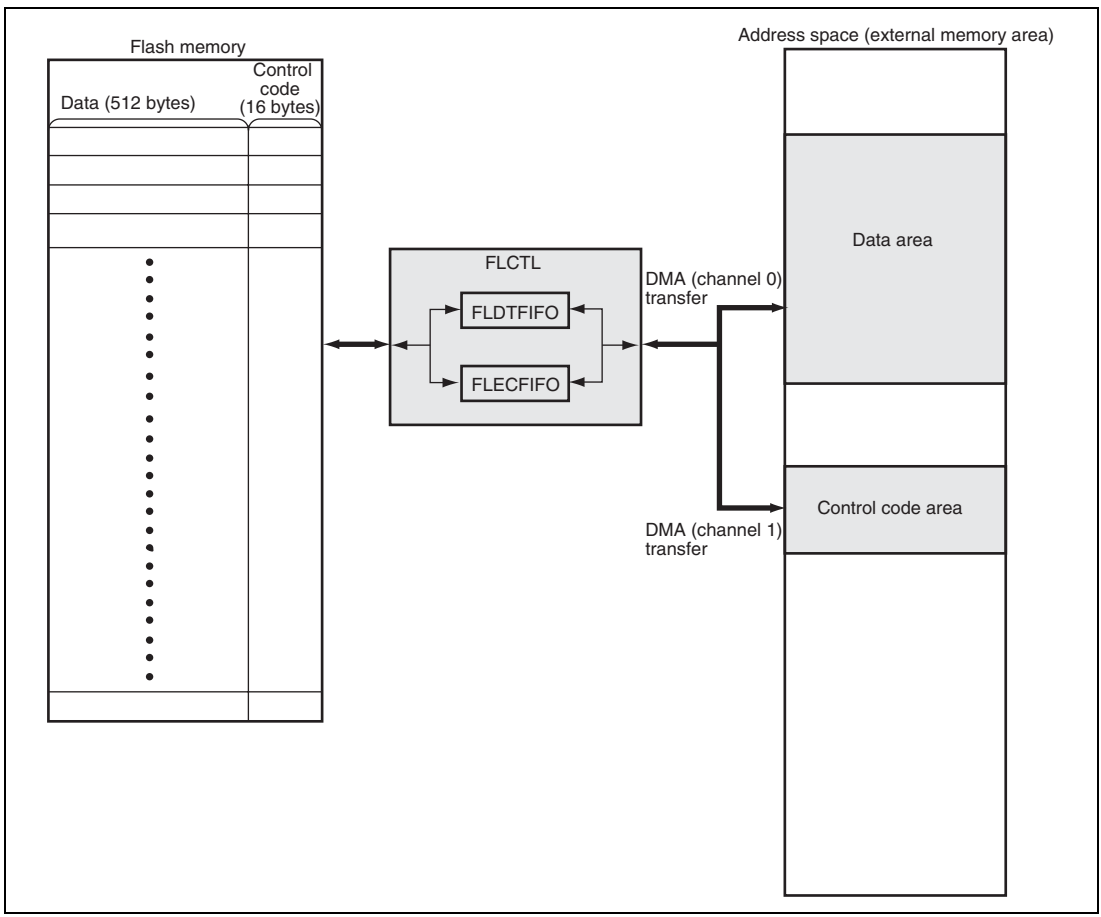


Figure 26.8 Diagram of DMA Transfer between Sector (Data and Control Code) in Flash Memory and Memory in Address Space

(1) Physical Sector

Figure 26.9 shows the relationship between the physical sector address and flash memory address of NAND-type flash memory.

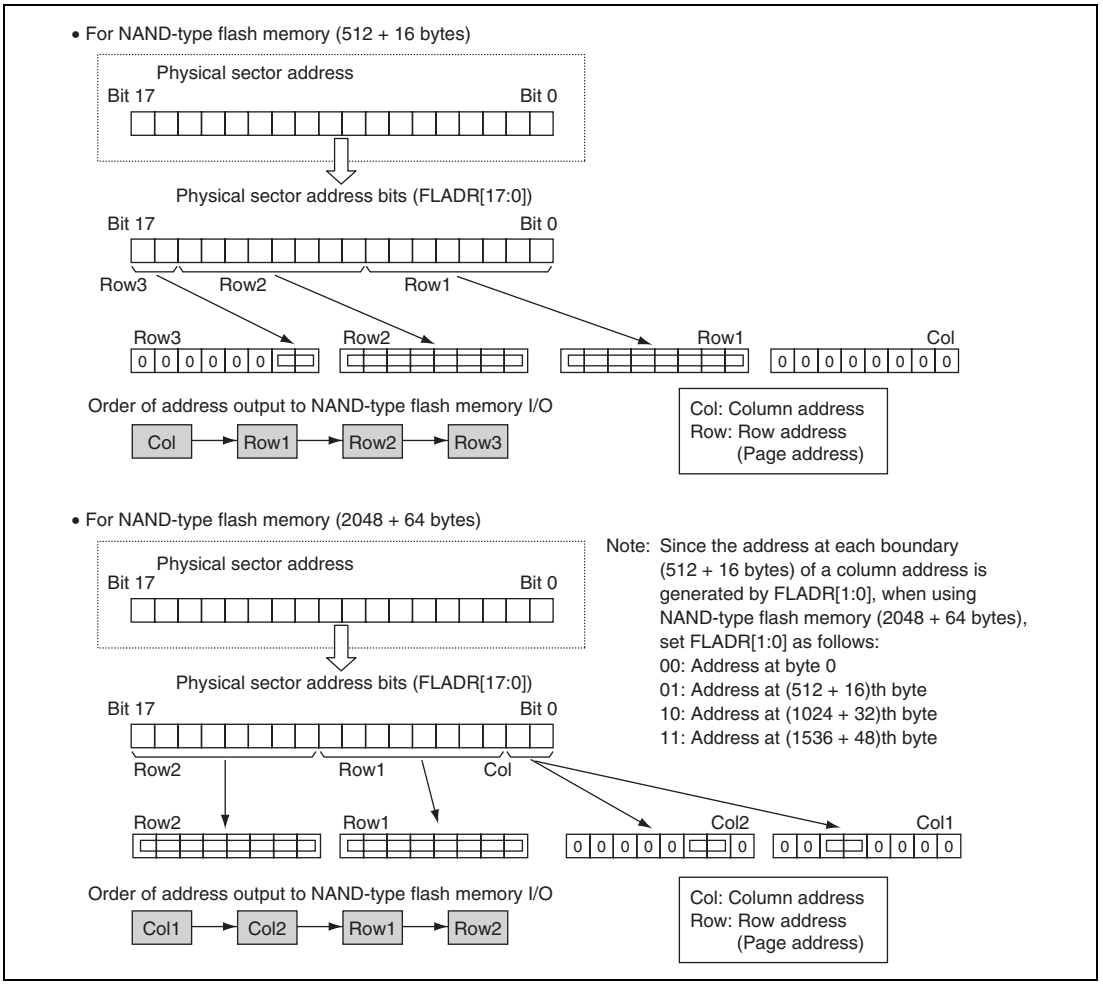


Figure 26.9 Example of Sector Number and NAND-Type Flash Memory Address Expansion

(2) Continuous Sector Access

Continuous physical sectors can be read from or written to by specifying the start physical sector address of NAND-type flash memory and the number of sectors to be transferred. Figure 26.10 shows an example of physical sector specification register and transfer count specification register settings when transferring logical sectors 0 to 40, which are not contiguous because of an unusable sector in NAND-type flash memory.

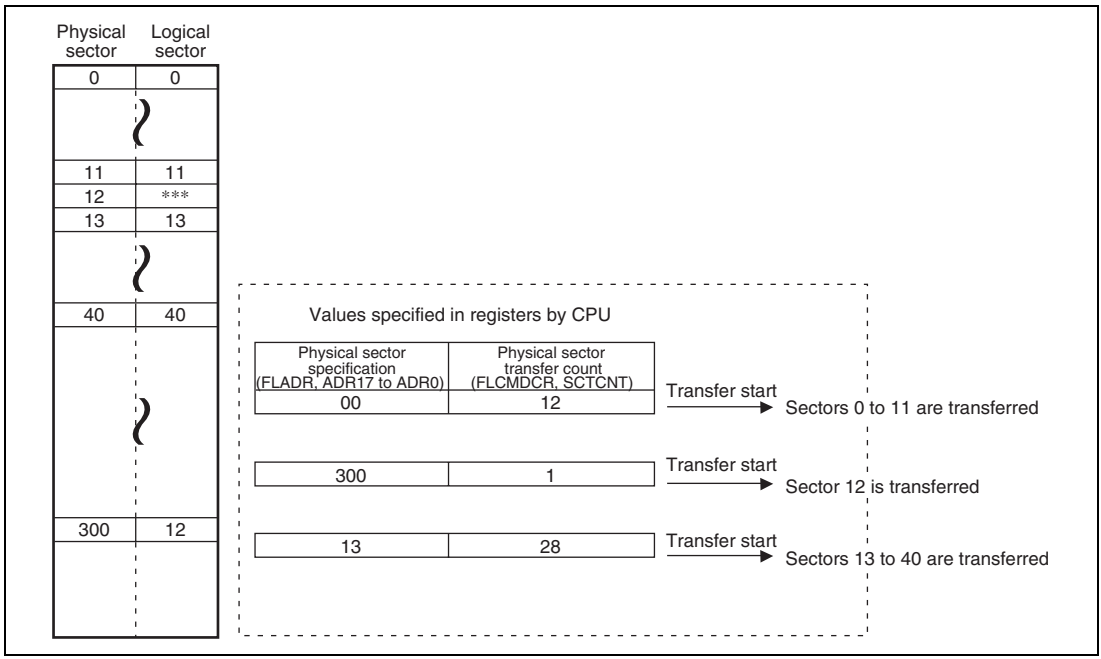


Figure 26.10 Sector Access when Unusable Sector Is in Continuous Sectors

26.4.4 Error Correcting by ECC

Although the FLCTL generates and adds the ECC at writing to the flash memory in sector access mode and detects ECC errors at reading, it does not perform error correction. Error correction should be processed by software.

26.4.5 Status Read

The FLCTL can read the status register of a NAND-type flash memory. The data in the status register of a NAND-type flash memory is input through the FD7 to FD0 pins and stored in the bits STAT7 to STAT0 in FLBSYCNT. The bits STAT7 to STAT0 in FLBSYCNT can be read by the CPU. If a program error or erase error is detected when the status register value is stored in the bits STAT7 to 0 in FLBSYCNT, the STERB bit in FLINTDMACR is set to 1 and generates an interrupt to the CPU if the STERINTE bit in FLINTDMACR is enabled.

(1) Status Read of NAND-Type Flash Memory (512 + 16 Bytes)

The status read of NAND-type flash memory can be performed by inputting the command H'70 to NAND-type flash memory. When the DOSR bit in FLCMDCR is set to 1 and writing is performed in command access mode or sector access mode, the FLCTL automatically inputs H'70 to NAND-type flash memory and status read is performed. During the status read of NAND-type flash memory, the FD7 to FD0 pins indicate the following information as shown in table 26.4.

Table 26.4 Status Read of NAND-Type Flash Memory (512 + 16 Bytes)

Bit	Status (Definition)	Description
STAT7	Write protection	0: Cannot be written 1: Can be written
STAT6	Ready/busy	0: Busy state 1: Ready state
STAT5 to STAT1	Reserved	0
STAT0	Write/erase	0: Pass 1: Fail

(2) Status Read of NAND-Type Flash Memory (2048 + 64 Bytes)

The status read of NAND-type flash memory can be performed by inputting the command H'70 to NAND-type flash memory. When the DOSR bit in FLCMDCR is set to 1 and writing is performed in command access mode or sector access mode, the FLCTL automatically inputs H'70 to NAND-type flash memory and status read is performed. During the status read of NAND-type flash memory, the FD7 to FD0 pins indicate the following information as shown in table 26.5.

Table 26.5 Status Read of NAND-Type Flash Memory (2048 + 64 Bytes)

Bit	Status (Definition)	Description
STAT7	Write protection	0: Cannot be written 1: Can be written
STAT6	Ready/busy	0: Busy state 1: Ready state
STAT5	Ready/busy	0: Busy state 1: Ready state
STAT4 to STAT1	Reserved	0
STAT0	Write/erase	0: Pass 1: Fail

26.5 Example of Register Setting

The examples of setting and starting registers in each access mode are shown below.

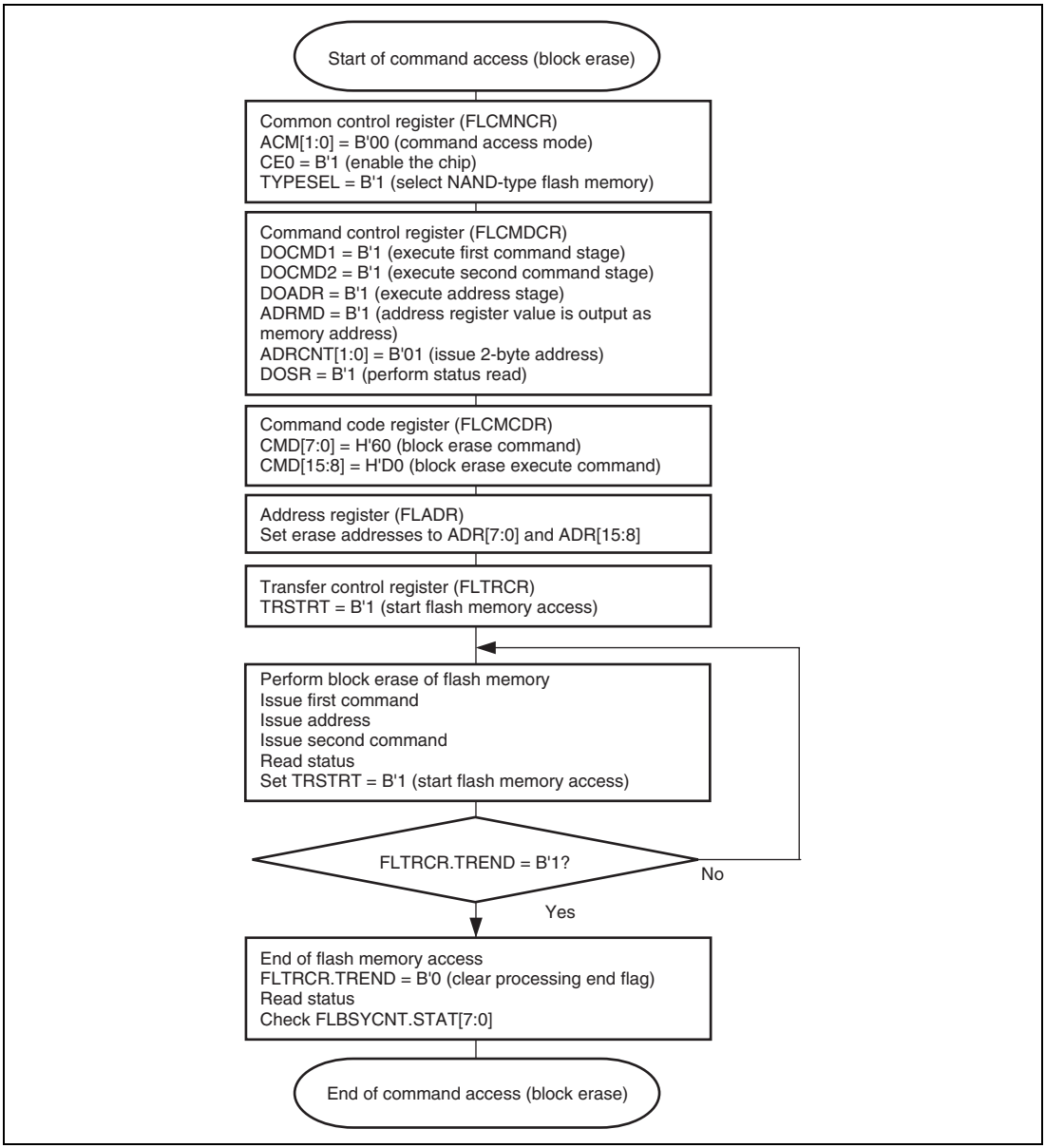


Figure 26.11 NAND Command Access (Block Erase)

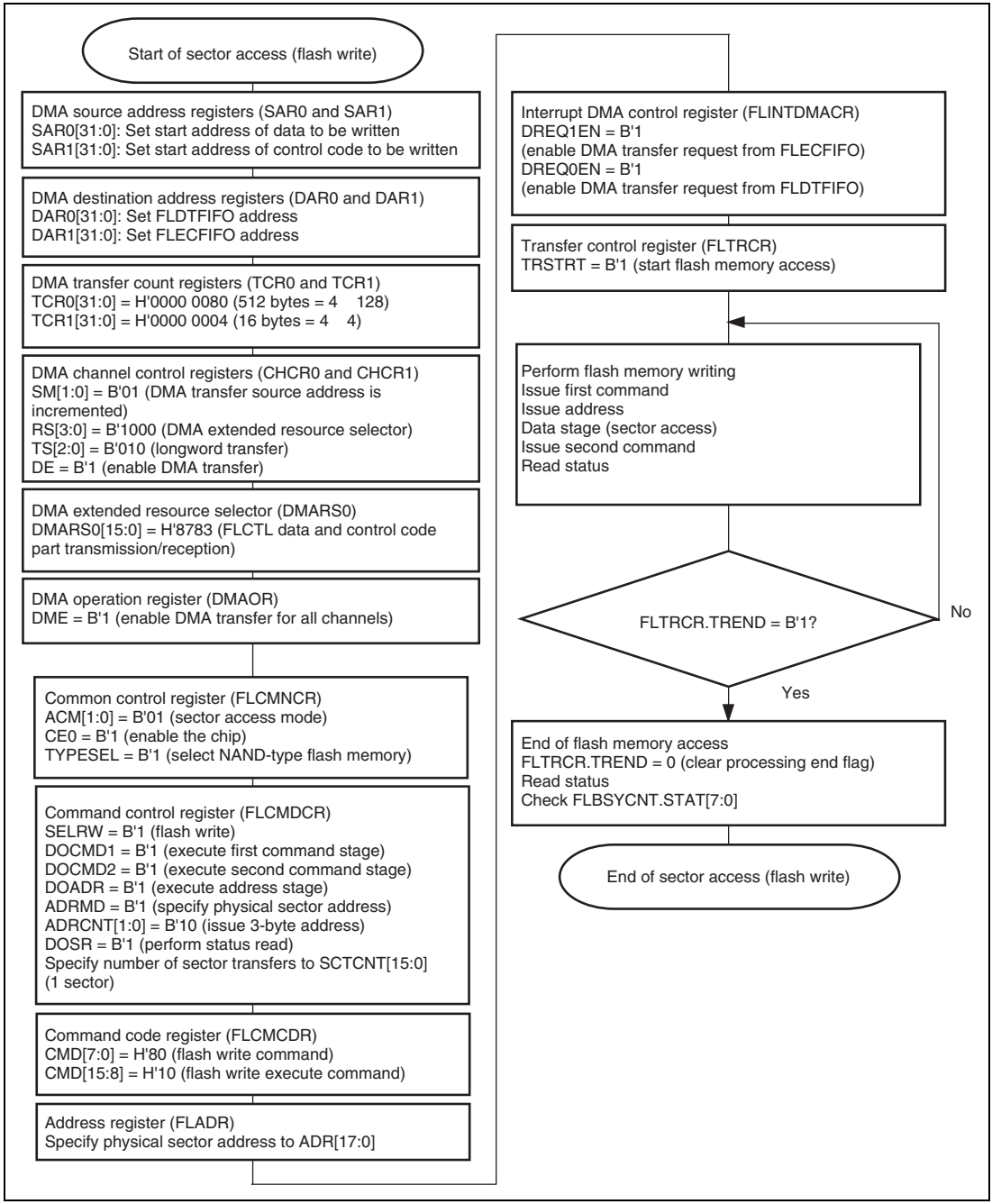


Figure 26.12 NAND Sector Access (Flash Write) Using DMAC

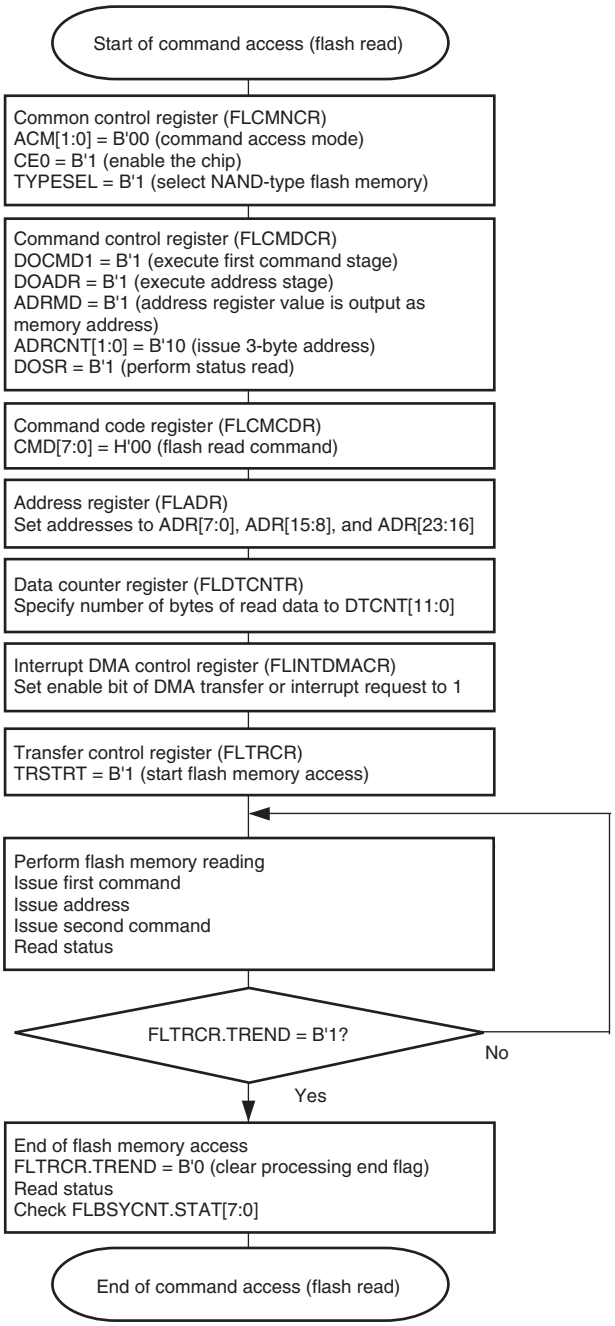


Figure 26.13 NAND Command Access (Flash Read)

26.6 Interrupt Processing

The FLCTL has six interrupt sources. Each of the interrupt sources has its corresponding interrupt flag. The interrupt request is generated independently if the interrupt is enabled by the interrupt enable bit. The status error and ready/busy timeout error use the common FLSTE interrupt.

Table 26.6 FLCTL Interrupt Requests

Interrupt Source	Interrupt Flag	Enable Bit	Description
FLSTE interrupt	STERB	STERINTE	Status error
	BTOERB	RBERINTE	Ready/busy timeout error
	ECERB	ECERINTE	ECC error
FLTEND interrupt	TREND	TEINTE	Transfer end
FLTRQ0 interrupt	TRREQF0	TRINTE0	FIFO0 transfer request
FLTRQ1 interrupt	TRREQF1	TRINTE1	FIFO1 transfer request

26.7 DMA Transfer Settings

The FLCTL can request DMA transfers separately to the data sector, FLDTFIFO, and control code sector, FLECFIFO. Table 26.7 shows whether DMA transfer is enabled or disabled in each access mode.

Table 26.7 DMA Transfer Settings

	Sector Access Mode	Command Access Mode
FLDTFIFO	Enabled	Enabled
FLECFIFO	Enabled	Disabled

For details on DMAC settings, see section 15, Direct Memory Access Controller0 (DMAC0).

Section 27 Audio Codec Interface (HAC)

The HAC, the audio codec digital controller interface, supports bidirectional data transfer compliant with Audio Codec 97 (AC'97) Version 2.1. The HAC provides serial transmission to/reception from the AC97 codec. Each channel of the HAC can be connected to a single audio codec device.

The HAC carries out data extraction from/insertion into audio frames. For data slots within both receive and transmit frames, the PIO transfer by the CPU or the DMA transfer by the DMAC can be used.

27.1 Features

The HAC has the following features:

- Supports digital interface to a single AC'97 version 2.1 Audio Codec
- PIO transfer of status slots 1 and 2 in RX frames
- PIO transfer of command slots 1 and 2 in TX frames
- PIO transfer of data slots 3 and 4 in RX frames
- PIO transfer of data slots 3 and 4 in TX frames
- Selectable 16-bit or 20-bit DMA transfer of data slots 3 and 4 in RX frames
- 16-bit DMA transfer of data slots 3 and 4 in TX frames
- Accommodates various sampling rates by qualifying slot data with tag bits and monitoring the TX frame request bits of RX frames
- Generates data ready, data request, overrun and underrun interrupts
- Supports cold reset, warm reset, and power-down mode

Figure 27.1 shows a block diagram of the HAC.

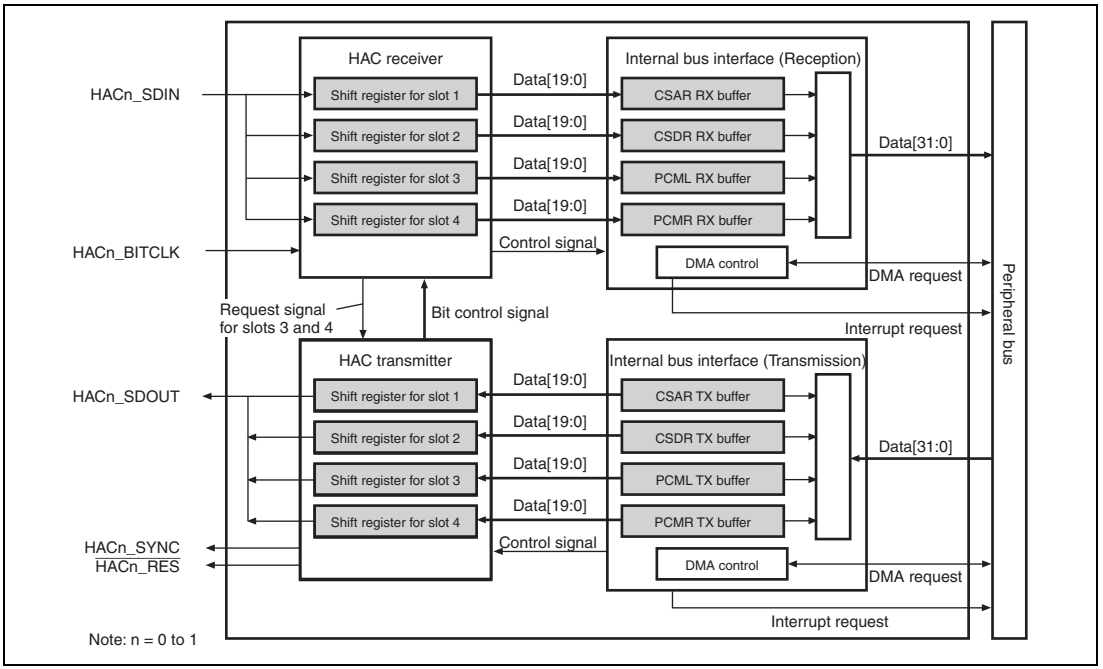


Figure 27.1 Block Diagram

27.2 Input/Output Pins

Table 27.1 describes the HAC pin configuration.

Table 27.1 Pin Configuration

Pin Name	Pin Count	I/O	Function
HAC0_BITCLK	1	Input	Serial data clock
HAC0_SDIN	1	Input	RX frame serial input data
HAC0_SDOUT	1	Output	TX frame serial output data
HAC0_SYNC	1	Output	Frame sync
HAC1_BITCLK	1	Input	Serial data clock
HAC1_SDIN	1	Input	RX frame serial input data
HAC1_SDOUT	1	Output	TX frame serial output data
HAC1_SYNC	1	Output	Frame sync
$\overline{\text{HAC\_RES}}$	1	Output	Reset (negative logic signal) (common to channels 0 and 1)

27.3 Register Descriptions

The following shows the HAC registers. In this manual, the registers are not discriminated by the channel.

Table 27.2 Register Configuration (1)

Channel	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Control and status register 0	HACCR0	R/W	H'FFE4 0008	H'1FE4 0008	32	Pck
0	Command/status address register 0	HACCSAR0	R/W	H'FFE4 0020	H'1FE4 0020	32	Pck
0	Command/status data register 0	HACCSDR0	R/W	H'FFE4 0024	H'1FE4 0024	32	Pck
0	PCM left channel register 0	HACPCML0	R/W	H'FFE4 0028	H'1FE4 0028	32	Pck
0	PCM right channel register 0	HACPCMR0	R/W	H'FFE4 002C	H'1FE4 002C	32	Pck
0	TX interrupt enable register 0	HACTIER0	R/W	H'FFE4 0050	H'1FE4 0050	32	Pck
0	TX status register 0	HACTSR0	R/W	H'FFE4 0054	H'1FE4 0054	32	Pck
0	RX interrupt enable register 0	HACRIER0	R/W	H'FFE4 0058	H'1FE4 0058	32	Pck
0	RX status register 0	HACRSR0	R/W	H'FFE4 005C	H'1FE4 005C	32	Pck
0	HAC control register 0	HACACR0	R/W	H'FFE4 0060	H'1FE4 0060	32	Pck
1	Control and status register 1	HACCR1	R/W	H'FFE5 0008	H'1FE5 0008	32	Pck
1	Command/status address register 1	HACCSAR1	R/W	H'FFE5 0020	H'1FE5 0020	32	Pck
1	Command/status data register 1	HACCSDR1	R/W	H'FFE5 0024	H'1FE5 0024	32	Pck
1	PCM left channel register 1	HACPCML1	R/W	H'FFE5 0028	H'1FE5 0028	32	Pck
1	PCM right channel register 1	HACPCMR1	R/W	H'FFE5 002C	H'1FE5 002C	32	Pck
1	TX interrupt enable register 1	HACTIER1	R/W	H'FFE5 0050	H'1FE5 0050	32	Pck
1	TX status register 1	HACTSR1	R/W	H'FFE5 0054	H'1FE5 0054	32	Pck
1	RX interrupt enable register 1	HACRIER1	R/W	H'FFE5 0058	H'1FE5 0058	32	Pck
1	RX status register 1	HACRSR1	R/W	H'FFE5 005C	H'1FE5 005C	32	Pck
1	HAC control register 1	HACACR1	R/W	H'FFE5 0060	H'1FE5 0060	32	Pck

Table 27.3 Register Configuration (2)

Channel	Register Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiple Exceptions	Sleep by SLEEP Instruction	Module Standby	Light Sleep
0	Control and status register 0	HACCR0	H'0000 0200	Retained	Retained	Retained	Retained
0	Command/status address register 0	HACCSAR0	H'0000 0000	Retained	Retained	Retained	Retained
0	Command/status data register 0	HACCSDR0	H'0000 0000	Retained	Retained	Retained	Retained
0	PCM left channel register 0	HACPCML0	H'0000 0000	Retained	Retained	Retained	Retained
0	PCM right channel register 0	HACPCMR0	H'0000 0000	Retained	Retained	Retained	Retained
0	TX interrupt enable register 0	HACTIER0	H'0000 0000	Retained	Retained	Retained	Retained
0	TX status register 0	HACTSR0	H'F000 0000	Retained	Retained	Retained	Retained
0	RX interrupt enable register 0	HACRIER0	H'0000 0000	Retained	Retained	Retained	Retained
0	RX status register 0	HACRSR0	H'0000 0000	Retained	Retained	Retained	Retained
0	HAC control register 0	HACACR0	H'8400 0000	Retained	Retained	Retained	Retained
1	Control and status register 1	HACCR1	H'0000 0200	Retained	Retained	Retained	Retained
1	Command/status address register 1	HACCSAR1	H'0000 0000	Retained	Retained	Retained	Retained
1	Command/status data register 1	HACCSDR1	H'0000 0000	Retained	Retained	Retained	Retained
1	PCM left channel register 1	HACPCML1	H'0000 0000	Retained	Retained	Retained	Retained
1	PCM right channel register 1	HACPCMR1	H'0000 0000	Retained	Retained	Retained	Retained
1	TX interrupt enable register 1	HACTIER1	H'0000 0000	Retained	Retained	Retained	Retained
1	TX status register 1	HACTSR1	H'F000 0000	Retained	Retained	Retained	Retained
1	RX interrupt enable register 1	HACRIER1	H'0000 0000	Retained	Retained	Retained	Retained
1	RX status register 1	HACRSR1	H'0000 0000	Retained	Retained	Retained	Retained
1	HAC control register 1	HACACR1	H'8400 0000	Retained	Retained	Retained	Retained

27.3.1 Control and Status Register (HACCR)

HACCR is a 32-bit read/write register for controlling input/output and monitoring the interface status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CR	—	—	—	CDRT	WMRT	—	—	—	—	ST	—	—	—	—	—
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	W	W	R	R	R	R	W	R	R	R	R	R

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	CR	0	R	Codec Ready 0: The HAC-connected codec is not ready. 1: The HAC-connected codec is ready.
14 to 12	—	All 0	R	Reserved These bits are always read as 0. Write prohibited.
11	CDRT	0	W	HAC Cold Reset Use a cold reset only after power-on, or only to exit from the power-down mode by the power-down command. [Write] 0: Always write 0 to this bit before writing 1 again. (When this bit is changed from 0 to 1, a cold reset is performed.) 1: Performs a cold reset on the HAC-connected codec. [Read] Always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
10	WMRT	0	W	HAC Warm Reset Use a warm reset only after power-up, or only to exit from the power-down mode by the power-down command. [Write] 0: Always write 0 to this bit before writing 1 again. (When this bit is changed from 0 to 1, a warm reset is performed.) 1: Performs a warm reset on the HAC-connected codec. [Read] Always read as 0.
9	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
8 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	ST	0	W	Start Transfer [Write] 0: Stops data transmission/reception at the end of the current frame. Do not take this action to terminate transmission/reception in normal operation. When terminating transmission/reception in normal operation, refer to the following description. 1: Starts data transmission/reception. [Read] Always read as 0.
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

To place the off-chip codec device into the power-down mode, write 1 to bit 12 of the register index 26 in the off-chip codec via the HAC. When entering the power-down mode, the off-chip codec stops HAC_BITCLK and suspends the normal operation. The off-chip codec acts in the same manner at power-on. To resume the normal operation, perform a cold reset or a warm reset on the off-chip codec.

27.3.2 Command/Status Address Register (HACCSAR)

HACCSAR is a 32-bit read/write register that specifies the address of the codec register to be read /written. When requesting a write to/read from a codec register, write the command register address to HACCSAR and set the ST bit in the HACCR register to 1. The HAC then transmits this register address to the codec via slot 1.

After the codec has responded to a read request (HACRSR.STARY = 1), the status address received via slot 1 can be read out from HACCSAR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RW	CA6/ SA6	CA5/ SA5	CA4/ SA4
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CA3/ SA3	CA2/ SA2	CA1/ SA1	CA0/ SA0	SLR EQ3	SLR EQ4	SLR EQ5	SLR EQ6	SLR EQ7	SLR EQ8	SLR EQ9	SLR EQ10	SLR EQ11	SLR EQ12	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	RW	0	R/W	Codec Read/Write Command 0: Notifies the off-chip codec device of a write access to the register specified in the address field (CA6/SA6 to CA0/SA0). Write the data to HACCSDR in advance. When HACACR.TX12_ATOMIC is 1, the HAC transmits HACCSAR and HACCSDR as a pair in the same TX frame. When HACACR.TX12_ATOMIC is 0, transmission of HACCSAR and HACCSDR in the same TX frame is not guaranteed. 1: Notifies the off-chip codec device of a read access to the register specified in the address field (CA6/SA6 to CA0/SA0).

Bit	Bit Name	Initial Value	R/W	Description
18	CA6/SA6	0	R/W	Codec Control Register Addresses 6 to 0/Codec Status Register Addresses 6 to 0
17	CA5/SA5	0	R/W	
16	CA4/SA4	0	R/W	[Write]
15	CA3/SA3	0	R/W	Specify the address of the codec register to be written.
14	CA2/SA2	0	R/W	
13	CA1/SA1	0	R/W	[Read]
12	CA0/SA0	0	R/W	Indicate the status address received via slot 1, corresponding to the codec register whose data has been returned in HACCSDR.
11	SLREQ3	0	R	Slot Requests 3 to 12
10	SLREQ4	0	R	Valid only in the RX frame. Indicate whether the codec is requesting slot data in the next TX frame. Automatically set by hardware, and correspond to bits 11 to 2 of slot 1 in the RX frame.
9	SLREQ5	0	R	
8	SLREQ6	0	R	
7	SLREQ7	0	R	0: Slot data is requested.
6	SLREQ8	0	R	1: Slot data is not requested.
5	SLREQ9	0	R	
4	SLREQ10	0	R	
3	SLREQ11	0	R	
2	SLREQ12	0	R	
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

27.3.3 Command/Status Data Register (HACCSDR)

HACCSDR is a 32-bit read/write data register used for accessing the codec register. Write the command data to HACCSDR and set the ST bit in the HACCR register to 1. The HAC then transmits the data to the codec via slot 2.

After the codec has responded to a read request (HACRSR.STDRY = 1), the status data received via slot 2 can be read out from HACCSDR. In both read and write, HACCSAR stores the related codec register address.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	CD15/SD15	CD14/SD14	CD13/SD13	CD12/SD12
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CD11/SD11	CD10/SD10	CD9/SD9	CD8/SD8	CD7/SD7	CD6/SD6	CD5/SD5	CD4/SD4	CD3/SD3	CD2/SD2	CD1/SD1	CD0/SD0	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	CD15/SD15	0	R/W	Command Data 15 to 0/Status Data 15 to 0 Write data to these bits and then write the codec register address in HACCSAR. The HAC then transmits the data to the codec. Read these bits to get the contents of the codec register indicated by HACCSAR.
18	CD14/SD14	0	R/W	
17	CD13/SD13	0	R/W	
16	CD12/SD12	0	R/W	
15	CD11/SD11	0	R/W	
14	CD10/SD10	0	R/W	
13	CD9/SD9	0	R/W	
12	CD8/SD8	0	R/W	
11	CD7/SD7	0	R/W	
10	CD6/SD6	0	R/W	
9	CD5/SD5	0	R/W	
8	CD4/SD4	0	R/W	
7	CD3/SD3	0	R/W	
6	CD2/SD2	0	R/W	
5	CD1/SD1	0	R/W	
4	CD0/SD0	0	R/W	
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

27.3.4 PCM Left Channel Register (HACPCML)

HACPCML is a 32-bit read/write register used for accessing the left channel of the codec in digital audio recording or stream playback. To transmit the PCM playback left channel data to the codec, write the data to HACPCML. To receive the PCM record left channel data from the codec, read HACPCML. The data is left justified when accommodating a codec with ADC/DAC resolution of 20 bits or less.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	D19	D18	D17	D16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 0	D19 to D0	All 0	R/W	Data 19 to 0 Write the PCM playback left channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record left channel data from the codec.

In 16-bit packed DMA mode, HACPCML is defined as follows:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LD15	LD14	LD13	LD12	LD11	LD10	LD9	LD8	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	LD15 to LD0	All 0	R/W	Left Data 15 to 0 Write the PCM playback left channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record left channel data from the codec.
15 to 0	RD15 to RD0	All 0	R/W	Right Data 15 to 0 Write the PCM playback right channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record right channel data from the codec.

27.3.5 PCM Right Channel Register (HACPCMR)

HACPCMR is a 32-bit read/write register used for accessing the right channel of the codec in digital audio recording or stream playback. To transmit the PCM playback right channel data to the codec, write the data to HACPCMR. To receive the PCM record right channel data from the codec, read HACPCMR. The data is left justified when accommodating a codec with ADC/DAC resolution of 20 bits or less.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	D19	D18	D17	D16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 0	D19 to D0	All 0	R/W	Data 19 to 0 Write the PCM playback right channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record right channel data from the codec.

27.3.6 TX Interrupt Enable Register (HACTIER)

HACTIER is a 32-bit read/write register that enables or disables HAC TX interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	PLTF RQIE	PRTF RQIE	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PLTF UNIE	PRTF UNIE	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29	PLTFRQIE	0	R/W	PCML TX Request Interrupt Enable 0: Disables PCML TX request interrupts. 1: Enables PCML TX request interrupts.
28	PRTFRQIE	0	R/W	PCMR TX Request Interrupt Enable 0: Disables PCMR TX request interrupts. 1: Enables PCMR TX request interrupts.
27 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PLTFUNIE	0	R/W	PCML TX Underrun Interrupt Enable 0: Disables PCML TX underrun interrupts. 1: Enables PCML TX underrun interrupts.
8	PRTFUNIE	0	R/W	PCMR TX Underrun Interrupt Enable 0: Disables PCMR TX underrun interrupts. 1: Enables PCMR TX underrun interrupts.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

27.3.7 TX Status Register (HACTSR)

HACTSR is a 32-bit read/write register that indicates the status of the HAC TX controller. Writing 0 to the bit will initialize it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMD AMT	CMD DMT	PLT FRQ	PRT FRQ	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PLT FUN	PRT FUN	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W*[2]	Description
31	CMDAMT	1	R/W	Command Address Empty 0: CSAR TX buffer contains untransmitted data. 1: CSAR TX buffer is empty and ready to store data.*[1]
30	CMDDMT	1	R/W	Command Data Empty 0: CSDR TX buffer contains untransmitted data. 1: CSDR TX buffer is empty and ready to store data.*[1]
29	PLTFRQ	1	R/W	PCML TX Request 0: PCML TX buffer contains untransmitted data. 1: PCML TX buffer is empty and needs to store data. In DMA mode, writing to HACPCML will automatically clear this bit to 0.
28	PRTFRQ	1	R/W	PCMR TX Request 0: PCMR TX buffer contains untransmitted data. 1: PCMR TX buffer is empty and needs to store data. In DMA mode, writing to HACPCMR will automatically clear this bit to 0.
27 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*[2]	Description
9	PLTFUN	0	R/W	PCML TX Underrun 0: No PCML TX underrun has occurred. 1: PCML TX underrun has occurred because the codec has requested slot 3 data but new data is not written to HACPCML.
8	PRTFUN	0	R/W	PCMR TX Underrun 0: No PCMR TX underrun has occurred. 1: PCMR TX underrun has occurred because the codec has requested slot 4 data but new data is not written to HACPCMR.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. CMDAMT and CMDDMT have no associated interrupts. Poll these bits until they are read as 1 before writing a new command to HACCSAR/HACCSDR. When bit 19 (RW) of HACCSAR is 0 and TX12_ATOMIC is 1, take the following steps:

a. Initialize CMDDMT and CMDAMT before first accessing a codec register after HAC initialization by any reset event.

b. After making the settings in HACCSDR and HACCSAR, poll CMDDMT and CMDAMT until they are cleared to 1, and then initialize these bits.

c. Now the next write to a register is available.

2. These bits are readable/writable. Writing 0 to the bit initializes it but writing 1 has no effect.

27.3.8 RX Interrupt Enable Register (HACRIER)

HACRIER is a 32-bit read/write register that enables or disables HAC RX interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	STARYIE	STDRYIE	PLRFRQIE	PRRFRQIE	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PLRFOVIE	PRRFOVIE	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22	STARYIE	0	R/W	Status Address Ready Interrupt Enable 0: Disables status address ready interrupts. 1: Enables status address ready interrupts.
21	STDRYIE	0	R/W	Status Data Ready Interrupt Enable 0: Disables status data ready interrupts. 1: Enables status data ready interrupts.
20	PLRFRQIE	0	R/W	PCML RX Request Interrupt Enable 0: Disables PCML RX request interrupts. 1: Enables PCML RX request interrupts.
19	PRRFRQIE	0	R/W	PCMR RX Request Interrupt Enable 0: Disables PCMR RX request interrupts. 1: Enables PCMR RX request interrupts.
18 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PLRFOVIE	0	R/W	PCML RX Overrun Interrupt Enable 0: Disables PCML RX overrun interrupts. 1: Enables PCML RX overrun interrupts.

Bit	Bit Name	Initial Value	R/W	Description
12	PRRFOVIE	0	R/W	PCMR RX Overrun Interrupt Enable 0: Disables PCMR RX overrun interrupts. 1: Enables PCMR RX overrun interrupts.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

27.3.9 RX Status Register (HACRSR)

HACRSR is a 32-bit read/write register that indicates the status of the HAC RX controller. Writing 0 to the bit will initialize it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	STARY	STDRY	PLR FRQ	PRR FRQ	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PLR FOV	PRR FOV	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W*	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22	STARY	0	R/W	Status Address Ready 0: HACCSAR (status address) is not ready. 1: HACCSAR (status address) is ready.
21	STDRY	0	R/W	Status Data Ready 0: HACCSDR (status data) is not ready. 1: HACCSDR (status data) is ready.
20	PLRFRQ	0	R/W	PCML RX Request 0: PCML RX data is not ready. 1: PCML RX data is ready and must be read. In DMA mode, reading HACPCML automatically clears this bit to 0.

Bit	Bit Name	Initial Value	R/W*	Description
19	PRRFRQ	0	R/W	PCMR RX Request 0: PCMR RX data is not ready. 1: PCMR RX data is ready and must be read. In DMA mode, reading HACPCMR automatically clears this bit to 0.
18 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PLRFOV	0	R/W	PCML RX Overrun 0: No PCML RX data overrun has occurred. 1: PCML RX data overrun has occurred because the HAC has received new data from slot 3 before PCML data is not read out.
12	PRRFOV	0	R/W	PCMR RX Overrun 0: No PCMR RX data overrun has occurred. 1: PCMR RX data overrun has occurred because the HAC has received new data from slot 4 before PCMR data is not read out.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * This register is readable/writable. Writing 0 to the bit initializes it but writing 1 has no effect.

27.3.10 HAC Control Register (HACACR)

HACACR is a 32-bit read/write register used for controlling the HAC interface.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	DMA RX16	DMA TX16	—	—	TX12_ ATOMIC	—	RXDMAL _EN	TXDMAL _EN	RXDMAR _EN	TXDMAR _EN	—	—	—	—	—
Initial value:	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
30	DMARX16	0	R/W	16-bit RX DMA Enable 0: Disables 16-bit packed RX DMA mode. Enables the RXDMAL_EN and RXDMAR_EN settings. 1: Enables 16-bit packed RX DMA mode. Disables the RXDMAL_EN and RXDMAR_EN settings.
29	DMATX16	0	R/W	16-bit TX DMA Enable 0: Setting prohibited. Set this bit to 1 and use 16-bit packed TX DMA mode. 1: Enables 16-bit packed TX DMA mode. Disables the TXDMAL_EN and TXDMAR_EN settings.
28, 27	—	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
26	TX12_ATOMIC	1	R/W	TX Slot 1 and 2 Atomic Control 0: Transmits TX data in HACCSAR and that in HACCSDR separately. (Setting prohibited) 1: Transmits TX data in HACCSAR and that in HACCSDR in the same frame if bit 19 in HACCSAR is 0 (write). (HACCSAR must be written last.)
25	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
24	RXDMAL_EN	0	R/W	RX DMA Left Enable 0: Disables 20-bit RX DMA for HACPCML. 1: Enables 20-bit RX DMA for HACPCML.
23	TXDMAL_EN	0	R/W	TX DMA Left Enable 0: Disables 20-bit TX DMA for HACPCML. 1: Setting prohibited.
22	RXDMAR_EN	0	R/W	RX DMA Right Enable 0: Disables 20-bit RX DMA for HACPCMR. 1: Enables 20-bit RX DMA for HACPCMR.
21	TXDMAR_EN	0	R/W	TX DMA Right Enable 0: Disables 20-bit TX DMA for HACPCMR. 1: Setting prohibited.
20 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

27.4 AC 97 Frame Slot Structure

Figure 27.2 shows the AC97 frame slot structure. This LSI supports slots 0 to 4 only. Slots 5 to 12 (hatched area) are out of scope.

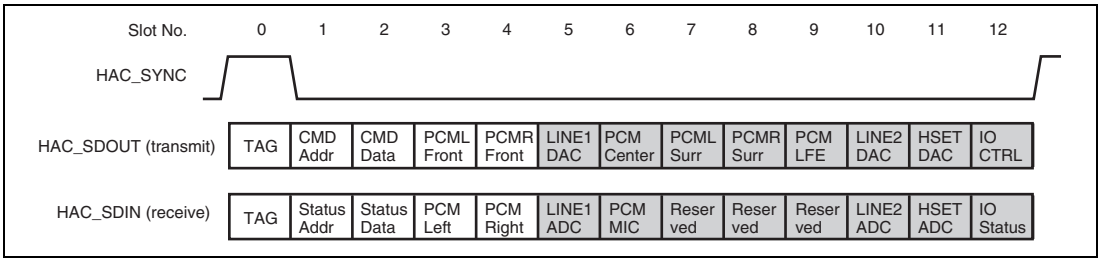


Figure 27.2 AC97 Frame Slot Structure

Table 27.4 AC97 Transmit Frame Structure

Slot	Name	Description
0	SDATA_OUT TAG	Codec IDs and Tags indicating valid data
1	Control CMD Addr write port	Read/write command and register address
2	Control DATA write port	Register write data
3	PCM L DAC playback	Left channel PCM output data
4	PCM R DAC playback	Right channel PCM output data
5	Modem Line 1 DAC	Modem 1 output data (unsupported)*
6	PCM Center	Center channel PCM data (unsupported)*
7	PCM Surround L	Surround left channel PCM data (unsupported)*
8	PCM Surround R	Surround right channel PCM data (unsupported)*
9	PCM LFE	LFE channel PCM data (unsupported)*
10	Modem Line 2 DAC	Modem 2 output data (unsupported)*
11	Modem handset DAC	Modem handset output data (unsupported)*
12	Modem IO control	Modem control IO output (unsupported)*

Note: * There is no register for unsupported functions.

Table 27.5 AC97 Receive Frame Structure

Slot	Name	Description
0	SDATA_IN TAG	Tags indicating valid data
1	Status ADDR read port	Register address and slot request
2	Status DATA read port	Register read data
3	PCM L ADC record	Left channel PCM input data
4	PCM R ADC record	Right channel PCM input data
5	Modem Line 1 ADC	Modem 1 input data (unsupported)*
6	Dedicated Microphone ADC	Optional PCM data (unsupported)*
7 to 9	Reserved	Reserved
10	Modem Line 2 ADC	Modem 2 input data (unsupported)*
11	Modem handset input DAC	Modem handset input data (unsupported)*
12	Modem IO status	Modem control IO input (unsupported)*

Note: * There is no register for unsupported functions.

27.5 Operation

27.5.1 Receiver

The HAC receiver receives serial audio data input on the HAC_SDIN pin, synchronous to HAC_BITCLK. From slot 0, the receiver extracts tag bits that indicate which other slots contain valid data. It will update the receive data only when receiving valid slot data indicated by the tag bits.

Supporting data only in slots 1 to 4, the receiver ignores tag bits and data related to slots 5 to 12. It loads valid slot data to the corresponding shift register to hold the data for PIO or DMA transfer, and sets the corresponding status bits. It is possible to read 20-bit data within a 32-bit register using PIO.

In the case of RX overrun, the new data will overwrite the current data in the RX buffer of the HAC.

27.5.2 Transmitter

The HAC transmitter outputs serial audio data on the HAC_SDOUT pin, synchronous to HAC_BIT_CLK. The transmitter sets the tag bits in slot 0 to indicate which slots in the current frame contain valid data. It loads data slots to the current TX frame in response to the corresponding slot request bits from the previous RX frame.

The transmitter supports data only in slots 1 to 4. The TX buffer holds data that has been transferred using PIO or DMA, and sets the corresponding status bit. It is possible to write 20-bit data within a 32-bit register using PIO.

In the case of a TX underrun, the HAC will transmit the current TX buffer data until the next data arrives.

27.5.3 DMA

The HAC supports DMA transfer for slots 3 and 4 of both the RX and TX frames. Specify the DMA transfer slot data size for RX frames as 16 or 20 bits with the DMARX16 bit in HACACR. The slot data size for TX frames must be 16 bits, so clear bit DMATX16 in HACACR to 0.

When the data size is 20 bits, transfer of data slots 3 and 4 requires two local bus access cycles. Since each of the receiver and transmitter has its DMA request, the stereo mode generates a DMA request for slots 3 and 4 separately. The mono mode generates a DMA request for just one slot.

When the data size is 16 bits, data from slots 3 and 4 are packed into a single 32-bit quantity (left data and right data are in PCML), which requires only one external bus access cycle.

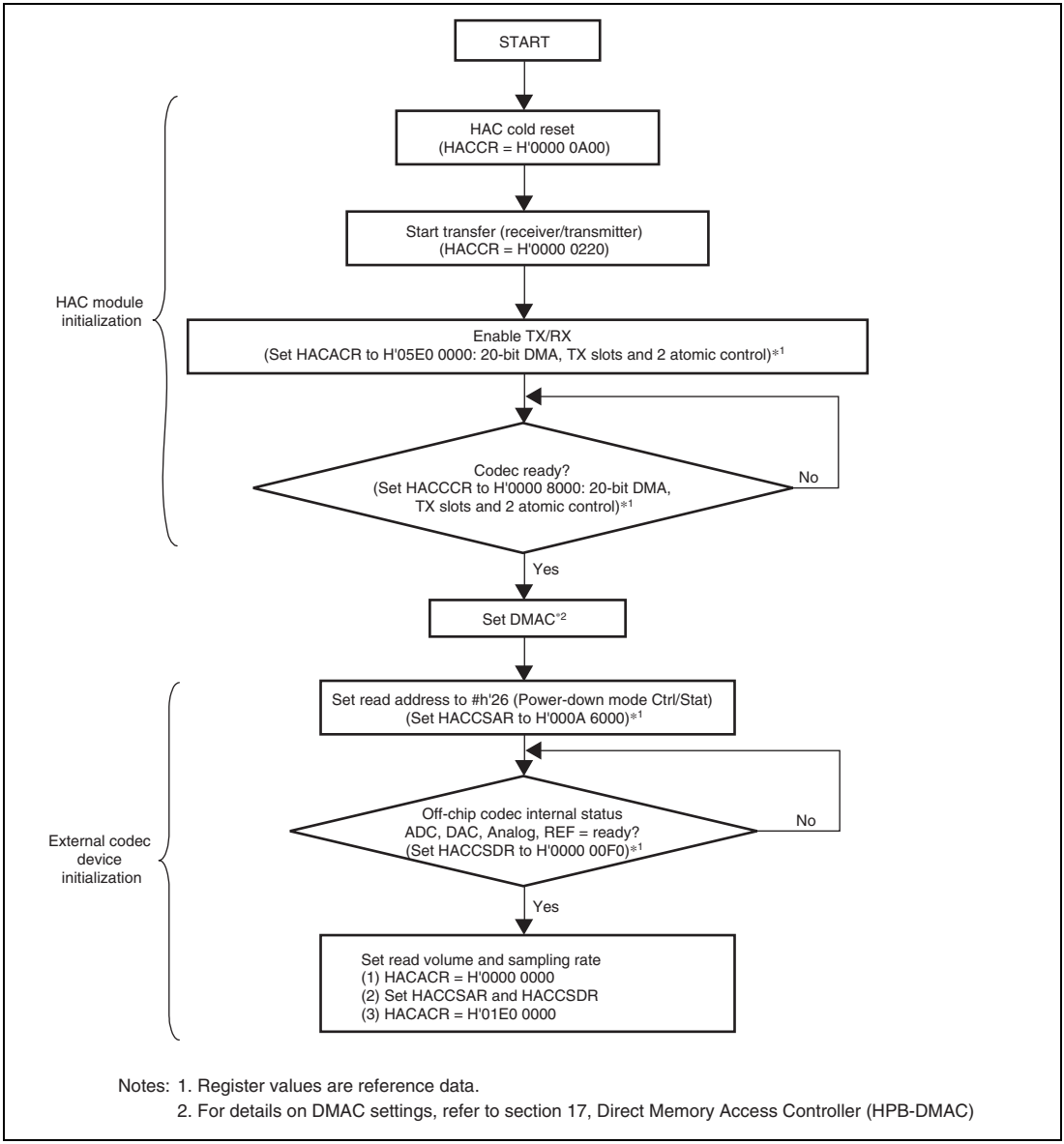
It may be necessary to halt a DMA transfer before the end count is reached, depending on system applications. If so, clear the corresponding DMA bit in HACACR to 0 (DMA disabled). To resume a DMA transfer, reprogram the DMAC and then set the corresponding DMA bit to 1 (DMA enabled).

27.5.4 Interrupts

Interrupts can be used for flag events from the receiver and transmitter. Make the setting for each interrupt in the corresponding interrupt enable register. Interrupts include a request to the CPU to read/write slot data, overrun and underrun. To get the interrupt source, read the status register. Writing 0 to the bit will clear the corresponding interrupt.

27.5.5 Initialization Sequence

Figure 27.3 shows an example of the initialization sequence.



Notes: 1. Register values are reference data.
2. For details on DMAC settings, refer to section 17, Direct Memory Access Controller (HPB-DMAC)

Figure 27.3 Initialization Sequence

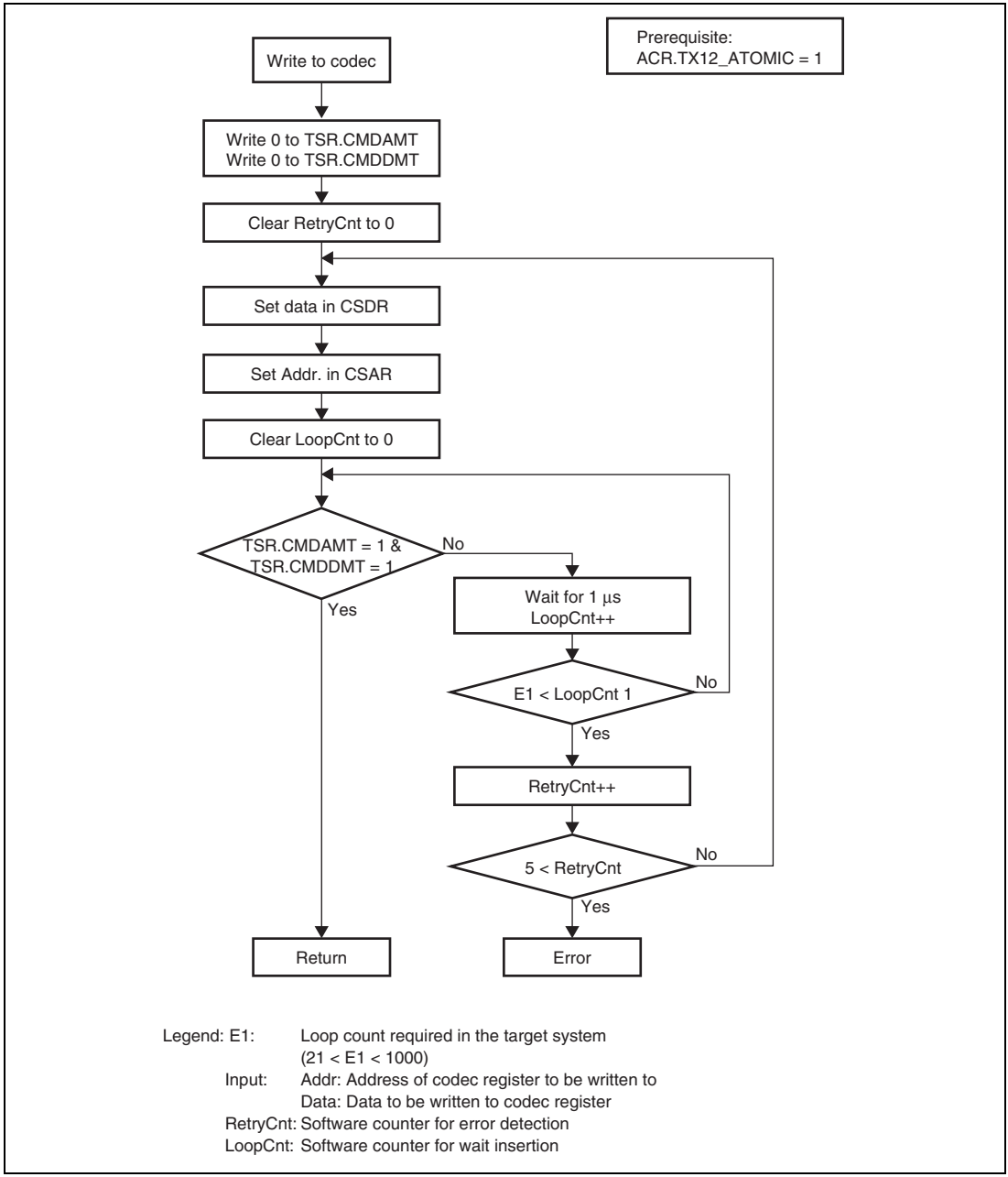


Figure 27.4 Sample Flowchart for Off-Chip Codec Register Write

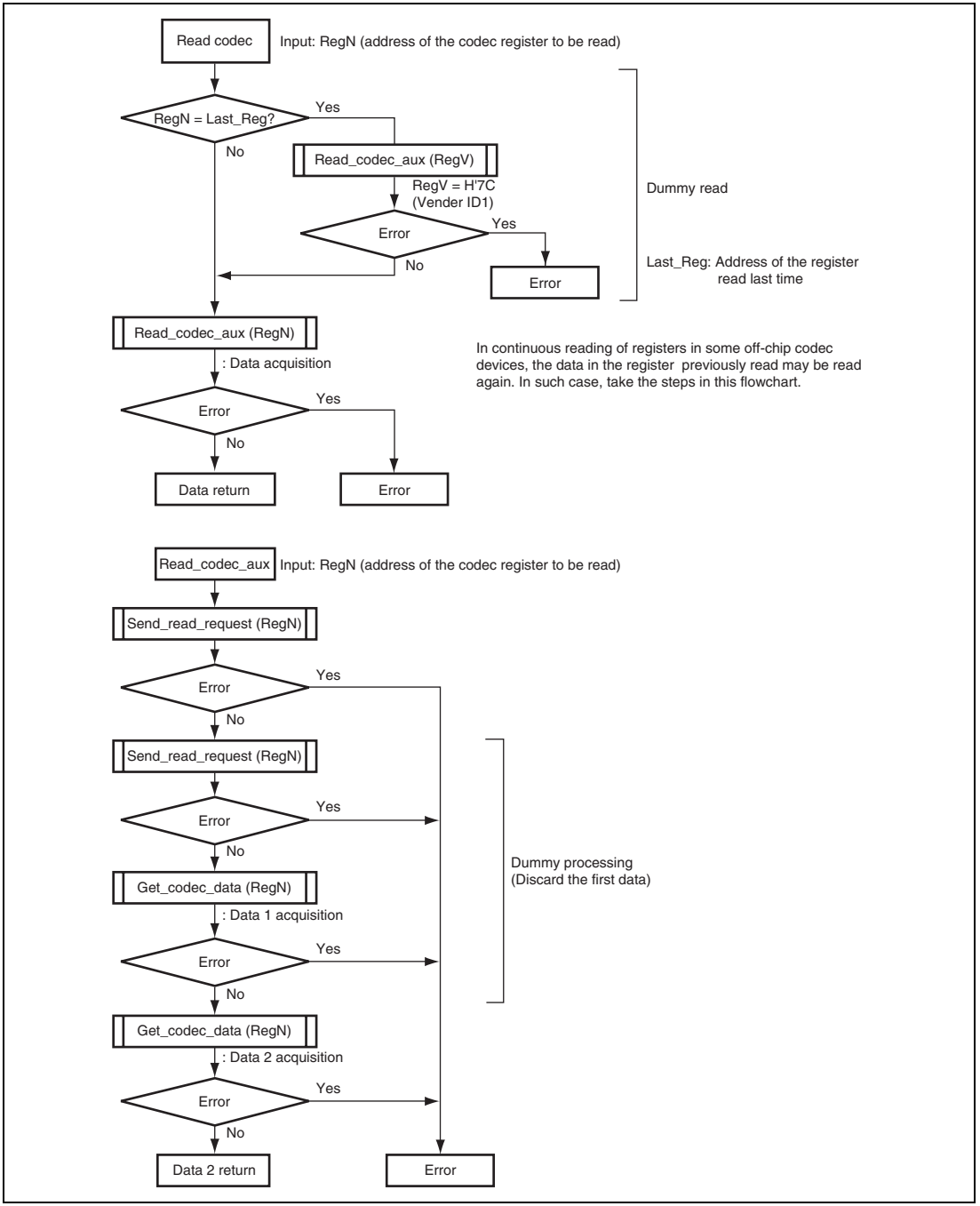


Figure27.5 Sample Flowchart for Off-Chip Codec Register Read (1)

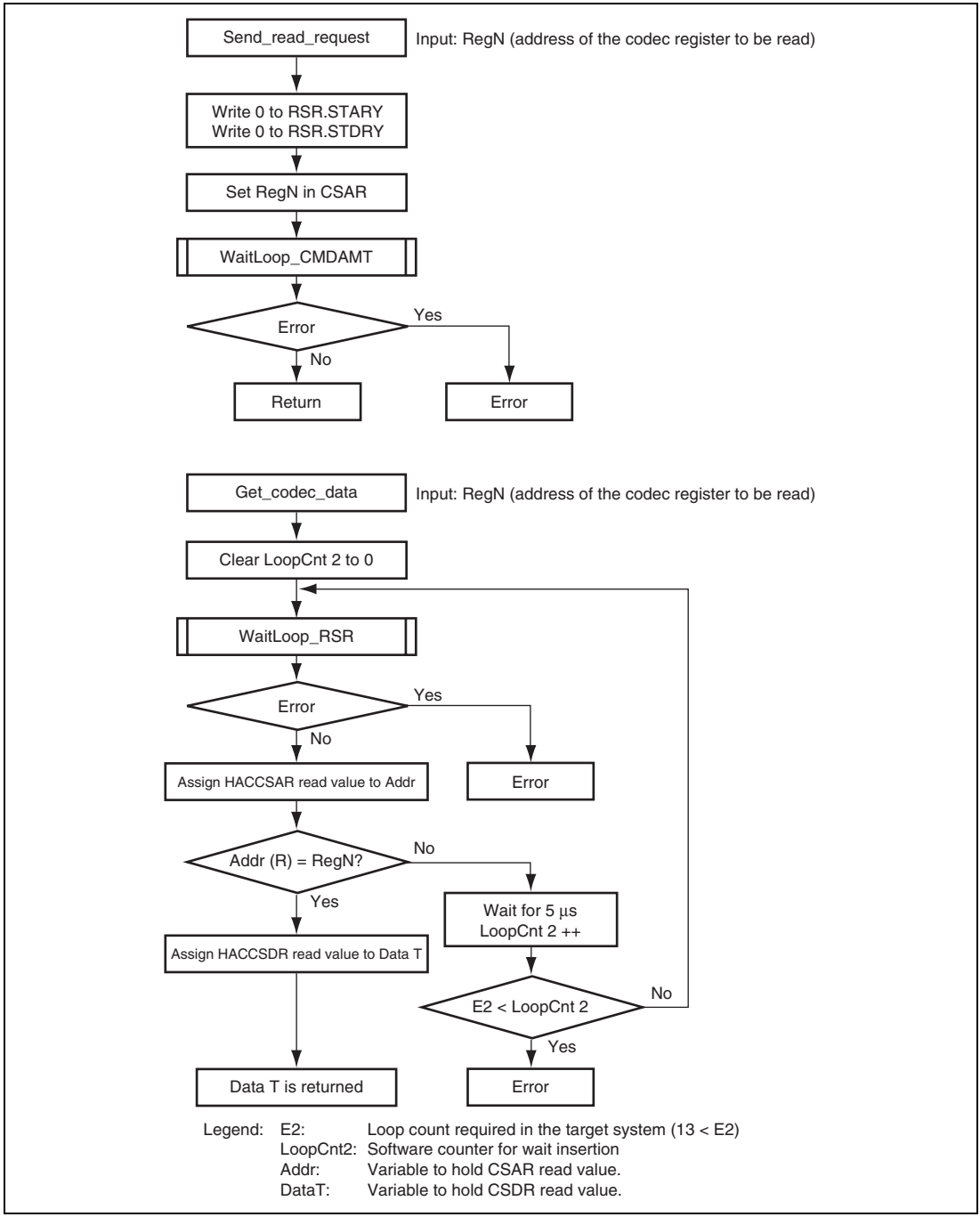
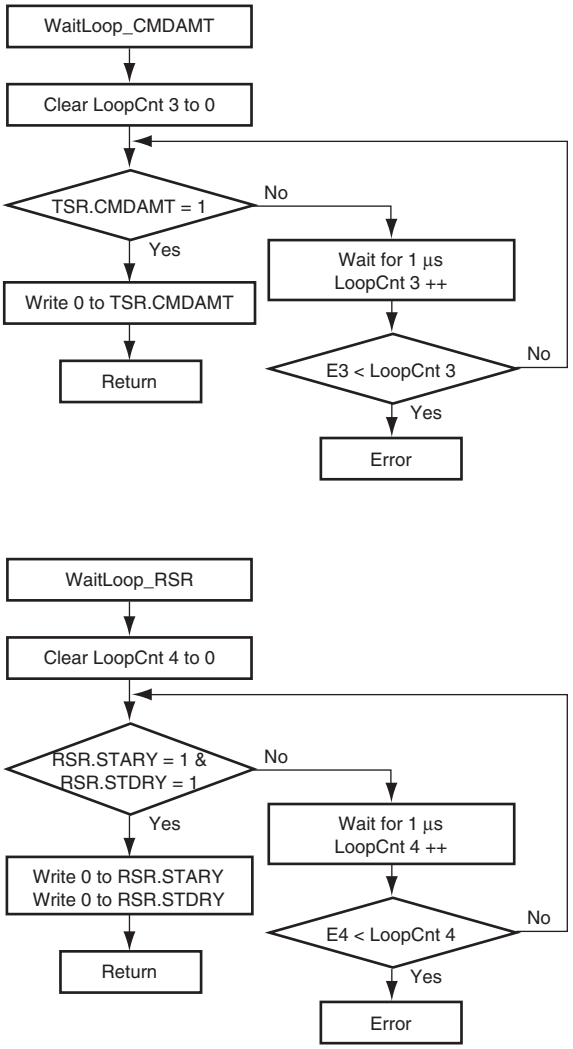


Figure 27.6 Sample Flowchart for Off-Chip Codec Register Read (2)



Legend: E3, E4: Loop count required in the target system
(21 < E3, 21 < E4 < 1000)
LoopCnt3: Software counter for wait insertion
LoopCnt4: Software counter for wait insertion

Figure 27.7 Sample Flowchart for Off-Chip Codec Register Read (3)

27.5.6 Power-Down Mode

Power-Down mode has following modes or functions.

(1) Sleep mode

The operation of HAC is continued.

(2) Light Sleep mode

The HAC continues operation, but the operation of receiving/transmitting using the DMAC is not guaranteed. In such cases, following methods are needed, before enter light sleep mode

i. Check all data transfers are finished.
   Both the transmitting data and the receiving data should be empty. So they are necessary to wait until all bits of HACTSR[31:28] (TX status register) are set to 1. And it is also necessary to wait until HACRSR[22] and HACRSR[21] are set to 1 or read out the HACPCMR/HACPCML (receive data) until HACRSR[20] and HACRSR[19] are set to 1.
ii. Disable all DMA requests from the HAC.
iii. Write a 0 to the ST bit in the HACCR (control status register) to conclude the transfer.

(3) Module standby mode

If the HAC enters module standby mode while operating, correct operation is not guaranteed. , Following procedure is needed before enter module standby mode.

i. Check that all data transfers are finished.
   The procedure is same as the case of light sleep mode.
ii. Disable all DMA and interrupt requests from the HAC.
iii. Set the state of the codec to power-down mode.
iv. Write 0 to the ST bit in HACCR (control status register) to finish the transfer.
v. Set the MSTP16 and MSTP17 bit in MSTPCR0 (standby control register) to 1.
   After releasing module standby mode, registers should be set again according to the procedure when using the HAC module again.

(4) Changing clock frequency

When clock frequency (Pck) is changed while the HAC is operating, the operation is continued. But there is a possibility that underflow or overflow occurs by frequency change. So before changing the clock frequency while the HAC is operating, following procedure is needed.

i. Check that all data transfers are finished.
   The procedure is same as the case of light sleep mode.
ii. Disable all DMA and interrupt requests from the HAC.
iii. Set the state of the codec to power-down mode.
iv. Write 0 to the ST bit in the HACCR (control status register) to conclude the transfer.

After changing the clock frequency, registers should be set again according to the procedure when using the HAC module again.

27.5.7 Notes

The HAC_SYNC signal is generated by the HAC to indicate the position of slot 0 within a frame. When using the two channels of the HAC concurrently, connect the $\overline{\text{HAC\_RES}}$ pin to the reset pins of the two codecs.

27.5.8 Reference

AC'97 Component Specification, Revision 2.1

Section 28 Serial Sound Interface (SSI) Module

This LSI incorporates four channels of serial sound interface (SSI) modules that send or receive audio data to or from a variety of devices. In addition to the common formats, they support burst and multi-channel mode.

28.1 Features

The SSI has the following features.

- Number of channels: Four channels
- Operating modes: Compressed mode and non-compressed mode
  The compressed mode is used for continuous bit stream transfer
  The non-compressed mode supports all serial audio streams divided into channels.
- The SSI module is configured as any of a transmitter or receiver. The serial bus format (refer to table 28.3) can be used in the compressed and non-compressed mode.
- Asynchronous transfer between the data buffer and the shift register
- Division ratios of the serial bus interface clock can be selected.
- Data transmission/reception can be controlled from the DMAC or SSI interrupt.

Figure 28.1 is a block diagram of the SSI module.

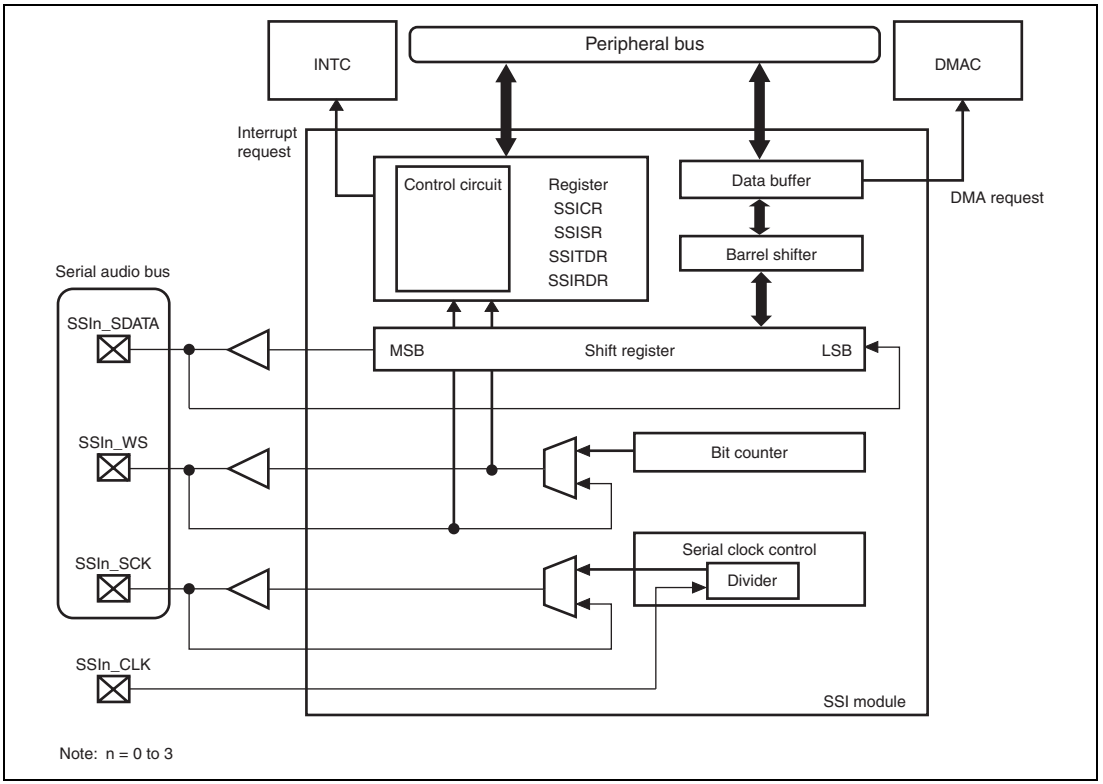


Figure 28.1 Block Diagram of SSI Module

28.2 Input/Output Pins

Table 28.1 lists the pin configurations relating to the SSI module.

Table 28.1 Pin Configuration

Name	Number of Pins	I/O	Function
SSIn_SCK	1	I/O	Serial bit clock
SSIn_WS	1	I/O	Word select
SSIn_SDATA	1	I/O	Serial data input/output
SSIn_CLK	1	Input	Divider input clock (oversampling clock 256/384/512fs input)

Note: n = 0 to 3

28.3 Register Descriptions

The SSI has the following registers. In this manual, the register description is not discriminated by the channel.

Table 28.2 Register Configuration (1)

Channel	Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Control register 0	SSICR0	R/W	H'FFE0 0000	H'1FE0 0000	32	Pck
0	Status register 0	SSISR0	R/W*	H'FFE0 0004	H'1FE0 0004	32	Pck
0	Transmit data register 0	SSITDR0	R/W	H'FFE0 0008	H'1FE0 0008	32	Pck
0	Receive data register 0	SSIRDR0	R	H'FFE0 000C	H'1FE0 000C	32	Pck
1	Control register 1	SSICR1	R/W	H'FFE1 0000	H'1FE1 0000	32	Pck
1	Status register 1	SSISR1	R/W*	H'FFE1 0004	H'1FE1 0004	32	Pck
1	Transmit data register 1	SSITDR1	R/W	H'FFE1 0008	H'1FE1 0008	32	Pck
1	Receive data register 1	SSIRDR1	R	H'FFE1 000C	H'1FE1 000C	32	Pck
2	Control register 2	SSICR2	R/W	H'FFE2 0000	H'1FE2 0000	32	Pck
2	Status register 2	SSISR2	R/W*	H'FFE2 0004	H'1FE2 0004	32	Pck
2	Transmit data register 2	SSITDR2	R/W	H'FFE2 0008	H'1FE2 0008	32	Pck
2	Receive data register 2	SSIRDR2	R	H'FFE2 000C	H'1FE2 000C	32	Pck
3	Control register 3	SSICR3	R/W	H'FFE3 0000	H'1FE3 0000	32	Pck
3	Status register 3	SSISR3	R/W*	H'FFE3 0004	H'1FE3 0004	32	Pck
3	Transmit data register 3	SSITDR3	R/W	H'FFE3 0008	H'1FE3 0008	32	Pck
3	Receive data register 3	SSIRDR3	R	H'FFE3 000C	H'1FE3 000C	32	Pck

Note: Do not write to other than the above address. Otherwise, the write operation cannot be guaranteed. The value when read may be undefined.

* Bits 26 and 27 of this register is readable/writable. The other bits are read only. For details, see section 28.3.2, Status Register (SSISR).

Table 28.3 Register Configuration (2)

Channel	Register Name	Abbr.	Power-on Reset by $\overline{\text{PRESET}}$ Pin/ WDT/H-UDI	Manual Reset by WDT/ Multiple Exception	Sleep by SLEEP Instruction	Module Standby	Light Sleep
0	Control register 0	SSICR0	H'0000 0000	Retained	Retained	Retained	Retained
0	Status register 0	SSISR0	H'xxxx xxxx	Retained	Retained	Retained	Retained
0	Transmit data register 0	SSITDR0	H'0000 0000	Retained	Retained	Retained	Retained
0	Receive data register 0	SSIRDR0	H'0000 0000	Retained	Retained	Retained	Retained
1	Control register 1	SSICR1	H'0000 0000	Retained	Retained	Retained	Retained
1	Status register 1	SSISR1	H'xxxx xxxx	Retained	Retained	Retained	Retained
1	Transmit data register 1	SSITDR1	H'0000 0000	Retained	Retained	Retained	Retained
1	Receive data register 1	SSIRDR1	H'0000 0000	Retained	Retained	Retained	Retained
2	Control register 2	SSICR2	H'0000 0000	Retained	Retained	Retained	Retained
2	Status register 2	SSISR2	H'xxxx xxxx	Retained	Retained	Retained	Retained
2	Transmit data register 2	SSITDR2	H'0000 0000	Retained	Retained	Retained	Retained
2	Receive data register 2	SSIRDR2	H'0000 0000	Retained	Retained	Retained	Retained
3	Control register 3	SSICR3	H'0000 0000	Retained	Retained	Retained	Retained
3	Status register 3	SSISR3	H'xxxx xxxx	Retained	Retained	Retained	Retained
3	Transmit data register 3	SSITDR3	H'0000 0000	Retained	Retained	Retained	Retained
3	Receive data register 3	SSIRDR3	H'0000 0000	Retained	Retained	Retained	Retained

28.3.1 Control Register (SSICR)

SSICR is a 32-bit readable/writable register that controls interrupts, selects each polarity status, and sets operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN	CHNL1	CHNL0	DWL2	DWL1	DWL0	SWL2	SWL1	SWL0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	BREN	CKDV2	CKDV1	CKDV0	MUEN	CPEN	TRMD	EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	DMEN	0	R/W	DMA Enable Enables or disables the DMA request. 0: DMA request disabled. 1: DMA request enabled.
27	UIEN	0	R/W	Underflow Interrupt Enable 0: Underflow interrupt disabled 1: Underflow interrupt enabled
26	OIEN	0	R/W	Overflow Interrupt Enable 0: Overflow interrupt disabled 1: Overflow interrupt enabled
25	IIEN	0	R/W	Idle Mode Interrupt Enable 0: Idle mode interrupt disabled 1: Idle mode interrupt enabled
24	DIEN	0	R/W	Data Interrupt Enable 0: Data interrupt disabled 1: Data interrupt enabled

Bit	Bit Name	Initial Value	R/W	Description
23	CHNL1	0	R/W	Channels
22	CHNL0	0	R/W	These bits indicate the number of channels in each system word. These bits are ignored if CPEN = 1. 00: 1 channel per system word 01: 2 channels per system word 10: 3 channels per system word 11: 4 channels per system word
21	DWL2	0	R/W	Data Word Length
20	DWL1	0	R/W	These bits indicate the number of bits in a data word. These bits are ignored if CPEN = 1.
19	DWL0	0	R/W	000: 8 bits 001: 16 bits 010: 18 bits 011: 20 bits 100: 22 bits 101: 24 bits 110: 32 bits 111: Setting prohibited
18	SWL2	0	R/W	System Word Length
17	SWL1	0	R/W	These bits indicate the number of bits in a system word. These bits are ignored if CPEN = 1.
16	SWL0	0	R/W	000: 8 bits 001: 16 bits 010: 24 bits 011: 32 bits 100: 48 bits 101: 64 bits 110: 128 bits 111: 256 bits
15	SCKD	0	R/W	Serial Bit Clock Direction 0: Serial bit clock input, slave mode 1: Serial bit clock output, master mode

Bit	Bit Name	Initial Value	R/W	Description
14	SWSD	0	R/W	Serial WS Direction 0: Serial word select input, slave mode 1: Serial word select output, master mode
13	SCKP	0	R/W	Serial Bit Clock Polarity 0: SSI_WS and SSI_SDATA change on falling edge of SSI_SCK (sampled on rising edge of SCK) 1: SSI_WS and SSI_SDATA change on rising edge of SSI_SCK (sampled on falling edge of SCK)
12	SWSP	0	R/W	Serial WS Polarity The function of this bit depends on whether the SSI module is in non-compressed mode or compressed mode. CPEN = 0 (Non compressed mode): 0: SSI_WS is low for the first system word, high for the second system word 1: SSI_WS is high for the first system word, low for the second system word CPEN = 1 (Compressed mode): 0: SSI_WS is active high flow control. WS = high means data should be transferred, low means data should not be transferred. 1: SSI_WS is active low flow control. WS = low means data should be transferred, high means data should not be transferred. Note: Do not change this bit when EN is 1.

	SCKP = 0	SCKP = 1
SSI_SDATA input sampling timing in receive mode (TRMD = 0)	SSI_SCK rising edge	SSI_SCK falling edge
SSI_SDATA output change timing in transmit mode (TRMD = 1)	SSI_SCK falling edge	SSI_SCK rising edge
SSI_WS input sampling timing in slave mode (SWSD = 0)	SSI_SCK rising edge	SSI_SCK falling edge
SSI_WS output change timing in master mode (SWSD = 1)	SSI_SCK falling edge	SSI_SCK rising edge

Bit	Bit Name	Initial Value	R/W	Description
11	SPDP	0	R/W	Serial Padding Polarity This bit is ignored if CPEN = 1. 0: Padding bits are low 1: Padding bits are high When MUEN = 1, the padding bits are low level. (The MUTE function takes precedence.)
10	SDTA	0	R/W	Serial Data Alignment This bit is ignored if CPEN = 1. 0: Serial data is transmitted/ received first, followed by padding bits. 1: Padding bits are transmitted/ received first, followed by serial data.
9	PDTA	0	R/W	Parallel Data Alignment This bit is ignored if CPEN = 1. If the data word length = 32, 16 or 8 then this bit has no meaning. This bit is applied to SSIRDR in receive mode and to SSITDR in transmit mode. 0: Parallel data (SSITDR or SSIRDR) is left aligned 1: Parallel data (SSITDR or SSIRDR) is right aligned • DWL = 000 (data word length: 8 bits), PDTA ignored All data bits in SSIRDR or SSITDR are used on the audio serial bus. Four data words are transmitted/received in each 32-bit access. The first data word is derived from bits 7 to 0, the second from bits 15 to 8, the third from bits 23 to 16 and the last data word is stored in bits 31 to 24. • DWL = 001 (data word length: 16 bits), PDTA ignored All data bits in SSIRDR or SSITDR are used on the audio serial bus. Two data words are transmitted/received in each 32-bit access. The first data word is derived from bits 15 to 0 and the second data word is stored in bits 31 to 16.

Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	• DWL = 010, 011, 100, 101 (data word length: 18, 20, 22 and 24 bits), PDTA = 0 (left aligned) • The data bits which are used in SSIRDR or SSITDR are the following: Bits 31 to (32 – number of bits having data word length specified by DWL). If DWL = 011 then data word length is 20 bits and bits 31 to 12 are used of either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 010, 011, 100, 101 (data word length: 18, 20, 22 and 24 bits), PDTA = 1 (right aligned) The data bits which are used in SSIRDR or SSITDR are the following: Bits (number of bits having data word length specified by DWL - 1) to 0. If DWL = 011 then data word length is 20 bits and bits 19 to 0 are used of either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 110 (data word length: 32 bits), PDTA ignored All data bits in SSIRDR or SSITDR are used on the audio serial bus.
8	DEL	0	R/W	Serial Data Delay When CPEN = 1, set this bit to 1. 0: 1 clock cycle delay between SSI_WS and SSI_SDATA 1: No delay between SSI_WS and SSI_SDATA
7	BREN	0	R/W	Burst Mode Enable 0: Burst mode is disabled. 1: Burst mode is enabled. When CPEN = 0, clear this bit to 0. Burst mode is used only in compressed mode (CPEN = 1) and transmit mode. When burst mode is enabled the SSI_SCK signal is gated. Clock pulses are output only when there is valid serial data being output on SSI_SDATA.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	CKDV2	0	R/W	Serial Oversampling Clock Division Ratio
	CKDV1	0	R/W	These bits define the division ratio between
	CKDV0	0	R/W	oversampling clock (SSI_CLK) and the serial bit clock (SSI_SCK). These bits are ignored if SCKD = 0. The serial bit clock is used for the shift register and is provided on the SSI_SCK pin. 000: (Serial bit clock frequency = oversampling clock frequency/1) 001: (Serial bit clock frequency = oversampling clock frequency/2) 010: (Serial bit clock frequency = oversampling clock frequency/4) 011: (Serial bit clock frequency = oversampling clock frequency/8) 100: (Serial bit clock frequency = oversampling clock frequency/16) 101: (Serial bit clock frequency = oversampling clock frequency/6) 110: (Serial bit clock frequency = oversampling clock frequency/12) 111: Setting prohibited
3	MUEN	0	R/W	Serial Data Disabled Bit 0: The SSI module is not muted 1: The SSI module is muted Note: This bit can be used to stop output (low-level output) or to enable output, but there is no synchronization with changes in the SSI_WS signal.
2	CPEN	0	R/W	Compressed Mode Enable 0: Compressed mode disabled 1: Compressed mode enabled Note: In compressed mode (CPEN=1), use operation mode other than slave transmitter (SWSD=0 and TRMD=1). Do not change this bit when EN = 1.

Bit	Bit Name	Initial Value	R/W	Description
1	TRMD	0	R/W	Transmit/Receive Mode Select 0: The SSI module is in receive mode 1: The SSI module is in transmit mode
0	EN	0	R/W	SSI Module Enable 0: The SSI module is disabled 1: The SSI module is enabled

28.3.2 Status Register (SSISR)

SSISR is configured by status flags that indicate the operating status of the SSI module and bits that indicate the current channel number and word number.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ	—	—	—	—	—	—	—	—
Initial value:	—	—	—	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R/W*	R/W*	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CHNO1	CHNO0	SWNO	IDST
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	0	0	1	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	0 or 1	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
28	DMRQ	0	R	DMA Request Status Flag This status flag allows the CPU to see the status of the DMA request of SSI module. TRMD = 0 (Receive Mode): • If DMRQ = 1 then SSIRDR has unread data. • If SSIRDR is read then DMRQ = 0 until there is new unread data. TRMD = 1 (Transmit Mode): • If DMRQ = 1, SSITDR requests data to be written to continue the transmission onto the audio serial bus. • Once data is written to SSITDR then DMRQ = 0 until further transmit data is requested.

Bit	Bit Name	Initial Value	R/W	Description
27	UIRQ	0	R/W *	Underflow Error Interrupt Status Flag This status flag indicates that the data has been supplied at a lower rate than the required rate. This bit is set to 1 regardless of the setting of UIEN bit. In order to clear it to 0, write 0 in it. If UIRQ = 1 and UIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): If UIRQ = 1, it indicates that SSIRDR was read out before DMRQ and DIRQ bits would indicate the existence of new unread data. In this instance, the same received data may be stored twice by the host, which can lead to destruction of multi-channel data. When TRMD = 1 (Transmit Mode): If UIRQ = 1, it indicates that the transmitted data was not written in SSITDR. By this, the same data may be transmitted one time too often, which can lead to destruction of multi-channel data. Consequently, erroneous SSI data will be output, which makes this error more serious than underflow in the receive mode. Note: When underflow error occurs, the data in the data buffer will be transmitted until the next data is written in.

Bit	Bit Name	Initial Value	R/W	Description
26	OIRQ	0	R/W *	Overflow Error Interrupt Status Flag This status flag indicates that the data has been supplied at a higher rate than the required rate. This bit is set to 1 regardless of the setting of OIEN bit. In order to clear it to 0, write 0 in it. If OIRQ = 1 and OIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): If OIRQ = 1, it indicates that the previous unread data had not been read out before new unread data was written in SSIRDR. This may cause the loss of data, which can lead to destruction of multi-channel data. When TRMD = 1 (Transmit Mode): If OIRQ = 1, it indicates that SSITDR had data written in before the data in SSITDR was transferred to the shift register. This may cause the loss of data, which can lead to destruction of multi-channel data. Note: When overflow error occurs, the data in the data buffer will be overwritten by the next data sent from the SSI interface.
25	IIRQ	0	R	Idle Mode Interrupt Status Flag This status flag indicates whether the SSI module is in the idle status. This bit is set to 1 regardless of the setting of IIEN bit, so that polling will be possible. The interrupt can be masked by clearing the IIEN bit to 0, but writing 0 in this bit will not clear the interrupt. If IIRQ = 1 and IIEN = 1, then an interrupt will be generated. 0: The SSI module is not in the idle status. 1: The SSI module is in the idle status. In the idle state, the serial bus operation is stopped after the SSI module is activated.

Bit	Bit Name	Initial Value	R/W	Description
24	DIRQ	0	R	Data Interrupt Status Flag This status flag indicates that the SSI module requires that data be either read out or written in. This bit is set to 1 regardless of the setting of DIEN bit, so that polling will be possible. The interrupt can be masked by clearing DIEN bit to 0, but writing 0 in this bit will not clear the interrupt. If DIRQ = 1 and DIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): 0: No unread data exists in SSIRDR. 1: Unread data exists in SSIRDR. When TRMD = 1 (Transmit Mode): 0: The transmit buffer is full. 1: The transmit buffer is empty, and requires that data be written in SSITDR.
23 to 4	—	0 or 1	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
3	CHNO1	0	R	Channel Number The number indicates the current channel. They have no significance when the data word length is set to 8 or 16 bits. 00: 0 channels 01: 1 channel 10: 2 channels 11: 3 channels When TRMD = 0 (Receive Mode): This bit indicates to which channel the current data in SSIRDR belongs. When the data in SSIRDR is updated by transfer from the shift register, this value will change. When TRMD = 1 (Transmit Mode): This bit indicates the data of which channel should be written in SSITDR. When data is copied to the shift register, regardless whether the data is written in SSITDR, this value will change.
2	CHNO0	0	R	

Bit	Bit Name	Initial Value	R/W	Description
1	SWNO	1	R	Serial Word Number The number indicates the current word number. They have no significance when the data word length is set to 8 or 16 bits. When TRMD = 0 (Receive Mode): This bit indicates which system word the current data in SSIRDR is. Regardless whether the data has been read out from SSIRDR, when the data in SSIRDR is updated by transfer from the shift register, this value will change. When TRMD = 1 (Transmit Mode): This bit indicates which system word should be written in SSITDR. When data is copied to the shift register, regardless whether the data is written in SSITDR, this value will change.
0	IDST	0 or 1	R	Idle Mode Status Flag Indicates that the serial bus activity has ceased. This bit is cleared if EN = 1 and the serial bus is currently active. This bit can be set to 1 automatically under the following conditions. SSI = Serial bus master transmitter (SWSD = 1 and TRMD = 1): This bit is set to 1 if no more data has been written to SSITDR and the current system word has been completed. It can also be set to 1 by clearing the EN bit after sufficient data has been written to SSITDR to complete the system word currently being output. SSI = Serial bus master receiver (SWSD = 1 and TRMD = 0): This bit is set to 1 if the EN bit is cleared and the current system word is completed. SSI = slave transmitter/ receiver (SWSD = 0): This bit is set to 1 if the EN bit is cleared and the current system word is completed. Note: If the external device stops the serial bus clock before the current system word is completed then this bit will never be set.

Note: * These bits are readable/writable bits. If writing 0, these bits are initialized, although writing 1 is ignored.

28.3.3 Transmit Data Register (SSITDR)

SSITDR is a 32-bit register that stores data to be transmitted.

Data written to SSITDR is transferred to the shift register as it is required for transmission. If the data word length is less than 32 bits then its alignment should be as defined by the PDTA control bit in SSICR.

Reading this register will return the data in the buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

28.3.4 Receive Data Register (SSIRDR)

SSIRDR is a 32-bit register that stores the received data.

Data in SSIRDR is transferred from the shift register as each data word is received. If the data word length is less than 32 bits then its alignment should be as defined by the PDTA control bit in SSICR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

28.4 Operation

28.4.1 Bus Format

The SSI module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode.

The bus formats can be one of eight major modes as shown in table 28.3.

Table 28.3 Bus Formats of SSI Module

Bus Format	TPMD	CPEN	SCKD	SWSD	EN	MUEN	DIEN	IIEN	OIEN	UIEN	DEL	PDTA	SDTA	SPDP	SWSP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-Compressed Slave Receiver	0	0	0	0	Control bits						Configuration bits								
Non-Compressed Slave Transmitter	1	0	0	0															
Non-Compressed Master Receiver	0	0	1	1															
Non-Compressed Master Transmitter	1	0	1	1															
Compressed Slave Receiver	0	1	0/1	0	Control bits						Ignored				Configu-ration bits		Ignored		
Compressed Master Receiver	0	1	0/1	1															
Compressed Master Transmitter	1	1	0/1	1															

28.4.2 Non-Compressed Modes

The non-compressed mode is designed to support all serial audio streams which are split into channels. It can support Philips, Sony and Matsushita modes as well as many more variants on these modes.

(1) Slave Receiver

This mode allows the SSI module to receive serial data from another device. The clock and word select signals used for the serial data stream are also supplied from an external device. If these signals do not conform to the format as specified in the SSI module then operation is not guaranteed.

(2) Slave Transmitter

This mode allows the SSI module to transmit serial data to another device. The clock and word select signals used for the serial data stream are also supplied from an external device. If these signals do not conform to the format as specified in the SSI module then operation is not guaranteed.

(3) Master Receiver

This mode allows the SSI module to receive serial data from another device. The clock and word select signals are internally derived from the SSI_CLK input clock. The format of these signals is as defined in the SSI module. If the incoming data does not conform to the defined format then operation is not guaranteed.

(4) Master Transmitter

This mode allows the SSI module to transmit serial data to another device. The clock and word select signals are internally derived from the SSI_CLK input clock. The format of these signals is as defined in the configuration bits in the SSI module.

(5) Configuration Fields (Word Length Related)

All configuration bits relating to the word length of SSICR are valid in non-compressed modes.

There are many configurations that the SSI module can support and it is not sensible to show all of the serial data formats in this document. Some of the combinations are shown below for the popular formats by Philips, Sony, and Matsushita.

1. Philips Format

Figures 28.2 and 28.3 show the supported Philips protocol both with padding and without. Padding occurs when the data word length is smaller than the system word length.

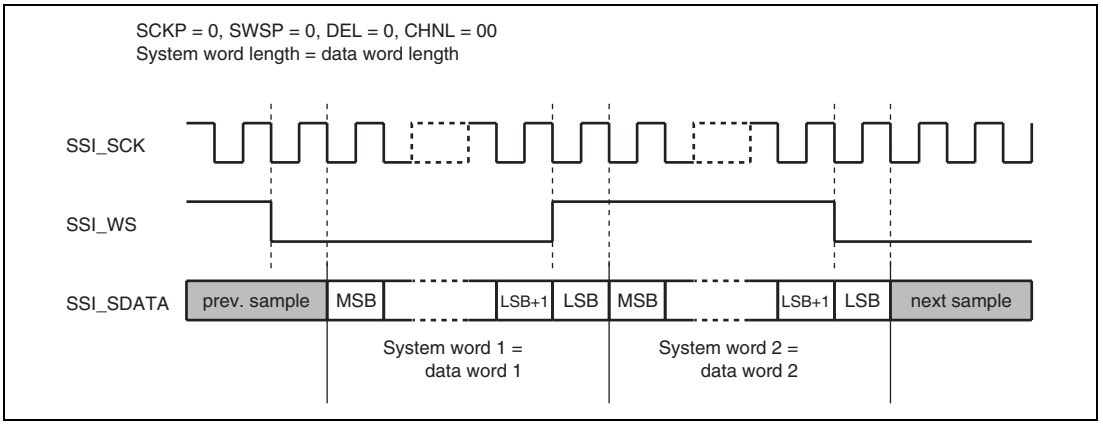


Figure 28.2 Philips Format (with no Padding)

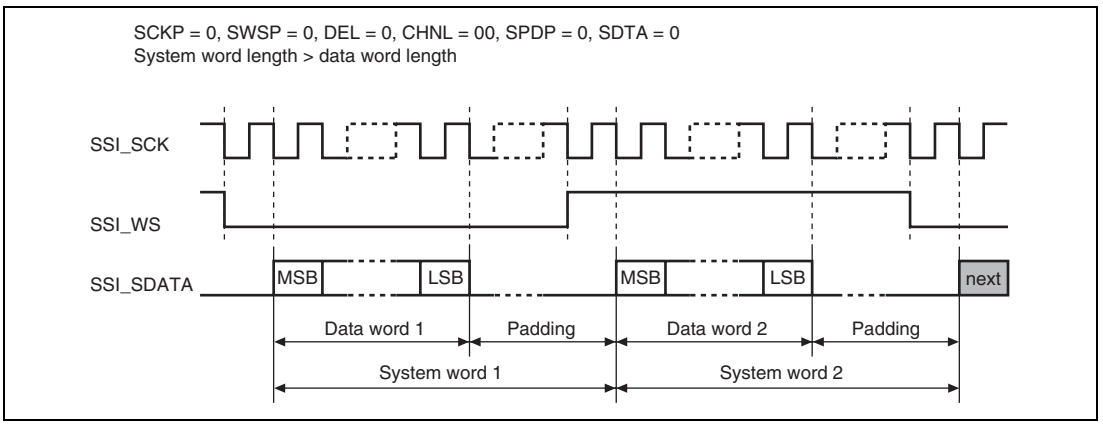


Figure 28.3 Philips Format (with Padding)

Figure 28.4 shows the format used by Sony. Figure 28.5 shows the format used by Matsushita. Padding is assumed in both cases, but may not be present in a final implementation if the system word length equals the data word length.

2. Sony Format

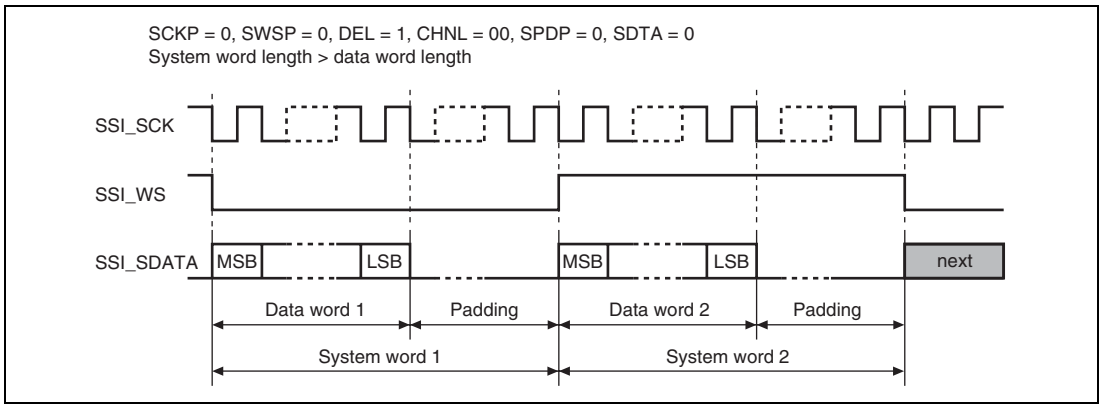


Figure 28.4 Sony Format (with Serial Data First, Followed by Padding Bits)

3. Matsushita Format

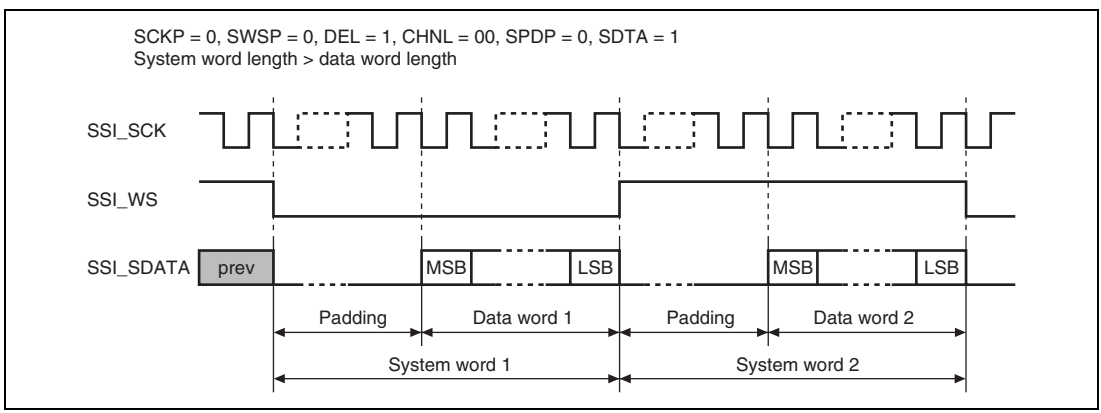


Figure 28.5 Matsushita Format (with Padding Bits First, Followed by Serial Data)

(6) Multi-Channel Formats

There is an extend format of the definition of the specification by Philips and allows more than 2 channels to be transferred within two system words.

The SSI module supports the transfer of 4, 6 and 8 channels by the use of the CHNL, SWL and DWL bits. It is important that the system word length (SWL) is greater than or equal to the number of channels (CHNL) times the data word length (DWL).

Table 28.4 shows the number of padding bits for each of the valid configurations. If a setup is not valid it does not have a number in the following table and has instead a dash.

Table 28.4 Number of Padding Bits for Each Valid Configuration

Padding Bits Per System Word			DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

In the case of the SSI module configured as a transmitter then each word that is written to SSITDR is transmitted in order on the serial audio bus.

In the case of the SSI module configured as a receiver each word received on the serial audio bus is presented for reading in order by SSIRDR.

Figures 28.6 to 28.8 show how 4, 6 and 8 channels are transferred on the serial audio bus.

Note that there are no padding bits in the first example, serial data is transmitted/received first and followed by padding bits in the second example, and padding bits are transmitted/received first and followed by serial data in the third example. This selection is purely arbitrary.

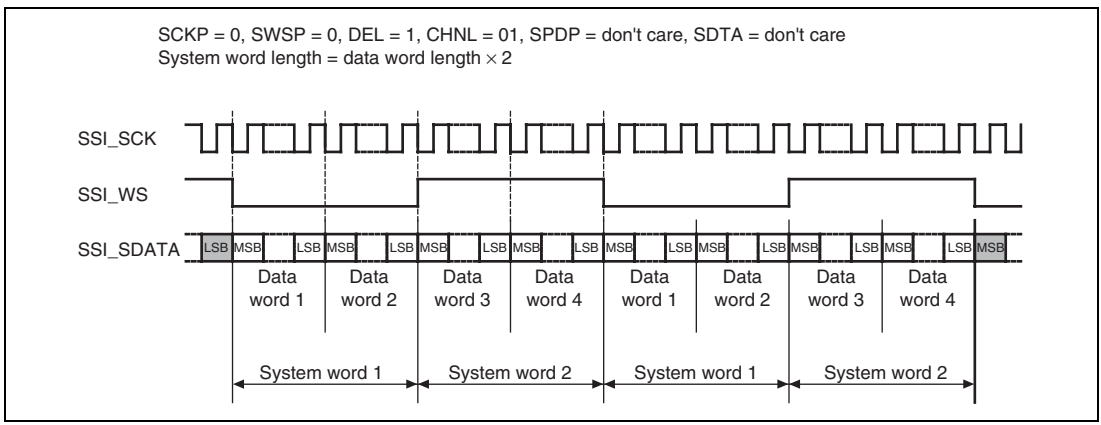


Figure 28.6 Multi-channel Format (4 Channels, No Padding)

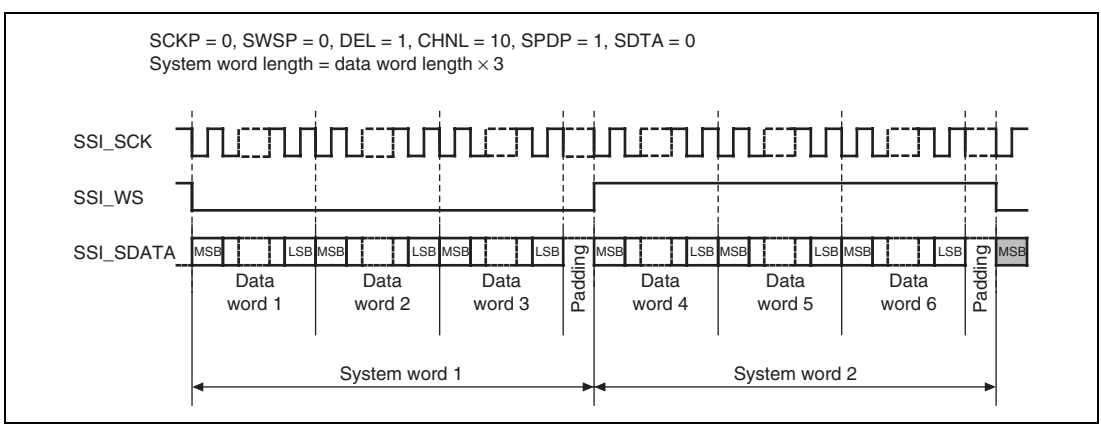


Figure 28.7 Multi-channel Format (6 Channels with High Padding)

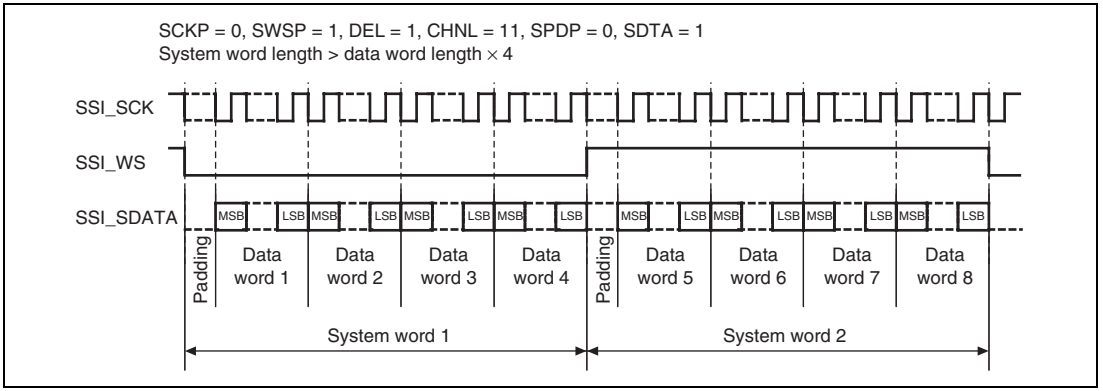


Figure 28.8 Multi-channel Format (8 Channels, Serial Data First, Followed by Padding Bits, with Padding)

(7) Configuration Fields - Signal Format Fields

There are several more configuration bits in non-compressed mode which will now be demonstrated. These bits are NOT mutually exclusive, however some configurations will probably not be useful for any other device.

They are demonstrated by referring to the following basic sample format shown in figure 28.9.

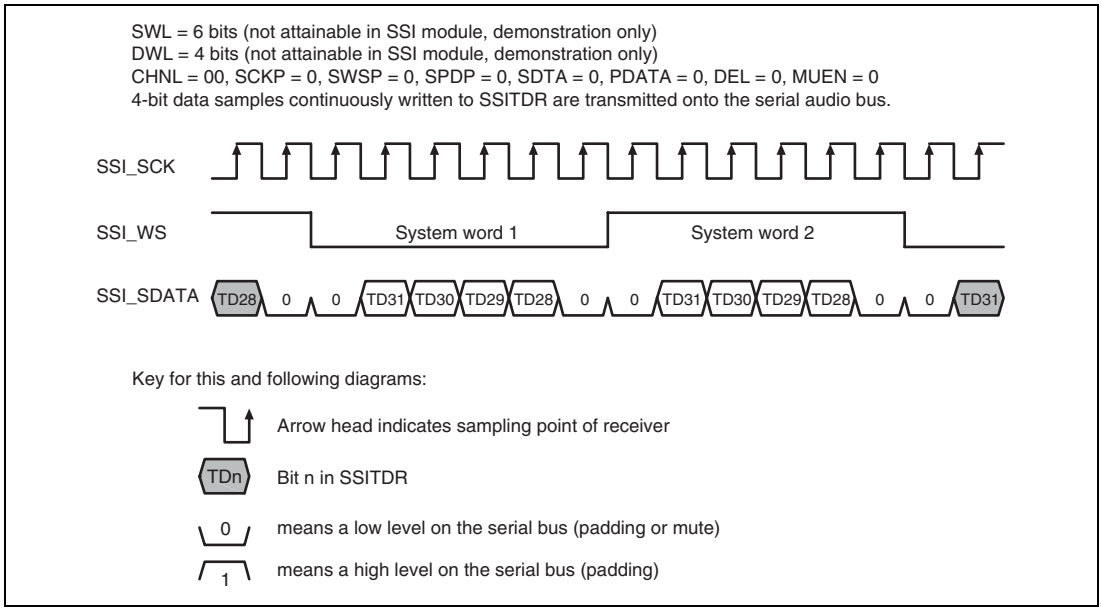


Figure 28.9 Basic Sample Format
(Transmit Mode with Example System/Data Word Length)

In figure 28.9, system word length of 6 bits and a data word length of 4 bits are used. Neither of these are possible with the SSI module but are used only for clarification of the other configuration bits.

1. Inverted Clock

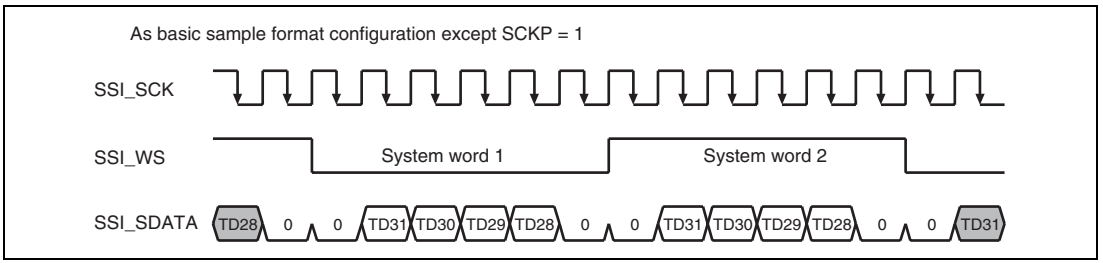


Figure 28.10 Inverted Clock

2. Inverted Word Select

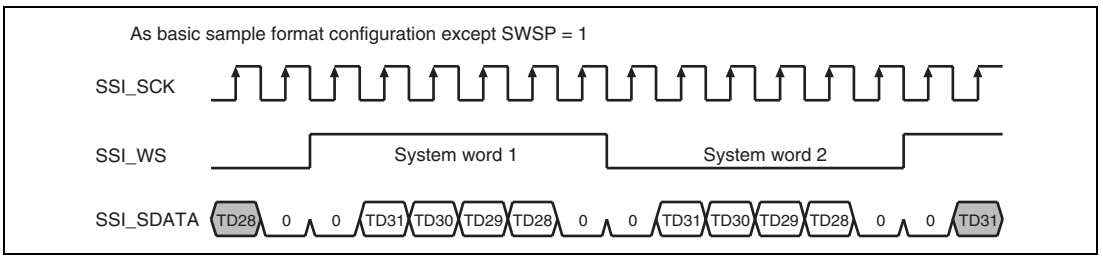


Figure 28.11 Inverted Word Select

3. Inverted Padding Polarity

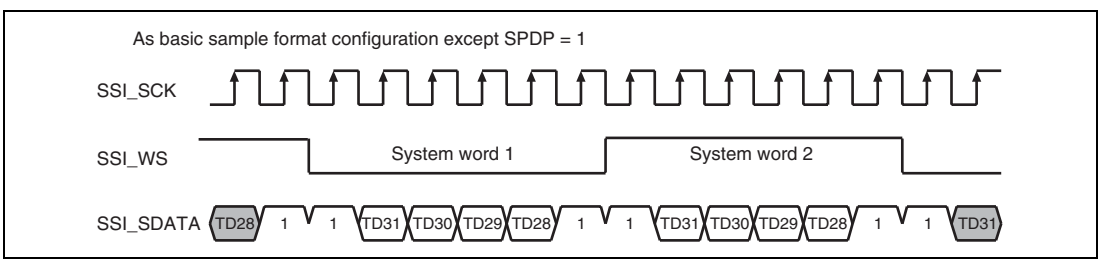


Figure 28.12 Inverted Padding Polarity

4. Padding Bits First, Followed by Serial Data, with Delay

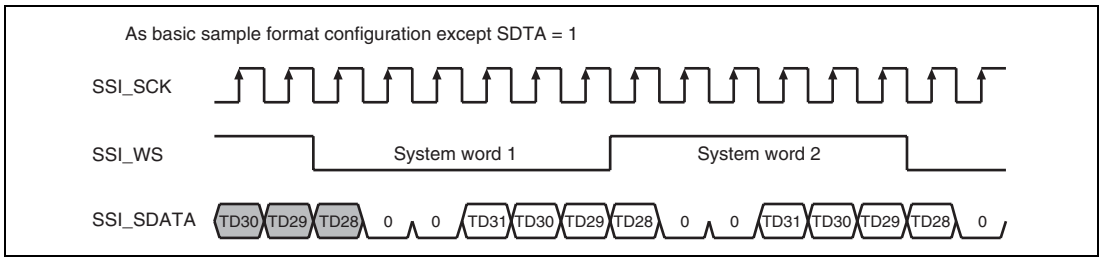


Figure 28.13 Padding Bits First, Followed by Serial Data, with Delay

5. Padding Bits First, Followed by Serial Data, without Delay

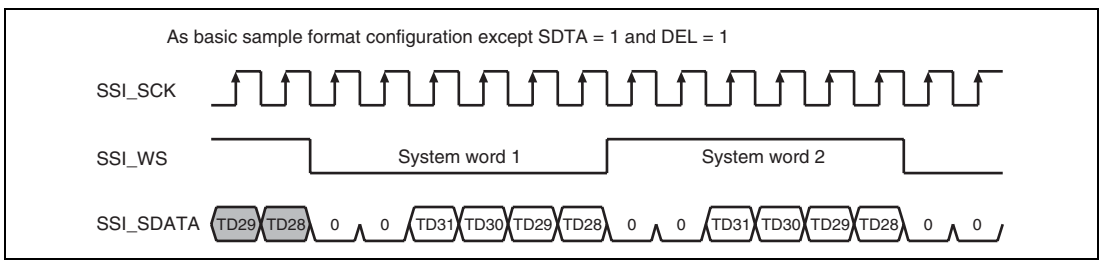


Figure 28.14 Padding Bits First, Followed by Serial Data, without Delay

6. Serial Data First, Followed by Padding Bits, without Delay

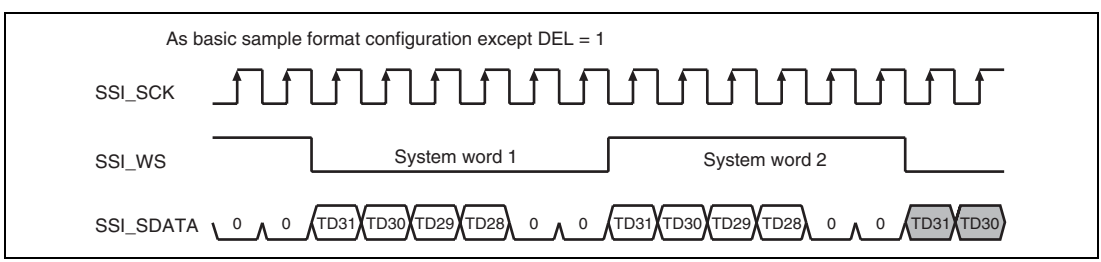


Figure 28.15 Serial Data First, Followed by Padding Bits, without Delay

7. Parallel Right Aligned with Delay

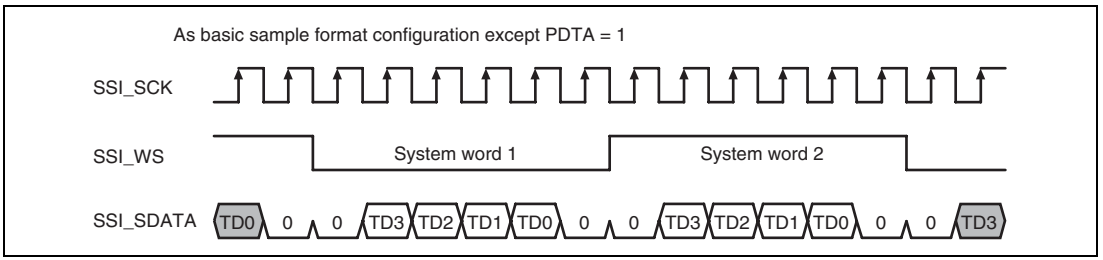


Figure 28.16 Parallel Right Aligned with Delay

8. Mute Enabled

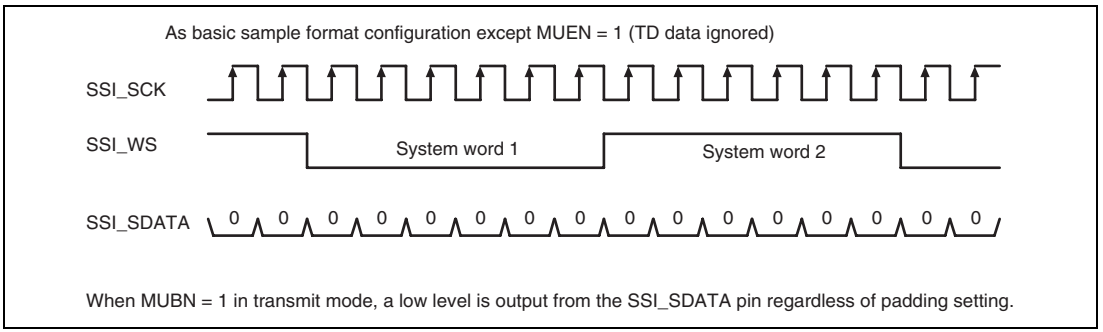


Figure 28.17 Mute Enabled

28.4.3 Compressed Modes

The compressed mode is used to transfer a continuous bit stream. This would typically be a compressed bit stream which requires downstream decoding.

When burst mode is not enabled, there is no concept of a data word. However in order to receive and transmit it is necessary to transfer between the serial bus and word formatted memory. Therefore the word boundary selection is arbitrary during receive/transmit and must be dealt with by another module. When burst mode is enabled then data bits being transmitted can be identified by virtue of the fact that the serial bit clock output is only activated when there is a word to be output and only the required number of clock pulses necessary to clock out each 32-bit word are generated. The serial bit clock stops at a low level when SSICR.SCKP = 0, and at a high level when SSICR.SCKP = 1. Note burst mode is only valid in the context of the SSI module being a transmitter of data. Burst mode data cannot be received by this module.

Data is transmitted and received in blocks of 32 bits, and the first bit received/transmitted bit is bit 31 when stored in memory.

The word select pin in this mode does not act as a system word start signal as in non-compressed mode, but instead is used to indicate that the receiver can receive another data burst, or the transmitter can transmit another data burst.

Figures 28.18 and 28.19 show the compressed mode data transfer, with burst mode disabled, and enabled, respectively.

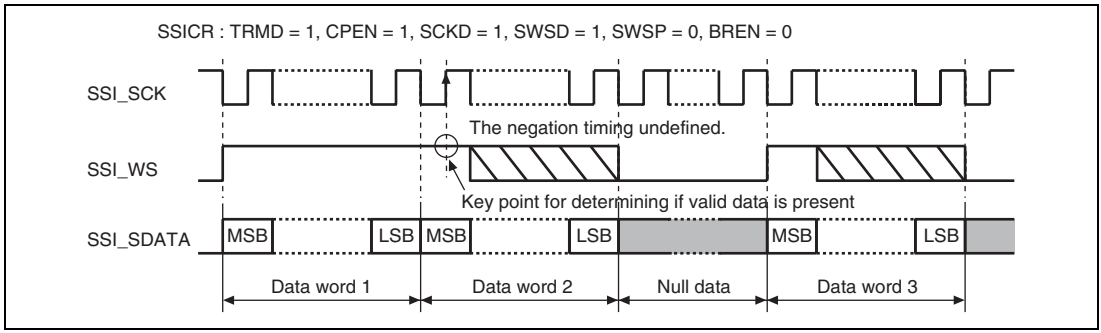


Figure 28.18 Compressed Data Format, Master Transmitter, Burst Mode Disabled

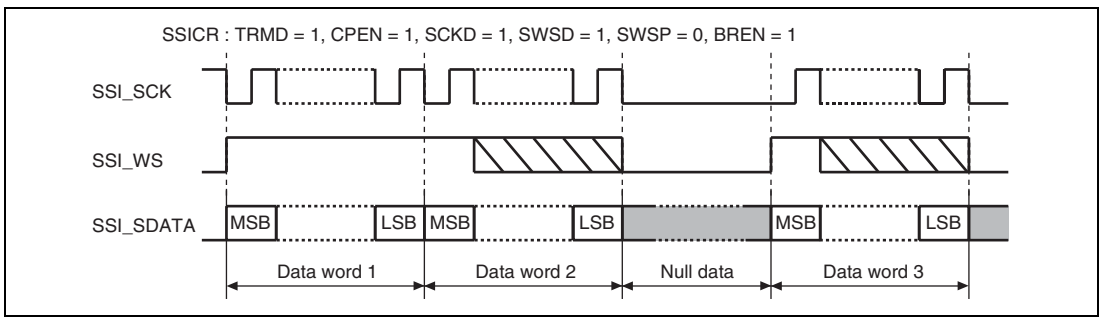


Figure 28.19 Compressed Data Format, Master Transmitter, and Burst Mode Enabled

(1) Slave Receiver

This mode allows the module to receive a serial bit stream from another device and store it in memory.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an input flow control. Assuming that SSICR.SWSP = 0 if SSI_WS is high then the module will receive the bit stream in blocks of 32 bits, one data bit per clock. If SSI_WS goes low then the module will complete the current 32-bit block and then stop any further reception, until SSI_WS goes high again.

(2) Slave Transmitter

This mode cannot be used.

(3) Master Receiver

This mode allows the SSI module to receive a serial bit stream from another device and store it in memory.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an output flow control. The module always asserts the word select signal to indicate it can receive more data continuously. It is the responsibility of the transmitting device to ensure it can transmit data to the SSI module in time to ensure no data is lost.

(4) Master Transmitter

This mode allows the module to transmit a serial bit stream from internal memory to another device.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an output flow control. The module always asserts the word select signal to indicate it will transmit more data continuously. Word select signal is not asserted until the first word is ready to transmit however. It is the responsibility of the receiving device to ensure it can receive the serial data in time to ensure no data is lost.

When the configuration for data transfer is completed, the SSI module can work with the minimum interaction with CPU. The CPU specifies settings for the SSI module and DMAC then handles overflow/underflow interrupts if required.

28.4.4 Operation Modes

There are three modes of operation: configuration, enabled and disabled. Figure 28.20 shows the transition diagram between these operation modes.

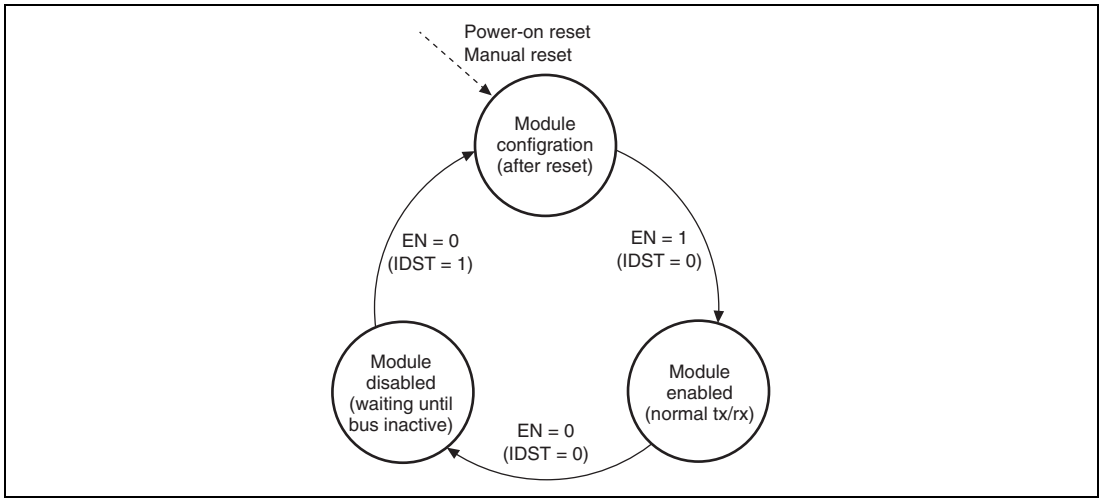


Figure 28.20 Transition Diagram between Operation Modes

(1) Configuration Mode

This mode is entered after the module is released from reset. All required settings in the control register should be defined in this mode, before the SSI module is enabled by setting the EN bit.

Setting the EN bit causes the SSI module to enter the module enabled mode.

(2) Module Enabled Mode:

Operation of the module in this mode depends on the selected operating mode. For details, see section 28.4.5, Transmit Operation and section 28.4.6, Receive Operation.

28.4.5 Transmit Operation

Transmission can be controlled in one of two ways: either DMA or an interrupt driven.

DMA driven is preferred to reduce the CPU load. In DMA control mode, an underflow or overflow of data or DMAC transfer end is notified by using an interrupt.

The alternative is using the interrupts that the SSI module generates to supply data as required. This mode has a higher interrupt load as the SSI module is only double buffered and will require data to be written at least every system word period.

When disabling the SSI module, the SSI clock* must be supplied continuously until the module enters in the idle state, indicated by the IIRQ bit.

Figure 28.21 shows the transmit operation in the DMA controller mode. Figure 28.22 shows the transmit operation in the interrupt controller mode.

Note: * SCKD = 0: Clock input through the SSI_SCK pin
SCKD = 1: Clock input through the SSI_CLK pin

(1) Transmission Using DMA Controller

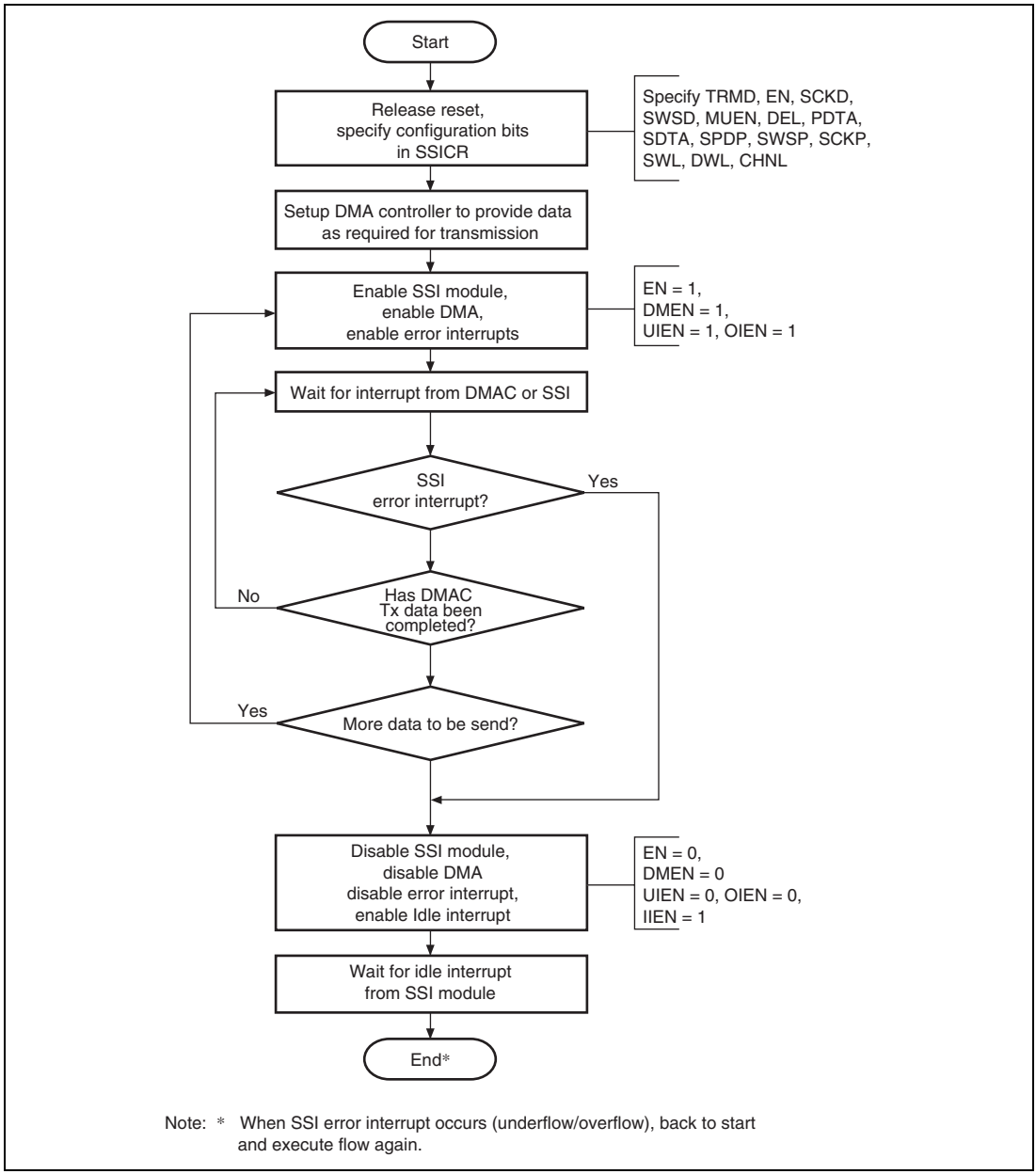


Figure 28.21 Transmission Using DMA Controller

(2) Transmission using Interrupt Data Flow Control

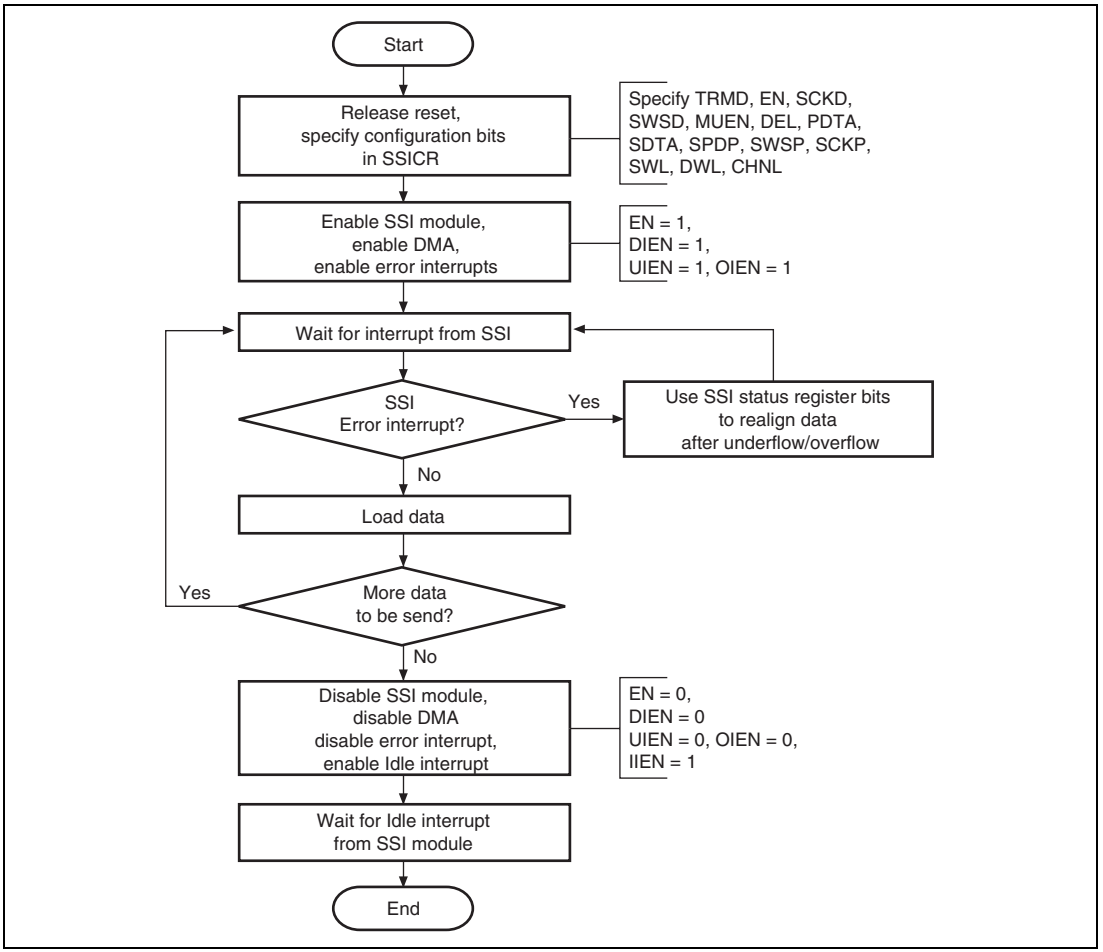


Figure 28.22 Transmission using Interrupt Data Flow Control

28.4.6 Receive Operation

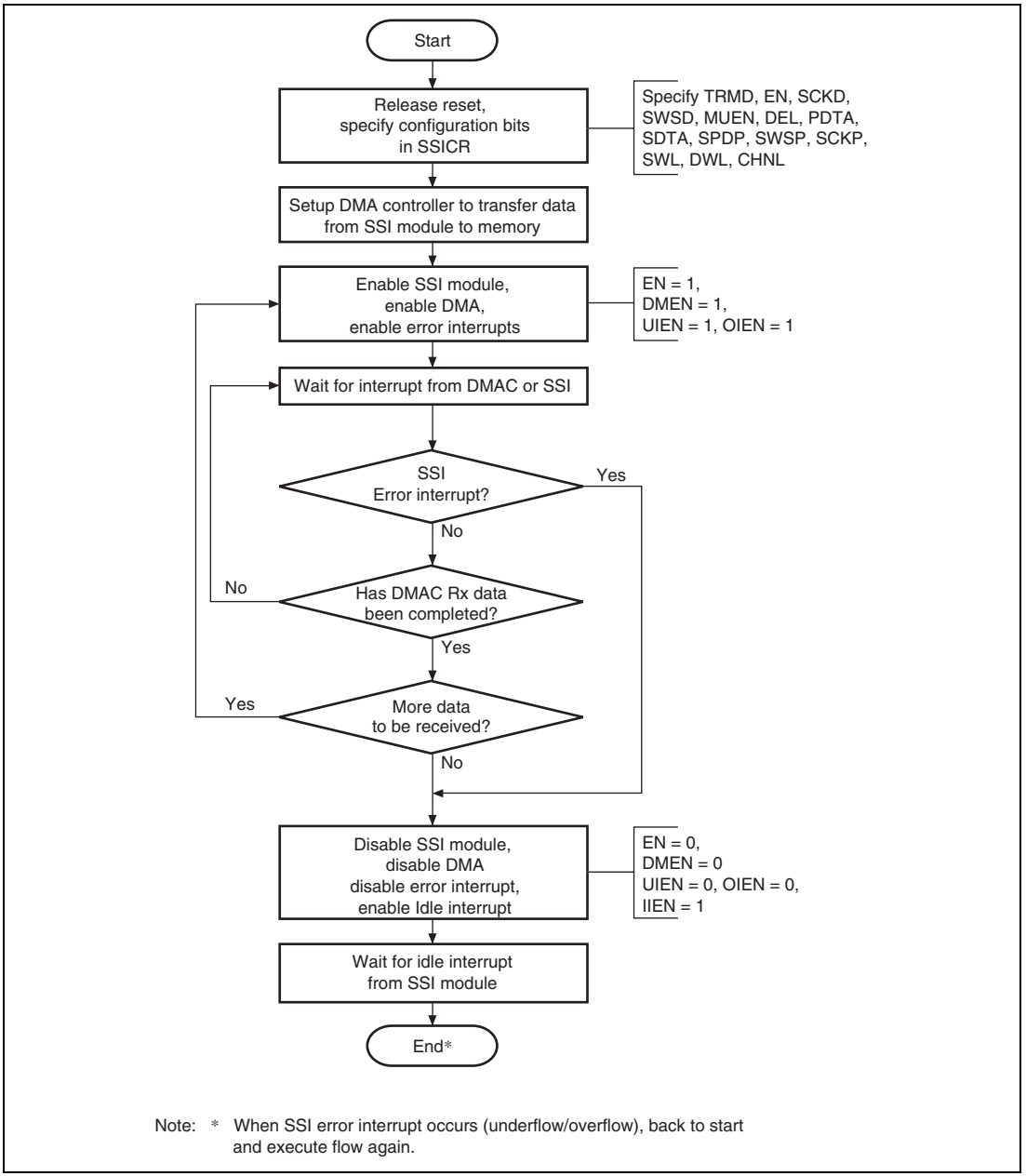
As with transmission the reception can be controlled in one of two ways: either DMA or an interrupt driven.

Figures 28.23 and 28.24 show the flow of operation.

When disabling the SSI module, the SSI clock must be supplied continuously until the module enters in the idle state, which is indicated by the IIRQ bit in SSISR.

Note: * SCKD = 0: Clock input through the SSI_SCK pin
SCKD = 1: Clock input through the SSI_CLK pin

(1) Reception Using DMA Controller



Note: * When SSI error interrupt occurs (underflow/overflow), back to start and execute flow again.

Figure 28.23 Reception using DMA Controller

(2) Reception using Interrupt Data Flow Control

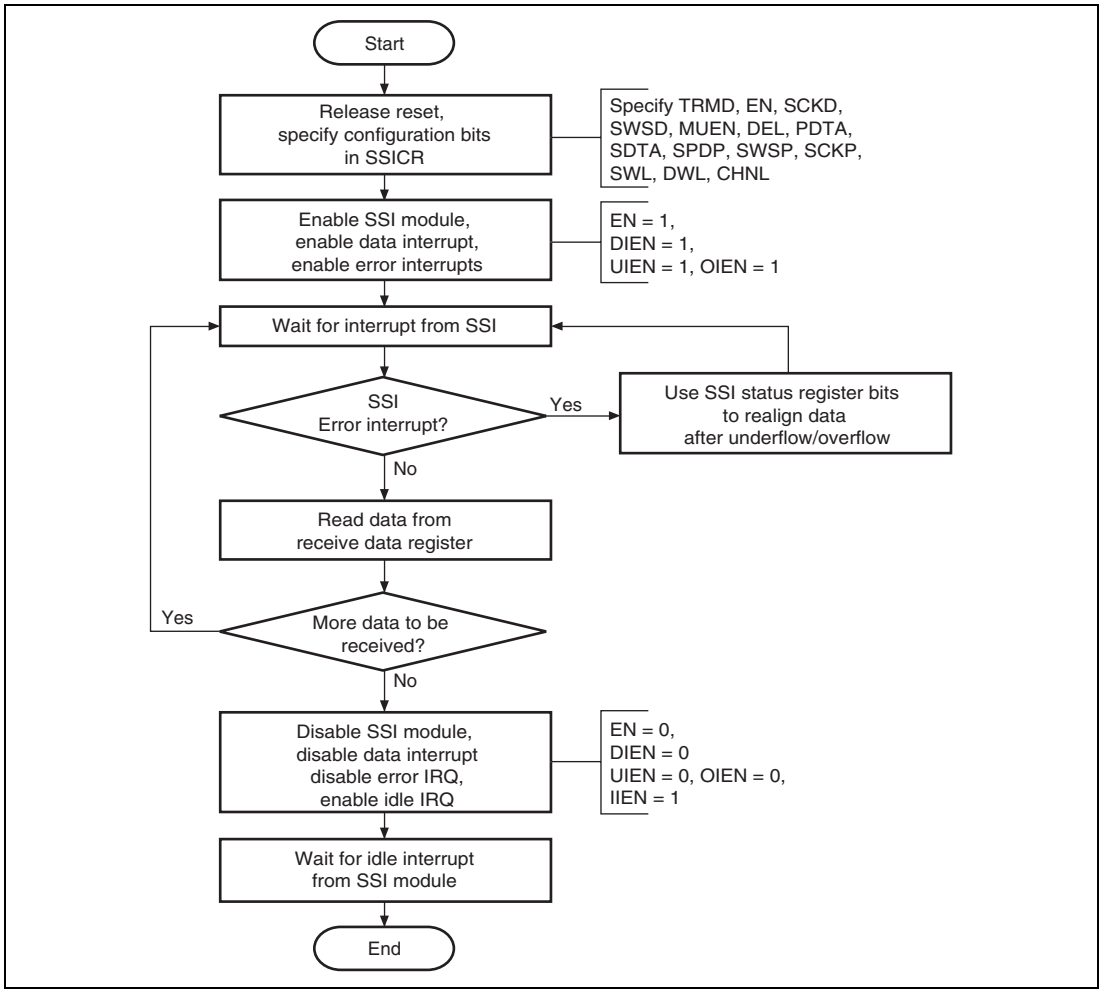


Figure 28.24 Reception using Interrupt Data Flow Control

When an underflow or overflow error condition is met (SSISR.UIRQ = 1 or SSISR.OIRQ = 1), the CHNO[1:0] and SWNO bits in SSISR can be used to recover the SSI module to a known status. When an underflow or overflow occurs, the CPU can read the number of channels and the number of system words to determine what point the serial audio stream has currently reached. In the transmitter case, the CPU can skip forward through the data it wants to transmit until the transmit of the data for which the SSI module is expecting to transmit next is enabled, and so resynchronize with the audio data stream. In the receiver case, the CPU can skip forward storing null sample data

until it is ready to store the sample data that the SSI module will receive next to ensure consistency of the number of received data, and so resynchronize with the audio data stream.

28.4.7 Procedures of Pausing and Resuming at Transmission

For pausing and resuming at transmission, operate with following procedures.

(1) To repeat transmitting and pausing without setting DMAC again

i. Set the DMEN bit in SSICR register to 0 (DMA request disabled) to disable the DMA transfer
ii. Wait for the DIRQ bit in SSISR register becomes 1 (the transmit buffer is empty) by polling or interrupt, etc.
iii. Set the EN bit in SSICR register to 0 (SSI module disabled) to stop the transfer.
iv. Check the IDST bit in SSISR register is set to 1 before resuming.
v. Set both the EN (SSI module enabled) and DMEN (DMA request enabled) bit in SSICR register to 1 at the same time.

(2) To set DMAC again after SSI paused

i. Set the DMEN bit in SSICR register to 0 (DMA request disabled) to disable the DMA transfer
ii. Wait for the DIRQ bit in SSISR register becomes 1 (the transmit buffer is empty) by polling or interrupt, etc.
iii. Set the EN bit in SSICR register to 0 (SSI module disabled) to stop the transfer.
iv. Enforce the DMA stopping by the DSTPR of the BBG/DMAC.
v. Check the IDST bit in SSISR register is set to 1 before resuming.
vi. Set each register of the BBG/DMAC and start the transfer.
vii. Set both the EN (SSI module enabled) and DMEN (DMA request enabled) bits in SSICR register to 1 at the same time.

28.4.8 Serial Bit Clock Control

This function is used to control and select which clock is used for the serial bus interface.

If the serial bit clock direction is set to input (SSICR.SCKD = 0), the SSI module is in clock slave mode, then the bit clock that is used in the shift register is derived from the SSI_SCK pin.

If the serial bit clock direction is set to output (SSICR.SCKD = 1), the SSI module is in clock master mode, and the shift register uses the bit clock derived from the SSI_CLK input pin or its clock divided. This input clock is then divided by the ratio in the serial oversampling clock division ratio (CKDV) bits in SSICR and used as the bit clock in the shift register.

In either case, the SSI_SCK pin output is the same as the bit clock.

28.5 Power-Down Mode

Power-Down mode has following modes or functions.

(1) Sleep mode

The operation of SSI is continued.

(2) Light Sleep mode

SSI continues operation, but the operation of receiving/transmitting using the DMAC is not guaranteed. In such cases, following procedure is needed before the state becomes to the light sleep mode.

i. Check the SSI module is idle.
   It is waiting until the SSISR[0] becomes 1.
ii. Check the transmitting buffer and the receiving buffer are empty.
   In transmitting mode, it is waiting until SSISR[24] becomes 1.
   In receiving mode, it is reading until SSISR[24] becomes 0.
iii. Disable all DMA requests from the SSI.
iv. Write a 0 to the EN bit in the SSICR (control status register) to stop the operation of the SSI module.

(3) Module standby mode

When the state becomes to the module standby mode while the SSI runs, the operation is not guaranteed. Before the state becomes to the module standby mode, following methods are needed:

i. Check the SSI module is idle.
ii. Check that all data transfers are finished.
   The procedure is same as the case of light sleep mode.
iii. Disable all DMA and interrupt requests from the SSI.
iv. Write a 0 to the EN bit in the SSICR (module control register) to finish the operation of the SSI module.
v. Set the MSTP20 and MSTP21 bit in the MSTPCR0 (standby control register) to 1.
   After releasing module standby mode, registers should be set again according to the procedure when using the SSI module again.

(4) Changing clock frequency

When clock frequency (Pck) is changed while the SSI is operating, the operation is continued. But there is a possibility that underflow or overflow occurs by frequency change. So before changing the clock frequency while the SSI is operating, following procedure is needed.

i. Check the SSI module is idle.
ii. Check that all data transfers are finished.
    The procedure is same as the case of light sleep mode.
iii. Disable all DMA and interrupt requests from the SSI.
iv. Set the state of the codec to power-down mode.
v. Write a 0 to the EN bit in the SSICR (control status register) to stop the operation of the SSI module.
    After changing the clock frequency, registers should be set again according to the procedure when using the SSI module again.

28.6 Usage Note

28.6.1 Restrictions when an Overflow Occurs during Receive DMA Operation

If an overflow occurs during receive DMA operation, the module must be reactivated.

The receive buffer of SSI has 32-bit common register to the left channel and right channel. If an overflow occurs under the condition of control register (SSICR) data-word length (DWL2 to 0) is 32-bit and system-word length (SWL2 to 0) is 32-bit, SSI has received the data at right channel that should be received at left channel.

If an overflow occurrence is confirmed through an overflow error interrupt or overflow error status flag (the OIRQ bit in SSISR), disable the DMA transfer of the SSI to halt its operation by writing 0 to the EN bit and DMEN bit in SSICR (then terminate the DMAC setting). And clear the overflow status flag by writing 0 to the OIRQ bit, set the DMA again to restart transfer.

28.6.2 Note when stopping the SSI module

As a protocol, the SSI transmission ends normally when the EN bit in SSICR register (SSI module enable bit) is 0 and a rising edge of the SSIn_WS (word select signal) is detected. When the master side of the communication stops first, the SSI module doesn't end normally because a rising edge of the SSIn_WS is not generated by the master. So the EN bit in SSICR register (SSI module enable bit) should be set to 0 before the master side stops.

28.6.3 Limitations when Operating in Slave Mode

When using the SH7786 in slave mode, make sure to end data transfer (EN = 0 in SSICR) before stopping input of the word select signal (SSI_WS).

In slave mode, data transfer ends when the EN bit in SSICR is cleared to 0 (transfer stop setting) and the falling edge of the word select signal (SSI_WS) is detected. If input of the word select signal stops too soon, it will not be possible to detect the falling edge of the word select signal and data transfer cannot end correctly.

28.6.4 Limitations when Changing Settings

Operation of the SSI_SCK and SSI_WS signals cannot be guaranteed immediately after the settings of the configuration bits in the control register (SSICR) are changed.

Do not change these settings dynamically if it could have an unwanted effect on the connected device.

28.6.5 Pin Function Setting for the SSI Module

Before setting or activating the SSI module, set the peripheral module select registers and the port control registers in terms of the SSI0 and SSI1 channels as described in section 30, General Purpose I/O Ports (GPIO).

Section 29 I²C Bus Interface

29.1 Features

This LSI has one channel of the I²C bus interface conforming to and providing the Philips I²C bus (Inter IC bus) interface functions. However, the configuration of registers that control the I²C bus differs partly from the Philips register configuration. The I²C bus interface has the following features:

- Supports the Philips I²C bus interface
- Multi-master compatible
- Seven- or ten-bit address compatible master
- Seven-bit slave address
- Fast mode compatible
- Variable clock frequencies

Figure 29.1 shows a block diagram for the I²C bus interface.

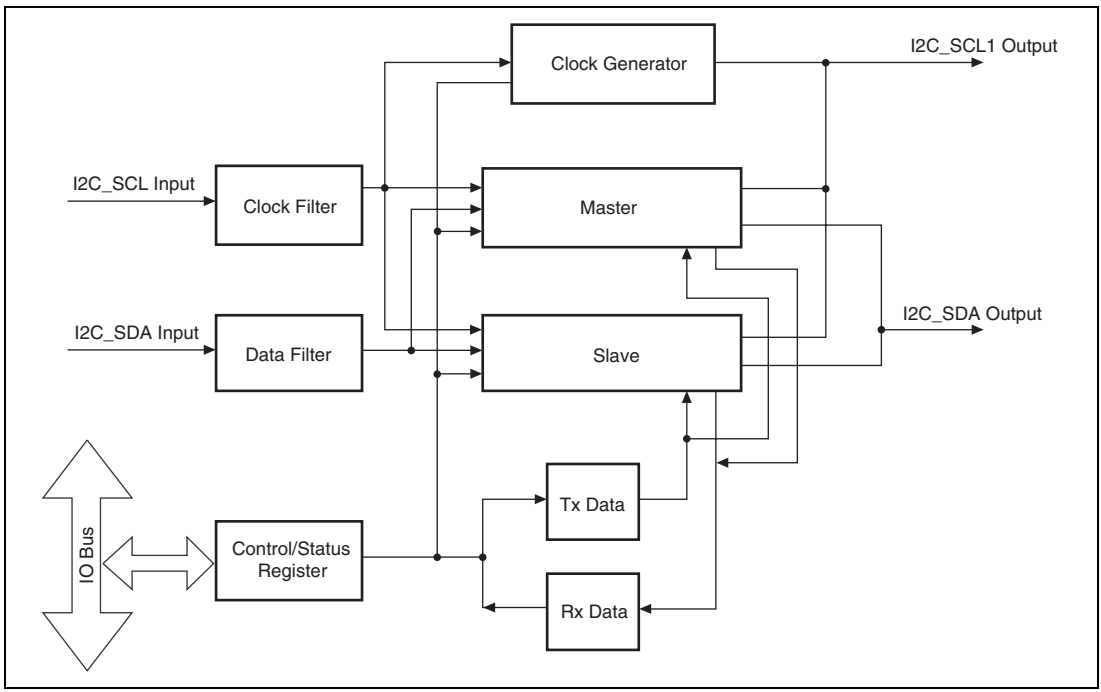


Figure 29.1 Block Diagram of I²C Bus Interface

29.2 Input/Output Pins

Table 29.1 lists the pins used in the I²C bus interface.

Table 29.1 I²C Bus Interface

Pin Name	I/O	Function
I2C0_SCL	I/O	I²C serial clock input/output pin*
I2C0_SDA	I/O	I²C serial data input/output pin*
I2C1_SCL	I/O	I²C serial clock input/output pin*
I2C1_SDA	I/O	I²C serial data input/output pin*

Note: * The SCL and SDA pins on the I²C bus interface are open drain pins.

The I²C buffer in this LSI is a 3.3 V LVTTL interface and is not compliant with the I²C timing standard (5V-torelance). When turning off the I/O power source (3.3 V) of this LSI, turn off the power source of the pull-up resistors connected to the pins of I²C.

29.3 Register Descriptions

The I²C bus interface includes the following registers.

Table 29.2 Register Configuration (1)

Register Name	Abbreviation	R/W	P4 Address	Area7 Address	Size
Slave control register 0	ICSCR0	R/W	H'FFE8 0000	H'1FE8 0000	32
Master control register 0	ICMCR0	R/W	H'FFE8 0004	H'1FE8 0004	32
Slave status register 0	ICSSR0	R/(W)*1	H'FFE8 0008	H'1FE8 0008	32
Master status register 0	ICMSR0	R/(W)*2	H'FFE8 000C	H'1FE8 000C	32
Slave interrupt enable register 0	ICSIER0	R/W	H'FFE8 0010	H'1FE8 0010	32
Master interrupt enable register 0	ICMIER0	R/W	H'FFE8 0014	H'1FE8 0014	32
Clock control register 0	ICCCR0	R/W	H'FFE8 0018	H'1FE8 0018	32
Slave address register 0	ICSAR0	R/W	H'FFE8 001C	H'1FE8 001C	32
Master address register 0	ICMAR0	R/W	H'FFE8 0020	H'1FE8 0020	32
Receive data register 0	ICRXD0	R/W	H'FFE8 0024	H'1FE8 0024	32
Transmit data register 0	ICTXD0	R/W	H'FFE8 0024	H'1FE8 0024	32
Slave control register 1	ICSCR1	R/W	H'FFE9 0000	H'1FE9 0000	32
Master control register 1	ICMCR1	R/W	H'FFE9 0004	H'1FE9 0004	32
Slave status register 1	ICSSR1	R/(W)*1	H'FFE9 0008	H'1FE9 0008	32
Master status register 1	ICMSR1	R/(W)*2	H'FFE9 000C	H'1FE9 000C	32
Slave interrupt enable register 1	ICSIER1	R/W	H'FFE9 0010	H'1FE9 0010	32
Master interrupt enable register 1	ICMIER1	R/W	H'FFE9 0014	H'1FE9 0014	32
Clock control register 1	ICCCR1	R/W	H'FFE9 0018	H'1FE9 0018	32
Slave address register 1	ICSAR1	R/W	H'FFE9 001C	H'1FE9 001C	32
Master address register 1	ICMAR1	R/W	H'FFE9 0020	H'1FE9 0020	32
Receive data register 1	ICRXD1	R/W	H'FFE9 0024	H'1FE9 0024	32
Transmit data register 1	ICTXD1	R/W	H'FFE9 0024	H'1FE9 0024	32

Table 29.2 Register Configuration (2)

Register Name	Abbreviation	Power-on reset by $\overline{\text{PRESET}}$ Pin/ WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction	Module Standby
Slave control register 0/1	ICSCR0/1	H'0000 0000	Retained	Retained	Retained
Master control register 0/1	ICMCR0/1	H'0000 0000*[3]	Retained	Retained	Retained
Slave status register 0/1	ICSSR0/1	H'0000 0000	Retained	Retained	Retained
Master status register 0/1	ICMSR0/1	H'0000 0000	Retained	Retained	Retained
Slave interrupt enable register 0/1	ICSIER0/1	H'0000 0000	Retained	Retained	Retained
Master interrupt enable register 0/1	ICMIER0/1	H'0000 0000	Retained	Retained	Retained
Clock control register 0/1	ICCCR0/1	H'0000 0000	Retained	Retained	Retained
Slave address register 0/1	ICSAR0/1	H'0000 0000	Retained	Retained	Retained
Master address register 0/1	ICMAR0/1	H'0000 0000	Retained	Retained	Retained
Receive data register 0/1	ICRXD0/1	Undefined	Retained	Retained	Retained
Transmit data register 0/1	ICTXD0/1	Undefined	Retained	Retained	Retained

Notes: 1. Only 0 can be written to bits 4 to 0 to clear the flags.
2. Only 0 can be written to bits 6 to 0 to clear the flags.
3. The value of bits 6 and bit 5 are undefined.

29.3.1 Slave Control Register (ICSCR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SDBS	SIE	GCAE	FNA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
3	SDBS	0	R/W	Slave Data Buffer Select This bit is used to select the data buffer. The double-buffer mode and single-buffer mode are available. When this bit is set to 0, the double-buffer mode is selected. During a reception, as long as both buffers are full and the SDR flag has not been cleared, SCL is held low. When the SDR flag is cleared, the low level state of SCL is released. When this bit is set to 1, the single-buffer mode is selected. SCL will be held low from the timing when the receive data register acquires the data packet until the SDR flag is cleared. 0: Double-buffer mode 1: Single-buffer mode
2	SIE	0	R/W	Slave Interface Enable This bit must be set for the slave operation. If this bit is low, the slave interface is reset.
1	GCAE	0	R/W	General Call Acknowledgement Enable When a master requires a slave to issue an acknowledgement, this bit must be set to 1
0	FNA	0	R/W	Forced Non Acknowledgement In the slave receive mode, the level of this bit is sent to the transmitting device as the acknowledge signal. This bit is set to 0 during the period that the data packet is being received, and set to 1 on completion of data reception. Forced non acknowledgement is returned to the master during slave reception. When the slave has received the last byte of data in a data packet, the slave communicates with the master by sending a nack, meaning that the acknowledgement is not driven. The master issues a stop on the bus after receiving a nack. The setting of this bit does not affect the acknowledgement of the slave address.

29.3.2 Slave Status Register (ICSSR)

The status bits (bits 0 to 4) in the slave status register are cleared by writing 0 to the respective status bit positions. The individual bits are held 1 until 0 is written to (other than the GCAR and STM bits).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	GCAR	STM	SSR	SDE	SDT	SDR	SAR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
6	GCAR	0	R	General Call Address Received Indicates that the address received from the bus is a general call address (00H). This status bit does not cause an interrupt. This bit is automatically cleared by hardware when the SIE bit (bit 2 in the slave control register) is set to 0 or when the SSR bit (bit 4 in this register) is set to 1.
5	STM	0	R	Slave Transmit Mode Indicates whether the current slave transmit mode is read or write. When this bit is set to 1, the mode is read. When this bit is set to 0, the mode is write. This status bit does not cause an interrupt. This bit is automatically cleared by hardware when the SIE bit (bit 2 in the slave control register) is set to 0 or when the SSR bit (bit 4 in the slave status register) is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	SSR	0	R/W*	Slave Stop Received A stop condition has been output on the bus. This status bit becomes active after the rising edge of SDA during the stop bit.
3	SDE	0	R/W*	Slave Data Empty Indicates that data to be transmitted has been loaded into the shift register. At the start of byte data transmission, the contents of the ICTXD register are loaded into a shift register ready for outputting data on the bus. This status bit indicates that data has been loaded and the ICTXD register is again ready for further data. This status bit becomes active on the falling edge of SCL before the first data bit. During the single-buffer mode, this bit must be reset every time new data has been written to the ICTXD register. This is because the slave holds SCL low to stop the bus while this bit is set to 1 even if a slave transmission cycle is started.
2	SDT	0	R/W*	Slave Data Transmitted A byte of data has been transmitted to the bus. This bit becomes active after the falling edge of SCL during the last data bit.
1	SDR	0	R/W*	Slave Data Received A byte of data has been received from the bus and is ready for read in the receive data register. This bit becomes active after the falling edge of SCL during the last data bit. During the single-buffer mode, this bit must be reset after data has been read from the ICRXD register. When SDBS is set to 1, SCL will be held low from the timing when the receive data register acquires the data packet until the SDR flag is cleared.

Bit	Bit Name	Initial Value	R/W	Description
0	SAR	0	R/W*	Slave Address Received Indicates that the slave has recognized its own address on the bus (defined by the contents of the slave address register). If the general call acknowledgement enable bit is enabled in the slave control register, then this status bit is also set to 1 even if the address on the bus is a general call address. In this case, the GCAR bit in this register is used to determine whether or not the address is a general call address. The STM bit indicates whether the access is read (high) or write (low). This status becomes active after the falling edge of SCL during the last address bit. The slave holds SCL low during the start of the ACK phase until the software resets this status bit.

Note: * This bit can be read from or written to. Writing 0 clears this bit to 0 and writing 1 is ignored.

29.3.3 Slave Interrupt Enable Register (ICSIER)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SSRE	SDEE	SDTE	SDRE	SARE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
4	SSRE	0	R/W	Slave Stop Received Interrupt Enable 0: Disables the SSR interrupt. 1: Enables the SSR interrupt.
3	SDEE	0	R/W	Slave Data Empty Interrupt Enable 0: Disables the SDE interrupt. 1: Enables the SDE interrupt.
2	SDTE	0	R/W	Slave Data Transmitted Interrupt Enable 0: Disables the SDT interrupt. 1: Enables the SDT interrupt.
1	SDRE	0	R/W	Slave Data Received Interrupt Enable 0: Disables the SDR interrupt. 1: Enables the SDR interrupt.
0	SARE	0	R/W	Slave Address Received Interrupt Enable 0: Disables the SAR interrupt. 1: Enables the SAR interrupt.

29.3.4 Slave Address Register (ICSAR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SAD D0_6	SAD D0_5	SAD D0_4	SAD D0_3	SAD D0_2	SAD D0_1	SAD D0_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
6 to 0	SADD0_6 to SADD0_0	All 0	R/W	Slave Address This is the unique 7-bit address allocated to the slave on the I²C bus. The slave interface compares this address with the first seven bits transmitted as the slave address, at the beginning of a data packet transmission.

29.3.5 Master Control Register (ICMCR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	—	—	0	0	0	0	0	0	—	—	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MDBS	FSCL	FSDA	OBPC	MIE	TSBE	FSB	ESG
Initial value:	0	0	0	0	0	0	0	0	0	—	—	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0 or 1	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
7	MDBS	0	R/W	Master Data Buffer Select This bit is used to select the data buffer. The double-buffer mode and singe-buffer mode are available. When this bit is set to 0, the double-buffer mode is selected. During a reception, as long as both buffers are full and the MDR flag has not been cleared, SCL is held low. When the MDR flag is cleared, the low level state of SCL is released. When this bit is set to 1, the single-buffer mode is selected. SCL will be held low from the timing when the receive data register acquires the data packet until the MDR flag is cleared. 0: Double-buffer mode 1: Single-buffer mode
6	FSCL	—	R/W	Forced SCL This bit controls the status of the I2C_SCL pin (reading reflects the current level on the I²C bus). When the OBPC bit is set, this bit directly controls the SCL line on the bus. During a read cycle, the level on this bit (which includes the reset level) will change depending on the level on I2C_SCL since it reflects the level on the I2C_SCL.

Bit	Bit Name	Initial Value	R/W	Description
5	FSDA	—	R/W	Forced SDA This bit controls the status of the I2C_SDA pin (reading reflects the busy status level on the I2C_SDA). When the OBPC bit is set then this bit directly controls the SDA line on the bus. During a read cycle, the level of this bit (which includes the reset level) will show the busy status of the I²C bus (1 for busy; 0 for not busy).
4	OBPC	0	R/W	Override Bus Pin Control When this bit is set to 1, the FSDA and FSCL bits in this register control SDA and SCL directly. This mode is used for testing purposes only.
3	MIE	0	R/W	Master Interface Enable When this bit is set to 1, the master interface is enabled.
2	TSBE	0	R/W	Start Byte Transmission Enable When this bit is set to 1, the master transmit is issuing a start byte (01H) on the bus after. The start byte is used for interfacing to slower microcontroller compatible with I²C bus interfaces.
1	FSB	0	R/W	Forced Stop onto the Bus When this bit is set to 1, the master transmits a STOP condition on the bus at the end of the current transfer. If ESG is also set, the master immediately transmits a START condition and begins transmitting a new data packet. If ESG is not set, state the master enters the idle state.
0	ESG	0	R/W	Enable Start Generation When this bit is set to 1, the master starts transmission of a data packet. If the bus is idle when ESG is set, the master transmits a START condition on the bus and then transmits the slave address. If the master is transferring data when ESG is set, at the end of that data byte transfer, the master transmits a repeated START condition before transmitting the slave address. When transmitting a data packet, the software must reset this bit when the slave address has been transmitted, otherwise a repeated START condition is transmitted after every transmission is completed.

29.3.6 Master Status Register (ICMSR)

The status bits (bits 0 to 6) in the master status register are cleared by writing 0 to the respective status bit positions. The individual status bits are held 1 until a reset by writing 0 to the appropriate bit position.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	MNR	MAL	MST	MDE	MDT	MDR	MAT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
6	MNR	0	R/W*	Master Nack Received When this bit is set to 1, this bit indicates that the master has received a nack response (the SDA line is high during the acknowledge cycle on the bus) to either an address or data transmission.
5	MAL	0	R/W*	Master Arbitration Lost In a multi-master system, when this bit is set to 1, it indicates that the master has lost arbitration to one of other masters on the bus. At this point, MIE is reset and the master interface is disabled.
4	MST	0	R/W*	Master Stop Transmitted When this bit is set to 1, it indicates that the master has sent a STOP condition on the bus. A STOP condition can be sent either as a result of the setting of the forced stop bit in the control register, or from a nack being received from a slave during a slave receive data packet.

Bit	Bit Name	Initial Value	R/W	Description
3	MDE	0	R/W*	Master Data Empty At the start of a byte data byte transmission, the contents of the transmit data register are loaded into a shift register ready for transmitting on the bus. When this bit is set to 1, it indicates that the transmit data register is available for further data by setting this register. During master transmit mode, the MDE bit is set at the same timing as the MAT bit is also set after transmission of the slave address. In this case, you need to set the MDT and MAT bits after the ICMCR's ESG bit is cleared. The clearing will restart the data transmission.
2	MDT	0	R/W*	Master Data Transmitted Byte data has been sent to the slave on the bus. This status bit becomes active after the falling edge of SCL during the last data bit.
1	MDR	0	R/W*	Master Data Received Byte data has been received from the bus and is in the receive data register. This status bit becomes active after the falling edge of SCL during the last data bit. During single-buffer mode, this status bit must be reset after data has been read from the receive data register. When MDBS is set to 1, SCL will be held low from the timing when the receive data register acquires the data packet until the MDR flag is cleared. During master reception mode, the MDR bit is set at the same timing as the MAT bit set after transmission of the salve address. In this case, you must clear the MDR and MAT bits after the ICMCR's ESG bit is cleared. Clearing will start the data reception
0	MAT	0	R/W*	Master Address Transmitted The master has been transmitted the slave address byte of a data packet. This bit becomes active after the falling edge of SCL during the ack bit of after the address.

Note: * This bit can be read from or written to. Writing 0 clears this bit to 0 and writing 1 is ignored.

29.3.7 Master Interrupt Enable Register (ICMIER)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	MNRE	MALE	MSTE	MDEE	MDTE	MDRE	MATE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
6	MNRE	0	R/W	Master Nack Received Interrupt Enable 0: Disables the MNR interrupt. 1: Enables the MNR interrupt.
5	MALE	0	R/W	Master Arbitration Lost Interrupt Enable 0: Disables the MAL interrupt. 1: Enables the MAL interrupt.
4	MSTE	0	R/W	Master Stop Transmitted Interrupt Enable 0: Disables the MST interrupt. 1: Enables the MST interrupt.
3	MDEE	0	R/W	Master Data Empty Interrupt Enable 0: Disables the MDE interrupt. 1: Enables the MDE interrupt.
2	MDTE	0	R/W	Master Data Transmitted Interrupt Enable 0: Disables the MDT interrupt. 1: Enables the MDT interrupt.
1	MDRE	0	R/W	Master Data Received Interrupt Enable 0: Disables the MDR interrupt. 1: Enables the MDR interrupt.
0	MATE	0	R/W	Master Address Transmitted Interrupt Enable 0: Disables the MAT interrupt. 1: Enables the MAT interrupt.

29.3.8 Master Address Register (ICMAR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SAD D1_6	SAD D1_5	SAD D1_4	SAD D1_3	SAD D1_2	SAD D1_1	SAD D1_0	STM1
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
7 to 1	SADD1_6 to SADD1_0	All 0	R/W	Slave Address These bits are the address of the slave which the master communicates with.
0	STM1	0	R/W	Slave Transfer Mode This bit specifies the mode in which the slave operates. Bit STM1 sets the operating mode (transmit or receive mode) of the slave, which is an external slave device whose address matches the slave address (SADD1) sent from the master. The slave device is automatically set to transmit/receive mode by hardware on reception of the STM1 signal. When this bit is set to 1, it indicates a read operation, when this bit is cleared to 0, it indicates a write operation.

29.3.9 Clock Control Register (ICCCR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SCGD						CDF	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
7 to 2	SCGD	All 0	R/W	SCL Clock Generation Divider When operating in master mode, the SCL clock is generated from the internal clock using SCGD as the ratio. The slave will also operate on the clock generated from the internal clock when SCL is held low to hold the bus up when an overflow occurs. SCGD must be specified in both master and slave modes. The formula expressing the relationship is: Equation 2 SCL rate calculation SCLfreq = IICck / (20 + (SCGD * 8)) IICck: I²C internal clock frequency Suggested settings for CDF and SCGD for various CPU speeds and the two I²C bus speeds are given in table 29.3.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	CDF	All 0	R/W	Clock Division Factor The internal clock used in most blocks in the I²C module is a divided peripheral Bus clock. The internal I²C clock is generated from the peripheral Bus clock using the CDF as the division ratio: Equation 1 I²C internal clock frequency calculation IICck = Pck / (1 + CDF) Pck: peripheral clock The minimum time to ensure adequate setup and hold times on the SDA line relative to the SCL line on the bus. The clock frequency is to ensure that the glitch filtering will operate with glitches of up to 50 ns as described in the fast mode I²C specification.

Note: CDF must be set so that the clock frequency (IICck) is lower than 20 MHz.

Table 29.3 Recommended Settings for CDF and SCGD*

I/O Bus Clock Frequency	100 kHz		400 kHz	
	CDF	SCGD	CDF	SCGD
50 MHz	2	19	2	3
Error	-3.10%		-5.30%	

Note: * These recommended values are for the SCL rate.

29.3.10 Receive and Transmit Data Registers (ICRXD and ICTXD)

Reading from or writing to these registers access different physical internal registers. When data is to be transmitted, the contents of the shift register are loaded via TXD. After data has been received into the shift register from the I²C bus, it is then loaded into RXD.

(1) Receive Data Register (ICRXD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RXD							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
7 to 0	RXD	All 0	R	Read—Receive Data Data received by master or slave.

(2) Transmit Data Register (ICTXD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXD							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved When reading these bits, the value of bits 7 to 0 is reflected by each byte. The write value should always be 0.
7 to 0	TXD	All 0	W	Write—Transmit Data Data transmitted by master or slave.

29.4 Operations

29.4.1 Data and Clock Filters

These blocks filter out glitches on signals coming from the I²C bus. Glitches up to one internal clock period in width are rejected (For details on the internal clock frequency see section 29.3.9, Clock Control Register (ICCCR)). This is for the faster I²C bit rate (400 kHz) but does not violate the slower I²C bus rate specification.

These blocks also resynchronize bus signals with the internal clock.

29.4.2 Clock Generator

The clock generator has two functions. Firstly, it generates the SCL (I²C bus clock) according to commands from of the master or slave interface. Secondly, it controls the internal clock rate, used by filtering blocks and the master and slave interfaces. This clock functions as a clock enable signal of the registers in these blocks.

29.4.3 Master/Slave Interfaces

These two interfaces run independently and in parallel. The master interface controls the transmission of address and data on the I²C bus. The slave interface monitors the I²C bus and takes part in transmissions if its programmed address is seen on the bus. The interfaces communicate with the control/status registers independently. There is only one interrupt line output from the I²C module. The interrupt source is either the master or the slave.

29.4.4 Software Status Interlocking

In order that the software interface to the I²C module be as robust as possible, various status interlocks are built into the operation of the master and slave interfaces. The status bits involved are:

(1) MDR and SDR

MDR and SDR are set to 1 when data is received. Clear the status after reading the receive data register. If data is received while MDR and SDR are set, hardware recognizes that unread data remains in the receive data register and automatically holds SCL at low level and suspends data transmission. In this case, transmission can be resumed by clearing the status after reading the receive data.

Consequently, when receiving data continuously, be sure to clear the status of MDR and SDR after reading the receive data register.

(2) MDE and SDE

If the MDE or SDE status bits are still set data in the transmit data register is to be transmitted on the I²C bus by the slave or master, the SCL line must be held low until the MDE and SDE status bits are reset. The MDE or SDE status bit being set indicates that the data currently held in the Transmit Data Register has already been transmitted on the I²C bus.

The software must clear this status bit when it writes to the transmit data register which is ready to transmit subsequent data bytes. This is not required for the first byte of data to be transmitted on the bus.

(3) MAL

When the master loses arbitration, the MAL bit (of the master status register) is set and the MIE bit (of the master control register) is reset. At this point, master mode is invalid and the I²C bus interface enters the slave mode. When master operation is restarted, data transfer from the master begins after the MAL bit has been cleared.

(4) SAR

The SAR status bit is set when the slave identifies its address on the I²C bus. At this point the slave interface forces the SCL line low until the SAR status bit is reset.

This is particularly important when a slave transmit is about to take place on the bus, and the slave will transmit the data from the transmit data register. The software responds to the SAR status by writing the required data into the transmit data register and then resetting the SAR status bit. This allows the slave interface to continue the access.

When the slave is about to receive data, the software may be reading data loaded in a previous access from the receive data register. In this case the valid data still held in the receive data register is overwritten. However, this is avoided using the SAR status bit. After the software has read data in the receive data register, reset the SAR bit (if it is set). Then overwriting the receive data register is avoided.

29.4.5 I²C Bus Data Format

Figure 29.2 shows a timing chart for the I²C bus interface. Table 29.4 describes the meaning of each symbol in figure 29.2.

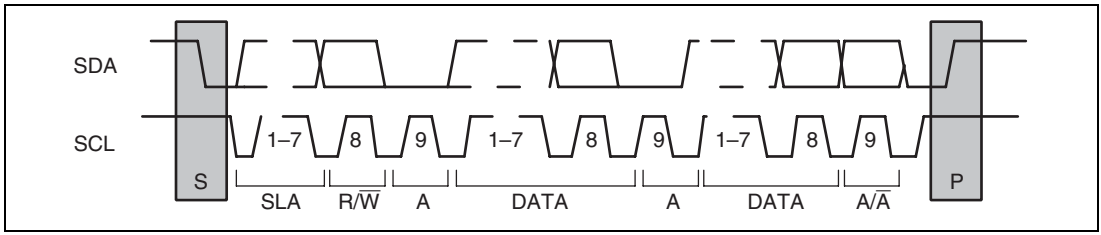


Figure 29.2 I²C Bus Timing

Table 29.4 Description on Symbols of I²C Bus Data Format

Symbol	Description
S	Indicates a start condition. A master device changes SDA from high to low while SCL is high level.
SLA	Indicates a slave address. A slave address is used when a master device selects a slave device.
R/$\overline{W}$	Indicates the direction of data transmission. If the R/$\overline{W}$ bit is 1, the data flows from the slave to the master device. If the bit is 0, the data flows from the master to the slave device.
A	Indicates data acknowledge. Data receiving device makes SDA low level (the slave device returns a data acknowledge signal in master transmission mode, and vice versa).
DATA	Indicates transmit or receive data. The data length is eight bits, which are transferred in the MSB first.
P	Indicates a stop condition. A master device changes SDA from low to high while SCL is high.

29.4.6 7-Bit Address Format

Figure 29.3 shows the format of data transfer from a master to a slave device (master data transmit format). Figure 29.4 shows the data transfer format (master data receive format) when a master device reads the second and the following byte data from a slave device.

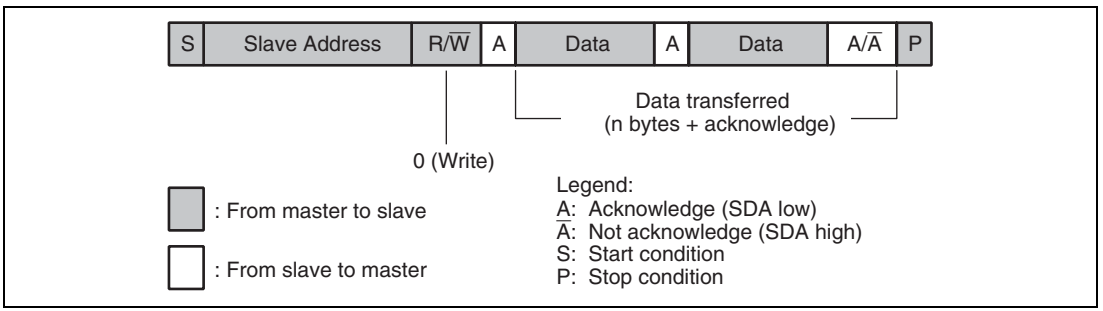


Figure 29.3 Master Data Transmit Format

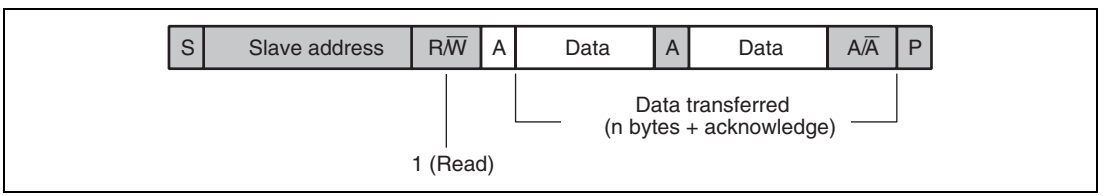


Figure 29.4 Master Data Receive Format

Figure 29.5 shows the combined format when the data transfer direction changes during one transfer. When changing the direction after the first transfer, the repeated START condition (Sr), slave address and R/W bits are transmitted. In this case, the R/W bit is set to the direction opposite to the first transfer direction. The repeated START condition is issued by the master at the end of a transmit or receive cycle if the enable start generation bit in the master control register has been set.

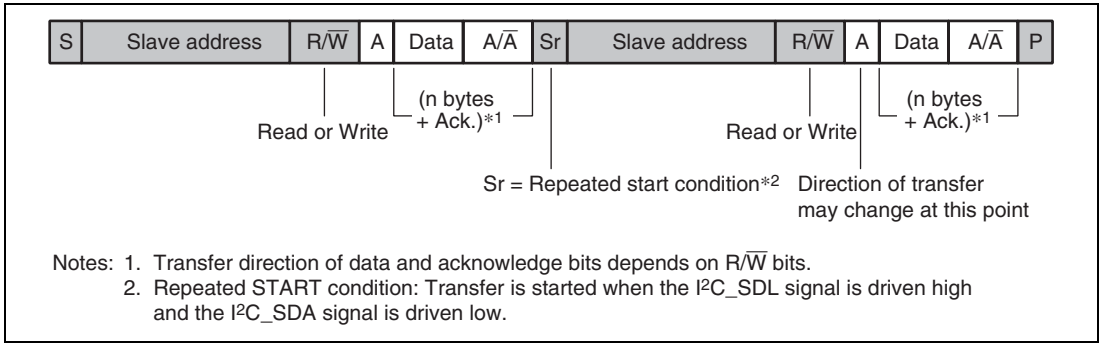


Figure 29.5 Combination Transfer Format of Master Transfer

29.4.7 10-Bit Address Format

Description is given below on the 10-bit address transfer format supported in master mode. This format has three transfer methods as the 7-bit address transfer format.

Figure 29.6 shows the data transmit format. The set value in the master address register is output in one byte following the first START condition (S). The value set in the transmit data register (TXD) is transmitted as a slave address in the second byte. Data on and after the third byte is transferred in the same way as the 7-bit address data.

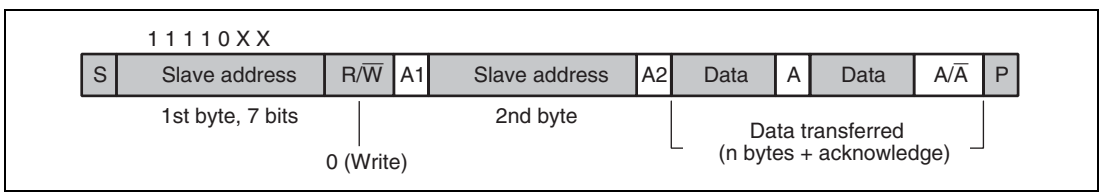


Figure 29.6 10-Bit Address Data Transmit Format

Figure 29.7 shows the data receive format. Two bytes of an address is transmitted a repeated START in the same way as in the data transmit format. Then, repeated START condition (Sr) is transmitted and the value set in the address register is output. At this time, STM1 must be set to 1 (receive mode). Data is transferred in the same way as in the 7-bit address data receive format.

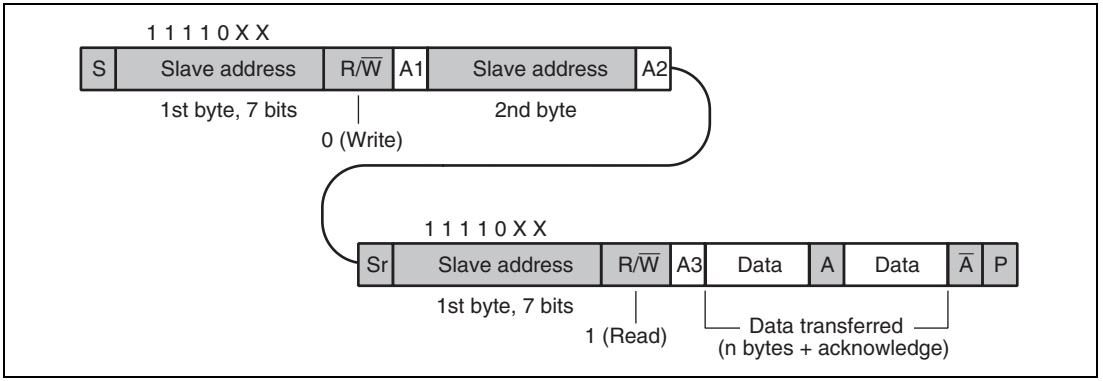


Figure 29.7 10-Bit Address Data Receive Format

Figure 29.8 shows the data transmit/receive combined format.

In the data transmit/receive combined format, data is transmitted after an address is transmitted with the first two bytes. Then, the repeated START condition (Sr) is transmitted instead of STOP condition (P). After Sr is transmitted, the procedure is the same as that in the data receive format.

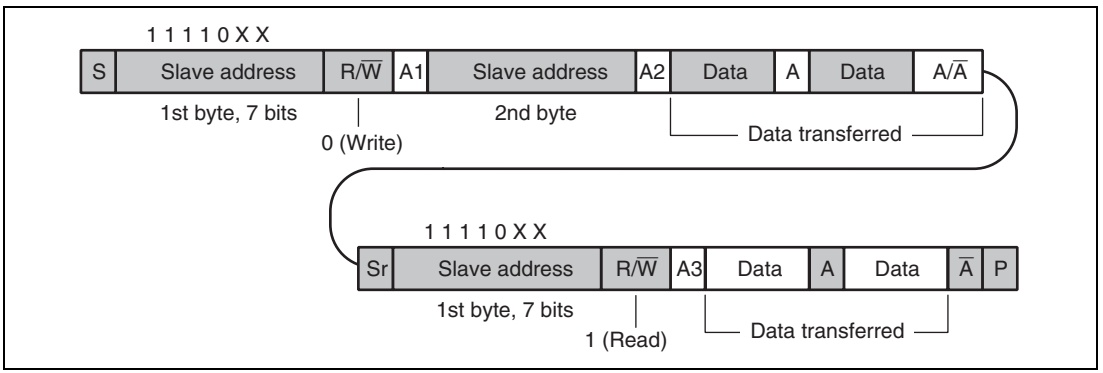


Figure 29.8 10-Bit Address Transmit/Receive Combined Format

29.4.8 Master Transmit Operation

The transmit procedure and operation in master transmit mode are described below. Figures 29.9 shows the timing chart in master transmit mode. Setting the MDBS bit in the master control register allows the I²C to operate in single-buffer mode.

1. For initial setting, set the clock control register and the master interrupt enable register according to the slave address, transmit data, and the transmit speed. Since slave mode is also required even when master mode is used, set the device address in the slave address register.
2. Monitor the FSDA bit in the master control register. Confirm that this bit is low, meaning that other I²C devices are not using the bus. After confirmation, set the MIE (bit 3) and ESG (bit 0) bits in the master control register to 1 to start master transmission.
3. After the transmit START condition, slave address, and data transfer direction bits are transmitted, an interrupt due to the MAT and MDE bits in the master status register is generated at the timing of (1) in figure 29.9. At this time, clear the ESG bit to 0. To suspend the data transmission, the master device will hold SCL low until the MDE bit is cleared.
4. An interrupt due to the SAR bit is generated at the timing of (3) shown in figure 29.9. If the IRQ handling in the slave device is delayed, the slave device extends the SCL period to suspend data transmission (at the timing of (7) in figure 29.9). The slave device drives SDA low at the ninth clock and returns ACK.
5. Data is transmitted in units of nine bits: 8-bit data and 1-bit ACK. An interrupt of MDE (bit 3) is generated at the ninth clock before data transfer (at the timing of (2) in figure 29.9). An interrupt of MDT (bit 2) is generated at the eighth clock after 1-byte data transfer (at the timing of (4) in figure 29.9). Clear MDE to 0 after setting transmit data. An interrupt of SDR (slave data receive) of the slave device is generated at the eighth clock (at the timing of (6) in figure 29.9). Clear SDR after the slave device reads the receive data. If this processing is delayed, the slave device extends the SCL period to suspend data transmit (at the timing of (8) in figure 29.9).
6. To end data transfer, an interrupt of MNR (bit 6) in the master status register is generated at the ninth clock (at the timing of (5) in figure 29.9) when ACK from the slave device is 1 (Nack). The master device receives this Nack and outputs data transfer end condition. When data transmission ends on the master device side, set FSB (bit 1) in the master control register to 1 to output the suspend condition. After the I²C module fetches FSB on completion of transmission or reception of the last of byte data, it enters the stop state. Therefore in order to stop the communication after the predetermined number of byte data is transferred, the FSB bit needs to be set before the last byte data transfer is started.
7. The FSB bit needs to be set before the last byte data is transferred. In master transmit mode, after the last byte data is set, the MST (master stop transmitted) bit is checked by either

interrupt or polling. At the same time MNR (master NACK received) bit must be checked. If NACK is returned, an error routine is executed to retransmit the last byte data.

Signal level changes of (1) to (6) in figure 29.9 are generated after the falling edge of the clock.

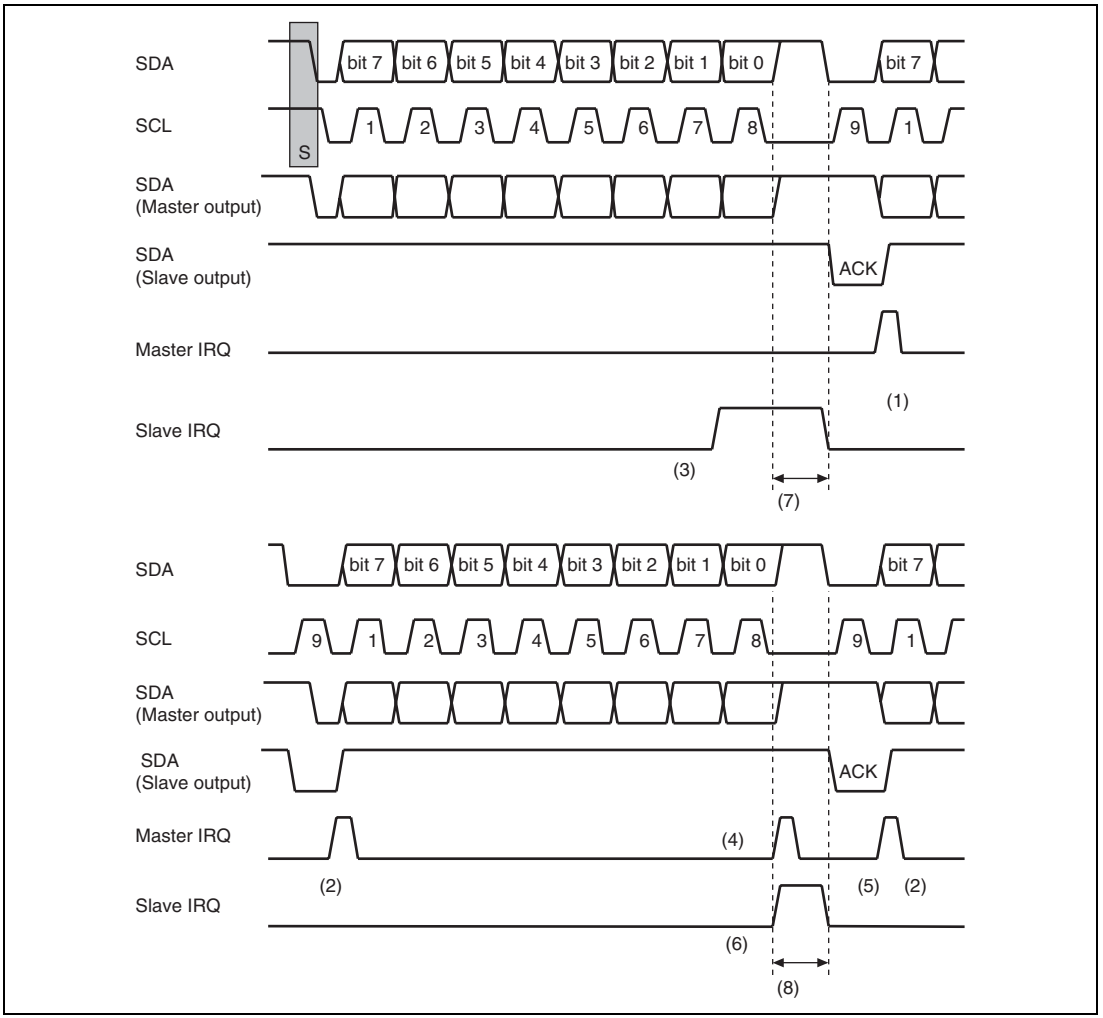


Figure 29.9 Data Transmit Mode Operation Timing

29.4.9 Master Receive Operation

The data receive procedure and operation in master receive mode are described below. Figure 29.10 shows the timing chart in master receive mode. Setting the MDBS bit in the master control register allows the I²C to operate in single-buffer mode.

1. In master receive mode, as to transmit of a slave address and a 1-bit signal indicating the data transfer direction, operation is the same as that in master transmit mode. At this time, set the data transfer direction to 1 (reception).
2. The slave device automatically enters the data transmit mode according to the signal that indicates the data transfer direction, and transmits 1-byte data in synchronization with the SCL clock output from the master device. The master device generates an interrupt of MDR (bit 1) at the eighth clock (at the timing of (2) in figure 29.10). Clear the MDR bit after the master device reads receive data. If this processing is delayed, the slave device extends the SCL period to suspend data transmission (at the timing of (3) in figure 29.10).
3. The slave device generates an interrupt of the status SDT (bit 2) indicating 1-byte data transfer end at the eighth clock (at the timing of (2) in figure 29.10) and an interrupt of the status SDE (bit 3) indicating data empty at the ninth clock (at the timing of (1) in figure 29.10). Clear SDE after writing slave transmit data to TXD.
4. To end data transfer, set FSB (bit 1) in the master control register of the master device and output suspend condition. After the I²C module fetches FSB on completion of transmission or reception of the last of byte data, it enters the stop state. . Therefore in order to stop the communication after predetermined number of byte data is transferred, FSB bit needs to be set before the last byte data transfer is started. After confirmation of the last byte data reception, though the master receiver finishes the receive transaction, the protocol layer will inform the slave transmitter or retransmission if the last byte is incorrect.

Signal level changes of (1) to (3) in figure 29.10 are generated after the falling edge of the clock.

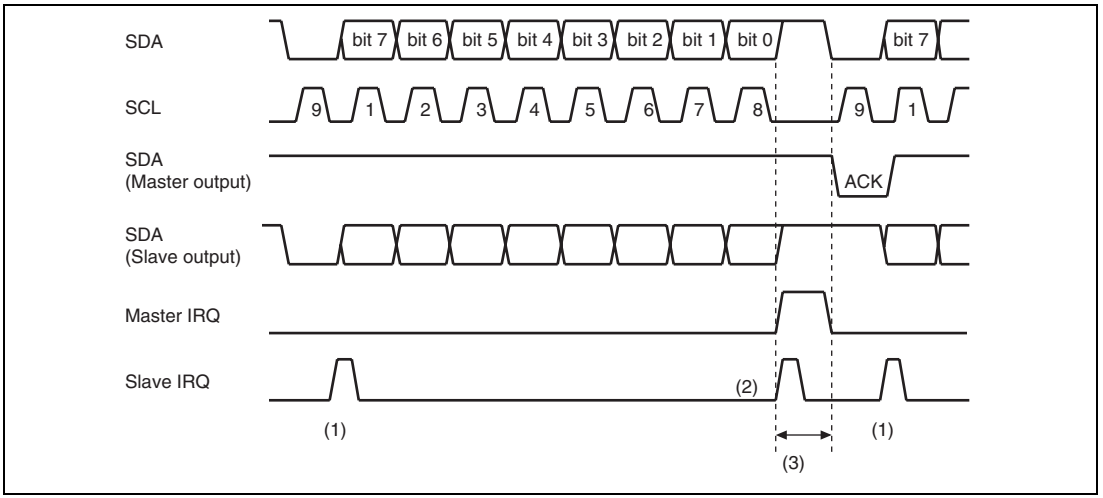


Figure 29.10 Data Receive Mode Operation Timing

29.5 Programming Examples

29.5.1 Master Transmitter

In order to set up the master interface to transmit a data packet on the I²C bus, follow the following procedures:

(1) Load Clock Control Register

1. SCL clock generation divider (SCGD) = 03h
   (SCL frequency of 400 kHz)
2. Clock division ratio (CDF) = 2h
   (The IO bus clock is 50 MHz and the I²C's internal clock IICck is 16.7 MHz.)

(2) Load Master Control Register (First Data Byte and Address)

1. Master address register = address of slave being accessed and STM1 bit (write mode: 0)
2. Transmit data register = first data byte to be transmitted
3. Master control register = 89h
   (MDBS = 1, MIE = 1, ESG = 1)

(3) Wait for Outputting Address

1. Wait for master event (an interrupt of the MAT and MDE bits in the master status register).
2. Set the master control register to 88h (To suspend the data transmission, the master device will hold the SCL low until the MDE bit is cleared.)
   If only one byte of data is transmitted, set the master control register to 8Ah, meaning that the stop generation is enabled. This generates a stop on the bus as soon as one byte has been transmitted.
3. Reset the MAT bit.

(4) Monitor Transmission of Data

1. Wait for master event, MDE in the master status register.
2. Transmit data register = subsequent data.
3. Reset the MDE bit.
   Clear MDE after setting the last byte to be transmitted. After the last byte data is transmitted, MDE is generated. To clear the MDE, you must set the master control register to 8Ah.
   (Set the force stop control bit).

(5) Wait for End of Transmission

1. Wait for the master event, MST in the master status register.
2. Reset the MST bit after confirming MNR (Master NACK Received).

29.5.2 Master Receiver

To set up the master interface to receive a data packet on the I²C bus, follow the following procedures:

(1) Load Clock Control Register

1. SCL clock generation divider (SCGD) = 03h
   (SCL frequency of 400 kHz).
2. Clock division ratio (CDF) = 2h
   (The IO bus clock is 50 MHz and the I²C's internal clock IICck is 16.7 MHz.)

(2) Load Master Control Register and Address

1. Set master address register to address of slave being accessed and STM1 bit (read mode: 1).
2. Set master control register to 89h
   (MDBS = 1, MIE = 1, ESG = 1).

(3) Wait for Outputting Address

1. Wait for master event (an interrupt of the MAT and MDR bits in the master status register).
2. Set the master control register to 88h
   (To suspend the data transmission, the master device will hold the SCL low until the MDR bit is cleared).
   If only one byte of data is received, set the master control register to 8Ah, meaning that the stop generation is enabled. This generates a stop on the bus as soon as one byte has been received.
3. Reset the MAT bit.

(4) Monitor Reception of Data

1. Wait for master event (the MDR bit in the master status register).
2. Read data from the received data register.
   If the next byte of data is the second to last byte to be transmitted by the slave device, the following applies to the receive interrupt (that is, MDR interrupt) in the second to last byte.

3. Set the master control register to 8Ah
   (Set the force stop control bit).
4. Reset the MDR bit.

(5) Wait for End of Reception

1. Handle the receive interrupt (MDR) in the last byte: that is, read the data and clear the MDR.
2. Wait for master event, MST in the master status register.
3. Reset the MST bit.

29.5.3 Master Transmitter—Restart—Master Receiver

In order to set up the master interface to transmit a data packet on the I²C bus, issue a restart, then read byte data back from the slave, follow the following procedures:

(1) Load Clock Control Register

1. Set the SCL clock generation divider (SCGD) to 03h
   (SCL frequency of 400 kHz).
2. Set the clock division (CDF) to 2h
   (The I/O bus clock is 50 MHz and the I²C's internal clock IICck is 16.7 MHz.)

(2) Load Master Control Register and Address

1. Set the master address register to address of slave being accessed and STM1 bit (writes mode: 0).
2. Set the master control register to 89h
   (MDBS = 1, MIE = 1, ESG = 1).

(3) Wait for Outputting Address

1. Wait for master event (an interrupt of the MAT and MDE bits in the master status register).
2. Set the master address register to address of slave being accessed and STM1 bit (read mode: 1).
   When the enable start generation bit in the master control register is still set, at the end of the byte transmission the master will issue a restart. Since the new address has been loaded above the bus direction will be changed.
3. Reset the MAT bit.

(4) Wait for Outputting Address

1. Wait for master event (an interrupt of the MAT and MDR bits in the master status register).
2. Set the master control register to 88h (To suspend stop the data transmission, the master device will hold the SCL low until the MDR bit is cleared.)
3. Reset the MAT bit.

(5) Monitor of Data

1. Wait for master event, the MDR bit in the master status register.
   Read data from the received data register.
   If the next byte of data is the second to last byte but one to be transmitted by the slave device, the following applies to a receive interrupt (that is, MDR interrupt) in the second to last byte
2. Set the master control register to 8Ah
   (set the force stop control bit).
3. Reset the MDR bit.

(6) Wait for End of Reception

1. Handle the receive interrupt (MDR) in the last byte: that is, read the data and clear the MDR.
2. Wait for the master event (the MST bit in the master status register).
3. Reset the MST bit.

Section 30 General Purpose I/O Ports (GPIO)

30.1 Features

This LSI has ten general-purpose ports (A to J), which provide 9 groups of 60 input/output pins each.

The general-purpose I/O (GPIO) port pins are multiplexed with the pins of peripheral modules to select whether the pins are used by the GPIO or the peripheral modules.

The GPIO has the following features.

- Each port pin is a multiplexed pin, for which the port control register can specify the pin function and control the pull-up MOS of the pin.
- Each port has a data register that stores data for the pins.
- GPIO interrupts are supported*.

Note: * For the ports which can be used as GPIO interrupt pins, refer to table 30.1. For GPIO interrupt settings, refer to section 10, Interrupt Controller (INTC).

Table 30.1 Multiplexed Pins Controlled by Port Control Registers

Pin Name	Port	GPIO	Selectable Module	GPIO Interrupt
CDE/ETH_MAGIC	A	PA7 input/output	DU/ETH	—
DISP/ETH_LINK	A	PA6 input/output	DU/ETH	—
DR5/ETH_TX_ER	A	PA5 input/output	DU/ETH	—
DR4/ETH_TX_EN	A	PA4 input/output	DU/ETH	—
DR3/ETH_TXD3	A	PA3 input/output	DU/ETH	—
DR2/ETH_TXD2	A	PA2 input/output	DU/ETH	—
DR1/ETH_TXD1	A	PA1 input/output	DU/ETH	Available
DR0/ETH_TXD0	A	PA0 input/output	DU/ETH	Available
$\overline{\text{VSYNC}}$/HSPI_CLK	B	PB7 input/output	DU/HSPI	—
ODDF/$\overline{\text{HSPI\_CS}}$	B	PB6 input/output	DU/HSPI	—
DG5/ETH_MDIO	B	PB5 input/output	DU/ETH	—
DG4/ETH_RX_CLK	B	PB4 input/output	DU/ETH	—
DG3/ETH_MDC	B	PB3 input/output	DU/ETH	—
DG2/ETH_COL	B	PB2 input/output	DU/ETH	—

Pin Name	Port	GPIO	Selectable Module	GPIO Interrupt
DG1/ETH_TX_CLK	B	PB1 input/output	DU/ETH	—
DG0/ETH_CRS	B	PB0 input/output	DU/ETH	—
DCLKIN/HSPI_RX	C	PC7 input/output	DU/HSPI	—
$\overline{\text{HSYNC}}$/HSPI_TX	C	PC6 input/output	DU/HSPI	—
DB5/ETH_RXD3	C	PC5 input/output	DU/ETH	—
DB4/ETH_RXD2	C	PC4 input/output	DU/ETH	—
DB3/ETH_RXD1	C	PC3 input/output	DU/ETH	—
DB2/ETH_RXD0	C	PC2 input/output	DU/ETH	—
DB1/ETH_RX_DV	C	PC1 input/output	DU/ETH	—
DB0/ETH_RX_ER	C	PC0 input/output	DU/ETH	—
DCLKOUT	D	PD7 input/output	DU	—
SCIF1_SCK	D	PD6 input/output	SCIF	—
SCIF1_RXD	D	PD5 input/output	SCIF	—
SCIF1_TXD	D	PD4 input/output	SCIF	—
DACK1/$\overline{\text{BACK}}$/FALE	D	PD3 input/output	LBSC/LBSC/FLTCTL	—
DACK0/FCLE	D	PD2 input/output	LBSC/DU/ETH	—
$\overline{\text{DREQ1}}$/$\overline{\text{BREQ}}$/USB_OVC1	D	PD1 input/output	DMAC/LBSC/USB	—
$\overline{\text{DREQ0}}$/USB_OVC0	D	PD0 input/output	DMAC/USB	—
USB_PENC1	E	PE7 input/output	USB	—
USB_PENC0	E	PE6 input/output	USB	—
HAC1_SDOUT/SSI1_SDATA/SDIF1CMD	F	PF7 input/output	HAC/SSI/SDIF	—
HAC1_SDIN/SSI1_SCK/SDIF1CD	F	PF6 input/output	HAC/SSI/SDIF	—
HAC1_SYNC/SSI1_WS/SDIF1WP	F	PF5 input/output	HAC/SSI/SDIF	—
HAC1_BITCLK/SSI1_CLK/SDIF1CLK	F	PF4 input/output	HAC/SSI/SDIF	—
HAC0_SDOUT/SSI0_SDATA/SDIF1D3	F	PF3 input/output	HAC/SSI/SDIF	—
HAC0_SDIN/SSI0_SCK/SDIF1D2	F	PF2 input/output	HAC/SSI/SDIF	—
HAC0_SYNC/SSI0_WS/SDIF1D1	F	PF1 input/output	HAC/SSI/SDIF	Available
HAC0_BITCLK/SSI0_CLK/SDIF1D0	F	PF0 input/output	HAC/SSI/SDIF	Available
SCIF3_SCK/SSI2_SDATA	G	PG7 input/output	SCIF/SSI	—
SCIF3_RXD/TCLK/SSI2_SCK	G	PG6 input/output	SCIF/TMU/SSI	—
SCIF3_TXD/$\overline{\text{HAC\_RES}}$/SSI2_WS	G	PG5 input/output	SCIF/HAC/SSI	—
MODE7/DACK3/SDIF0CMD	H	PH7 input/output	LBSC/SDIF	

Pin Name	Port	GPIO	Selectable Module	GPIO Interrupt
MODE6/DACK2/$\overline{\text{SDIF0CD}}$	H	PH6 input/output	LBSC/SDIF	
MODE5/$\overline{\text{DREQ3}}$/SDIF0WP	H	PH5 input/output	DMAC/SDIF	
MODE4/$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{DREQ2}}$/SDIF0CLK	H	PH4 input/output	SCIF/DMAC/SDIF	
MODE3/$\overline{\text{SCIF0\_RTS}}$/$\overline{\text{IRL7}}$/SDIF0D3	H	PH3 input/output	SCIF/INTC/SDIF	
MODE2/SCIF0_SCK/$\overline{\text{IRL6}}$/SDIF0D2	H	PH2 input/output	SCIF/INTC/SDIF	—
MODE1/SCIF0_RXD/$\overline{\text{IRL5}}$/SDIF0D1	H	PH1 input/output	SCIF/INTC/SDIF	Available
MODE0/SCIF0_TXD/$\overline{\text{IRL4}}$/SDIF0D0	H	PH0 input/output	SCIF/INTC/SDIF	Available
MODE14/SCIF5_SCK/$\overline{\text{FRB}}$	J	PJ7 input/output	SCIF/FLCTL	—
MODE13/SCIF5_RXD/$\overline{\text{IOIS16}}$	J	PJ6 input/output	SCIF/LBSC	—
MODE12/SCIF5_TXD/$\overline{\text{CE2B}}$	J	PJ5 input/output	SCIF/LBSC	—
MODE11/DRAK3/$\overline{\text{CE2A}}$	J	PJ4 input/output	DMAC/LBSC	—
MODE10/SCIF4_SCK/DRAK2/SSI3_WS	J	PJ3 input/output	SCIF/DMAC/SSI	—
MODE9/SCIF4_RXD/DRAK1/SSI3_SDATA	J	PJ2 input/output	SCIF/DMAC/SSI	Available
MODE8/SCIF4_TXD/DRAK0/SSI3_SCK/ $\overline{\text{FSE}}$	J	PJ1 input/output	SCIF/DMAC/SSI/FLCTL	Available

Note: The module that uses this pin is selected by the peripheral module select registers 1 and 2 (P1MSELR and P2MSELR).

30.2 Register Descriptions

The following registers are provided to control the GPIO ports.

Table 30.2 Register Configuration (1)

Register Name	Abbr.	R/W	P4 Address*1	Area 7 Address*1	Access Size*2	Sync Clock
Port A control register	PACR	R/W	H'FFCC 0000	H'1FCC 0000	16	Pck
Port B control register	PBCR	R/W	H'FFCC 0002	H'1FCC 0002	16	Pck
Port C control register	PCCR	R/W	H'FFCC 0004	H'1FCC 0004	16	Pck
Port D control register	PDCR	R/W	H'FFCC 0006	H'1FCC 0006	16	Pck
Port E control register	PECR	R/W	H'FFCC 0008	H'1FCC 0008	16	Pck
Port F control register	PFCR	R/W	H'FFCC 000A	H'1FCC 000A	16	Pck
Port G control register	PGCR	R/W	H'FFCC 000C	H'1FCC 000C	16	Pck
Port H control register	PHCR	R/W	H'FFCC 000E	H'1FCC 000E	16	Pck
Port J control register	PJCR	R/W	H'FFCC 0010	H'1FCC 0010	16	Pck
Port A data register	PADR	R/W	H'FFCC 0020	H'1FCC 0020	8	Pck
Port B data register	PBDR	R/W	H'FFCC 0022	H'1FCC 0022	8	Pck
Port C data register	PCDR	R/W	H'FFCC 0024	H'1FCC 0024	8	Pck
Port D data register	PDDR	R/W	H'FFCC 0026	H'1FCC 0026	8	Pck
Port E data register	PEDR	R/W	H'FFCC 0028	H'1FCC 0028	8	Pck
Port F data register	PFDR	R/W	H'FFCC 002A	H'1FCC 002A	8	Pck
Port G data register	PGDR	R/W	H'FFCC 002C	H'1FCC 002C	8	Pck
Port H data register	PHDR	R/W	H'FFCC 002E	H'1FCC 002E	8	Pck
Port J data register	PJDR	R/W	H'FFCC 0030	H'1FCC 0030	8	Pck
Port A pull-up control register	PAPUPR	R/W	H'FFCC 0040	H'1FCC 0040	8	Pck
Port B pull-up control register	PBPUPR	R/W	H'FFCC 0042	H'1FCC 0042	8	Pck
Port C pull-up control register	PCPUPR	R/W	H'FFCC 0044	H'1FCC 0044	8	Pck
Port D pull-up control register	PDPUPR	R/W	H'FFCC 0046	H'1FCC 0046	8	Pck
Port E pull-up control register	PEPUPR	R/W	H'FFCC 0048	H'1FCC 0048	8	Pck
Port H pull-up control register	PHPUPR	R/W	H'FFCC 004E	H'1FCC 004E	8	Pck
Port J pull-up control register	PJPUPR	R/W	H'FFCC 0050	H'1FCC 0050	8	Pck
Input pin pull-up control register 1	PPUPR1	R/W	H'FFCC 0060	H'1FCC 0060	16	Pck
Input pin pull-up control register 2	PPUPR2	R/W	H'FFCC 0062	H'1FCC 0062	16	Pck

Register Name	Abbr.	R/W	P4 Address*[1]	Area 7 Address*[1]	Access Size*[2]	Sync Clock
Peripheral module select register 1	P1MSELR	R/W	H'FFCC 0080	H'1FCC 0080	16	Pck
Peripheral module select register 2	P2MSELR	R/W	H'FFCC 0082	H'1FCC 0082	16	Pck

Notes: 1. The P4 area address uses the P4 area of the logical address area. The area 7 address is accessed from area 7 of the physical address space using the TLB.
2. There are 8-bit and 16-bit registers. The registers must be accessed in the designate size.

Table 30.2 Register Configuration (2)

Register Name	Abbr.	Power-on Reset by $\overline{\text{RESET}}$ Pin / WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep/Light Sleep by SLEEP Instruction	Module Standby
Port A control register	PACR	H'FFFF	Retained	Retained	Retained
Port B control register	PBCR	H'FFFF	Retained	Retained	Retained
Port C control register	PCCR	H'FFFF	Retained	Retained	Retained
Port D control register	PDCR	H'FF00	Retained	Retained	Retained
Port E control register	PECR	H'0FFF	Retained	Retained	Retained
Port F control register	PFCR	H'FFFF	Retained	Retained	Retained
Port G control register	PGCR	H'FFFF	Retained	Retained	Retained
Port H control register	PHCR	H'FFFF	Retained	Retained	Retained
Port J control register	PJCR	H'3FC3	Retained	Retained	Retained
Port A data register	PADR	H'xx	Retained	Retained	Retained
Port B data register	PBDR	H'xx	Retained	Retained	Retained
Port C data register	PCDR	H'xx	Retained	Retained	Retained
Port D data register	PDDR	H'xx	Retained	Retained	Retained
Port E data register	PEDR	H'00	Retained	Retained	Retained
Port F data register	PFDR	H'xx	Retained	Retained	Retained
Port G data register	PGDR	H'x0	Retained	Retained	Retained
Port H data register	PHDR	H'xx	Retained	Retained	Retained
Port J data register	PJDR	H'xx	Retained	Retained	Retained
Port A pull-up control register	PAPUPR	H'FF	Retained	Retained	Retained
Port B pull-up control register	PBPUPR	H'FF	Retained	Retained	Retained
Port C pull-up control register	PCPUPR	H'FF	Retained	Retained	Retained
Port D pull-up control register	PDPUPR	H'FF	Retained	Retained	Retained
Port E pull-up control register	PEPUPR	H'FF	Retained	Retained	Retained
Port H pull-up control register	PHPUPR	H'FF	Retained	Retained	Retained
Port J pull-up control register	PJPUPR	H'FF	Retained	Retained	Retained
Input pin pull-up control register 1	PPUPR1	H'FFFF	Retained	Retained	Retained
Input pin pull-up control register 2	PPUPR2	H'FFFF	Retained	Retained	Retained
Peripheral module select register 1	P1MSELR	H'0000	Retained	Retained	Retained
Peripheral module select register 2	P2MSELR	H'0000	Retained	Retained	Retained

30.2.1 Port A Control Register (PACR)

PACR is a 16-bit readable/writable register that selects the pin function and controls the input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA7 MD1	PA7 MD0	PA6 MD1	PA6 MD0	PA5 MD1	PA5 MD0	PA4 MD1	PA4 MD0	PA3 MD1	PA3 MD0	PA2 MD1	PA2 MD0	PA1 MD1	PA1 MD0	PA0 MD1	PA0 MD0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PA7MD1	1	R/W	PA7 Mode
14	PA7MD0	1	R/W	00: DU/ETH module (CDE/ETH_MAGIC)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PA6MD1	1	R/W	PA6 Mode
12	PA6MD0	1	R/W	00: DU/ETH module (DISP/ETH_LINK)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PA5MD1	1	R/W	PA5 Mode
10	PA5MD0	1	R/W	00: DU/ETH module (DR5/ETH_TX_ER)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9	PA4MD1	1	R/W	PA4 Mode
8	PA4MD0	1	R/W	00: DU/ETH module (DR4/ETH_TX_EN)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7	PA3MD1	1	R/W	PA3 Mode
6	PA3MD0	1	R/W	00: DU/ETH module (DR3/ETH_TXD3)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Bit	Bit Name	Initial Value	R/W	Description
5	PA2MD1	1	R/W	PA2 Mode
4	PA2MD0	1	R/W	00: DU/ETH module (DR2/ETH_TXD2)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PA1MD1	1	R/W	PA1 Mode
2	PA1MD0	1	R/W	00: DU/ETH module (DR1/ETH_TXD1)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PA0MD1	1	R/W	PA0 Mode
0	PA0MD0	1	R/W	00: DU/ETH module (DR0/ETH_TXD0)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Note: * The module that uses this pin is selected by the peripheral module select registers 1 (P1MSELR).

30.2.2 Port B Control Register (PBCR)

PBCR is a 16-bit readable/writable register that selects the pin function and controls the input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB7 MD1	PB7 MD0	PB6 MD1	PB6 MD0	PB5 MD1	PB5 MD0	PB4 MD1	PB4 MD0	PB3 MD1	PB3 MD0	PB2 MD1	PB2 MD0	PB1 MD1	PB1 MD0	PB0 MD1	PB0 MD0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PB7MD1	1	R/W	PB7 Mode
14	PB7MD0	1	R/W	00: DU/HSPI module ($\overline{\text{VSYNC}}$/HSPI_CLK)*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
13	PB6MD1	1	R/W	PB6 Mode
12	PB6MD0	1	R/W	00: DU/HSPI module (ODDF/$\overline{\text{HSPI\_CS}}$)*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
11	PB5MD1	1	R/W	PB5 Mode
10	PB5MD0	1	R/W	00: DU/Ether module (DG5_ETH_MDIO)*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
9	PB4MD1	1	R/W	PB4 Mode
8	PB4MD0	1	R/W	00: DU/Ether module (DG4/ETH_RX_CLK)*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
7	PB3MD1	1	R/W	PB3 Mode
6	PB3MD0	1	R/W	00: DU/Ether module (DG3/ETH_MDC)*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)

Bit	Bit Name	Initial Value	R/W	Description
5	PB2MD1	1	R/W	PB2 Mode
4	PB2MD0	1	R/W	00: DU/Ether module (DG2_ETH_COL)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PB1MD1	1	R/W	PB1 Mode
2	PB1MD0	1	R/W	00: DU/Ether module (DG1/ETH_TX_CLK)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PB0MD1	1	R/W	PB0 Mode
0	PB0MD0	1	R/W	00: DU/Ether module (DG0_ETH_CRS)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Note: * The module that uses this pin is selected by the peripheral module select registers 1 (P1MSELR).

30.2.3 Port C Control Register (PCCR)

PCCR is a 16-bit readable/writable register that selects the pin function and controls the input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC7 MD1	PC7 MD0	PC6 MD1	PC6 MD0	PC5 MD1	PC5 MD0	PC4 MD1	PC4 MD0	PC3 MD1	PC3 MD0	PC2 MD1	PC2 MD0	PC1 MD1	PC1 MD0	PC0 MD1	PC0 MD0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PC7MD1	1	R/W	PC7 Mode
14	PC7MD0	1	R/W	00: DU/HSPI module ($\overline{\text{DCLKIN}}$/HSPI_RX)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PC6MD1	1	R/W	PC6 Mode
12	PC6MD0	1	R/W	00: DU/HSPI module ($\overline{\text{HSYNC}}$/HSPI_TX)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PC5MD1	1	R/W	PC5 Mode
10	PC5MD0	1	R/W	00: DU/Ether module (DB5/ETH_RXD3)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9	PC4MD1	1	R/W	PC4 Mode
8	PC4MD0	1	R/W	00: DU/Ether module (DB4/ETH_RXD2)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7	PC3MD1	1	R/W	PC3 Mode
6	PC3MD0	1	R/W	00: DU/Ether module (DB3/ETH_RXD1)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Bit	Bit Name	Initial Value	R/W	Description
5	PC2MD1	1	R/W	PC2 Mode
4	PC2MD0	1	R/W	00: DU/Ether module (DB2/ETH_RXD0)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PC1MD1	1	R/W	PC1 Mode
2	PC1MD0	1	R/W	00: DU/Ether module (DB1/ETH_RX_DV)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PC0MD1	1	R/W	PC0 Mode
0	PC0MD0	1	R/W	00: DU/Ether module (DB0/ETH_RX_ER)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Note: * The module that uses this pin is selected by the peripheral module select registers 1 (P1MSELR).

30.2.4 Port D Control Register (PDCR)

PDCR is a 16-bit readable/writable register that selects the pin function and controls the input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD7 MD1	PD7 MD0	PD6 MD1	PD6 MD0	PD5 MD1	PD5 MD0	PD4 MD1	PD4 MD0	PD3 MD1	PD3 MD0	PD2 MD1	PD2 MD0	PD1 MD1	PD1 MD0	PD0 MD1	PD0 MD0
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PD7MD1	1	R/W	PD7 Mode
14	PD7MD0	1	R/W	00: DU module (DCLKOUT) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PD6MD1	1	R/W	PD6 Mode
12	PD6MD0	1	R/W	00: SCIF module (SCIF1_SLK) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PD5MD1	1	R/W	PD5 Mode
10	PD5MD0	1	R/W	00: SCIF module (SCIF1_RXD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9	PD4MD1	1	R/W	PD4 Mode
8	PD4MD0	1	R/W	00: SCIF module (SCIF1_TXD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7	PD3MD1	0	R/W	PD3 Mode
6	PD3MD0	0	R/W	00: DMAC/LBSC/FLCTL module (DACK1/$\overline{\text{BACK}}$/FALE)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Bit	Bit Name	Initial Value	R/W	Description
5	PD2MD1	0	R/W	PD2 Mode
4	PD2MD0	0	R/W	00: DMAC/FLCTL module (DACK0/FCLE)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PD1MD1	0	R/W	PD1 Mode
2	PD1MD0	0	R/W	00: DMAC/LBSC/USB module ($\overline{\text{DREQ1}}$/$\overline{\text{BREQ}}$/USB_OVC1)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PD0MD1	0	R/W	PD0 Mode
0	PD0MD0	0	R/W	00: DMAC/USB module ($\overline{\text{DREQ0}}$/USB_OVC0)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Note: * The module that uses this pin is selected by the peripheral module select registers 1 (P1MSELR).

30.2.5 Port E Control Register (PECR)

PECR is a 16-bit readable/writable register that selects the pin function and controls the input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE7MD1	PE7MD0	PE6MD1	PE6MD0	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PE7MD1	0	R/W	PE7 Mode
14	PE7MD0	0		00: USB module (USB_PENC1) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PE6MD1	0	R/W	PE6 Mode
12	PE6MD0	0	R/W	00: USB module (USB_PENC0) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11 to 0	—	All 1	R/W	Reserved These bits are always read as 1, and the write value should always be 1.

30.2.6 Port F Control Register (PFCR)

PFCR is a 16-bit readable/writable register that selects the pin function and controls the input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF7 MD1	PF7 MD0	PF6 MD1	PF6 MD0	PF5 MD1	PF5 MD0	PF4 MD1	PF4 MD0	PF3 MD1	PF3 MD0	PF2 MD1	PF2 MD0	PF1 MD1	PF1 MD0	PF0 MD1	PF0 MD0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PF7MD1	1	R/W	PF7 Mode
14	PF7MD0	1	R/W	00: HAC/SSI/SDIF module (HAC1_SDOUT/SSI1_SDATA/SDIF1CMD)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PF6MD1	1	R/W	PF6 Mode
12	PF6MD0	1	R/W	00: HAC/SSI/SDIF module (HAC1_SDIN/SSI1_SCK/SDIF1CD)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PF5MD1	1	R/W	PF5 Mode
10	PF5MD0	1	R/W	00: HAC/SSI/SDIF module (HAC1_SYNC/SSI1_WS/SDIF1WP)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9	PF4MD1	1	R/W	PF4 Mode
8	PF4MD0	1	R/W	00: HAC/SSI/SDIF module (HAC1_BITCLK/SSI1_CLK/SDIF1CLK)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Bit	Bit Name	Initial Value	R/W	Description
7	PF3MD1	1	R/W	PF3 Mode
6	PF3MD0	1	R/W	00: HAC/SSI/SDIF module (HAC0_SDOUT/SSI0_SDATA/SDIF1D3)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PF2MD1	1	R/W	PF2 Mode
4	PF2MD0	1	R/W	00: HAC/SSI/SDIF module (HAC0_SDIN/SSI0_SCK/SDIF1D2)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PF1MD1	1	R/W	PF1 Mode
2	PF1MD0	1	R/W	00: HAC/SSI/SDIF module (HAC0_SYNC/SSI0_WS/SDIF1D1)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PF0MD1	1	R/W	PF0 Mode
0	PF0MD0	1	R/W	00: HAC/SSI/SDIF module (HAC0_BITCLK/SSI0_CLK/SDIF1D0)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Note: * The module that uses this pin can be selected by the peripheral module select register 2 (P2MSELR).

30.2.7 Port G Control Register (PGCR)

PGCR is a 16-bit readable/writable register that selects the pin function and controls the input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG7 MD1	PG7 MD0	PG6 MD1	PG6 MD0	PG5 MD1	PG5 MD0	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PG7MD1	1	R/W	PG7 Mode
14	PG7MD0	1	R/W	00: SCIF/SSI module (SCIF3_SCK/SSI2_SDATA)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PG6MD1	1	R/W	PG6 Mode
12	PG6MD0	1	R/W	00: SCIF/TMU/SSI module (SCIF3_RXD/TCLK/SSI2_SCK)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PG5MD1	1	R/W	PG5 Mode
10	PG5MD0	1	R/W	00: SCIF/HAC/SSI module (SCIF3_TXD/$\overline{\text{HAC\_RES}}$/SSI2_WS)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9 to 0	—	All 1	R	Reserved These bits are always read as 1, and the write value should always be 1.

Note: * The module that uses this pin can be selected by the peripheral module select register 1 (P1MSELR).

30.2.8 Port H Control Register (PHCR)

PHCR is a 16-bit readable/writable register that selects the pin function and controls the input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH7 MD1	PH7 MD0	PH6 MD1	PH6 MD0	PH5 MD1	PH5 MD0	PH4 MD1	PH4 MD0	PH3 MD1	PH3 MD0	PH2 MD1	PH2 MD0	PH1 MD1	PH1 MD0	PH0 MD1	PH0 MD0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PH7MD1	1	R/W	PH7 Mode
14	PH7MD0	1	R/W	00: DMAC/SDIF module (MODE7/DACK3/SDIF0CMD)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PH6MD1	1	R/W	PH6 Mode
12	PH6MD0	1	R/W	00: DMAC/SDIF module (MODE6/DACK2/$\overline{\text{SDIF0CD}}$)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PH5MD1	1	R/W	PH5 Mode
10	PH5MD0	1	R/W	00: DMAC/SDIF module (MODE5/$\overline{\text{DREQ3}}$/SDIF0WP)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9	PH4MD1	1	R/W	PTH4 Mode
8	PH4MD0	1	R/W	00: SCIF/DMAC/SDIF module (MODE4/$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{DREQ2}}$/SDIF0CLK)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Bit	Bit Name	Initial Value	R/W	Description
7	PH3MD1	1	R/W	PH3 Mode
6	PH3MD0	1	R/W	00: SCIF/INTC/SDIF module (MODE3/$\overline{\text{SCIF0\_RTS}}$/$\overline{\text{IRL7}}$/SDIF0D3)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PH2MD1	1	R/W	PH2 Mode
4	PH2MD0	1	R/W	00: SCIF/INTC/SDIF module (MODE2/SCIF0_SCK/$\overline{\text{IRL6}}$/SDIF0D2)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PH1MD1	1	R/W	PH1 Mode
2	PH1MD0	1	R/W	00: SCIF/INTC/SDIF module (MODE1/SCIF0_RXD/$\overline{\text{IRL5}}$/SDIF0D1)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PH0MD1	1	R/W	PH0 Mode
0	PH0MD0	1	R/W	00: SCIF/INTC/SDIF module (MODE0/SCIF0_TXD/$\overline{\text{IRL4}}$/SDIF0D0)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

Note: * The module that uses this pin can be selected by the peripheral module select register 2 (P2MSELR).

30.2.9 Port J Control Register (PJCR)

PJCR is a 16-bit readable/writable register that selects the pin function and controls the input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ7 MD1	PJ7 MD0	PJ6 MD1	PJ6 MD0	PJ5 MD1	PJ5 MD0	PJ4 MD1	PJ4 MD0	PJ3 MD1	PJ3 MD0	PJ2 MD1	PJ2 MD0	PJ1 MD1	PJ1 MD0	—	—
Initial value:	0	0	1	1	1	1	1	1	1	1	0	0	0	0	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PJ7MD1	0	R/W	PJ7 Mode
14	PJ7MD0	0	R/W	00: SCIF/FLCTL module (MODE14/SCIF5_SCK/$\overline{\text{FRB}}$)*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
13	PJ6MD1	1	R/W	PJ6 Mode
12	PJ6MD0	1	R/W	00: SCIF/LBSC module (MODE13/SCIF5_RXD/$\overline{\text{IOIS16}}$)*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
11	PJ5MD1	1	R/W	PJ5 Mode
10	PJ5MD0	1	R/W	00: SCIF/LBSC module (MODE12/SCIF5_TXD/$\overline{\text{CE2B}}$)*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
9	PJ4MD1	1	R/W	PJ4 Mode
8	PJ4MD0	1	R/W	00: DMAC/LBSC module (MODE11/DRAK3/$\overline{\text{CE2A}}$)*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)

Bit	Bit Name	Initial Value	R/W	Description
7	PJ3MD1	1	R/W	PJ3 Mode
6	PJ3MD0	1	R/W	00: SCIF/DMAC/SSI module (MODE10/SCIF4_SCK/DRAK2/SSI3_WS)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PJ2MD1	0	R/W	PJ2 Mode
4	PJ2MD0	0	R/W	00: SCIF/DMAC/SSI module (MODE9/SCIF4_RXD/DRAK1/SSI3_SDATA)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PJ1MD1	0	R/W	PJ1 Mode
2	PJ1MD0	0	R/W	00: SCIF/DMAC/SSI/FLCTL module (MODE8/SCIF4_TXD/DRAK0/SSI3_SCK/$\overline{\text{FSE}}$)* 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1 to 0	—	All 1	R	Reserved These bits are always read as 1, and the write value should always be 1.

Note: * The module that uses this pin can be selected by the peripheral module select register 2 (P2MSELR).

30.2.10 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores port A data.

Bit:	7	6	5	4	3	2	1	0
	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT
Initial value:	x	x	x	x	x	x	x	x
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DT	Pin input	R/W	These bits store output data of a pin which is used as a general-purpose output port. When the pin functions as a general-purpose output port, reading the port will read out the value of the corresponding bit of this register. When the pin functions as a general-purpose input port, reading the port will read out the status of the corresponding pin.
6	PA6DT	Pin input	R/W	
5	PA5DT	Pin input	R/W	
4	PA4DT	Pin input	R/W	
3	PA3DT	Pin input	R/W	
2	PA2DT	Pin input	R/W	
1	PA1DT	Pin input	R/W	
0	PA0DT	Pin input	R/W	

30.2.11 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores port B data.

Bit:	7	6	5	4	3	2	1	0
	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT
Initial value:	x	x	x	x	x	x	x	x
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DT	Pin input	R/W	These bits store output data of a pin which is used as a general-purpose output port. When the pin functions as a general-purpose output port, reading the port will read out the value of the corresponding bit of this register. When the pin functions as a general-purpose input port, reading the port will read out the status of the corresponding pin.
6	PB6DT	Pin input	R/W	
5	PB5DT	Pin input	R/W	
4	PB4DT	Pin input	R/W	
3	PB3DT	Pin input	R/W	
2	PB2DT	Pin input	R/W	
1	PB1DT	Pin input	R/W	
0	PB0DT	Pin input	R/W	

30.2.12 Port C Data Register (PCDR)

PCDR is an 8-bit readable/writable register that stores port C data.

Bit:	7	6	5	4	3	2	1	0
	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT
Initial value:	x	x	x	x	x	x	x	x
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DT	Pin input	R/W	These bits store output data of a pin which is used as a general-purpose output port. When the pin functions as a general-purpose output port, reading the port will read out the value of the corresponding bit of this register. When the pin functions as a general-purpose input port, reading the port will read out the status of the corresponding pin.
6	PC6DT	Pin input	R/W	
5	PC5DT	Pin input	R/W	
4	PC4DT	Pin input	R/W	
3	PC3DT	Pin input	R/W	
2	PC2DT	Pin input	R/W	
1	PC1DT	Pin input	R/W	
0	PC0DT	Pin input	R/W	

30.2.13 Port D Data Register (PDDR)

PDDR is an 8-bit readable/writable register that stores port D data.

Bit:	7	6	5	4	3	2	1	0
	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT
Initial value:	x	x	x	x	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DT	Pin input	R/W	These bits store output data of a pin which is used as a general-purpose output port. When the pin functions as a general-purpose output port, reading the port will read out the value of the corresponding bit of this register. When the pin functions as a general-purpose input port, reading the port will read out the status of the corresponding pin.
6	PD6DT	Pin input	R/W	
5	PD5DT	Pin input	R/W	
4	PD4DT	Pin input	R/W	
3	PD3DT	0	R/W	
2	PD2DT	0	R/W	
1	PD1DT	0	R/W	
0	PD0DT	0	R/W	

30.2.14 Port E Data Register (PEDR)

PEDR is an 8-bit readable/writable register that stores port E data.

Bit:	7	6	5	4	3	2	1	0
	PE7DT	PE6DT	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DT	0	R/W	These bits store output data of a pin which is used as a general-purpose output port. When the pin functions as a general-purpose output port, reading the port will read out the value of the corresponding bit of this register. When the pin functions as a general-purpose input port, reading the port will read out the status of the corresponding pin.
6	PE6DT	0	R/W	
5 to 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

30.2.15 Port F Data Register (PFDR)

PFDR is an 8-bit readable/writable register that stores port F data.

Bit:	7	6	5	4	3	2	1	0
	PF7DT	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF0DT
Initial value:	x	x	x	x	x	x	x	x
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DT	Pin input	R/W	These bits store output data of a pin which is used as a general-purpose output port. When the pin functions as a general-purpose output port, reading the port will read out the value of the corresponding bit of this register. When the pin functions as a general-purpose input port, reading the port will read out the status of the corresponding pin.
6	PF6DT	Pin input	R/W	
5	PF5DT	Pin input	R/W	
4	PF4DT	Pin input	R/W	
3	PF3DT	Pin input	R/W	
2	PF2DT	Pin input	R/W	
1	PF1DT	Pin input	R/W	
0	PF0DT	Pin input	R/W	

30.2.16 Port G Data Register (PGDR)

PGDR is an 8-bit readable/writable register that stores port G data.

Bit:	7	6	5	4	3	2	1	0
	PG7DT	PG6DT	PG5DT	—	—	—	—	—
Initial value:	x	x	x	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	PG7DT	Pin input	R/W	These bits store output data of a pin which is used as a general-purpose output port. When the pin functions as a general-purpose output port, reading the port will read out the value of the corresponding bit of this register. When the pin functions as a general-purpose input port, reading the port will read out the status of the corresponding pin.
6	PG6DT	Pin input	R/W	
5	PG5DT	Pin input	R/W	
4 to 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

30.2.17 Port H Data Register (PHDR)

PHDR is an 8-bit readable/writable register that stores port H data.

Bit:	7	6	5	4	3	2	1	0
	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT
Initial value:	x	x	x	x	x	x	x	x
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PH7DT	Pin input	R/W	These bits store output data of a pin which is used as a general-purpose output port. When the pin functions as a general-purpose output port, reading the port will read out the value of the corresponding bit of this register. When the pin functions as a general-purpose input port, reading the port will read out the status of the corresponding pin.
6	PH6DT	Pin input	R/W	
5	PH5DT	Pin input	R/W	
4	PH4DT	Pin input	R/W	
3	PH3DT	Pin input	R/W	
2	PH2DT	Pin input	R/W	
1	PH1DT	Pin input	R/W	
0	PH0DT	Pin input	R/W	

30.2.18 Port J Data Register (PJDR)

PJDR is an 8-bit readable/writable register that stores port J data.

Bit:	7	6	5	4	3	2	1	0
	PJ7DT	PJ6DT	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	—
Initial value:	0	x	x	x	x	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7DT	0	R/W	These bits store output data of a pin which is used as a general-purpose output port. When the pin functions as a general-purpose output port, reading the port will read out the value of the corresponding bit of this register. When the pin functions as a general-purpose input port, reading the port will read out the status of the corresponding pin.
6	PJ6DT	Pin input	R/W	
5	PJ5DT	Pin input	R/W	
4	PJ4DT	Pin input	R/W	
3	PJ3DT	Pin input	R/W	
2	PJ2DT	0	R/W	
1	PJ1DT	0	R/W	
0	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.

30.2.19 Port A Pull-Up Control Register (PAPUPR)

PAPUPR is an 8-bit readable/writable register that performs the pull-up control for each of the port A7 to A0 (PA7 to PA0) pins when the pins are used by peripheral modules. When the port a pins are used by the GPIO, the setting for this register is ignored.

Bit:	7	6	5	4	3	2	1	0
	PA7 PUPR	PA6 PUPR	PA5 PUPR	PA4 PUPR	PA3 PUPR	PA2 PUPR	PA1 PUPR	PA0 PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PUPR	1	R/W	Pull-up of the Port An pin can be controlled independently. 0: PAn pull-up off 1: PAn pull-up on
6	PA6PUPR	1	R/W	
5	PA5PUPR	1	R/W	
4	PA4PUPR	1	R/W	
3	PA3PUPR	1	R/W	
2	PA2PUPR	1	R/W	
1	PA1PUPR	1	R/W	
0	PA0PUPR	1	R/W	

Note: n = 7 to 0

30.2.20 Port B Pull-Up Control Register (PBPUPR)

PBPUPR is an 8-bit readable/writable register that performs the pull-up control for each of the port B7 to B0 (PB7 to PB0) pins when the pins are used by peripheral modules. When the port a pins are used by the GPIO, the setting for this register is ignored.

Bit:	7	6	5	4	3	2	1	0
	PB7 PUPR	PB6 PUPR	PB5 PUPR	PB4 PUPR	PB3 PUPR	PB2 PUPR	PB1 PUPR	PB0 PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PUPR	1	R/W	Pull-up of the Port Bn pin can be controlled independently. 0: PBn pull-up off 1: PBn pull-up on
6	PB6PUPR	1	R/W	
5	PB5PUPR	1	R/W	
4	PB4PUPR	1	R/W	
3	PB3PUPR	1	R/W	
2	PB2PUPR	1	R/W	
1	PB1PUPR	1	R/W	
0	PB0PUPR	1	R/W	

Note: n = 7 to 0

30.2.21 Port C Pull-Up Control Register (PCPUPR)

PCPUPR is an 8-bit readable/writable register that performs the pull-up control for each of the port C7 to C0 (PC7 to PC0) pins when the pins are used by peripheral modules. When the port a pins are used by the GPIO, the setting for this register is ignored.

Bit:	7	6	5	4	3	2	1	0
	PC7 PUPR	PC6 PUPR	PC5 PUPR	PC4 PUPR	PC3 PUPR	PC2 PUPR	PC1 PUPR	PC0 PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PUPR	1	R/W	Pull-up of the Port Cn pin can be controlled independently. 0: PCn pull-up off 1: PCn pull-up on
6	PC6PUPR	1	R/W	
5	PC5PUPR	1	R/W	
4	PC4PUPR	1	R/W	
3	PC3PUPR	1	R/W	
2	PC2PUPR	1	R/W	
1	PC1PUPR	1	R/W	
0	PC0PUPR	1	R/W	

Note: n = 7 to 0

30.2.22 Port D Pull-Up Control Register (PDPUPR)

PDPUPR is an 8-bit readable/writable register that performs the pull-up control for each of the port D7 to D0 (PD7 to PD0) pins when the pins are used by peripheral modules. When the port a pins are used by the GPIO, the setting for this register is ignored.

Bit:	7	6	5	4	3	2	1	0
	PD7 PUPR	PD6 PUPR	PD5 PUPR	PD4 PUPR	PD3 PUPR	PD2 PUPR	PD1 PUPR	PD0 PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PUPR	1	R/W	Pull-up of the Port Dn pin can be controlled independently. 0: PDn pull-up off 1: PDn pull-up on
6	PD6PUPR	1	R/W	
5	PD5PUPR	1	R/W	
4	PD4PUPR	1	R/W	
3	PD3PUPR	1	R/W	
2	PD2PUPR	1	R/W	
1	PD1PUPR	1	R/W	
0	PD0PUPR	1	R/W	

Note: n = 7 to 0

30.2.23 Port E Pull-Up Control Register (PEPUPR)

PLPUPR is an 8-bit readable/writable register that performs the pull-up control for each of the port E7 to E4 (PE7 to PE4) pins when the port L pins are used by peripheral modules. When the port L pins are used by the GPIO, the setting for this register is ignored.

Bit:	7	6	5	4	3	2	1	0
	PE7 PUPR	PE6 PUPR	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	PE7PUPR	1	R/W	Pull-up of each Port En pin can be controlled independently. 0: PEn pull-up off 1: PEn pull-up on
6	PE6PUPR	1	R/W	
5 to 0	—	All 1	R	Reserved These bits are always read as 1, and the write value should always be 1.

Note: n = 7, 6

30.2.24 Port F Pull-Up Control Register (PFPUPR)

PFPUPR is an 8-bit readable/writable register that performs the pull-up control for each of the port F7 to F0 (PF7 to PF0) pins when the pins are used by peripheral modules. When the port a pins are used by the GPIO, the setting for this register is ignored.

Bit:	7	6	5	4	3	2	1	0
	PF7 PUPR	PF6 PUPR	PF5 PUPR	PF4 PUPR	PF3 PUPR	PF2 PUPR	PF1 PUPR	PF0 PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PF7PUPR	1	R/W	Pull-up of the Port Fn pin can be controlled independently. 0: PFn pull-up off 1: PFn pull-up on
6	PF6PUPR	1	R/W	
5	PF5PUPR	1	R/W	
4	PF4PUPR	1	R/W	
3	PF3PUPR	1	R/W	
2	PF2PUPR	1	R/W	
1	PF1PUPR	1	R/W	
0	PF0PUPR	1	R/W	

Note: n = 7 to 0

30.2.25 Port G Pull-Up Control Register (PGPUPR)

PGPUPR is an 8-bit readable/writable register that performs the pull-up control for each of the port G7 to G0 (PG7 to PG0) pins when the pins are used by peripheral modules. When the port a pins are used by the GPIO, the setting for this register is ignored.

Bit:	7	6	5	4	3	2	1	0
	PG7 PUPR	PG6 PUPR	PG5 PUPR	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	PG7PUPR	1	R/W	Pull-up of the Port Gn pin can be controlled independently. 0: PGn pull-up off 1: PGn pull-up on
6	PG6PUPR	1	R/W	
5	PG5PUPR	1	R/W	
4 to 0	—	All 1	R	Reserved These bits are always read as 1, and the write value should always be 1.

Note: n = 7 to 5

30.2.26 Port H Pull-Up Control Register (PHPUPR)

PHPUPR is an 8-bit readable/writable register that performs the pull-up control for each of the port H7 to H0 (PH7 to PH0) pins when the pins are used by peripheral modules. When the port a pins are used by the GPIO, the setting for this register is ignored.

Bit:	7	6	5	4	3	2	1	0
	PH7 PUPR	PH6 PUPR	PH5 PUPR	PH4 PUPR	PH3 PUPR	PH2 PUPR	PH1 PUPR	PH0 PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PH7PUPR	1	R/W	Pull-up of the Port Hn pin can be controlled independently. 0: PHn pull-up off 1: PHn pull-up on
6	PH6PUPR	1	R/W	
5	PH5PUPR	1	R/W	
4	PH4PUPR	1	R/W	
3	PH3PUPR	1	R/W	
2	PH2PUPR	1	R/W	
1	PH1PUPR	1	R/W	
0	PH0PUPR	1	R/W	

Note: n = 7 to 0

30.2.27 Port J Pull-Up Control Register (PJPUPR)

PJPUPR is an 8-bit readable/writable register that performs the pull-up control for each of the port J7 to J0 (PJ7 to PJ0) pins when the pins are used by peripheral modules. When the port a pins are used by the GPIO, the setting for this register is ignored.

Bit:	7	6	5	4	3	2	1	0
	PJ7 PUPR	PJ6 PUPR	PJ5 PUPR	PJ4 PUPR	PJ3 PUPR	PJ2 PUPR	PJ1 PUPR	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7PUPR	1	R/W	Pull-up of the Port Jn pin can be controlled independently. 0: PJn pull-up off 1: PJn pull-up on
6	PJ6PUPR	1	R/W	
5	PJ5PUPR	1	R/W	
4	PJ4PUPR	1	R/W	
3	PJ3PUPR	1	R/W	
2	PJ2PUPR	1	R/W	
1	PJ1PUPR	1	R/W	
0	—	1	R	Reserved This bit is always read as 1, and the write value should always be 1.

Note: n = 7 to 1

30.2.28 Input-Pin Pull-Up Control Register 1 (PPUPR1)

PPUPR1 is a 16-bit readable/writable register that performs the pull-up control for the pin corresponding to each bit of the register field.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RDY PUP	—	—
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 1	R/W	Reserved These bits are always read as 1, and the write value should always be 1.
2	RDYPUP	1	R/W	Controls pull-up of the $\overline{\text{RDY}}$ pin 0: $\overline{\text{RDY}}$ pull-up off 1: $\overline{\text{RDY}}$ pull-up on
1, 0	—	All 1	R/W	Reserved These bits are always read as 1, and the write value should always be 1.

30.2.29 Input-Pin Pull-Up Control Register 2 (PPUPR2)

PPUPR2 is a 16-bit readable/writable register that performs the pull-up control for the pin corresponding to each bit of the register field.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SCL0 PUP	SDA0 PUP	SCL1 PUP	SDA1 PUP	—	NMI PUP	IRL3 PUP	IRL2 PUP	IRL1 PUP	IRL0 PUP
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 1	R	Reserved These bits are always read as 1, and the write value should always be 1.
9	SCL0PUP	1	R/W	Controls pull-up of the I2C_SCL0/SCIF2_RXD pin 0: I2C_SCL0/SCIF2_RXD pull-up off 1: I2C_SCL0/SCIF2_RXD pull-up on
8	SDA0PUP	1	R/W	Controls pull-up of the I2C_SDA0/SCIF2_TXD pin 0: I2C_SDA0/SCIF2_TXD pull-up off 1: I2C_SDA0/SCIF2_TXD pull-up on
7	SCL1PUP	1	R/W	Controls pull-up of the I2C_SCL1/SCIF2_SCK pin 0: I2C_SCL1/SCIF2_SCK pull-up off 1: I2C_SCL1/SCIF2_SCK pull-up on
6	SDA1PUP	1	R/W	Controls pull-up of the I2C_SDA1/IRQOUT pin 0: I2C_SDA1/IRQOUT pull-up off 1: I2C_SDA1/IRQOUT pull-up on
5	—	1	R	Reserved This bit is always read as 1, and the write value should always be 1.
4	NMIPUP	1	R/W	Controls pull-up of the NMI pin 0: NMI pull-up off 1: NMI pull-up on
3	IRL3PUP	1	R/W	Controls pull-up of the $\overline{\text{IRL3}}$ pin 0: $\overline{\text{IRL3}}$ pull-up off 1: $\overline{\text{IRL3}}$ pull-up on
2	IRL2PUP	1	R/W	Controls pull-up of the $\overline{\text{IRL2}}$ pin 0: $\overline{\text{IRL2}}$ pull-up off 1: $\overline{\text{IRL2}}$ pull-up on
1	IRL1PUP	1	R/W	Controls pull-up of the $\overline{\text{IRL1}}$ pin 0: $\overline{\text{IRL1}}$ pull-up off 1: $\overline{\text{IRL1}}$ pull-up on
0	IRL0PUP	1	R/W	Controls pull-up of the $\overline{\text{IRL0}}$ pin 0: $\overline{\text{IRL0}}$ pull-up off 1: $\overline{\text{IRL0}}$ pull-up on

30.2.30 Peripheral Module Select Register 1 (P1MSELR)

P1MSELR is a 16-bit readable/writable register. This register can be used to select the module that uses multiplexed pins. For details of pin multiplexing, see table 30.1, Multiplexed Pins Controlled by Port Control Registers.

This register is valid only when peripheral modules are selected by PACR to PHCR, PJCR of the GPIO.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	P1MSEL14	P1MSEL13	P1MSEL12	P1MSEL11	P1MSEL10	P1MSEL9	P1MSEL8	P1MSEL7	P1MSEL6	P1MSEL5	P1MSEL4	P1MSEL3	P1MSEL2	P1MSEL1	P1MSEL0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.
14	P1MSEL14	0	R/W	Out of the modules DMAC and FLCTL, selects the one which uses the pin FCLE/DACK0. 0: FLCTL 1: DMAC
13	P1MSEL13	0	R/W	Out of the modules DMAC, LBSC and FLCTL, selects the one which uses the pin FALE/DACK1/BACK. 00: FLCTL 01: LBSC 10: DMAC0 11: Setting prohibited
12	P1MSEL12	0	R/W	
11	P1MSEL11	0	R/W	Out of the modules DMAC and USB, selects the one which uses the pin DREQ0/USB_OVC0. 0: USB 1: DMAC0

Bit	Bit Name	Initial Value	R/W	Description
10	P1MSEL10	0	R/W	Out of the modules DMAC, LBSC and USB, selects the one which uses the pin DREQ1/BREQ/USB_OVC1. 00: USB 01: DMAC0 10: LBSC 11: Setting prohibited
9	P1MSEL9	0	R/W	
8	P1MSEL8	0	R/W	Out of the modules SCIF and SSI, selects the one which uses the pin SCIF3_SCK/SSI2_SDATA. 0: SCIF3 1: SSI2
7	P1MSEL7	0	R/W	Out of the modules SCIF, TMU and SSI, selects the one which uses the pin SCIF3_RXD/TCLK/SSI2_SCK. 00: SCIF 01: TMU 10: SSI2 11: Setting prohibited
6	P1MSEL6	0	R/W	
5	P1MSEL5	0	R/W	Out of the modules SCIF, HAC and SSI, selects the one which uses the pin SCIF3_TXD/HAC_RES/SSI2_WS. 00: SCIF3 01: HAC 10: SSI2 11: Setting prohibited
4	P1MSEL4	0	R/W	
3	P1MSEL3	0	R/W	Out of the modules DU and HSPI, selects the one which uses the pins ODDF/HSPI_CS, VSYNC/HSPI_CLK and DCLKIN/HSPI_RX. 0: DU 1: HSPI

Bit	Bit Name	Initial Value	R/W	Description
2	P1MSEL2	0	R/W	Out of the modules DU and Eth, selects the one which uses the pins DR0/ETH_TXD0, DR1/ETH_TXD1, DR2/ETH_TXD2, DR3/ETH_TXD3, DR4/ETH_TX_EN, DR5/ETH_TX_ER, DG0/ETH_CRS, DG1/ETH_TX_CLK, DG2/ETH_COL, DG3/ETH_MDC, DG4/ETH_RX_CLK, DG5/ETH_MDIO, DB0/ETH_RX_ER, DB1/ETH_RX_DV, DB2/ETH_RXD0, DB3/ETH_RXD1, DB4/ETH_RXD2, DB5/ETH_RXD3, DISP/ETH_LINK, CDE/ETH_MAGIC. 0: DU 1: Eth
1	P1MSEL1	0	R/W	Out of the modules I2C and SCIF/INTC, selects the one which uses the pins I2C_SCL0/SCIF2_RXD, I2C_SDA0/SCIF2_TXD, I2C_SCL1/SCIF2_SCK and I2C_SDA1/$\overline{\text{IRQOUT}}$. 0: I2C 1: SCIF2 (I2C_SDA1/$\overline{\text{IRQOUT}}$ is assigned to INTC)
0	P1MSEL0	0	R/W	Out of the modules STATUS and SSI, selects the one which uses the pins STATUS0/SSI2_CLK and STATUS1/SSI3_CLK. 0: STATUS 1: SSI2/3

30.2.31 Peripheral Module Select Register 2 (P2MSELR)

P2MSELR is a 16-bit readable/writable register. This register can be used to select the module that uses multiplexed pins. For details of pin multiplexing, see table 30.1, Multiplexed Pins Controlled by Port Control Registers.

This register is valid only when peripheral modules are selected by PACR to PHCR, PJCR of the GPIO.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	P2M SEL15	P2M SEL14	P2M SEL13	P2M SEL12	P2M SEL11	P2M SEL10	P2M SEL9	P2M SEL8	P2M SEL7	P2M SEL6	P2M SEL5	P2M SEL4	P2M SEL3	P2M SEL2	P2M SEL1	P2M SEL0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	P2MSEL15	0	R/W	Out of the modules HAC, SSI and SDIF, selects the one which uses the pins HAC1_BITCLK/SSI1_CLK/SDIF1CLK, HAC1_SYNC/SSI1_WS/SDIF1WP, HAC1_SDIN/SSI1_SCK/SDIF1CD and HAC1_SDOUT/SSI1_SDATA/SDIF1CMD. 00: HAC1 01: SSI1 10: SDIF1 11: Setting prohibited
14	P2MSEL14	0	R/W	
13	P2MSEL13	0	R/W	Out of the modules HAC, SSI and SDIF, selects the one which uses the pins HAC0_BITCLK/SSI0_CLK/SDIF1D0, HAC0_SYNC/SSI0_WS/SDIF1D1, HAC0_SDIN/SSI0_SCK/SDIF1D2 and HAC0_SDOUT/SSI0_SDATA/SDIF1D3. 00: HAC0 01: SSI0 10: SDIF1 11: Setting prohibited
12	P2MSEL12	0	R/W	
11	P2MSEL11	0	R/W	Out of the modules SCIF and FLCTL, selects the one which uses the pin MODE14/$\overline{\text{FRB}}$/SCIF5_SCK. 0: Selects FLCTL 1: Selects SCIF5

Bit	Bit Name	Initial Value	R/W	Description
10	P2MSEL10	0	R/W	Out of the modules SCIF and LBSC, selects the one which uses the pins MODE13/SCIF5_RXD/$\overline{\text{IOIS16}}$ and MODE12/SCIF5_TXD/$\overline{\text{CE2B}}$. 0: Selects SCIF5 1: Selects LBSC
9	P2MSEL9	0	R/W	Out of the modules SCIF, DMAC and SSI, selects the one which uses the pin MODE10/SCIF4_SCK/DRAK2/SSI3_WS. 00: SCIF4 01: DMAC0 10: SSI3 11: Setting prohibited
8	P2MSEL8	0	R/W	
7	P2MSEL7	0	R/W	Out of the modules SCIF/DMAC and LBSC, selects the one which uses the pin MODE11/DRAK3/$\overline{\text{CE2A}}$. 0: DMAC0 1: LBSC
6	P2MSEL6	0	R/W	Out of the modules SCIF, DMAC, SSI and FLCTL, selects the one which uses the pins MODE8/$\overline{\text{FSE}}$/SCIF4_TXD/DRAK0/SSI3_SCK and MODE9/SCIF4_RXD/DRAK1/SSI3_SDATA. 00: FLCTL 01: DMAC0 10: SCIF4 11: SSI3
5	P2MSEL5	0	R/W	
4	P2MSEL4	0	R/W	Out of the modules DMAC and SDIF, selects the one which uses the pins MODE7/DACK3/SDIF0CMD, MODE6/DACK2/$\overline{\text{SDIF0CD}}$ and MODE5/$\overline{\text{DREQ3}}$/SDIF0WP. 0: DMAC0 1: SDIF0

Bit	Bit Name	Initial Value	R/W	Description
3	P2MSEL3	0	R/W	Out of the modules SCIF, DMAC and SDIF, selects the one which uses the pin MODE4/SCIF0_CTS/DREQ2/SDIF0CLK. 00: SCIF0 01: DMAC0 10: SDIF0 11: Setting prohibited
2	P2MSEL2	0	R/W	
1	P2MSEL1	0	R/W	Out of the modules SCIF, INTC and SDIF, selects the one which uses the pins MODE2/SCIF0_SCK/IRL6/SDIF0D2, MODE1/SCIF0_RXD/IRL5/SDIF0D1 and MODE0/SCIF0_TXD/IRL4/SDIF0D0. 00: SCIF0 01: INTC 10: SDIF0 11: Setting prohibited
0	P2MSEL0	0	R/W	

30.3 Usage Examples

Setting procedure examples are described below.

30.3.1 Port Output Function

To output the data of port data registers (PADR to PJDR) from the GPIO output port, write B'01 to the corresponding two bits in port control registers (PACR to PJCR).

In this case, for each output port, the settings of the port pull-up control registers (PAPUPR to PJPUPR), peripheral module select register 1 (P1MSELR) and peripheral module select register 2 (P2MSELR) are invalid.

Figure 30.1 shows an example of operation timing diagram when port A is used as an output port.

The output data is written to port data registers (PADR to PJDR) and then the data is output via the corresponding port pins after one peripheral clock (Pck).

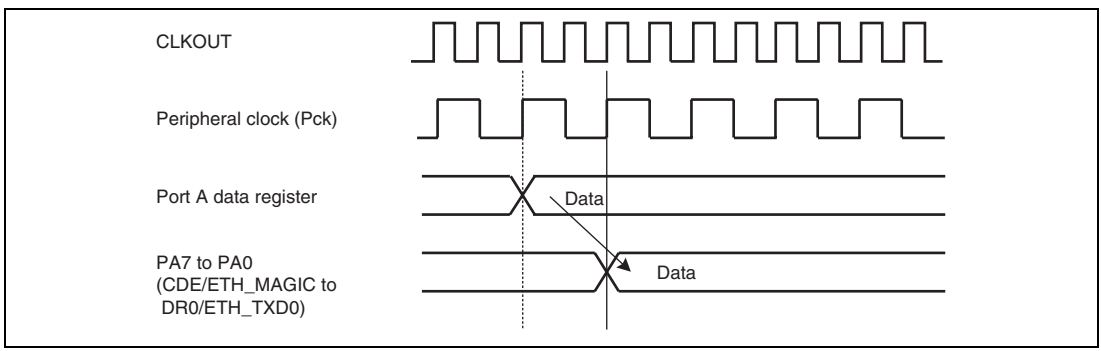


Figure 30.1 Port A Data Output Timing Diagram

30.3.2 Port Input function

To input the data via the GPIO port, write B'10 or B'11 to the corresponding two bits in port control registers (PACR to PJCR). B'10 should be written when the pull-up MOS is off, and B'11 when the pull-up MOS is on. The input data to each port can be read out from the corresponding bit in port data registers (PADR to PJCR).

In this case, for each output port, the settings of port pull-up control registers (PAPUPR to PJPUPR), peripheral module select register 1 (P1MSELR) and peripheral module select register 2 (P2MSELR) are invalid.

Figure 30.2 shows an example of operation timing diagram when port A is used as an input port.

The input data from each port can be read out from corresponding port data register after the 2nd rising edge of the peripheral clock (Pck).

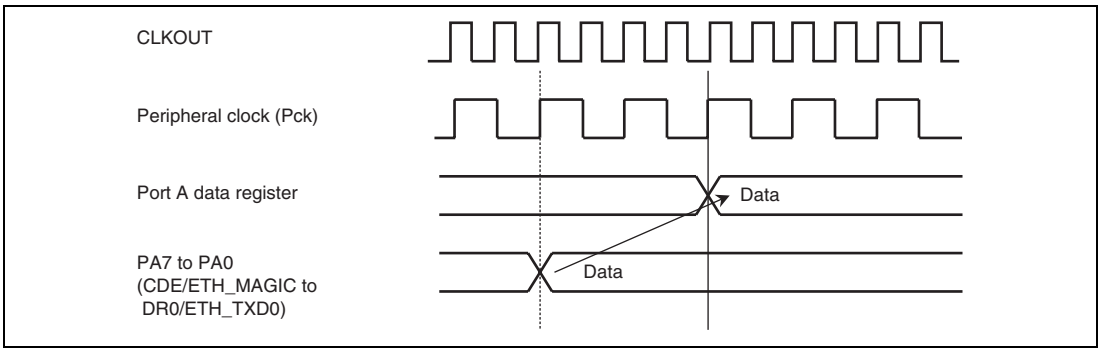


Figure 30.2 Port A Data Input Timing Diagram

30.3.3 Peripheral Module Function

The procedures for setting the peripheral module function are described below.

1. Select the peripheral module by using the peripheral module select register 1 (P1MSELR) and peripheral module select register 2 (P2MSELR).
2. When an input or input/output pin is used, it is necessary to set the pull-up MOS for each pin by using the port pull-up control registers (PAPUPR to PJPUPR). Write B'0 (when the pull-up MOS is off) or B'1 (when the pull-up MOS is on) to the corresponding bit. When an output port is used, the pull-up MOS is off regardless of the settings of the port pull-up control registers.
3. Write B'00 to the corresponding two bits in the port control registers (PACR to PJCR).
4. After the settings of the port, do setting of the peripheral module side according to its specs.

Section 31 User Break Controller (UBC)

The user break controller (UBC) provides versatile functions to facilitate program debugging. These functions help to ease creation of a self-monitor/debugger, which allows easy program debugging using this LSI alone, without using the in-circuit emulator. Various break conditions can be set in the UBC: instruction fetch or read/write access of an operand, operand size, data contents, address value, and program stop timing for instruction fetch.

31.1 Features

1. Each CPU core has independent UBC that is able to keep an individual setting.
2. The following break conditions can be set.

   Break channels: Two (channels 0 and 1) for each CPU core

   User break conditions can be set independently for channels 0 and 1, and can also be set as a single sequential condition for the two channels, that is, a sequential break. (Sequential break involves two cases such that the channel 0 break condition is satisfied in a certain bus cycle and then the channel 1 break condition is satisfied in a different bus cycle, and vice versa.)

- Address

  When 40 bits or 48 bits that containing ASID (8 bits or 16 bits) and 32-bit address are compared with the specified value, all the ASID bits can be compared or masked.

  32-bit address can be masked bit by bit, allowing the user to mask the address in desired page sizes such as lower 12 bits (4-Kbyte page) or lower 10 bits (1-Kbyte page).

- Data

  32 bits can be masked only for channel 1.

- Bus cycle

  The program can break either for instruction fetch (PC break) or operand access.

- Read or write access
- Operand sizes

  Byte, word, longword, and quadword are supported.

3. The user-designated exception handling routine for the user break condition can be executed.
4. Pre-instruction-execution or post-instruction-execution can be selected as the PC break timing.
5. A maximum of $2^{12} - 1$ repetition counts can be specified as the break condition (available only for channel 1).

Figure 31.1 shows the UBC block diagram.

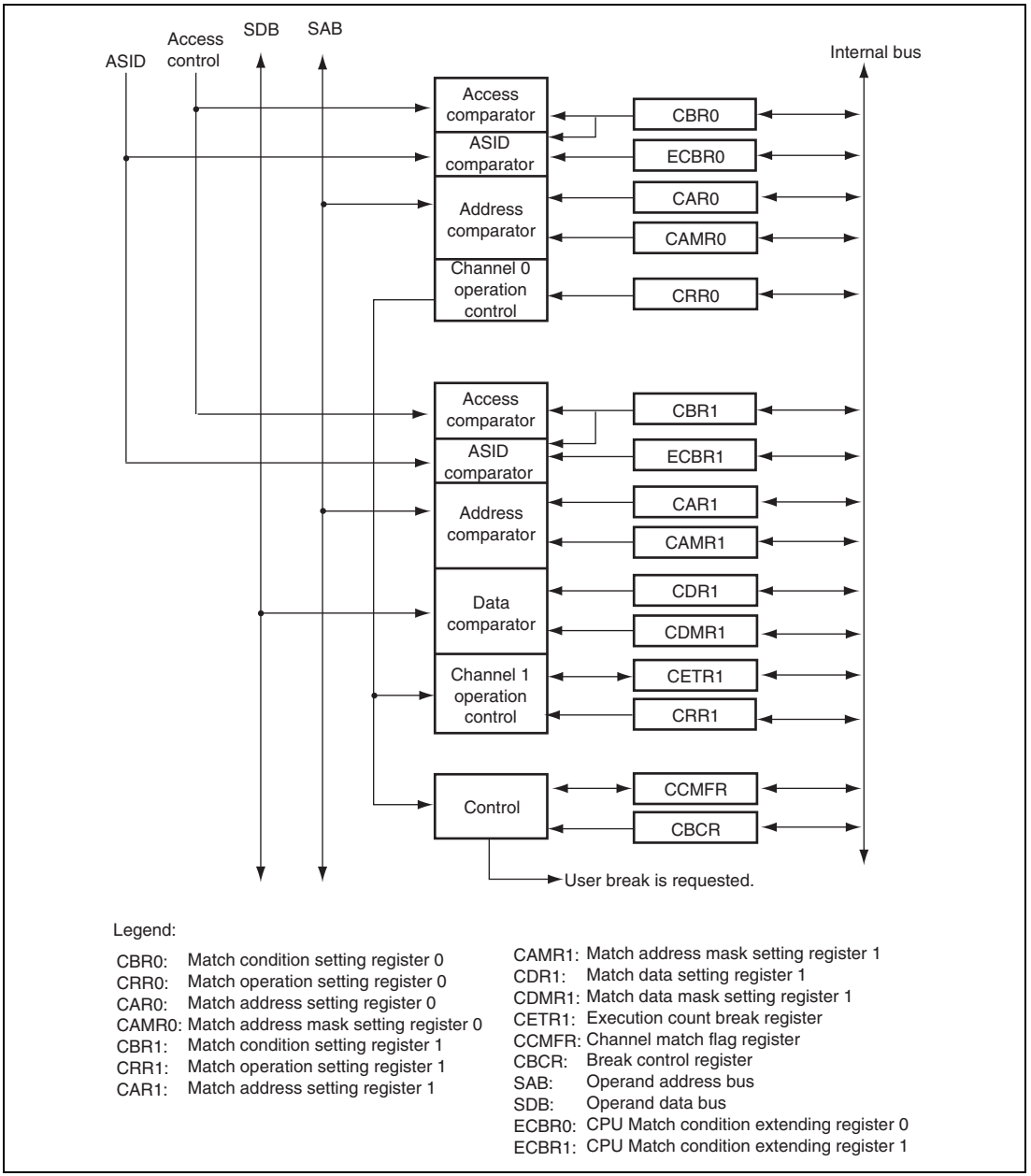


Figure 31.1 Block Diagram of UBC

31.2 Register Descriptions

The UBC has the following registers.

Table 31.1 Register Configuration

Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
Match condition setting register 0	CBR0	R/W	H'FF200000	H'1F200000	32
Match operation setting register 0	CRR0	R/W	H'FF200004	H'1F200004	32
Match address setting register 0	CAR0	R/W	H'FF200008	H'1F200008	32
Match address mask setting register 0	CAMR0	R/W	H'FF20000C	H'1F20000C	32
CPU match condition setting extension register 0	ECBR0	R/W	H'FF20001C	H'1F20001C	32
Match condition setting register 1	CBR1	R/W	H'FF200020	H'1F200020	32
Match operation setting register 1	CRR1	R/W	H'FF200024	H'1F200024	32
Match address setting register 1	CAR1	R/W	H'FF200028	H'1F200028	32
Match address mask setting register 1	CAMR1	R/W	H'FF20002C	H'1F20002C	32
Match data setting register 1	CDR1	R/W	H'FF200030	H'1F200030	32
Match data mask setting register 1	CDMR1	R/W	H'FF200034	H'1F200034	32
Execution count break register 1	CETR1	R/W	H'FF200038	H'1F200038	32
CPU match condition setting extension register 1	ECBR1	R/W	H'FF20003C	H'1F20003C	32
Channel match flag register	CCMFR	R/W	H'FF200600	H'1F200600	32
Break control register	CBCR	R/W	H'FF200620	H'1F200620	32

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 31.2 Register Status in Each Processing State

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sleep/ Light Sleep
Match condition setting register 0	CBR0	H'20000000	Retained	Retained
Match operation setting register 0	CRR0	H'00002000	Retained	Retained
Match address setting register 0	CAR0	Undefined	Retained	Retained
Match address mask setting register 0	CAMR0	Undefined	Retained	Retained
CPU match condition setting extension register 0	ECBR0	H'00000000	Retained	Retained
Match condition setting register 1	CBR1	H'20000000	Retained	Retained
Match operation setting register 1	CRR1	H'00002000	Retained	Retained
Match address setting register 1	CAR1	Undefined	Retained	Retained
Match address mask setting register 1	CAMR1	Undefined	Retained	Retained
Match data setting register 1	CDR1	Undefined	Retained	Retained
Match data mask setting register 1	CDMR1	Undefined	Retained	Retained
Execution count break register 1	CETR1	Undefined	Retained	Retained
CPU match condition setting extension register 1	ECBR1	H'00000000	Retained	Retained
Channel match flag register	CCMFR	H'00000000	Retained	Retained
Break control register	CBCR	H'00000000	Retained	Retained

The access size must be the same as the control register size. If the size is different, the register is not written to if attempted, and reading the register returns the undefined value. A desired break may not occur between the time when the instruction for rewriting the control register is executed and the time when the written value is actually reflected on the register. In order to confirm the exact timing when the control register is updated, read the data which has been written most recently. The subsequent instructions are valid for the most recently written register value.

31.2.1 Match Condition Setting Registers 0 and 1 (CBR0 and CBR1)

CBR0 and CBR1 are readable/writable 32-bit registers which specify the break conditions for channels 0 and 1, respectively. The following break conditions can be set in the CBR0 and CBR1: (1) whether or not to include the match flag in the conditions, (2) whether or not to include the ASID, and the ASID value when included (lower 8bits), (3) whether or not to include the data value, (4) operand size, (5) whether or not to include the execution count, (6) bus type, (7) instruction fetch cycle or operand access cycle, and (8) read or write access cycle.

- CBR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFE	AIE	MFI						AIV7	AIV6	AIV5	AIV4	AIV3	AIV2	AIV1	AIV0
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SZ			—	—	—	—	CD		ID		—	RW		CE
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MFE	0	R/W	Match Flag Enable Specifies whether or not to include the match flag value specified by the MFI bit of this register in the match conditions. When the specified match flag value is 1, the condition is determined to be satisfied. 0: The match flag is not included in the match conditions; thus, not checked. 1: The match flag is included in the match conditions.
30	AIE	0	R/W	ASID Enable*[3] Specifies whether or not to include the ASID (lower 8bits) specified by the AIV7 to AIV0 bits of this register and another ASID (upper 8bits) specified by the AIV15 to AIV8 bits of ECBR0 register in the match conditions. 0: The ASID (16bits) is not included in the match conditions; thus, not checked. 1: The ASID (16bits) is included in the match conditions.

Bit	Bit Name	Initial Value	R/W	Description
29 to 24	MFI	100000	R/W	Match Flag Specify Specifies the match flag to be included in the match conditions. 000000: MF0 bit of the CCMFR register 000001: MF1 bit of the CCMFR register Others: Reserved (setting prohibited) Note: The initial value is the reserved value, but when 1 is written into CBR0[0], MFI must be set to 000000 or 000001. And note that the channel 0 is not hit when MFE bit of this register is 1 and MFI bits are 000000 in the condition of CCRMF.MF0 = 0.
23 to 16	AIV7 to AIV0	All 0	R/W	ASID (lower 8bits) Specify Specifies the ASID (lower 8bits) value to be included in the match conditions.
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	SZ	000	R/W	Operand Size Select Specifies the operand size to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 000: The operand size is not included in the match conditions; thus, not checked (any operand size specifies the match condition).*[1] 001: Byte access 010: Word access 011: Longword access 100: Quadword access*[2] Others: Reserved (setting prohibited)
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD	00	R/W	Bus Select Specifies the bus to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 00: Operand bus for operand access Others: Reserved (setting prohibited)
5, 4	ID	00	R/W	Instruction Fetch/Operand Access Select Specifies the instruction fetch cycle or operand access cycle as the match condition. 00: Instruction fetch cycle or operand access cycle 01: Instruction fetch cycle 10: Operand access cycle 11: Instruction fetch cycle or operand access cycle
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2, 1	RW	00	R/W	Bus Command Select Specifies the read/write cycle as the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 00: Read cycle or write cycle 01: Read cycle 10: Write cycle 11: Read cycle or write cycle
0	CE	0	R/W	Channel Enable Validates/invalidates the channel. If this bit is 0, all the other bits of this register are invalid. 0: Invalidates the channel. 1: Validates the channel.

Notes: The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions for these instructions.

1. If the data value is included in the match conditions, be sure to specify the operand size.
2. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and the match data mask setting register.
3. The AIV15 to AIV8 bits (upper 8bits of ASID) of ECBR0 register are used for the comparing regardless of the AEX bit in MMUCR register (ASID compatible mode). When the AEX bit in MMUCR register is set to 0 (ASID 8-bit compatible mode) and the AIE bit in this register is set to 1, the AIV15 to AIV8 bits of ECBR0 register should be set to 0.

- CBR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFE	AIE	MFI						AIV7	AIV6	AIV5	AIV4	AIV3	AIV2	AIV1	AIV0
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DBE	SZ			ETBE	—	—	—	CD		ID		—	RW		CE
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MFE	0	R/W	Match Flag Enable Specifies whether or not to include the match flag value specified by the MFI bit of this register in the match conditions. When the specified match flag value is 1, the condition is determined to be satisfied. 0: The match flag is not included in the match conditions; thus, not checked. 1: The match flag is included in the match conditions.

Bit	Bit Name	Initial Value	R/W	Description
30	AIE	0	R/W	ASID Enable Specifies whether or not to include the ASID (lower 8bits) specified by the AIV7 to AIV0 bits of this register and another ASID (upper 8bits) specified by the AIV15 to AIV8 bits of ECBR1 register in the match conditions. 0: The ASID (16bits) is not included in the match conditions; thus, not checked. 1: The ASID (16bits) is included in the match conditions.
29 to 24	MFI	100000	R/W	Match Flag Specify Specifies the match flag to be included in the match conditions. 000000: The MF0 bit of the CCMFR register 000001: The MF1 bit of the CCMFR register Others: Reserved (setting prohibited) Note: The initial value is the reserved value, but when 1 is written into CBR1[0], MFI must be set to 000000 or 000001. And note that the channel 1 is not hit when MFE bit of this register is 1 and MFI bits are 000001 in the condition of CCRMF.MF1 = 0.
23 to 16	AIV7 to AIV0	All 0	R/W	ASID (lower 8bits) Specify Specifies the ASID (lower 8bits) value to be included in the match conditions.
15	DBE	0	R/W	Data Value Enable*[2] Specifies whether or not to include the data value in the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 0: The data value is not included in the match conditions; thus, not checked. 1: The data value is included in the match conditions.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	SZ	000	R/W	Operand Size Select Specifies the operand size to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 000: The operand size is not included in the match condition; thus, not checked (any operand size specifies the match condition).*[1] 001: Byte access 010: Word access 011: Longword access 100: Quadword access*[3] Others: Reserved (setting prohibited)
11	ETBE	0	R/W	Execution Count Value Enable Specifies whether or not to include the execution count value in the match conditions. If this bit is 1 and the match condition satisfaction count matches the value specified by the CETR1 register, the operation specified by the CRR1 register is performed. 0: The execution count value is not included in the match conditions; thus, not checked. 1: The execution count value is included in the match conditions.
10 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	CD	00	R/W	Bus Select Specifies the bus to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 00: Operand bus for operand access Others: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
5, 4	ID	00	R/W	Instruction Fetch/Operand Access Select Specifies the instruction fetch cycle or operand access cycle as the match condition. 00: Instruction fetch cycle or operand access cycle 01: Instruction fetch cycle 10: Operand access cycle 11: Instruction fetch cycle or operand access cycle
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2, 1	RW	00	R/W	Bus Command Select Specifies the read/write cycle as the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 00: Read cycle or write cycle 01: Read cycle 10: Write cycle 11: Read cycle or write cycle
0	CE	0	R/W	Channel Enable Validates/invalidates the channel. If this bit is 0, all the other bits in this register are invalid. 0: Invalidates the channel. 1: Validates the channel.

Notes: The AIV15 to AIV8 bits (upper 8bits of ASID) of ECBR1 register are used for the comparing regardless of the AEX bit in MMUCR register (ASID compatible mode). When the AEX bit in MMUCR register is set to 0 (ASID 8-bit compatible mode) and the AIE bit in this register is set to 1, the AIV15 to AIV8 bits of ECBR1 register should be set to 0.

1. If the data value is included in the match conditions, be sure to specify the operand size.
2. The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions for these instructions.
3. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and the match data mask setting register.

31.2.2 CPU Mach Condition Setting Extension Register (ECBR0, ECBR1)

ECBR0 and ECBR1 are readable/writable 32-bit registers which expand the break conditions for channels 0 and 1, respectively. The break condition can be set is (1) ASID (upper 8 bits).

- ECBR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AIV15	AIV14	AIV13	AIV12	AIV11	AIV10	AIV9	AIV8
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	AIV15 to AIV8	All 0	R/W	ASID (upper 8 bits) Specify Specifies the ASID (upper 8 bits) value to be included in the match conditions.

Note: The AIV15 to AIV8 bits (upper 8 bits of ASID) of this register are used for the comparing of ASID regardless of the AEX bit in the MMUCR register (ASID compatible mode). When the AEX bit in the MMUCR register is set to 0 (ASID 8-bit compatible mode) and the AIE bit in CBR0 register is set to 1, the AIV15 to AIV8 bits of this register should be set to 0.

- ECBR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AIV15	AIV14	AIV13	AIV12	AIV11	AIV10	AIV9	AIV8
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	AIV15 to AIV8	All 0	R/W	ASID (upper 8 bits) Specify Specifies the ASID (upper 8 bits) value to be included in the match conditions.

Note: The AIV15 to AIV8 bits (upper 8 bits of ASID) of this register are used for the comparing of ASID regardless of the AEX bit in MMUCR register (ASID compatible mode). When the AEX bit in MMUCR register is set to 0 (ASID 8-bit compatible mode) and the AIE bit in CBR1 is set to 1, the AIV15 to AIV8 bits of this register should be set to 0.

31.2.3 Match Operation Setting Registers 0 and 1 (CRR0 and CRR1)

CRR0 and CRR1 are readable/writable 32-bit registers which specify the operation to be executed when channels 0 and 1 satisfy the match condition, respectively. The following operations can be set in the CRR0 and CRR1 registers: (1) breaking at a desired timing for the instruction fetch cycle and (2) requesting a break.

- CRR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCB	BIE
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
12 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PCB	0	R/W	PC Break Select Specifies either before or after instruction execution as the break timing for the instruction fetch cycle. This bit is invalid for breaks other than the ones for the instruction fetch cycle. 0: Sets the PC break before instruction execution. 1: Sets the PC break after instruction execution.
0	BIE	0	R/W	Break Enable Specifies whether or not to request a break when the match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

- CRR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

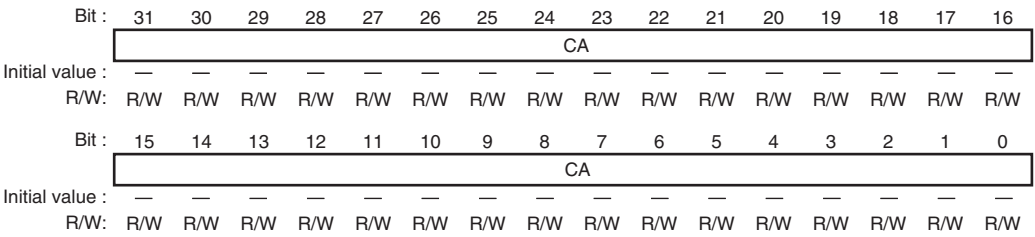
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCB	BIE
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
12 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PCB	0	R/W	PC Break Select Specifies either before or after instruction execution as the break timing for the instruction fetch cycle. This bit is invalid for breaks other than ones for the instruction fetch cycle. 0: Sets the PC break before instruction execution. 1: Sets the PC break after instruction execution.
0	BIE	0	R/W	Break Enable Specifies whether or not to request a break when the match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

31.2.4 Match Address Setting Registers 0 and 1 (CAR0 and CAR1)

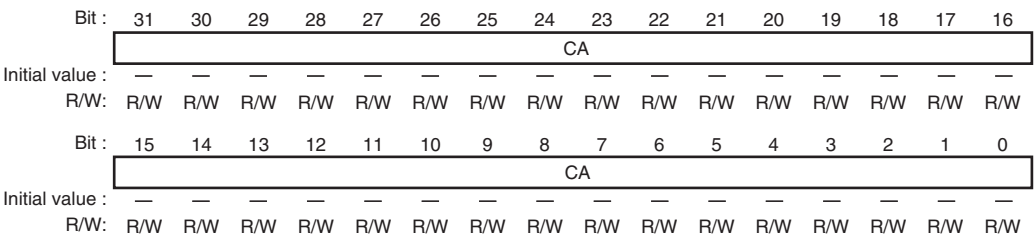
CAR0 and CAR1 are readable/writable 32-bit registers specifying the virtual address to be included in the break conditions for channels 0 and 1, respectively.

- CAR0



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CA	Undefined	R/W	Compare Address Specifies the address to be included in the break conditions. When the operand bus has been specified using the CBR0 register, specify the SAB address in CA[31:0].

- CAR1

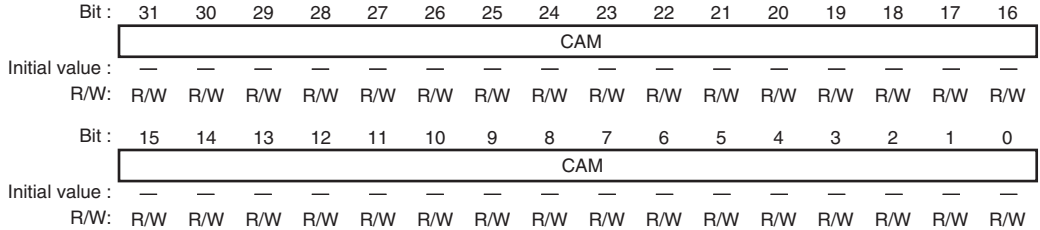


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CA	Undefined	R/W	Compare Address Specifies the address to be included in the break conditions. When the operand bus has been specified using the CBR1 register, specify the SAB address in CA[31:0].

31.2.5 Match Address Mask Setting Registers 0 and 1 (CAMR0 and CAMR1)

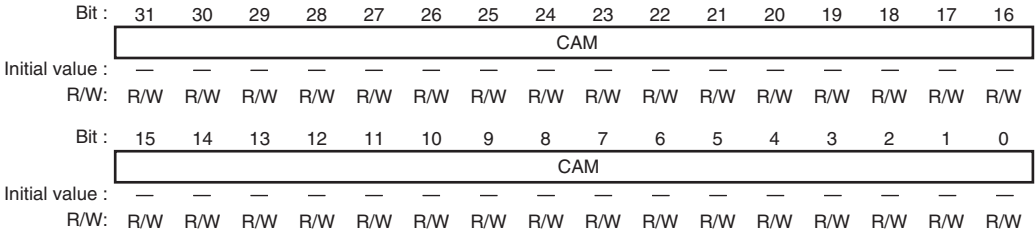
CMAR0 and CMAR1 are readable/writable 32-bit registers which specify the bits to be masked among the address bits specified by using the match address setting register of the corresponding channel. (Set the bits to be masked to 1.)

- CAMR0



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CAM	Undefined	R/W	Compare Address Mask Specifies the bits to be masked among the address bits which are specified using the CAR0 register. (Set the bits to be masked to 1.) 0: Address bits CA[n] are included in the break condition. 1: Address bits CA[n] are masked and not included in the break condition. [n] = any values from 31 to 0

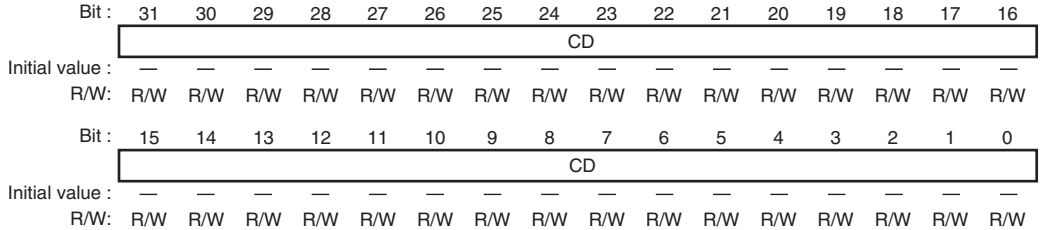
- CAMR1



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CAM	Undefined	R/W	Compare Address Mask Specifies the bits to be masked among the address bits which are specified using the CAR1 register. (Set the bits to be masked to 1.) 0: Address bits CA[n] are included in the break condition. 1: Address bits CA[n] are masked and not included in the break condition. [n] = any values from 31 to 0

31.2.6 Match Data Setting Register 1 (CDR1)

CDR1 is a readable/writable 32-bit register which specifies the data value to be included in the break conditions for channel 1.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CD	Undefined	R/W	Compare Data Value Specifies the data value to be included in the break conditions. When the operand bus has been specified using the CBR1 register, specify the SDB data value in CD[31:0].

Table 31.3 Settings for Match Data Setting Register

Bus and Size Selected Using CBR1	CD[31:24]	CD[23:16]	CD[15:8]	CD[7:0]
Operand bus (byte)	Don't care	Don't care	Don't care	SDB7 to SDB0
Operand bus (word)	Don't care	Don't care	SDB15 to SDB8	SDB7 to SDB0
Operand bus (longword)	SDB31 to SDB24	SDB23 to SDB16	SDB15 to SDB8	SDB7 to SDB0

Notes: 1. If the data value is included in the match conditions, be sure to specify the operand size.
2. The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions for these instructions.
3. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and match data mask setting register.

31.2.7 Match Data Mask Setting Register 1 (CDMR1)

CDMR1 is a readable/writable 32-bit register which specifies the bits to be masked among the data value bits specified using the match data setting register. (Set the bits to be masked to 1.)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDM															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDM															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CDM	Undefined	R/W	Compare Data Value Mask Specifies the bits to be masked among the data value bits specified using the CDR1 register. (Set the bits to be masked to 1.) 0: Data value bits CD[n] are included in the break condition. 1: Data value bits CD[n] are masked and not included in the break condition. [n] = any values from 31 to 0

31.2.8 Execution Count Break Register 1 (CETR1)

CETR1 is a readable/writable 32-bit register which specifies the number of the channel hits before a break occurs. A maximum value of $2^{12} - 1$ can be specified. When the execution count value is included in the match conditions by using the match condition setting register, the value of this register is decremented by one every time the channel is hit. When the channel is hit after the register value reaches H'001, a break occurs.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CET											
Initial value :	0	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	CET	Undefined	R/W	Execution Count Specifies the execution count to be included in the break conditions.

31.2.9 Channel Match Flag Register (CCMFR)

CCMFR is a readable/writable 32-bit register which indicates whether or not the match conditions have been satisfied for each channel. When a channel match condition has been satisfied, the corresponding flag bit is set to 1. To clear the flags, write the data containing value 0 for the bits to be cleared and value 1 for the other bits to this register. (The logical AND between the value which has been written and the current register value is actually written to the register.) A sequential operation using multiple channels is available by using these match flags.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MF1	MF0
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	MF1	0	R/W	Channel 1 Condition Match Flag This flag is set to 1 when the channel 1 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 1 match condition has not been satisfied. 1: Channel 1 match condition has been satisfied.
0	MF0	0	R/W	Channel 0 Condition Match Flag This flag is set to 1 when the channel 0 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 0 match condition has not been satisfied. 1: Channel 0 match condition has been satisfied.

31.2.10 Expanding Match Flag Register (CEMFR)

CEMFR is a readable/writable 32-bit register which can assert/clear flags by the match conditions of each channel. A sequential break is available by using this register.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MF1	MF0
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	MF1	0	R/W	Channel 1 Condition Match Flag This flag is set to 1 when the channel 1 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 1 match condition has not been satisfied. 1: Channel 1 match condition has been satisfied.
0	MF0	0	R/W	Channel 0 Condition Match Flag This flag is set to 1 when the channel 0 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 0 match condition has not been satisfied. 1: Channel 0 match condition has been satisfied.

31.2.11 Break Control Register (CBCR)

CBCR is a readable/writable 32-bit register which specifies whether or not to use the user break debugging support function. For details on the user break debugging support function, refer to section 31.4, User Break Debugging Support Function.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UBDE
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	UBDE	0	R/W	User Break Debugging Support Function Enable Specifies whether or not to use the user break debugging support function. 0: Does not use the user break debugging support function. 1: Uses the user break debugging support function.

31.3 Operation Description

31.3.1 Definition of Words Related to Accesses

"Instruction fetch" refers to an access in which an instruction is fetched. For example, fetching the instruction located at the branch destination after executing a branch instruction is an instruction access. "Operand access" refers to any memory access accompanying execution of an instruction. For example, accessing an address (PC + disp × 2 + 4) in the instruction MOV.W@(disp,PC),Rn is an operand access. "Data" is used in contrast to "address".

All types of operand access are classified into read or write access. Special care must be taken in using the following instructions.

- PREF, OCBP, and OCBWB: Instructions for a read access
- MOVCA.L and OCBI: Instructions for a write access
- TAS.B: Instruction for a single read access or a single write access

The operand access accompanying the PREF, OCBP, OCBWB, and OCBI instructions is access without the data value; therefore, do not include the data value in the match conditions for these instructions.

The operand size should be defined for all types of operand access. Available operand sizes are byte, word, longword, and quadword. For operand access accompanying the PREF, OCBP, OCBWB, MOVCA.L, and OCBI instructions, the operand size is defined as longword.

31.3.2 User Break Operation Sequence

The following describes the sequence from when the break condition is set until the user break exception handling is initiated.

1. Specify the operand size, bus, instruction fetch/operand access, and read/write as the match conditions using the match condition setting register (CBR0 or CBR1). Specify the break address using the match address setting register (CAR0 or CAR1), and specify the address mask condition using the match address mask setting register (CAMR0 or CAMR1). To include the ASID in the match conditions, set the AIE bit in the match condition setting register and specify the lower 8bits and upper 8bits value of the ASID by using the AIV7 to AIV0 bits in the same register and the AIV15 to AIV8 bits in CPU match condition extension register (ECBR0 or ECBR1), respectively. To include the data value in the match conditions, set the DBE bit in the match condition setting register; specify the break data using the match data setting register (CDR1); and specify the data mask condition using the match data mask setting register (CDMR1). To include the execution count in the match conditions, set the

ETBE bit of the match condition setting register; and specify the execution count using the execution count break register (CETR1). To use the sequential break, set the MFE bit of the match condition setting register; and specify the number of the first channel using the MFI bit.

2. Specify whether or not to request a break when the match condition is satisfied and the break timing when the match condition is satisfied as a result of fetching the instruction using the match operation setting register (CRR0 or CRR1). After having set all the bits in the match condition setting register except the CE bit and the other necessary registers, set the CE bit and read the match condition setting register again. This ensures that the set values in the control registers are valid for the subsequent instructions immediately after reading the register. Setting the CE bit of the match condition setting register in the initial state after reset via the control registers may cause an undesired break.
3. When the match condition has been satisfied, the corresponding condition match flag (MF1 or MF0) in the channel match flag register (CCMFR) is set. A break is also requested to the CPU according to the set values in the match operation setting register (CRR0 or CRR1). The CPU operates differently according to the BL bit value of the SR register: when the BL bit is 0, the CPU accepts the break request and executes the specified exception handling; and when the BL bit is 1, the CPU does not execute the exception handling.
4. The match flags (MF1 and MF0) can be used to confirm whether or not the corresponding match condition has been satisfied. Although the flag is set when the condition is satisfied, it is not cleared automatically; therefore, write 0 to the flag bit by issuing a memory store instruction to the channel match flag register (CCMFR) in order to use the flag again.
5. Breaks may occur virtually at the same time for channels 0 and 1. In this case, only one break request is sent to the CPU; however, the two condition match flags corresponding to these breaks may be set.
6. While the BL bit in the SR register is 1, no break requests are accepted. However, whether or not the condition has been satisfied is determined. When the condition is determined to be satisfied, the corresponding condition match flag is set.
7. If the sequential break conditions are set, the condition match flag is set every time the match conditions are satisfied for each channel. When the conditions have been satisfied for the first channel in the sequence but not for the second channel in the sequence, clear the condition match flag for the first channel in the sequence in order to release the first channel in the sequence from the match state.

31.3.3 Instruction Fetch Cycle Break

1. If the instruction fetch cycle is set in the match condition setting register (CBR0 or CBR1), the instruction fetch cycle is handled as a match condition. To request a break upon satisfying the match condition, set the BIE bit in the match operation setting register (CRR0 or CRR1) of the corresponding channel. Either before or after executing the instruction can be selected as the break timing according to the PCB bit value. If the instruction fetch cycle is specified as a match condition, be sure to clear the LSB to 0 in the match address setting register (CAR0 or CAR1); otherwise, no break occurs.
2. If pre-instruction-execution break is specified for the instruction fetch cycle, the break is requested when the instruction is fetched and determined to be executed. Therefore, this function cannot be used for the instructions which are fetched through overrun (i.e., the instructions fetched during branching or making transition to the interrupt routine but not executed). For priorities of pre-instruction-execution break and the other exceptions, refer to section 5, Exception Handling. If pre-instruction-execution break is specified for the delayed slot of the delayed branch instruction, the break is requested before the delayed branch instruction is executed. However, do not specify pre-instruction-execution break for the delayed slot of the RTE instruction.
3. If post-instruction-execution break is specified for the instruction fetch cycle, the break is requested after the instruction which satisfied the match condition has been executed and before the next instruction is executed. Similar to pre-instruction-execution break, this function cannot be used for the instructions which are fetched through overrun. For priorities of post-instruction-execution break and the other exceptions, refer to section 5, Exception Handling. If post-instruction-execution break is specified for the delayed branch instruction and its delayed slot, the break does not occur until the first instruction at the branch destination.
4. If the instruction fetch cycle is specified as the channel 1 match condition, the DBE bit of match condition setting register CBR1 becomes invalid, the settings of match data setting register CDR1 and match data mask setting register CDMR1 are ignored. Therefore, the data value cannot be specified for the instruction fetch cycle break.

31.3.4 Operand Access Cycle Break

1. Table 31.4 shows the relation between the operand sizes specified using the match condition setting register (CBR0 or CBR1) and the address bits to be compared for the operand access cycle break.

Table 31.4 Relation between Operand Sizes and Address Bits to be Compared

Selected Operand Size	Address Bits to be Compared
Quadword	Address bits A31 to A3
Longword	Address bits A31 to A2
Word	Address bits A31 to A1
Byte	Address bits A31 to A0
Operand size is not included in the match conditions	Address bits A31 to A3 for quadword access Address bits A31 to A2 for longword access Address bits A31 to A1 for word access Address bits A31 to A0 for byte access

The above table means that if address H'00001003 is set in the match address setting register (CAR0 or CAR1), for example, the match condition is satisfied for the following access cycles (assuming that all the other conditions are satisfied):

— Longword access to address H'00001000

— Word access to address H'00001002

— Byte access to address H'00001003

2. When the data value is included in the channel 1 match conditions:

   If the data value is included in the match conditions, be sure to select the quadword, longword, word, or byte as the operand size using the operand size select bit (SZ) of the match condition setting register (CBR1), and also set the match data setting register (CDR1) and the match data mask setting register (CDMR1). With these settings, the match condition is satisfied when both of the address and data conditions are satisfied. The data value and mask control for byte access, word access, and longword access should be set in bits 7 to 0, 15 to 0, and 31 to 0 in the bits CDR1 and CDMR1, respectively. For quadword access, 64-bit data is divided into the upper and lower 32-bit data units, and each unit is independently compared with the specified condition. When either the upper or lower 32-bit data unit satisfies the match condition, the match condition for the 64-bit data is determined to be satisfied.

3. The operand access accompanying the PREF, OCBP, OCBWB, and OCBI instructions are access without the data value; therefore, if the data value is included in the match conditions for these instructions, the match conditions will never be satisfied.

4. If the operand bus is selected, a break occurs after executing the instruction which has satisfied the conditions and immediately before executing the next instruction. However, if the data value is included in the match conditions, a break may occur after executing several instructions after the instruction which has satisfied the conditions; therefore, it is impossible to identify the instruction causing the break. If such a break has occurred for the delayed

branch instruction or its delayed slot, the break does not occur until the first instruction at the branch destination.

However, do not specify the operand break for the delayed slot of the RTE instruction. And if the data value is included in the match conditions, it is not allowed to set the break for the preceding the RTE instruction by one to six instructions.

31.3.5 Sequential Break

1. Sequential break conditions can be specified by setting the MFE and MFI bits in the match condition setting registers (CBR0 and CBR1). (Sequential break involves two cases such that channel 0 break condition is satisfied then channel 1 break condition is satisfied, and vice versa.) To use the sequential break function, clear the MFE bit of the match condition setting register and the BIE bit of the match operation setting register of the first channel in the sequence, and set the MFE bit and specify the number of the second channel in the sequence using the MFI bit in the match condition setting register of the second channel in the sequence. If the sequential break condition is set, the condition match flag is set every time the match condition is satisfied for each channel. When the condition has been satisfied for the first channel in the sequence but not for the second channel in the sequence, clear the condition match flag for the first channel in the sequence in order to release the first channel in the sequence from the match state.
2. For channel 1, the execution count break condition can also be included in the sequential break conditions.
3. If the match conditions for the first and second channels in the sequence are satisfied within a significantly short time, sequential operation may not be guaranteed in some cases, as shown below.

- When the Match Condition is Satisfied at the Instruction Fetch Cycle for Both the First and Second Channels in the Sequence:

Instruction B is 0 instruction after instruction A	Equivalent to setting the same addresses; do not use this setting.
Instruction B is one instruction after instruction A	Sequential operation is not guaranteed.
Instruction B is two or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the instruction fetch cycle for the first channel in the sequence whereas the match condition is satisfied at the operand access cycle for the second channel in the sequence:

Instruction B is 0 or one instruction after instruction A	Sequential operation is not guaranteed.
Instruction B is two or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the operand access cycle for the first channel in the sequence whereas the match condition is satisfied at the instruction fetch cycle for the second channel in the sequence:

Instruction B is 0 to five instructions after instruction A	Sequential operation is not guaranteed.
Instruction B is six or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the operand access cycle for both the first and second channels in the sequence:

Instruction B is 0 to five instructions after instruction A	Sequential operation is not guaranteed.
Instruction B is six or more instructions after instruction A	Sequential operation is guaranteed.

31.3.6 Program Counter Value to be Saved

When a break has occurred, the address of the instruction to be executed when the program restarts is saved in the SPC then the exception handling state is initiated. A unique instruction causing a break can be identified unless the data value is included in the match conditions.

1. When the instruction fetch cycle (before instruction execution) is specified as the match condition:

   The address of the instruction which has satisfied the match conditions is saved in the SPC. The instruction which has satisfied the match conditions is not executed, but a break occurs instead. However, if the match conditions are satisfied for the delayed slot instruction, the address of the delayed branch instruction is saved in the SPC.

2. When the instruction fetch cycle (after instruction execution) is specified as the match condition:

   The address of the instruction immediately after the instruction which has satisfied the match conditions is saved in the SPC. The instruction which has satisfied the match conditions is executed, then a break occurs before the next instruction. If the match conditions are satisfied for the delayed branch instruction or its delayed slot, these instructions are executed and the address of the branch destination is saved in the SPC.

3. When the operand access (address only) is specified as the match condition:

   The address of the instruction immediately after the instruction which has satisfied the break conditions is saved in the SPC. The instruction which has satisfied the match conditions are executed, then a break occurs before the next instruction. However, if the conditions are satisfied for the delayed slot, the address of the branch destination is saved in the SPC.

4. When the operand access (address and data) is specified as the match condition:

   If the data value is added to the match conditions, the instruction which has satisfied the match conditions is executed. A user break occurs before executing an instruction that is one through six instructions after the instruction which has satisfied the match conditions. The address of the instruction is saved in the SPC; thus, it is impossible to identify exactly where a break will occur. If the conditions are satisfied for the delayed slot instruction, the address of the branch destination is saved in the SPC. If a branch instruction follows the instruction which has satisfied the match conditions, a break may occur after the delayed instruction and delayed slot are executed. In this case, the address of the branch destination is also saved in the SPC.

31.4 User Break Debugging Support Function

By using the user break debugging support function, the branch destination address can be modified when the CPU accepts the user break request. Specifically, setting the UBDE bit of break control register CBCR to 1 allows branching to the address indicated by DBR instead of branching to the address indicated by the [VBR + offset]. Figure 31.2 shows the flowchart of the user break debugging support function.

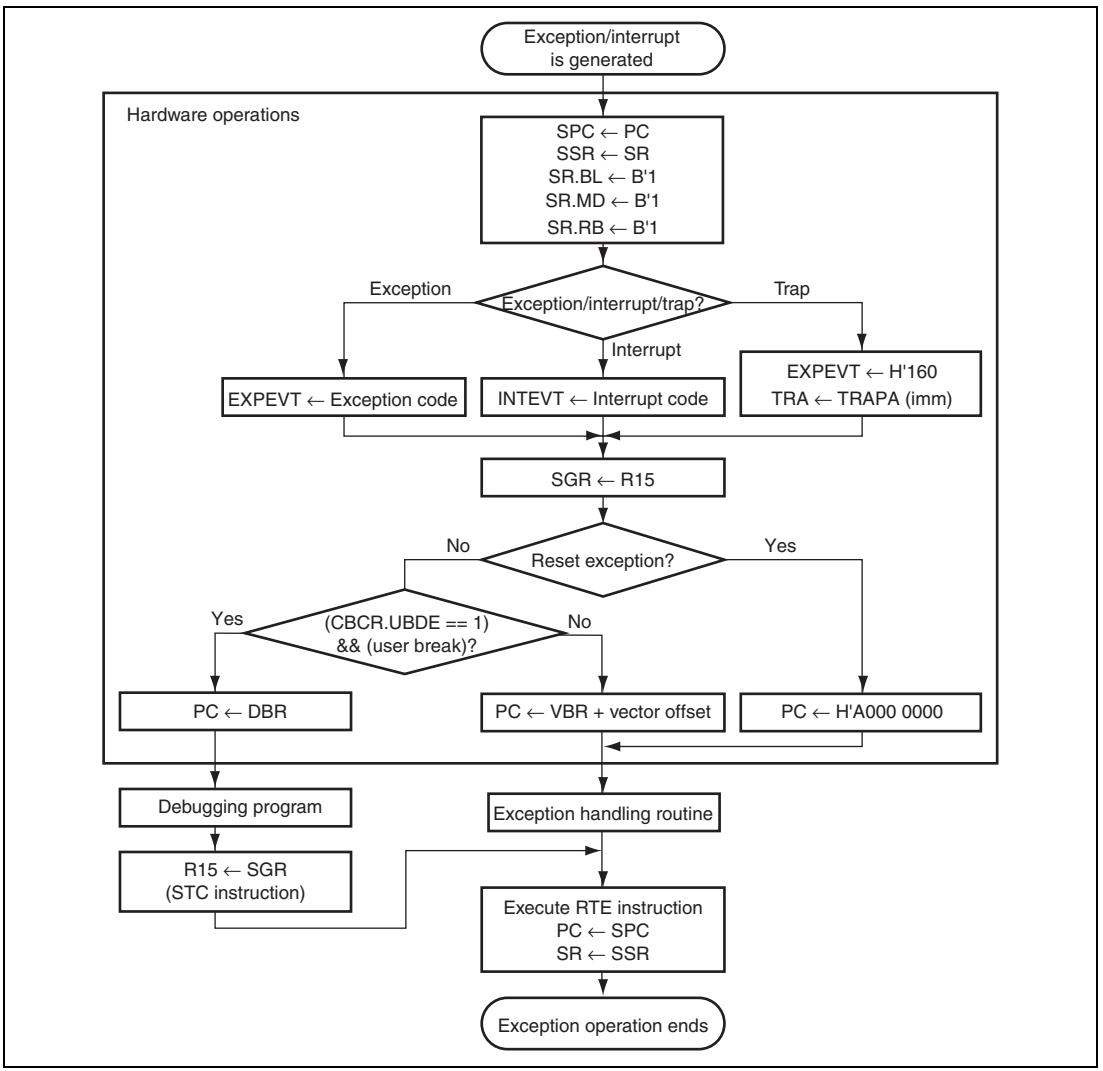


Figure 31.2 Flowchart of User Break Debugging Support Function

31.5 User Break Examples

(1) Match Conditions are Specified for an Instruction Fetch Cycle

- Example 1-1

  Register settings:

  CBR0 = H'00000013/CRR0 = H'00002003/CAR0 = H'00000404/

  CAMR0 = H'00000000/CBR1 = H'00000013/CRR1 = H'00002001/

  CAR1 = H'00008010/CAMR1 = H'00000006/CDR1 = H'00000000/

  CDMR1 = H'00000000/CETR1 = H'00000000/

  ECBR0 = H'00000000/ECBR1 = H'00000000/

  CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

    Address: H'00000404/Address mask: H'00000000

    Bus cycle: Instruction fetch (after executing the instruction)

    ASID is not included in the conditions.

  — Channel 1:

    Address: H'00008010/Address mask: H'00000006

    Data: H'00000000/Data mask: H'00000000/Execution count: H'00000000

    Bus cycle: Instruction fetch (before executing instruction)

    ASID, data values, and execution count are not included in the conditions.

  With the above settings, the user break occurs after executing the instruction at address H'00000404 or before executing the instruction at address H'00008010 to H'00008016.

- Example 1-2

  Register settings:

  CBR0 = H'40800013/CRR0 = H'00002000/CAR0 = H'00037226/CAMR0 = H'00000000/

  CBR1 = H'C0700013/CRR1 = H'00002001/CAR1 = H'0003722E/CAMR1 = H'00000000/

  CDR1 = H'00000000/CDMR1 = H'00000000/CETR1 = H'00000000/

  ECBR0 = H'00000000/ECBR1 = H'00000000/CBCR = H'00000000

  Specified conditions: Channel 0 → Channel1 sequential mode

  — Channel 0

    Address: H'00037226/Address mask: H'00000000/ASID: H'0080

    Bus cycle: Instruction fetch (before executing the instruction)

— Channel 1

Address: H'0003722E/Address mask: H'00000000/ASID: H'0070

Data: H'00000000/Data mask: H'00000000/Execution count: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00037226 where ASID is H'0080 before executing the instruction at address H'0003722E where ASID is H'0070.

- Example 1-3

Register settings:

CBR0 = H'00000013/CRR0 = H'00002001/CAR0 = H'00027128/CAMR0 = H'00000000/

CBR1 = H'00000013/CRR1 = H'00002001/CAR1 = H'00031415/CAMR1 = H'00000000/

CDR1 = H'00000000/CDMR1 = H'00000000/CETR1 = H'00000000/

ECBR0 = H'00000000/ECBR1 = H'00000000/CBCR = H'00000000

Specified conditions: Independent for channels 0 and 1

— Channel 0

Address: H'00027128/Address mask: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

ASID is not included in the conditions.

— Channel 1

Address: H'00031415/Address mask: H'00000000

Data: H'00000000/Data mask: H'00000000/Execution count: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

ASID, data values, and execution count are not included in the conditions.

With the above settings, the user break occurs for channel 0 before executing the instruction at address H'00027128. No user break occurs for channel 1 since the instruction fetch is executed only at even addresses.

- Example 1-4

Register settings:

CBR0 = H'40800013/CRR0 = H'00002000/CAR0 = H'00037226/CAMR0 = H'00000000/

CBR1 = H'C0700013/CRR1 = H'00002001/CAR1 = H'0003722E/CAMR1 = H'00000000/

CDR1 = H'00000000/CDMR1 = H'00000000/CETR1 = H'00000000/

ECBR0 = H'00000000/ECBR1 = H'00000000/CBCR = H'00000000

Specified conditions: Channel 0 → Channel 1 sequential mode

— Channel 0

Address: H'00037226/Address mask: H'00000000/ASID: H'0080

Bus cycle: Instruction fetch (before executing the instruction)

— Channel 1

Address: H'0003722E/Address mask: H'00000000/ASID: H'0070

Data: H'00000000/Data mask: H'00000000/Execution count: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00037226 where ASID is H'0080 and before executing the instruction at address H'0003722E where ASID is H'0070.

- Example 1-5

Register settings:

CBR0 = H'00000013/CRR0 = H'00002001/CAR0 = H'00000500/CAMR0 = H'00000000/

CBR1 = H'00000813/CRR1 = H'00002001/CAR1 = H'00001000/CAMR1 = H'00000000/

CDR1 = H'00000000/CDMR1 = H'00000000/CETR1 = H'00000005/

ECBR0 = H'00000000/ECBR1 = H'00000000/CBCR = H'00000000

Specified conditions: Independent for channels 0 and 1

— Channel 0

Address: H'00000500/Address mask: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

ASID is not included in the conditions.

— Channel 1

Address: H'00001000/Address mask: H'00000000

Data: H'00000000/Data mask: H'00000000/Execution count: H'00000005

Bus cycle: Instruction fetch (before executing the instruction)

Execution count: 5

ASID and data values are not included in the conditions.

With the above settings, the user break occurs for channel 0 before executing the instruction at address H'00000500. The user break occurs for channel 1 after executing the instruction at address H'00001000 four times; before executing the instruction five times.

- Example 1-6

  Register settings:

  CBR0 = H'40800013/CRR0 = H'00002003/CAR0 = H'00008404/CAMR0 = H'00000FFF/

  CBR1 = H'40700013/CRR1 = H'00002001/CAR1 = H'00008010/CAMR1 = H'00000006/

  CDR1 = H'00000000/CDMR1 = H'00000000/CETR1 = H'00000000/

  ECBR0 = H'00000000/ECBR1 = H'00000000/

  CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

    Address: H'00008404/Address mask: H'00000FFF/ASID: H'0080

    Bus cycle: Instruction fetch (after executing the instruction)

  — Channel 1

    Address: H'00008010/Address mask: H'00000006/ASID: H'0070

    Data: H'00000000/Data mask: H'00000000/Execution count: H'00000000

    Bus cycle: Instruction fetch (before executing the instruction)

    Data values and execution count are not included in the conditions.

  With the above settings, the user break occurs after executing the instruction at address H'00008000 to H'00008FFE where ASID is H'0080 or before executing the instruction at address H'00008010 to H'00008016 where ASID is H'0070.

(2) Match Conditions are Specified for an Operand Access Cycle

- Example 2-1

  Register settings:

  CBR0 = H'40800023/CRR0 = H'00002001/CAR0 = H'00123456/CAMR0 = H'00000000/

  CBR1 = H'4070A025/CRR1 = H'00002001/CAR1 = H'000ABCDE/

  CAMR1 = H'000000FF/CDR1 = H'0000A512/CDMR1 = H'00000000/

  CETR1 = H'00000000/

  ECBR0 = H'00000000/ECBR1 = H'00000000/

  CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

    Address: H'00123456/Address mask: H'00000000/ASID: H'0080

    Bus cycle: Operand bus, operand access, and read (operand size is not included in the conditions.)

— Channel 1

Address: H'000ABCDE/Address mask: H'000000FF/ASID: H'0070

Data: H'0000A512/Data mask: H'00000000/Execution count: H'00000000

Bus cycle: Operand bus, operand access, write, and word size

Execution count is not included in the conditions.

With these settings, the user break occurs for channel 0 for the following accesses: longword read access to address H'000123454, word read access to address H'000123456, byte read access to address H'000123456 where ASID is H'0080. The user break occurs for channel 1 when word H'A512 is written to address H'000ABC00 to H'000ABCFE where ASID is H'0070.

31.6 Usage Notes

1. A desired break may not occur between the time when the instruction for rewriting the UBC register is executed and the time when the written value is actually reflected on the register.

   After the UBC register is updated, execute one of the following three methods.

   A. Read the updated UBC register, and execute a branch using the RTE instruction.
   (It is not necessary that a branch using the RTE instruction is next to a reading UBC register.)

   B. Execute the ICBI instruction for any address (including non-cacheable area).
   (It is not necessary that the ICBI instruction is next to a reading UBC register.)

   C. Set 0 (initial value) to IRMCR.R1 before updating the UBC register and update with following sequence.

      a. Write the UBC register.

      b. Read the UBC register which is updated at 1.

      c. Write the value which is read at 2 to the UBC register.

Note: When two or more UBC registers are updated, executing these methods at each updating the UBC registers is not necessary. At only last updating the UBC register, execute one of these methods.

2. The PCB bit of the CRR0 and CRR1 registers is valid only when the instruction fetch is specified as the match condition.
3. If the sequential break conditions are set, the sequential break conditions are satisfied when the conditions for the first and second channels in the sequence are satisfied in this order. Therefore, if the conditions are set so that the conditions for channels 0 and 1 should be satisfied simultaneously for the same bus cycle, the sequential break conditions will not be satisfied, causing no break.
4. For the SLEEP instruction, do not allow the post-instruction-execution break where the instruction fetch cycle is the match condition. For the instructions preceding the SLEEP instruction by one to five instructions, do not allow the break where the operand access is the match condition.
5. If the user break and other exceptions occur for the same instruction, they are determined according to the specified priority. For the priority, refer to section 5, Exception Handling. If the exception having the higher priority occurs, the user break does not occur.

   — The pre-instruction-execution break is accepted prior to any other exception.

— If the post-instruction-execution break and data access break have occurred simultaneously with the re-execution type exception (including the pre-instruction-execution break) having a higher priority, only the re-execution type exception is accepted, and no condition match flags are set. When the exception handling has finished thus clearing the exception source, and when the same instruction has been executed again, the break occurs setting the corresponding flag.

— If the post-instruction-execution break or operand access break has occurred simultaneously with the completion-type exception (TRAPA) having a higher priority, then no user break occurs; however, the condition match flag is set.

6. When conditions have been satisfied simultaneously and independently for channels 0 and 1, resulting in identical SPC values for both of the breaks, the user break occurs only once. However, the condition match flags are set for both channels. For example,

   Instruction at address 110 (post-instruction-execution break for instruction fetch for channel 0) → SPC = 112, CCMFR.MF0 = 1

   Instruction at address 112 (pre-instruction-execution break for instruction fetch for channel 1) → SPC = 112, CCMFR.MF1 = 1

7. It is not allowed to set the pre-instruction-execution break or the operand break in the delayed slot instruction of the RTE instruction. And if the data value is included in the match conditions of the operand break, do not set the break for the preceding the RTE instruction by one to six instructions.

8. If the re-execution type exception and the post-instruction-execution break are in conflict for the instruction requiring two or more execution states, then the re-execution type exception occurs. Here, the CCMFR.MF0 (or CCMFR.MF1) bit may or may not be set to 1 when the break conditions have been satisfied.

Section 32 User Debugging Interface (H-UDI)

The H-UDI is a serial input/output interface which supports to a subset of JTAG (IEEE 1149.1). The H-UDI is used to connect emulators.

32.1 Features

The H-UDI is a serial input/output interface which supports to a subset of JTAG (IEEE 1149.1: IEEE Standard Test Access Port and Boundary-Scan Architecture). The H-UDI is used to connect emulators. Do not use the JTAG functions of this interface when using an emulator. For the method of connecting the emulator, see emulator manuals.

The H-UDI has six pins, the TCK, TMS, TDI, TDO, $\overline{\text{TRST}}$, and $\overline{\text{ASEBRK}}$/BRKACK pins. The pin functions except $\overline{\text{ASEBRK}}$/BRKACK, and serial transfer protocols conform to JTAG with the subset. Also, the H-UDI has six signals (AUDSYNC, AUDCK, and AUDATA3 to AUDATA0) used for emulator pins, and a signal (MPMD) for the chip mode select pin.

In the H-UDI in this LSI, the boundary-scan test access port (TAP) controller is separated from the TAP controller for other H-UDI function control. When the $\overline{\text{TRST}}$ is asserted (including when the power is turned on), the boundary-scan TAP controller is selected. Therefore, the switching command should be input to use the H-UDI functions. The boundary-scan TAP controller cannot be accessed through the CPU.

Figure 32.1 shows a block diagram of the H-UDI.

The H-UDI circuit has TAP controllers and four registers (SDBPR, SDBSR, SDIR, and SDINT). SDBPR supports the JTAG bypass mode, SDBSR supports the JTAG boundary scan mode, SDIR is used for commands, and SDINT is used for H-UDI interrupts. SDIR can be directly accessed through the TDI and TDO pins.

Without reset pins of the chip, the TAP controller, control registers, and boundary-scan TAP controller are reset when the $\overline{\text{TRST}}$ pin is set to low or when five or more TCK cycles are elapsed after TMS is set to 1. The other circuits are reset in a normal reset period, and initialized.

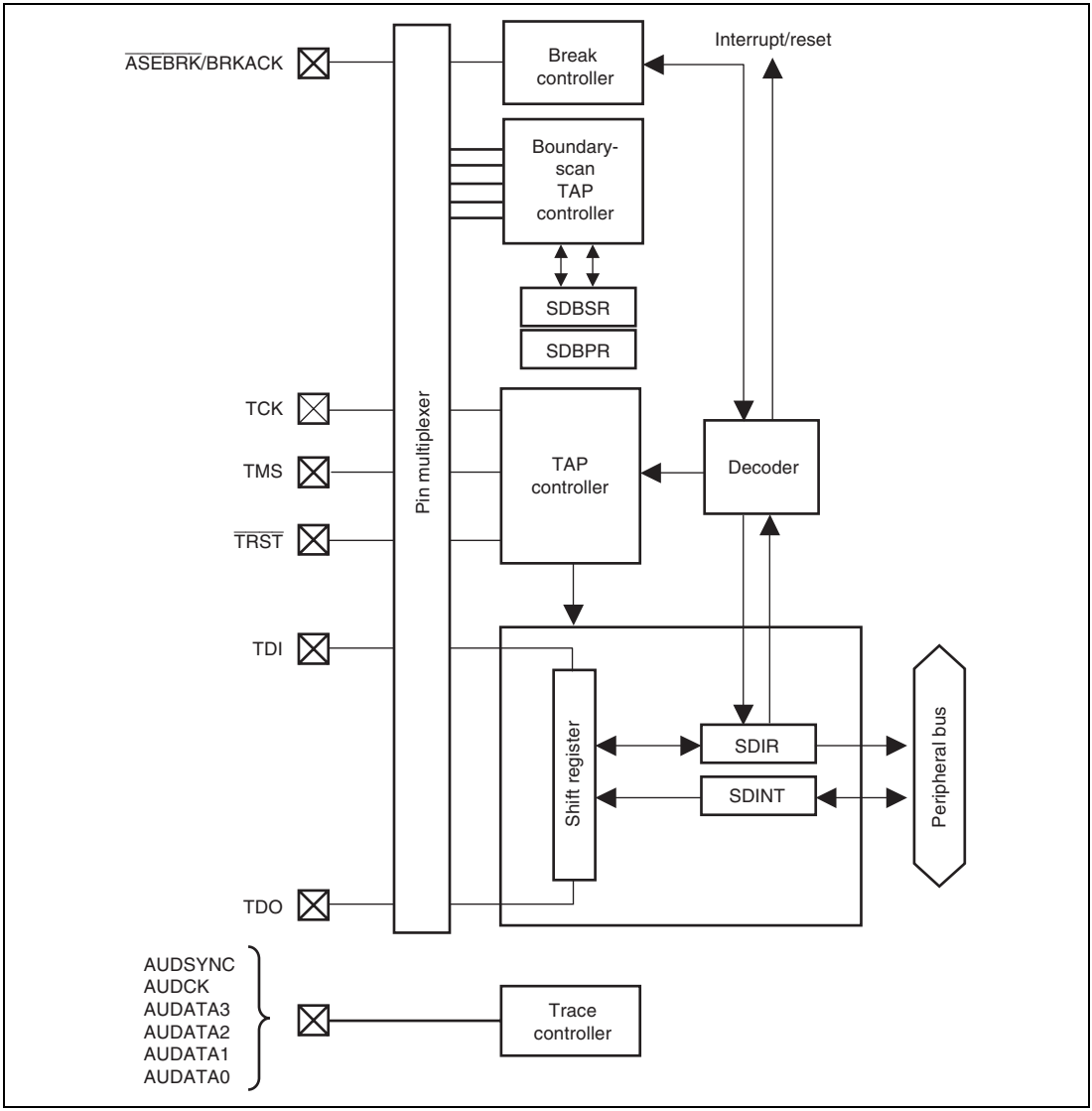


Figure 32.1 Block Diagram of H-UDI

32.2 Input/Output Pins

Table 32.1 shows the pin configuration of the H-UDI.

Table 32.1 Pin Configuration of H-UDI

Pin Name	Function	I/O	Description	When Not in Use
TCK	Clock	Input	The functions are the same as the serial clock input pin of JTAG. In synchronization with this signal, data is sent from the TDI pin to the H-UDI circuit or data is read from the TDO pin.	Open*[1]
TMS	Mode	Input	Mode Select Input Pin The meaning of data input from the TDI pin is determined by changing this signal in synchronization with TCK. The protocol supports JTAG (IEEE 1149.1) with the subset.	Open*[1]
$\overline{\text{TRST}}$*[2]	Reset	Input	H-UDI Reset Input Pin This signal is received without relating to TCK. The JTAG interface circuit is reset when this signal is at a low level. Regardless of whether JTAG is used or not, $\overline{\text{TRST}}$ should be set to a low level during a specific period when power is turned on. This does not conform to the IEEE standard.	Fixed to ground or connected to the $\overline{\text{PRESET}}$ pin*[3]
TDI	Data input	Input	Data Input Pin Data is sent to the H-UDI circuit by changing this signal in synchronization with TCK.	Open*[1]
TDO	Data output	Output	Data Output Pin Data is read from the H-UDI circuit by reading this signal in synchronization with TCK.	Open
$\overline{\text{ASEBRK}}$/ BRKACK	Emulator	I/O	Pins for emulators	Open*[1]
AUDSYNC, AUDCK, AUDATA3 to AUDATA0	Emulator	Output	Pins for emulators	Open
MPMD	Chip-mode	Input	Indicates whether the operating mode of this LSI is emulation support mode (MPMD = 0) or chip mode (MPMD = 1).	Open

Notes: 1. This pin is pulled up in the chip. In designing the board that can connect an emulator, or using interrupts or resets through the H-UDI, there is no problem with putting the pull-up resistor outside this LSI.

2. In designing the board that can connect an emulator, or using interrupts or resets through the H-UDI, the $\overline{\text{TRST}}$ pin should be set so that it can be held low while the $\overline{\text{PRESET}}$ signal is low after power is turned on and the $\overline{\text{TRST}}$ pin can be controlled independently.

3. This pin should be connected to ground, or connected to the same signal as the $\overline{\text{PRESET}}$, or a pin which operates in the same way as the $\overline{\text{PRESET}}$ pin. When this pin is connected to ground, the following problem occurs. Since the $\overline{\text{TRST}}$ pin is pulled up within this LSI, weak current runs when the pin is connected to ground outside the LSI. The current depends on a resistance of the pull-up for the port pin. Although this current does not affect the operation of this LSI, it consumes unnecessary power.

The TCK clock or the CPG of this LSI should be set so that the frequency of the TCK clock is less than that of the peripheral clock of this LSI. For details of the setting of the CPG, see section 18, Clock Pulse Generator (CPG).

32.3 Register Description

The H-UDI has the following registers.

Table 32.2 Register Configuration (1)

		CPU Side				
Register Name	**Abbreviation**	**R/W**	**Area P4 Address**	**Area 7 Address**	**Size**	**Sync Clock**
Instruction register	SDIR	R	H'FC11 0000	H'1C11 0000	16	Pck
Interrupt source register	SDINT	R/W	H'FC11 0018	H'1C11 0018	16	Pck
Boundary scan register	SDBSR	—	—	—	—	—
Bypass register	SDBPR	—	—	—	—	—

Table 32.3 Register Configuration (2)

		H-UDI Side		
Register Name	**Abbreviation**	**R/W**	**Size**	**Sync Clock**
Instruction register	SDIR	R/W*[1]	32	Pck
Interrupt source register	SDINT	W*[2]	32	Pck
Boundary scan register	SDBSR	R/W	—	—
Bypass register	SDBPR	R/W	1	—

Notes: 1. The read value from the H-UDI is always fixed to H'FFFF FFFD.
2. 1 can be written to the LSB by the H-UDI interrupt command.

Table 32.4 Register States in Each Processing State

Register Name	**Abbreviation**	**Power-On Reset**	**Manual Reset**	**Module Standby**	**Sleep**	**Light Sleep**
Instruction register	SDIR	H'0EFF	Retained	Retained	Retained	Retained
Interrupt source register	SDINT	H'0000	Retained	Retained	Retained	Retained

32.3.1 Instruction Register (SDIR)

SDIR is a 16-bit read-only register that can be read from the CPU. Commands are set via the serial input pin (TDI). SDIR is initialized by $\overline{\text{TRST}}$ or in the Test-Logic-Reset state of the TAP. This register can be written to by the H-UDI, regardless of the CPU mode. Operation is not guaranteed when a reserved command is set to this register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TI								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TI	0000 1110	R	Test Instruction Bits 7 to 0 0110 xxxx: H-UDI reset negate 0111 xxxx: H-UDI reset assert 101x xxxx: H-UDI interrupt 0000 1110: Initial state Other than above: Setting prohibited
7 to 0	—	All 1	R	Reserved These bits are always read as 1.

32.3.2 Interrupt Source Register (SDINT)

SDINT is a 16-bit register that can be read from or written to by the CPU. If the H-UDI interrupt command (Update-IR) is set to SDIR, the INTREQ bit is set to 1. When an H-UDI interrupt command is set in SDIR, SDINT is connected between the TDI and TDO pins, and becomes a 32-bit readable register. In this case, the upper 16 bits are 0 and the lower 16 bits are values specified in SDINT.

Only 0 can be written to the INTREQ bit by the CPU. While this bit is set to 1, an interrupt request continues to be generated. Therefore, clear this bit to 0 in an interrupt handler and read this bit again to confirm that this bit is cleared. This register is initialized by $\overline{\text{TRST}}$ or in the test-logic-reset state.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INTREQ
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INTREQ	0	R/W	Interrupt Request Indicates whether or not an interrupt by an H-UDI interrupt command has occurred. Clearing this bit to 0 by the CPU cancels an interrupt request. When writing 1 to this bit, the previous value is maintained.

32.3.3 Bypass Register (SDBPR)

SDBPR is a 1-bit register that supports the JTAG bypass mode. When the BYPASS command is set to the boundary-scan TAP controller, SDBPR is connected between the TDI and TDO pins. This register cannot be accessed through the CPU. This register is not initialized by a power-on reset or assertion of $\overline{\text{TRST}}$, but it is initialized to 0 by the Capture-DR state.

32.3.4 Boundary Scan Register (SDBSR)

SDBSR is a register that supports the JTAG boundary scan mode. SDBSR is a shift register that is located on the PAD, to control the input/output pins. By using the SAMPLE/PRELOAD and EXTEST commands, this register can perform the boundary scan test that supports the JTAG standard (IEEE 1149.1) with the subset. This register cannot be accessed through the CPU, regardless of chip mode. This register is not initialized by a power-on reset or assertion of $\overline{\text{TRST}}$.

Table 32.5 Boundary Scan Register Configuration

Number	Pin Name	Type	Number	Pin Name	Type
	From TDI		427	HAC1_SYNC/SSI1_WS/SDIF1WP	CONTROL
440	IRL0	OUTPUT	426	HAC1_SYNC/SSI1_WS/SDIF1WP	INPUT
439	IRL0	CONTROL	425	HAC1_BITCLK/SSI1_CLK/SDIF1CLK	OUTPUT
438	IRL0	INPUT	424	HAC1_BITCLK/SSI1_CLK/SDIF1CLK	CONTROL
437	NMI	OUTPUT	423	HAC1_BITCLK/SSI1_CLK/SDIF1CLK	INPUT
436	NMI	CONTROL	422	HAC0_SDOUT/SSI0_SDATA/SDIF1D3	OUTPUT
435	NMI	INPUT	421	HAC0_SDOUT/SSI0_SDATA/SDIF1D3	CONTROL
434	HAC1_SDOUT/SSI1_SDATA/SDIF1CMD	OUTPUT	420	HAC0_SDOUT/SSI0_SDATA/SDIF1D3	INPUT
433	HAC1_SDOUT/SSI1_SDATA/SDIF1CMD	CONTROL	419	HAC0_SDIN/SSI0_SCK/SDIF1D2	OUTPUT
432	HAC1_SDOUT/SSI1_SDATA/SDIF1CMD	INPUT	418	HAC0_SDIN/SSI0_SCK/SDIF1D2	CONTROL
431	HAC1_SDIN/SSI1_SCK/$\overline{\text{SDIF1CD}}$	OUTPUT	417	HAC0_SDIN/SSI0_SCK/SDIF1D2	INPUT
430	HAC1_SDIN/SSI1_SCK/$\overline{\text{SDIF1CD}}$	CONTROL	416	HAC0_SYNC/SSI0_WS/SDIF1D1	OUTPUT
429	HAC1_SDIN/SSI1_SCK/$\overline{\text{SDIF1CD}}$	INPUT	415	HAC0_SYNC/SSI0_WS/SDIF1D1	CONTROL
428	HAC1_SYNC/SSI1_WS/SDIF1WP	OUTPUT	414	HAC0_SYNC/SSI0_WS/SDIF1D1	INPUT

Number	Pin Name	Type	Number	Pin Name	Type
413	HAC0_BITCLK/SSI0_CLK/SDIF1D0	OUTPUT	388	I2C_SDA1/IRQOUT	CONTROL
412	HAC0_BITCLK/SSI0_CLK/SDIF1D0	CONTROL	387	I2C_SDA1/IRQOUT	INPUT
411	HAC0_BITCLK/SSI0_CLK/SDIF1D0	INPUT	386	I2C_SCL1/SCIF2_SCK	OUTPUT
410	DREQ1/BREQ/USB_OVC1	OUTPUT	385	I2C_SCL1/SCIF2_SCK	CONTROL
409	DREQ1/BREQ/USB_OVC1	CONTROL	384	I2C_SCL1/SCIF2_SCK	INPUT
408	DREQ1/BREQ/USB_OVC1	INPUT	383	I2C_SDA0/SCIF2_TXD	OUTPUT
407	IRL3	OUTPUT	382	I2C_SDA0/SCIF2_TXD	CONTROL
406	IRL3	CONTROL	381	I2C_SDA0/SCIF2_TXD	INPUT
405	IRL3	INPUT	380	I2C_SCL0/SCIF2_RXD	OUTPUT
404	IRL3	OUTPUT	379	I2C_SCL0/SCIF2_RXD	CONTROL
403	IRL2	CONTROL	378	I2C_SCL0/SCIF2_RXD	INPUT
402	IRL2	INPUT	377	SCIF3_TXD/HAC_RES/SSI2_WS	OUTPUT
401	IRL1	OUTPUT	376	SCIF3_TXD/HAC_RES/SSI2_WS	CONTROL
400	IRL1	CONTROL	375	SCIF3_TXD/HAC_RES/SSI2_WS	INPUT
399	IRL1	INPUT	374	SCIF3_RXD/TCLK/SSI2_SCK	OUTPUT
398	USB_PENC1	OUTPUT	373	SCIF3_RXD/TCLK/SSI2_SCK	CONTROL
397	USB_PENC1	CONTROL	372	SCIF3_RXD/TCLK/SSI2_SCK	INPUT
396	USB_PENC1	INPUT	371	SCIF3_SCK/SSI2_SDATA	OUTPUT
395	USB_PENC0	OUTPUT	370	SCIF3_SCK/SSI2_SDATA	CONTROL
394	USB_PENC0	CONTROL	369	SCIF3_SCK/SSI2_SDATA	INPUT
393	USB_PENC0	INPUT	368	STATUS1/SSI3_CLK	OUTPUT
392	DREQ0/USB_OVC0	OUTPUT	367	STATUS1/SSI3_CLK	CONTROL
391	DREQ0/USB_OVC0	CONTROL	366	STATUS1/SSI3_CLK	INPUT
390	DREQ0/USB_OVC0	INPUT	365	STATUS0/SSI2_CLK	OUTPUT
389	I2C_SDA1/IRQOUT	OUTPUT	364	STATUS0/SSI2_CLK	CONTROL

Number	Pin Name	Type	Number	Pin Name	Type
363	STATUS0/SSI2_CLK	INPUT	331	D21	CONTROL
362	D31	OUTPUT	330	D21	INPUT
361	D31	CONTROL	329	D20	OUTPUT
360	D31	INPUT	328	D20	CONTROL
359	D30	OUTPUT	327	D20	INPUT
358	D30	CONTROL	326	D19	OUTPUT
357	D30	INPUT	325	D19	CONTROL
356	D29	OUTPUT	324	D19	INPUT
355	D29	CONTROL	323	D18	OUTPUT
354	D29	INPUT	322	D18	CONTROL
353	D28	OUTPUT	321	D18	INPUT
352	D28	CONTROL	320	D17	OUTPUT
351	D28	INPUT	319	D17	CONTROL
350	D27	OUTPUT	318	D17	INPUT
349	D27	CONTROL	317	D16	OUTPUT
348	D27	INPUT	316	D16	CONTROL
347	D26	OUTPUT	315	D16	INPUT
346	D26	CONTROL	314	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	OUTPUT
345	D26	INPUT	313	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	CONTROL
344	D25	OUTPUT	312	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	INPUT
343	D25	CONTROL	311	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	OUTPUT
342	D25	INPUT	310	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	CONTROL
341	D24	OUTPUT	309	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	INPUT
340	D24	CONTROL	308	$\overline{\text{WE1}}$/$\overline{\text{FEW}}$	OUTPUT
339	D24	INPUT	307	$\overline{\text{WE1}}$/$\overline{\text{FEW}}$	CONTROL
338	D23	OUTPUT	306	$\overline{\text{WE1}}$/$\overline{\text{FEW}}$	INPUT
337	D23	CONTROL	305	$\overline{\text{WE0}}$/$\overline{\text{REG}}$	OUTPUT
336	D23	INPUT	304	$\overline{\text{WE0}}$/$\overline{\text{REG}}$	CONTROL
335	D22	OUTPUT	303	$\overline{\text{WE0}}$/$\overline{\text{REG}}$	INPUT
334	D22	CONTROL	302	D15	OUTPUT
333	D22	INPUT	301	D15	CONTROL
332	D21	OUTPUT	300	D15	INPUT

Number	Pin Name	Type	Number	Pin Name	Type
299	D14	OUTPUT	267	D4/FD4	INPUT
298	D14	CONTROL	266	D3/FD3	OUTPUT
297	D14	INPUT	265	D3/FD3	CONTROL
296	D13	OUTPUT	264	D3/FD3	INPUT
295	D13	CONTROL	263	D2/FD2	OUTPUT
294	D13	INPUT	262	D2/FD2	CONTROL
293	D12	OUTPUT	261	D2/FD2	INPUT
292	D12	CONTROL	260	D1/FD1	OUTPUT
291	D12	INPUT	259	D1/FD1	CONTROL
290	D11	OUTPUT	258	D1/FD1	INPUT
289	D11	CONTROL	257	D0/FD0	OUTPUT
288	D11	INPUT	256	D0/FD0	CONTROL
287	D10	OUTPUT	255	D0/FD0	INPUT
286	D10	CONTROL	254	$\overline{BS}$	OUTPUT
285	D10	INPUT	253	$\overline{BS}$	CONTROL
284	D9	OUTPUT	252	$\overline{BS}$	INPUT
283	D9	CONTROL	251	CLKOUTENB	OUTPUT
282	D9	INPUT	250	CLKOUTENB	CONTROL
281	D8	OUTPUT	249	CLKOUTENB	INPUT
280	D8	CONTROL	248	CLKOUT	OUTPUT
279	D8	INPUT	247	CLKOUT	CONTROL
278	D7/FD7	OUTPUT	246	CLKOUT	INPUT
277	D7/FD7	CONTROL	245	$\overline{RDY}$	OUTPUT
276	D7/FD7	INPUT	244	$\overline{RDY}$	CONTROL
275	D6/FD6	OUTPUT	243	$\overline{RDY}$	INPUT
274	D6/FD6	CONTROL	242	R/$\overline{W}$	OUTPUT
273	D6/FD6	INPUT	241	R/$\overline{W}$	CONTROL
272	D5/FD5	OUTPUT	240	R/$\overline{W}$	INPUT
271	D5/FD5	CONTROL	239	A25	OUTPUT
270	D5/FD5	INPUT	238	A25	CONTROL
269	D4/FD4	OUTPUT	237	A25	INPUT
268	D4/FD4	CONTROL	236	A24	OUTPUT

Number	Pin Name	Type	Number	Pin Name	Type
235	A24	CONTROL	203	$\overline{CS3}$	OUTPUT
234	A24	INPUT	202	$\overline{CS3}$	CONTROL
233	MODE11/DRAK3/$\overline{CE2A}$	OUTPUT	201	$\overline{CS3}$	INPUT
232	MODE11/DRAK3/$\overline{CE2A}$	CONTROL	200	$\overline{CS4}$	OUTPUT
231	MODE11/DRAK3/$\overline{CE2A}$	INPUT	199	$\overline{CS4}$	CONTROL
230	MODE12/SCIF5_TXD/$\overline{CE2B}$	OUTPUT	198	$\overline{CS4}$	INPUT
229	MODE12/SCIF5_TXD/$\overline{CE2B}$	CONTROL	197	$\overline{CS5}$	OUTPUT
228	MODE12/SCIF5_TXD/$\overline{CE2B}$	INPUT	196	$\overline{CS5}$	CONTROL
227	MODE13/SCIF5_RXD/$\overline{IOIS16}$	OUTPUT	195	$\overline{CS5}$	INPUT
226	MODE13/SCIF5_RXD/$\overline{IOIS16}$	CONTROL	194	$\overline{CS6}$	OUTPUT
225	MODE13/SCIF5_RXD/$\overline{IOIS16}$	INPUT	193	$\overline{CS6}$	CONTROL
224	MODE14/SCIF5_SCK/$\overline{FRB}$	OUTPUT	192	$\overline{CS6}$	INPUT
223	MODE14/SCIF5_SCK/$\overline{FRB}$	CONTROL	191	A23	OUTPUT
222	MODE14/SCIF5_SCK/$\overline{FRB}$	INPUT	190	A23	CONTROL
221	DACK1/$\overline{BACK}$/FALE	OUTPUT	189	A23	INPUT
220	DACK1/$\overline{BACK}$/FALE	CONTROL	188	A22	OUTPUT
219	DACK1/$\overline{BACK}$/FALE	INPUT	187	A22	CONTROL
218	DACK0/FCLE	OUTPUT	186	A22	INPUT
217	DACK0/FCLE	CONTROL	185	A21	OUTPUT
216	DACK0/FCLE	INPUT	184	A21	CONTROL
215	$\overline{RD}$/$\overline{FRAME}$/$\overline{FRE}$	OUTPUT	183	A21	INPUT
214	$\overline{RD}$/$\overline{FRAME}$/$\overline{FRE}$	CONTROL	182	A20	OUTPUT
213	$\overline{RD}$/$\overline{FRAME}$/$\overline{FRE}$	INPUT	181	A20	CONTROL
212	$\overline{CS0}$/$\overline{FCE0}$	OUTPUT	180	A20	INPUT
211	$\overline{CS0}$/$\overline{FCE0}$	CONTROL	179	A19	OUTPUT
210	$\overline{CS0}$/$\overline{FCE0}$	INPUT	178	A19	CONTROL
209	$\overline{CS1}$/A26	OUTPUT	177	A19	INPUT
208	$\overline{CS1}$/A26	CONTROL	176	A18	OUTPUT
207	$\overline{CS1}$/A26	INPUT	175	A18	CONTROL
206	$\overline{CS2}$	OUTPUT	174	A18	INPUT
205	$\overline{CS2}$	CONTROL	173	A17	OUTPUT
204	$\overline{CS2}$	INPUT	172	A17	CONTROL

Number	Pin Name	Type	Number	Pin Name	Type
171	A17	INPUT	143	A7	OUTPUT
170	A16	OUTPUT	142	A7	CONTROL
169	A16	CONTROL	141	A7	INPUT
168	A16	INPUT	140	A6	OUTPUT
167	A15	OUTPUT	139	A6	CONTROL
166	A15	CONTROL	138	A6	INPUT
165	A15	INPUT	137	A5	OUTPUT
164	A14	OUTPUT	136	A5	CONTROL
163	A14	CONTROL	135	A5	INPUT
162	A14	INPUT	134	A4	OUTPUT
161	A13	OUTPUT	133	A4	CONTROL
160	A13	CONTROL	132	A4	INPUT
159	A13	INPUT	131	A3	OUTPUT
158	A12	OUTPUT	130	A3	CONTROL
157	A12	CONTROL	129	A3	INPUT
156	A12	INPUT	128	A2	OUTPUT
155	A11	OUTPUT	127	A2	CONTROL
154	A11	CONTROL	126	A2	INPUT
153	A11	INPUT	125	A1	OUTPUT
152	A10	OUTPUT	124	A1	CONTROL
151	A10	CONTROL	123	A1	INPUT
150	A10	INPUT	122	A0	OUTPUT
149	A9	OUTPUT	121	A0	CONTROL
148	A9	CONTROL	120	A0	INPUT
147	A9	INPUT	119	MODE10/SCIF4_SCK/DRAK2/SSI3_WS	OUTPUT
146	A8	OUTPUT	118	MODE10/SCIF4_SCK/DRAK2/SSI3_WS	CONTROL
145	A8	CONTROL	117	MODE10/SCIF4_SCK/DRAK2/SSI3_WS	INPUT
144	A8	INPUT	116	MODE9/SCIF4_RXD/DRAK1/SSI3_SDATA	OUTPUT

Number	Pin Name	Type
115	MODE9/SCIF4_RXD/DRAK1/ SSI3_SDATA	CONTROL
114	MODE9/SCIF4_RXD/DRAK1/ SSI3_SDATA	INPUT
113	MODE8/SCIF4_TXD/DRAK0/ SSI3_SCK/$\overline{\text{FSE}}$	OUTPUT
112	MODE8/SCIF4_TXD/DRAK0/ SSI3_SCK/$\overline{\text{FSE}}$	CONTROL
111	MODE8/SCIF4_TXD/DRAK0/ SSI3_SCK/$\overline{\text{FSE}}$	INPUT
110	MODE7/DACK3/SDIF0CMD	OUTPUT
109	MODE7/DACK3/SDIF0CMD	CONTROL
108	MODE7/DACK3/SDIF0CMD	INPUT
107	MODE6/DACK2/$\overline{\text{SDIF0CD}}$	OUTPUT
106	MODE6/DACK2/$\overline{\text{SDIF0CD}}$	CONTROL
105	MODE6/DACK2/$\overline{\text{SDIF0CD}}$	INPUT
104	MODE5/$\overline{\text{DREQ3}}$/SDIF0WP	OUTPUT
103	MODE5/$\overline{\text{DREQ3}}$/SDIF0WP	CONTROL
102	MODE5/$\overline{\text{DREQ3}}$/SDIF0WP	INPUT
101	MODE4/$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{DREQ2}}$/ SDIF0CLK	OUTPUT
100	MODE4/$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{DREQ2}}$/ SDIF0CLK	CONTROL
99	MODE4/$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{DREQ2}}$/ SDIF0CLK	INPUT
98	MODE3/$\overline{\text{SCIF0\_RTS}}$/$\overline{\text{IRL7}}$/ SDIF0D3	OUTPUT
97	MODE3/$\overline{\text{SCIF0\_RTS}}$/$\overline{\text{IRL7}}$/ SDIF0D3	CONTROL
96	MODE3/$\overline{\text{SCIF0\_RTS}}$/$\overline{\text{IRL7}}$/ SDIF0D3	INPUT
95	MODE2/SCIF0_SCK/$\overline{\text{IRL6}}$/ SDIF0D2	OUTPUT
94	MODE2/SCIF0_SCK/$\overline{\text{IRL6}}$/ SDIF0D2	CONTROL
93	MODE2/SCIF0_SCK/$\overline{\text{IRL6}}$/ SDIF0D2	INPUT
92	MODE1/SCIF0_RXD/$\overline{\text{IRL5}}$ /SDIF0D1	OUTPUT
91	MODE1/SCIF0_RXD/$\overline{\text{IRL5}}$ /SDIF0D1	CONTROL
90	MODE1/SCIF0_RXD/$\overline{\text{IRL5}}$ /SDIF0D1	INPUT
89	MODE0/SCIF0_TXD/$\overline{\text{IRL4}}$/ SDIF0D0	OUTPUT
88	MODE0/SCIF0_TXD/$\overline{\text{IRL4}}$/ SDIF0D0	CONTROL
87	MODE0/SCIF0_TXD/$\overline{\text{IRL4}}$/ SDIF0D0	INPUT
86	DR0/ETH_TXD0	OUTPUT
85	DR0/ETH_TXD0	CONTROL
84	DR0/ETH_TXD0	INPUT
83	DR1/ETH_TXD1	OUTPUT
82	DR1/ETH_TXD1	CONTROL
81	DR1/ETH_TXD1	INPUT
80	DR2/ETH_TXD2	OUTPUT
79	DR2/ETH_TXD2	CONTROL
78	DR2/ETH_TXD2	INPUT

Number	Pin Name	Type	Number	Pin Name	Type
77	DR3/ETH_TXD3	OUTPUT	46	DB1/ETH_RX_DV	CONTROL
76	DR3/ETH_TXD3	CONTROL	45	DB1/ETH_RX_DV	INPUT
75	DR3/ETH_TXD3	INPUT	44	DB2/ETH_RXD0	OUTPUT
74	DR4/ETH_TX_EN	OUTPUT	43	DB2/ETH_RXD0	CONTROL
73	DR4/ETH_TX_EN	CONTROL	42	DB2/ETH_RXD0	INPUT
72	DR4/ETH_TX_EN	INPUT	41	DB3/ETH_RXD1	OUTPUT
71	DR5/ETH_TX_ER	OUTPUT	40	DB3/ETH_RXD1	CONTROL
70	DR5/ETH_TX_ER	CONTROL	39	DB3/ETH_RXD1	INPUT
69	DR5/ETH_TX_ER	INPUT	38	DB4/ETH_RXD2	OUTPUT
68	DG0/ETH_CRS	OUTPUT	37	DB4/ETH_RXD2	CONTROL
67	DG0/ETH_CRS	CONTROL	36	DB4/ETH_RXD2	INPUT
66	DG0/ETH_CRS	INPUT	35	DB5/ETH_RXD3	OUTPUT
65	DG1/ETH_TX_CLK	OUTPUT	34	DB5/ETH_RXD3	CONTROL
64	DG1/ETH_TX_CLK	CONTROL	33	DB5/ETH_RXD3	INPUT
63	DG1/ETH_TX_CLK	INPUT	32	DCLKOUT	OUTPUT
62	DG2/ETH_COL	OUTPUT	31	DCLKOUT	CONTROL
61	DG2/ETH_COL	CONTROL	30	DCLKOUT	INPUT
60	DG2/ETH_COL	INPUT	29	CDE/ETH_MAGIC	OUTPUT
59	DG3/ETH_MDC	OUTPUT	28	CDE/ETH_MAGIC	CONTROL
58	DG3/ETH_MDC	CONTROL	27	CDE/ETH_MAGIC	INPUT
57	DG3/ETH_MDC	INPUT	26	DISP/ETH_LINK	OUTPUT
56	DG4/ETH_RX_CLK	OUTPUT	25	DISP/ETH_LINK	CONTROL
55	DG4/ETH_RX_CLK	CONTROL	24	DISP/ETH_LINK	INPUT
54	DG4/ETH_RX_CLK	INPUT	23	DCLKIN/HSPI_RX	OUTPUT
53	DG5/ETH_MDIO	OUTPUT	22	DCLKIN/HSPI_RX	CONTROL
52	DG5/ETH_MDIO	CONTROL	21	DCLKIN/HSPI_RX	INPUT
51	DG5/ETH_MDIO	INPUT	20	$\overline{\text{HSYNC}}$/HSPI_TX	OUTPUT
50	DB0/ETH_RX_ER	OUTPUT	19	$\overline{\text{HSYNC}}$/HSPI_TX	CONTROL
49	DB0/ETH_RX_ER	CONTROL	18	$\overline{\text{HSYNC}}$/HSPI_TX	INPUT
48	DB0/ETH_RX_ER	INPUT	17	$\overline{\text{VSYNC}}$/HSPI_CLK	OUTPUT
47	DB1/ETH_RX_DV	OUTPUT	16	$\overline{\text{VSYNC}}$/HSPI_CLK	CONTROL

Number	Pin Name	Type
15	$\overline{\text{VSYNC}}$/HSPI_CLK	INPUT
14	ODDF/$\overline{\text{HSPI\_CS}}$	OUTPUT
13	ODDF/$\overline{\text{HSPI\_CS}}$	CONTROL
12	ODDF/$\overline{\text{HSPI\_CS}}$	INPUT
11	SCIF1_SCK	OUTPUT
10	SCIF1_SCK	CONTROL
9	SCIF1_SCK	INPUT
8	SCIF1_RXD	OUTPUT
7	SCIF1_RXD	CONTROL
6	SCIF1_RXD	INPUT
5	SCIF1_TXD	OUTPUT
4	SCIF1_TXD	CONTROL
3	SCIF1_TXD	INPUT
2	$\overline{\text{ASEBRK}}$/BRKACK	OUTPUT
1	$\overline{\text{ASEBRK}}$/BRKACK	CONTROL
0	$\overline{\text{ASEBRK}}$/BRKACK	INPUT
	To TDO	

Note: * Control means a low-active signal. When a low-active signal is driven low, the corresponding pin is driven by the output value.

32.4 Operation

32.4.1 Boundary-Scan TAP Controller (IDCODE, EXTEST, SAMPLE/PRELOAD, and BYPASS)

In the H-UDI in this LSI, the boundary-scan TAP controller is separated from the TAP controller for other H-UDI function control. When the $\overline{\text{TRST}}$ is asserted (including when the power is turned on), the boundary-scan TAP controller operates and the boundary scan function specified in JTAG can be used. By inputting the switching command, an H-UDI reset and H-UDI interrupts can be used. However, the following restrictions are put on this LSI.

- Clock-related pins (EXTAL, XTAL, USB_EXTAL, USB_XTAL and DCLKOUT) are out of the scope of the boundary scan test.
- Reset-related pin ($\overline{\text{PRESET}}$) is out of the scope of the boundary scan test.
- H-UDI-related pins (TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$, AUDSYNC, AUDCK, AUDATA3 to AUDATA0, and MPMD) are out of the scope of the boundary scan test.
- USB-IF-related pins are out of the scope of the boundary scan test without USB_OVC0, USB_OVC1, USB_PENC0 and USB_PENC1.
- DDR3-IF-related pins are out of the scope of the boundary scan test.
- PCIexpress-IF-related pins are out of the scope of the boundary scan test.
- Thermal-sensor-related pins are out of the scopde of the boundary scan test.
- During the boundary scan (IDCODE, EXTEST, SAMPLE/PRELOAD, BYPASS, and H-UDI switching command), the maximum frequency of TCK is 10 MHz.
- The access size from the H-UDI to the boundary-scan TAP controller is 8 bits.

The commands supported by the boundary-scan TAP controller are shown below.

Note: During the boundary scan, the MPMD and the $\overline{\text{PRESET}}$ pins should be fixed to high level. When the H-UDI operates in emulation support mode (MPMD = 0), the boundary scan function cannot be used. Figure 32.2 shows the sequence to switch from the boundary-scan TAP controller to the H-UDI.

Table 32.6 Commands Supported by Boundary-Scan TAP Controller

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
0	1	0	1	0	1	0	1	IDCODE
1	1	1	1	1	1	1	1	BYPASS
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	0	0	0	1	0	0	0	H-UDI switching command1
0	0	0	1	0	1	1	1	H-UDI switching command2
Other than above								Setting prohibited

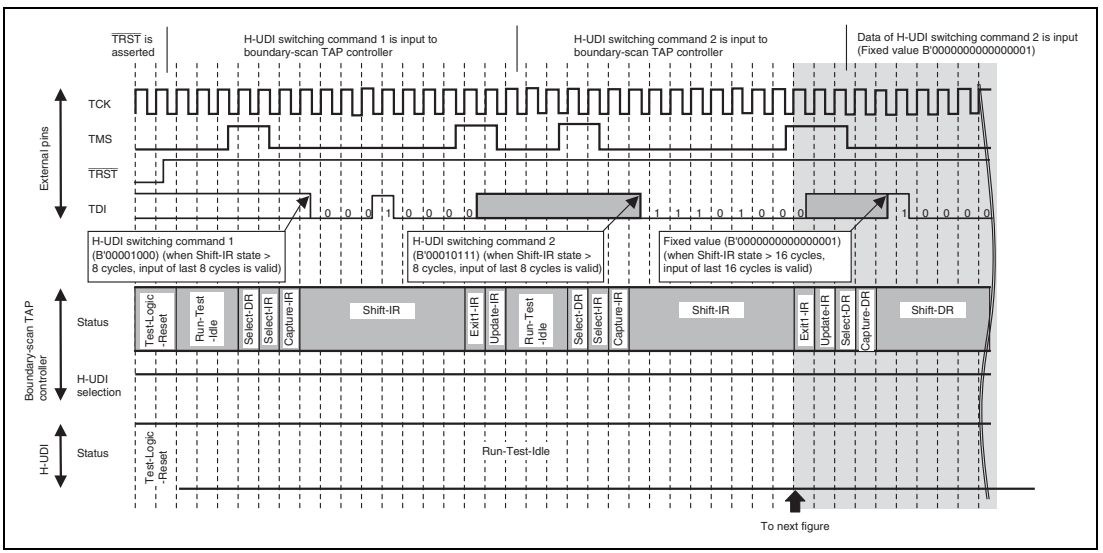


Figure 32.2 Sequence to Switch from Boundary-Scan TAP Controller to H-UDI (1)

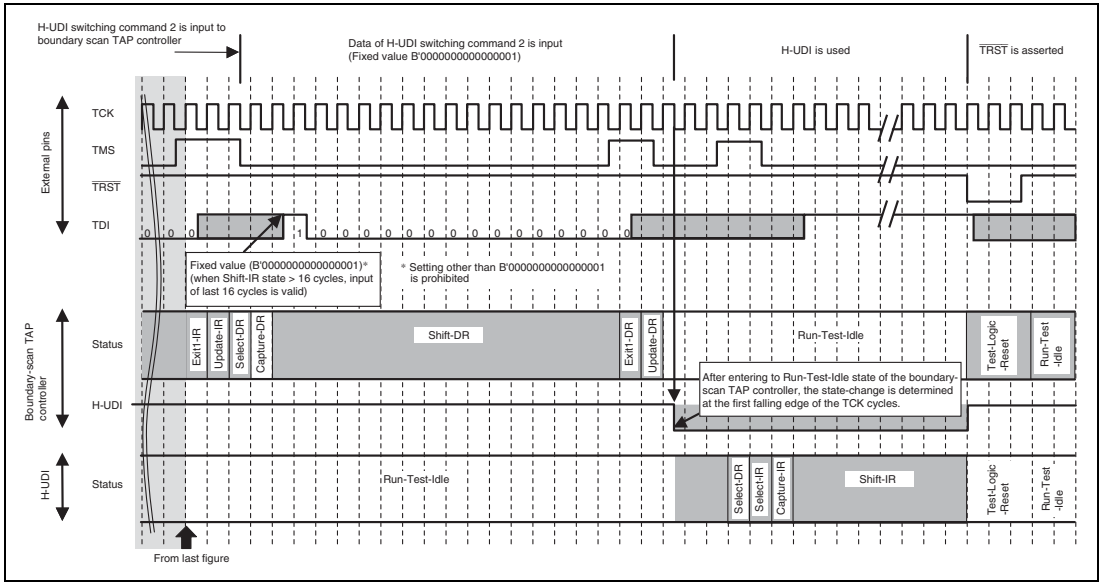


Figure 32.2 Sequence to Switch from Boundary-Scan TAP Controller to H-UDI (2)

32.4.2 TAP Control

Figure 32.3 shows the internal states of the TAP controller. The controller supports the state transitions specified in JTAG with the subset.

- The condition of transition is the TMS value at the rising edge of TCK.
- The TDI value is sampled at the rising edge of TCK and shifted at the falling edge of TCK.
- The TDO value is changed at the falling edge of TCK. The TDO is in the high impedance state other than in the Shift-DR or Shift-IR state.
- When $\overline{\text{TRST}}$ is changed to 0, the transition to the Test-Logic-Reset state is performed asynchronously with the TCK signal.

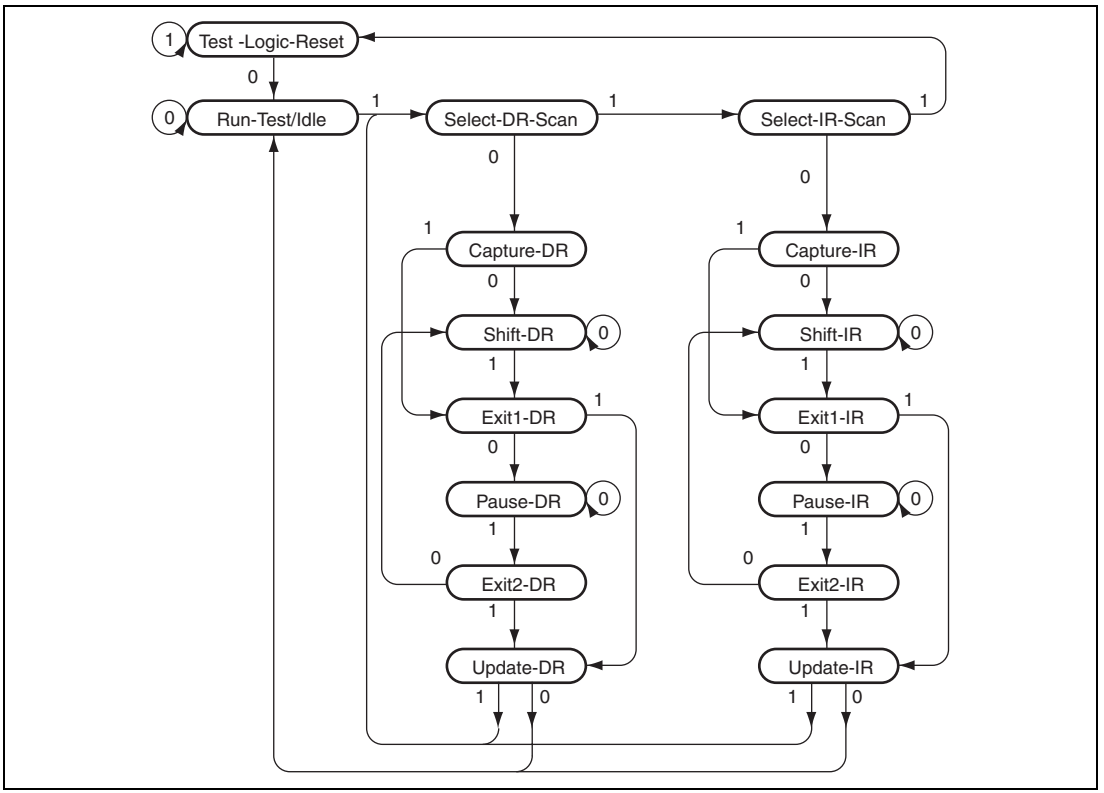


Figure 32.3 Diagram of Transitions of TAP Controller State

32.4.3 H-UDI Reset

The H-UDI is reset by a power-on reset by the SDIR command. To reset the H-UDI, send the H-UDI reset assert command from the H-UDI pin, and then send the H-UDI reset negate command (see figure 32.4). The time required between the H-UDI reset assert and H-UDI reset negate commands is the same as the time to keep the reset pin low in order to reset this LSI by a power-on reset. After the H-UDI reset assert command is set, the reset signal is asserted in the chip after four cycles at a peripheral clock (Pck). When the H-UDI reset negate command is set, the reset signal is negated in the chip after a reset hold period. (The minimum period is 17 cycles at a peripheral clock, and the maximum period is 42 cycles at a peripheral clock. For details, see section 18, Clock Pulse Generator (CPG).)

Note: The WDT/RST module is not initialized. However, the overflow counter of the WDT/RST module is initialized.

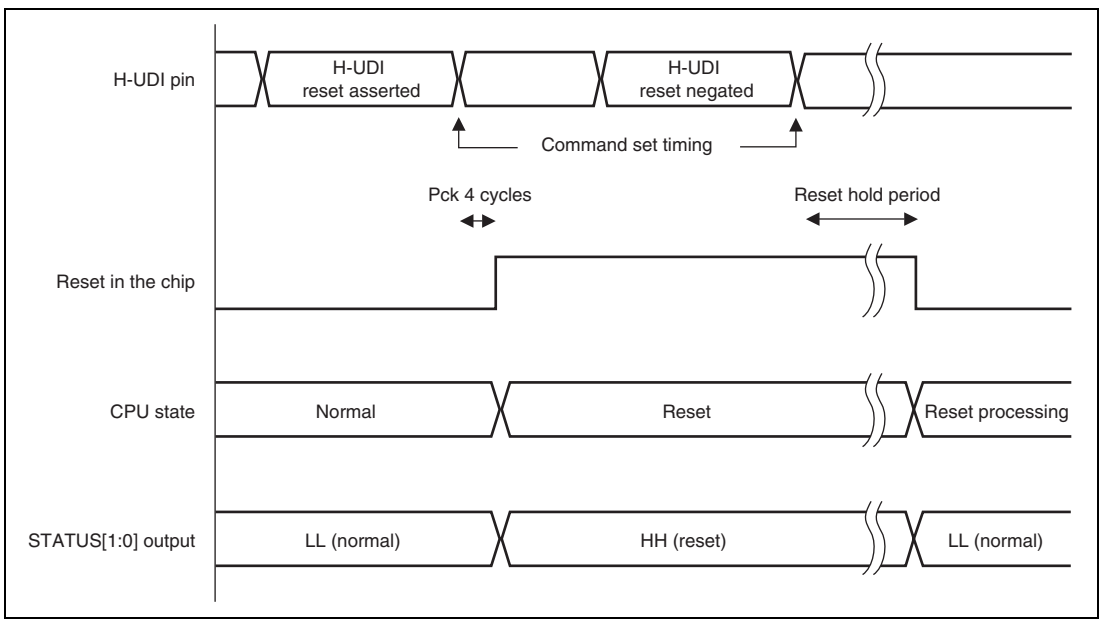


Figure 32.4 H-UDI Reset

32.4.4 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command value in SDIR through the H-UDI. The H-UDI interrupt function is general exception or interrupt operation, that is, execution is branched to the address based on VBR and is returned to the branch source by the RTE instruction. In this case, the exception code stored in INTEVT is H'600. Also, the priority of the H-UDI interrupt is controlled by bits 28 to 24 in INT2PRI4. For details, see section 10, Interrupt Controller (INTC).

An H-UDI interrupt request signal is asserted when the INTREQ bit in SDINT is set to 1 after setting the command (Update-IRQ). Since the interrupt request signal is not negated unless the INTREQ bit is cleared to 0 by software, the interrupt request cannot be missed. While the H-UDI interrupt command is set in SDIR, SDINT is connected between the TDI and TDO pins. For values read through the TDO pin and others, see section 32.3.2, Interrupt Source Register (SDINT).

32.5 Usage Notes

1. Once the SDIR command is set, it is not changed unless a command is written through the H-UDI, except the assertion of $\overline{\text{TRST}}$ or initialization by changing the TAP to the Test-Logic-Reset state.
2. Sleep mode and light sleep mode are released by an H-UDI interrupt or H-UDI reset, and these modes accept the interrupt and reset requests.
3. The H-UDI is used to connect an emulator. Therefore, the JTAG functions cannot be used when an emulator is used.

Section 33 Thermal Sensor

33.1 Features

This LSI provides a thermal sensor module that measures the temperature (Tj) inside the LSI.

The thermal sensor module has the following features.

- Generates interrupts through the INTC when internal temperature Tj of the LSI reaches 85°C, 105°C, 115°C, and 125°C.
- External pin VTHSENSE outputs a voltage correlated with internal temperature Tj.
  External pin VTHREF outputs a reference voltage used to convert the voltage output from the VTHSENSE pin into a digital signal.
  Internal temperature Tj can be calculated from VTHREF and VTHSENSE.

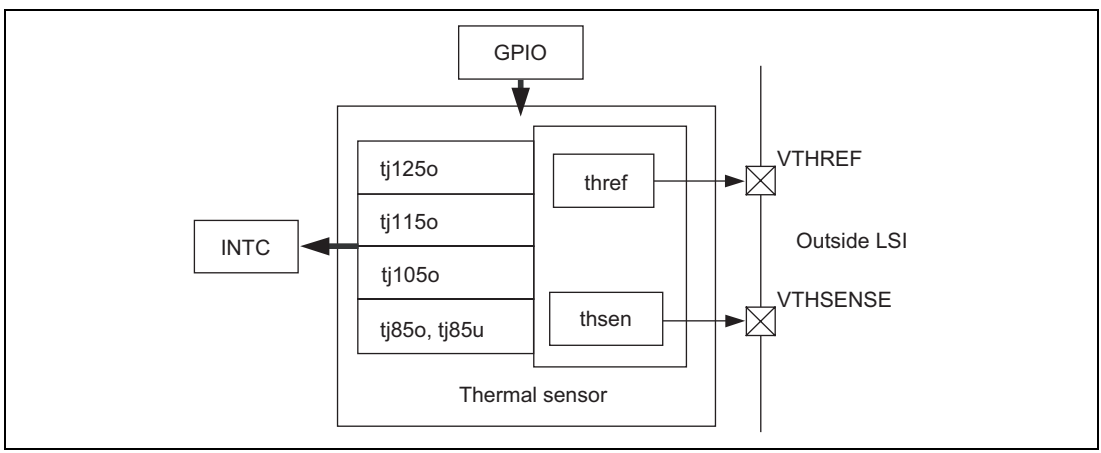


Figure 33.1 Block Diagram of Thermal Sensor Module

33.2 Output Pins

Table 33.1 shows the external output pins of the thermal sensor module.

Table 33.1 Output Pins of Thermal Sensor Module

Pin Name	Symbol	I/O	Function
Thermal sensor output voltage	VTHSENSE	Analog I/O	Outputs a voltage correlated with temperature Tj in the LSI.
Thermal sensor output reference voltage	VTHREF	Analog I/O	Outputs a reference voltage used to convert the voltage output from the VTHSENSE pin into a digital signal.

33.3 Registers

The thermal sensor module uses the following registers.

(1) GPIO Register

Table 33.2 Thermal Sensor GPIO Register Configuration

Register Name	Symbol	R/W	P4 Address*[1]	Area 7 Address*[1]	Access Size*[2]	Sync Clock
Thermal sensor output control register	THTAPR	R/W	H'FFCC 00A0	H'1FCC 00A0	16	Pck

Notes: 1. The P4 address is used when area P4 in the virtual address space is used, and the area 7 address is used when accessing the register through area 7 in the physical address space using the TLB.
2. This LSI has 16-bit access registers and 8-bit access registers. Access each register in the specified access size.

(2) INTC Register

For the register in the INTC, refer to section 10.3.5, Thermal Sensor Interrupt Register.

The register bits have the following functions.

Table 33.3 lists the signals sent from the thermal sensor to INT2THSC when bits tap11 to tap 8 in the thermal sensor output control register (THTAPR) are all cleared to 0 (P).

Table 33.3 Thermal Sensor INTC Register Configuration (1)

Bit	Temperature Detection Register Abbreviation	Thermal Sensor Interrupt Register	THS3E	THS2E	THS1E	THS0E
Bit for Tj >= 125	Tj125o	INT2THSC	1	1	1	1
Bit for Tj >= 115	Tj115o		0	1	1	1
Bit for Tj >= 105	Tj105o		0	0	1	1
Bit for Tj >= 85	Tj85o		0	0	0	1
Bit for Tj < 85	Tj85u (initial value)		0	0	0	0

Table 33.4 lists the signals sent from the thermal sensor to INT2THSC when bits tap11 to tap 8 in the thermal sensor output control register (THTAPR) are all set to 1 (N).

Table 33.4 Thermal Sensor Register Configuration (2)

Bit	Temperature Detection Register Abbreviation	Thermal Sensor Interrupt Register	THS3E	THS2E	THS1E	THS0E
Bit for Tj >= 125	Tj125o	INT2THSC	0	0	0	0
Bit for Tj >= 115	Tj115o		1	0	0	0
Bit for Tj >= 105	Tj105o		1	1	0	0
Bit for Tj >= 85	Tj85o		1	1	1	0
Bit for Tj < 85	Tj85u (initial value)		1	1	1	1

33.3.1 Thermal sensor output control register (THTAPR)

The following shows the GPIO register bits of the thermal sensor and their functions.

As the thermal sensor module is active in the initial state, the register should be set up regardless of whether the sensor is used.

Control register THTAPR in the thermal sensor module is a 16-bit readable/writable register.

This register selects the operating state (normal operation or idle state) of the thermal sensor module and controls interrupts and output voltages VTHSENSE and VTHREF. To use the thermal sensor, make appropriate settings in this register.

Note that bits 0 to 7 should be set to one of the values shown in the following table; if a value not shown is used, the electrical characteristics of the thermal sensor are not guaranteed.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	tap12	tap11	tap10	tap9	tap8	tap7	tap6	tap5	tap4	tap3	tap2	tap1	tap0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R/W	Reserved. The write value should always be 0. These bits are always read as 0.
12	tap12	0	R/W	Selects the operating state of the thermal sensor module. 0: Normal operation 1: Idle state. Select this setting when the thermal sensor is not used. When the idle state is selected, leave tap7 to tap0 at their initial values and do not write to them.
11 to 8	tap11 to tap8	All 0	R/W	These bits select the polarity (P or N) of internal signals THS0E to THS3E in the thermal sensor module. 0: P 1: N

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	tap7 to tap0	All 0	R/W	These settings determine the detection temperature of the temperature detection registers of the thermal sensor module. The following settings guarantee the detection temperature of the temperature detection registers. Make the settings listed below when using temperature detection registers Tj85o, Tj105o, Tj115o, and Tj125o. Do not use the initial settings or write values other than those listed below. tap7: 1 tap6: 0 tap5: 0 tap4: 1 tap3: 0 tap2: 1 tap1: 0 tap0: 0

33.4 Recommended External Connection Circuit

Figure 33.2 shows a recommended external circuit to be connected to the thermal sensor module.

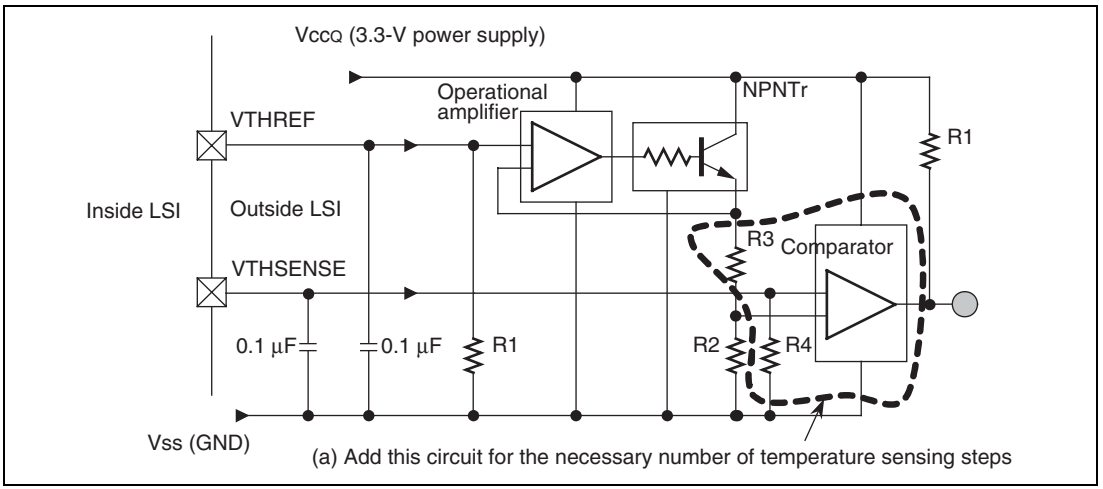


Figure 33.2 Recommended External Connection Circuit for Thermal Sensor Module

The recommended external connection circuit outputs a high level at a desired temperature by using VTHREF, which is affected by the same level of external noise as that affects VTHSENSE. This circuit thus cancels external noise effects and improves the accuracy of Tj.

The steps for digitizing measured temperature Tj (number of temperatures to be compared) can be increased by increasing the number of circuits indicated as (a) in the figure.

Circuit element values:

R1 = 549 Ω, R2 = 414 Ω, R13 = R4 = 560 kΩ

E96 type resistors with an error of ±0.1% are recommended for R1 and R2. The resistance values of these resistors affects the accuracy with which Tj is digitized. It is recommended that the resistors be made of a material that is not susceptible to the effects of temperature. It is also acceptable to use two or more resistors, connected in parallel, for R1 and R2. R3 and R4 are used as pull-down resistors, so accuracy is not an issue.

33.5 Electrical Characteristics

The following shows the electrical characteristics.

Table 33.5 External Output Pins of Thermal Sensor

Pin Name	Output Voltage Range [V]	Maximum Input/Output Current [μA]	Tj Accuracy
VTHSENSE	0.4 V (min) to 0.8 V (max)	± 20	When the recommended external connection circuit is used, the accuracy is ±125°C max. The temperature range to be detected is –40°C to 125°C. When the temperature exceeds 125°C, the accuracy cannot be guaranteed.
VTHREF	1.1 V (min) to 1.3 V (max)	± 20	

The following equation is used to obtain the temperature (Tj) from VTHSENSE and VTHREF.

Tj = (temperature coefficient 2 + VTHSENSE – VTHREF) / temperature coefficient 1 + 25 [°C]

VTHSENSE [V], temperature coefficient 1: (1.7042) [mV/°C], temperature coefficient 2: (0.7065) [V]

For example, inserting 125°C standing state VTHSENSE (0.7100) [V] and VTHREF (1.2460) [V] into the equation produces the following result.

Tj = ((0.7065) + (0.7100) – (1.2460)) × 1000 / 1.7042 + 25 = 125.0 [°C]

Tj, which is calculated using the above equation from the VTHSENSE and VTHREF output voltages, has a temperature range of –40°C to 125°C and an accuracy of ±5°C. Note that in order to obtain this level of accuracy, one 0.1 μF capacitor each should be connected between the VTHSENSE and VTHREF pins and their respective grounds, as shown in the recommended external connection circuit diagram. Note that the two capacitors should each be located as close as possible to their ground connections.

Regarding Tj accuracy when using the temperature detection registers of the thermal sensor and internal control by the SH7786, keep in mind that the operation of the internal circuits of the SH7786 is guaranteed within a temperature range of –40 to +85°C. Outside of this temperature range neither operation nor accuracy can be guaranteed. Also refer to item (3) in 33.6, Notes.

Table 33.6 Accuracy of Tj for Each Temperature Detection Register in Thermal Sensor

Abbreviation	Tj [°C]	Tj Accuracy
Tj125o	125°C >=	Not guaranteed
Tj115o	115°C >=	Not guaranteed
Tj105o	105°C >=	Not guaranteed
Tj85o	85°C >=	85 ± (5) °C >=
Tj85u (initial value)	85°C <	—

33.6 Usage Notes

(1) If the thermal sensor module is not used, either leave the VTHSENSE and VTHREF pins open or connect them to a ground.

(2) Tj, which is calculated from the VTHSENSE and VTHREF output voltages, has a temperature range of –40°C to 125°C and an accuracy of ±5°C. This accuracy assumes a case in which the VTHSENSE and VTHREF voltages are measured by identical directly connected measuring devices having a direct-connection accuracy of 0.1 mV or better. Regarding the accuracy of the measuring devices employed and cases where an external connection circuit is used, make sure the design makes allowance for error arising from the components used, etc.

(3) Regarding the Tj accuracy when using the temperature detection registers of the thermal sensor and internal control by the SH7786, neither operation nor accuracy can be guaranteed when Tj exceeds 85°C.

Bits are provided for Tj of 105°C, 115°C, and 125°C, but these are intended to be used as part of a failsafe mechanism to stop operation of the SH7786 and prevent thermal runaway from occurring in cases where the power consumption increases suddenly and the MCU is about to enter a thermal runaway state.

(4) The recommended external connection circuit outputs digital signals used to perform SH7786 temperature adjustment externally. It can be used, for example, as a failsafe circuit that cuts off power to the SH7786 when it is about to enter a thermal runaway state and has reached a high temperature at which control is no longer possible. Alternately, is can be connected to a device such as an ADC.

Section 34 Electrical Characteristics

34.1 Absolute Maximum Ratings

Table 34.1 Absolute Maximum Ratings*[1, 2]

Item		Symbol	Value	Unit
I/O, PCI Express, USB power supply voltage		V_{CCQ}	–0.3 to 4.6	V
		$V_{CCQ-PCIE}$		
		AV_{33}		
PCI Express I/O power supply voltage		$V_{CCQ-PCI15}$	–0.3 to 1.8	V
Internal power supply voltage		V_{DD}	–0.3 to 1.8	V
		$V_{DD-PCIE}$		
		V_{DDAI}		
		AV_{12}		
DDR power supply		$V_{CCQ-DDR15}$	–0.4 to 1.975	V
Input voltage	3.3V I/O (excluding USB)	V_{in}	–0.3 to V_{CCQ} +0.3*[3]	V
	USB (excluding VBUS)	AV_{in}	–0.3 to AV_{33} +0.3*[3]	
	USB VBUS		–0.3 to 5.5	
Operating temperature		T_{opr}	–20 to 85	°C
Storage temperature		T_{stg}	–55 to 125	°C

Notes: 1. The LSI may be permanently damaged if the maximum ratings are exceeded.

2. Supply to specified voltage to all power supplies and connect all GND (V_{SS}, V_{SSQ}, AG33, V_{SSAI}, and AG12) pins to a ground (0 V). Failure to do this can result in permanent damage to the MCU.

3. The voltage must not exceed an upper limit of 4.6 V.

34.2 DC Characteristics

Table 34.2 DC Characteristics

Conditions: T_a = –20 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		V_{CCQ}	3.0	3.3	3.6	V	Normal mode, sleep mode, module standby mode
		$V_{CCQ-PCIE}$					
		AV_{33}					
		$V_{CCQ-DDR15}$	1.425	1.5	1.575		
		$V_{CCQ-PCI15}$					
		V_{DD}	1.15	1.25	1.35		
		$V_{DD-PCIE}$					
		V_{DDAI}					
		AV_{12}					
		$MV_{REF0/1}$	0.49 × $V_{CCQ-DDR15}$	0.50 × $V_{CCQ-DDR15}$	0.51 × $V_{CCQ-DDR15}$		
Current dissipation	Normal operation	I_{DD}	—	1200	3800	mA	Ick = 533 MHz Bck = 188.9 MHz Pck = 66 MHz DDRck = 533 MHz
	Sleep mode		—	—	800		
	Normal operation	I_{CCQ}	—	100	120		
	Sleep mode		—	—	30		
	Normal operation	I_{DDAI}	—	—	5	mA	
	USB Normal operation	I_{AV12}	—	—	10	mA	USBCLK = 48 MHz
		I_{AV33}	—	—	10	mA	
	PCI Normal operation (1.2V)	$I_{CCQ-PCI12}$	—	—	600	mA	
	PCI Normal operation	$I_{CCQ-PCI5}$	—	—	200	mA	
		$I_{CCQ-PCIE}$	—	—	100	mA	
	DDR Normal operation	$I_{CCQ-DDR}$	—	—	450	mA	
	DDR backup operation		—	—	800	μA	No ODT No Vtt termination

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leak current	DDR pins (SDBUP/MBKPRST)	|Iin|	—	—	5	μA	V_{IN} = 0.5 to V_{CCQ} –0.5 V
	Other input pins		—	—	1		
Three-state leak current	All I/O pins (off condition)	|Isti|	—	—	1	μA	V_{IN} = 0.5 to V_{CCQ} –0.5 V
Pull-up resistance	Pins other than DDR, PCI, and USB	R_{pull}	20	—	180	kΩ	
Pin capacitance	USB pins	C_L	—	—	15	pF	
	DDR pins		—	—	10		
	Other pins		—	—	10		

Note: Regardless of whether or not the PLL is used, connect V_{DDAI} to the power supply, and V_{SSAI} to GND. The LSI may be damage when not filling this.

Table 34.3 3.3 V I/O Pin Characteristics (Excluding USB)

Conditions: T_a = –20 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input voltage	$\overline{\text{PRESET}}$, NMI, $\overline{\text{TRST}}$	V_{IH}	V_{CCQ} × 0.9	—	V_{CCQ} +0.3	V	V_{CCQ} = 3.0 to 3.6 V
	EXTAL		V_{CCQ} × 0.8	—	V_{CCQ} +0.3		
	Other input pins		2.0	—	V_{CCQ} +0.3		
	$\overline{\text{PRESET}}$, NMI, $\overline{\text{TRST}}$	V_{IL}	—	—	V_{CCQ} × 0.1	V	V_{CCQ} = 3.0 to 3.6 V
	EXTAL		—	—	V_{CCQ} × 0.2		
	Other input pins		—	—	V_{CCQ} × 0.2		
Output voltage	All output pins	V_{OH}	2.4	—	—	V	V_{CCQ} = 3.0 V
		V_{OL}	—	—	0.55		

Table 34.4 USB Interface Characteristics (High speed)

Conditions: T_a = –20 to +85°C, V_{CCQ} = 3.3 V, AV_{33} = 3.3 V, AV_{12} = 1.25 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristic	Differential input sensitivity	V_{HSCM}	150	—	—	mV	
	Squelch detection voltage	V_{HSSQ}	100	—	150		
	Disconnect voltage	V_{HSDSC}	525	—	625		
Output characteristic	Output current	I_{OUT}	—	17.78	—	mA	
	High-level output voltage	V_{HSOH}	360	—	440	mV	
	Low-level output voltage	V_{HSOL}	–10	—	10		
	Chirp J output voltage (differential voltage)	V_{CHIRPJ}	700	—	1100		
	Chirp K output voltage (differential voltage)	V_{CHIRPK}	500	—	900		

Table 34.5 USB Interface Characteristics (Full/Low speed)

Conditions: T_a = –20 to +85°C, V_{CCQ} = 3.3 V, AV_{33} = 3.3 V, AV_{12} = 1.25 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristic	High-level input voltage	V_{IH}	2.0	—	V_{CCQ} +0.3	V	
	Low-level input voltage	V_{IL}	–0.3	—	0.8		
	Differential input sensitivity	V_{DI}	0.2	—	—		
	Differential input common mode range	V_{CMI}	0.8	—	2.5		
Output characteristic	High-level output voltage	V_{OH}	2.8	—	V_{CCQ}	V	
	Low-level output voltage	V_{OL}	—	—	0.3		

Table 34.6 DDR3 Interface Characteristics

Conditions: T_a = –20 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristic	High-level input voltage	$V_{IH(DC)}$	MV_{REF} +0.1	—	$V_{CCQ-DDR15}$	V	
		$V_{IH(AC)}$	MV_{REF} +0.175	—	—		
	DC differential input voltage	$V_{IHDIFF(DC)}$	$V_{IX(DC)}$ +0.2	—	—		
	AC differential input voltage	$V_{IHDIFF(AC)}$	0.35	—	—		
	Low-level input voltage	$V_{IL(DC)}$	—	—	MV_{REF} –0.1		
		$V_{IL(AC)}$	—	—	MV_{REF} – 0.175		
	DC differential input voltage	$V_{ILDIFF(DC)}$	—	—	$V_{IX(DC)}$ –0.2		
	AC differential input voltage	$V_{ILDIFF(AC)}$	—	—	1.0		
Output characteristic	High-level output voltage	$V_{OH(DC)}$	0.8 × $V_{CCQ-DDR15}$	—	—	V	
		$V_{OH(AC)}$	V_{TT} +0.1 × $V_{CCQ-DDR15}$	—	—		
	Low-level output voltage	$V_{OL(DC)}$	—	—	0.2 × $V_{CCQ-DDR15}$		
		$V_{OL(AC)}$	—	—	V_{TT} –0.1 × $V_{CCQ-DDR15}$		
AC differential input crosspoint voltage		$V_{IX(AC)}$	$V_{CCQ-DDR15}$ /2 –0.15	—	$V_{CCQ-DDR15}$ /2 +0.15	V	
AC differential output crosspoint voltage		$V_{IX(AC)}$	$V_{CCQ-DDR15}$ /2 –0.125	—	$V_{CCQ-DDR15}$ /2 +0.125	V	

Table 34.7 DDR3 Interface ODT Characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
ODT resistance value (60 Ω)	R_{TT}	46.8	60	102.6	Ω	
		54	60	96		V_{DD} = 1.25 V $V_{CCQ-DDR15}$ = 1.5 V
ODT open value (60Ω)	ΔVM	–5		+5	%	

Table 34.8 PCI Express Interface Characteristics

Conditions: T_a = –20 to +85°C, $V_{CCQ-PCIE}$ = 3.3 V, $V_{CCQ-PCI15}$ = 1.5 V, V_{DD_PCIE} = 1.25 V

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Differential input voltage (Peak to Peak)	VRX-DIFFp-p	0.175	—	1.200	V	
Differential output voltage (Peak to Peak)	VTX-DIFFp-p	0.800	—	1.2	V	
DC common mode voltage difference between D+ and D–	VTX-CM-DC-LNE-DELTA	0.0	—	25	mV	
DC differential TX output impedance	ZTX-DIFF-DC	80	—	120	Ω	
DC differential input impedance	ZRX-DIFF-DC	80	—	120	Ω	
DC input impedance	ZRX-DC	80	—	120	Ω	

Table 34.9 Permissible Output Currents

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin; excluding DDR, USB and PCI pins)	I_{OL}	—	—	2	mA
Permissible output high current (per pin; excluding DDR, USB and PCI pins)	$–I_{OH}$	—	—	2	

Note: To protect chip reliability, do not exceed the output current values in table 34.9.

34.3 AC Characteristics

In principle, this LSI's input should be synchronous. Unless specified otherwise, ensure that the setup time and hold times for each input signal are observed.

Table 34.10 Clock Timing

Item		Symbol	Min.	Typ.	Max.	Unit
Operating frequency	CPU, FPU, cache, TLB	f	1	—	534	MHz
	DDR3-SDRAM bus		400	—	534	
	PCI Express bus		—	2500	—	
	USB bus		—	480	—	
	External bus		—	—	89	
	Peripheral modules		1	—	67	

34.3.1 Clock and Control Signal Timing

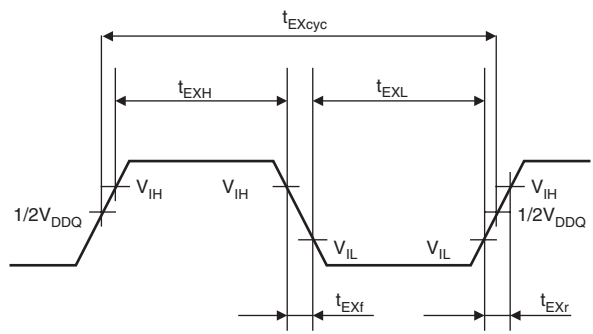
Table 34.11 Clock and Control Signal Timing

Conditions: V_{CCQ} = 3.0 to 3.6 V, V_{DD} = 1.25 V, Ta = –20 to +85°C, C_L = 30 pF

Item		Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	PLL1: × 64, PLL2 in operation*[4]	f_{EX}	12	17	MHz	34.1
	PLL1: × 32, PLL2 in operation*[5]		24	34		
EXTAL clock input cycle time	PLL1: × 64, PLL2 in operation*[4]	t_{EXcyc}	59	83	ns	34.1
	PLL1: × 32, PLL2 in operation*[5]		29	43		
EXTAL clock input low pulse width		t_{EXL}	3.5	—	ns	34.1
EXTAL clock input high pulse width		t_{EXH}	3.5	—	ns	34.1
EXTAL clock input rise time		t_{EXr}	—	4	ns	34.1
EXTAL clock input fall time		t_{EXf}	—	4	ns	34.1
CLKOUT clock output (with use of PLL1/PLL2)		f_{OP}	25	88.9	MHz	
CLKOUT clock output cycle time		t_{CKOcyc}	11.25	40	ns	34.2
CLKOUT clock output low pulse width		t_{CKOL1}	1	—	ns	34.2
CLKOUT clock output high pulse width		t_{CKOH1}	1	—	ns	34.2
CLKOUT clock output rise time		t_{CKOr}	—	4	ns	34.2
CLKOUT clock output fall time		t_{CKOf}	—	4	ns	34.2
CLKOUT clock output low pulse width		t_{CKOL2}	3	—	ns	34.3
CLKOUT clock output high pulse width		t_{CKOH2}	3	—	ns	34.3
Power-on oscillation settling time		t_{OSC1}	10	—	ms	34.4
Power-on oscillation settling time/mode (MODE14, MODE9, MODE3 to MODE0) settling time		t_{OSCMD}	10	—	ms	34.4
MODE (MODE13 to MODE10, MODE8 to MODE4) reset setup time		t_{MDRS}	5	—	t_{cyc}	34.6

Item		Symbol	Min.	Max.	Unit	Figure
MODE reset hold time	MODE13 to MODE10, MODE8 to MODE4	t_{MDRH}	20	—	ns	34.6
	MODE14, MODE9, MODE3 to MODE0					34.4
$\overline{\text{PRESET}}$ assert time		t_{RESW}	20	—	t_{cyc}	34.4
PLL synchronization settling time		t_{PLL}	400	—	μs	34.5
$\overline{\text{TRST}}$ reset hold time		t_{TRSTRH}	0	—	ns	34.4

Notes: 1. With a crystal resonator connected on EXTAL and XTAL, the maximum frequency is 34 MHz. When using a third-order overtone crystal resonator, a tank circuit must be connected externally.
2. The load capacitance connected on the CLKOUT pin should be not greater than 50 pF.
3. t_{cyc} is the period of one CLKOUT clock cycle.
4. This applies to clock operating modes 0, 1, and 2 (see table 18.2).
5. This applies to clock operating modes 3, 4, and 5 (see table 18.2).



Note: When the clock is input from the EXTAL pin.

Figure 34.1 EXTAL Clock Input Timing

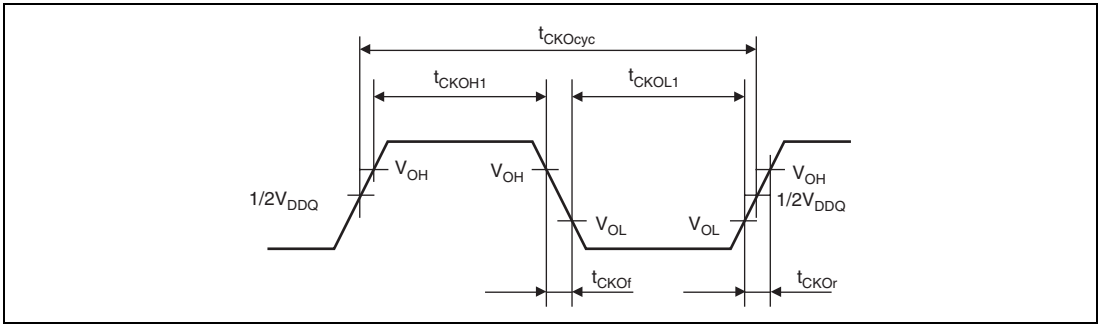


Figure 34.2 CLKOUT Clock Output Timing (1)

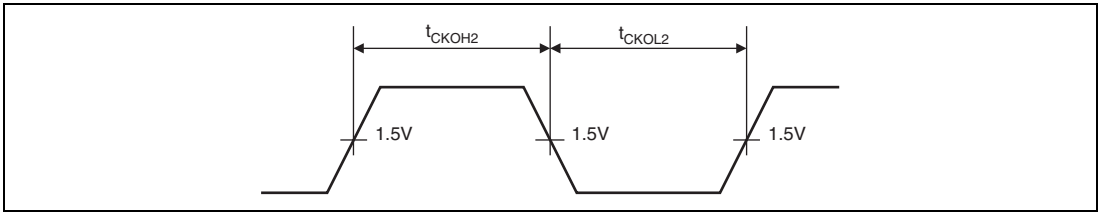
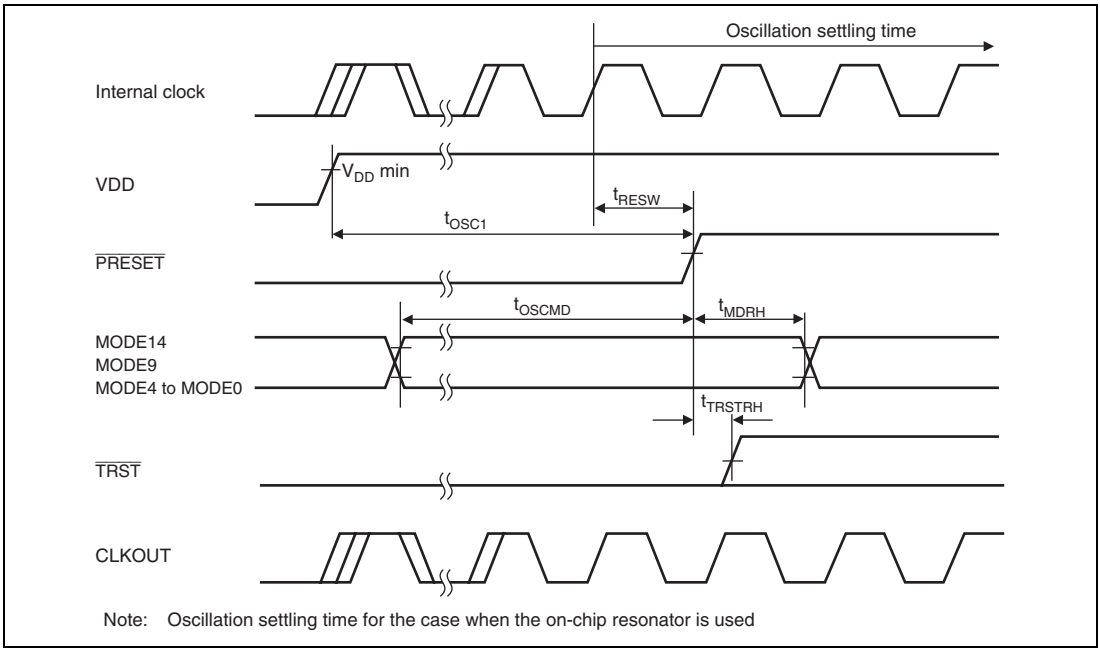


Figure 34.3 CLKOUT Clock Output Timing (2)



Note: Oscillation settling time for the case when the on-chip resonator is used

Figure 34.4 Power-On Oscillation Settling Time

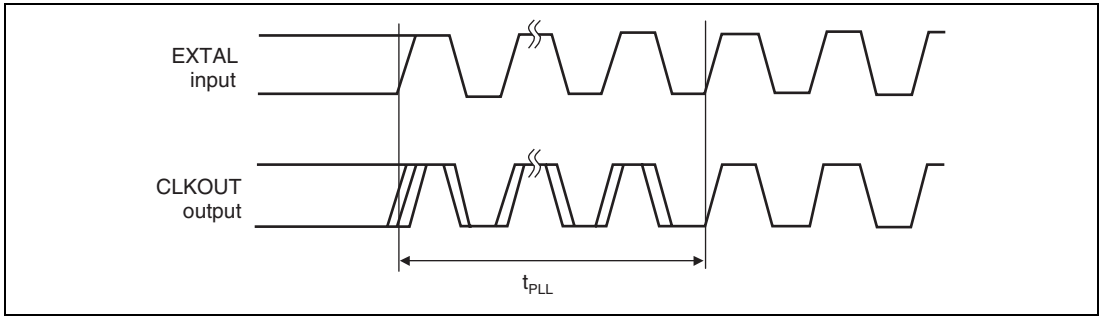
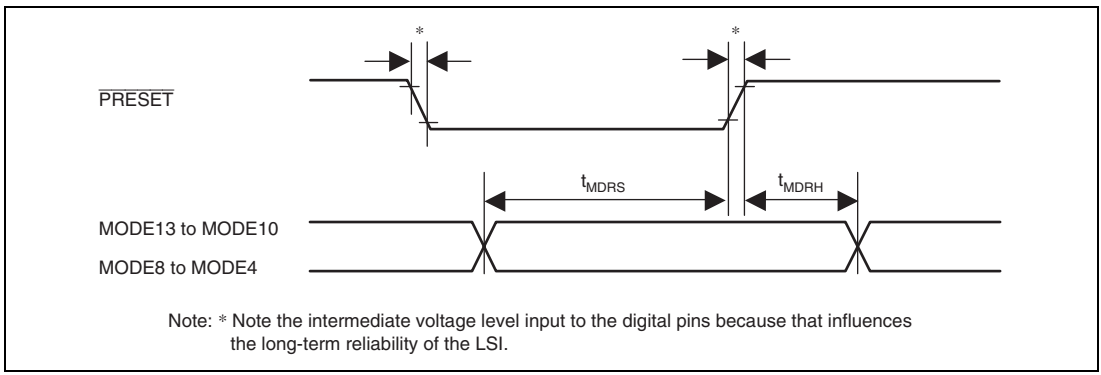


Figure 34.5 PLL Synchronization Settling Time



Note: * Note the intermediate voltage level input to the digital pins because that influences the long-term reliability of the LSI.

Figure 34.6 MODE Pin Setup/Hold Timing

34.3.2 Control Signal Timing

Table 34.12 Control Signal Timing

Conditions: V_{CCQ} = 3.0 to 3.6 V, V_{DD} = 1.25 V, Ta = –20 to +85°C, C_L = 30 pF

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	2.5	—	ns	34.7
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	1.5	—	ns	
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	8	ns	
Bus tri-state delay time	t_{BOFF1}	—	14	ns	
Bus buffer on time	t_{BON1}	—	14	ns	
STATUS0, STATUS1 delay time	t_{STD}	—	8	ns	34.8

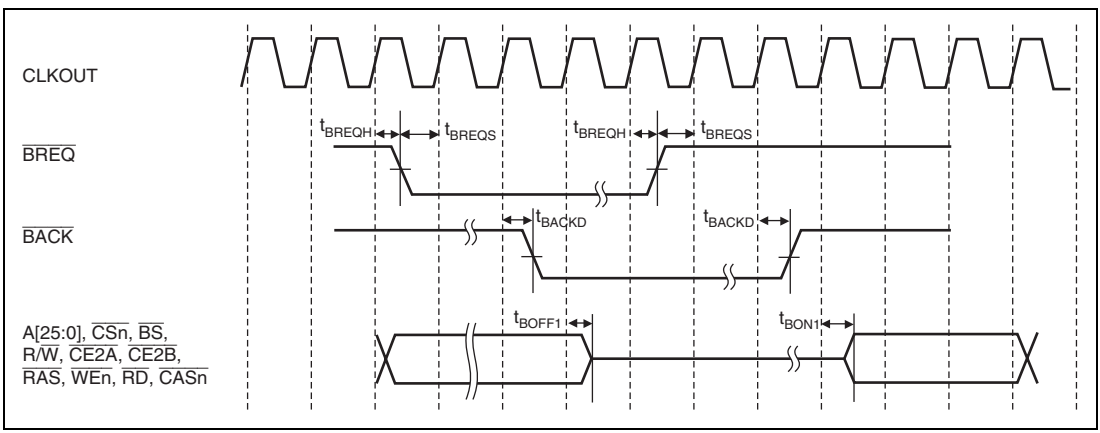


Figure 34.7 Control Signal Timing

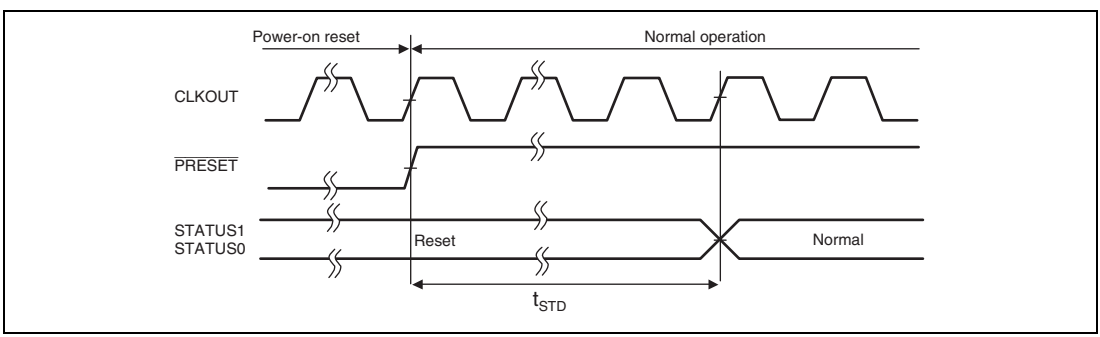


Figure 34.8 STATUS Pin Output Timing at Power-On Reset

34.3.3 Bus Timing

Table 34.13 Bus Timing

Conditions: V_{CCQ} = 3.0 to 3.6 V, V_{DD} = 1.25 V, T_a = –20 to +85°C, C_L = 30 pF

Item	Symbol	Min.	Max.	Unit	Notes
Address delay time	t_{AD}	1.5	7	ns	
$\overline{BS}$ delay time	t_{BSD}	1.5	7	ns	
$\overline{CS}$ delay time	t_{CSD}	1.5	7	ns	
R/$\overline{W}$ delay time	t_{RWD}	1.5	7	ns	
$\overline{RD}$ delay time	t_{RSD}	1.5	7	ns	
Read data setup time	t_{RDS}	2.5	—	ns	
Read data hold time	t_{RDH}	1.5	—	ns	
$\overline{WE}$ delay time (falling edge)	t_{WEDF}	1.5	7	ns	Relative to CLKOUT falling edge
$\overline{WE}$ delay time	t_{WED1}	1.5	7	ns	
Write data delay time	t_{WDD}	1.5	7	ns	
$\overline{RDY}$ setup time	t_{RDYS}	2.5	—	ns	
$\overline{RDY}$ hold time	t_{RDYH}	1.5	—	ns	
$\overline{FRAME}$ delay time	t_{FMD}	1.5	7	ns	MPX
$\overline{IOIS16}$ setup time	t_{IO16S}	2.5	—	ns	PCMCIA
$\overline{IOIS16}$ hold time	t_{IO16H}	1.5	—	ns	PCMCIA
$\overline{IOWR}$ delay time (falling edge)	t_{ICWSDF}	1.5	7	ns	PCMCIA
$\overline{IORD}$ delay time	t_{ICRSD}	1.5	7	ns	PCMCIA
DACK delay time	t_{DACD}	1.5	7	ns	
DACK delay time (falling edge)	t_{DACDF}	1.5	7	ns	

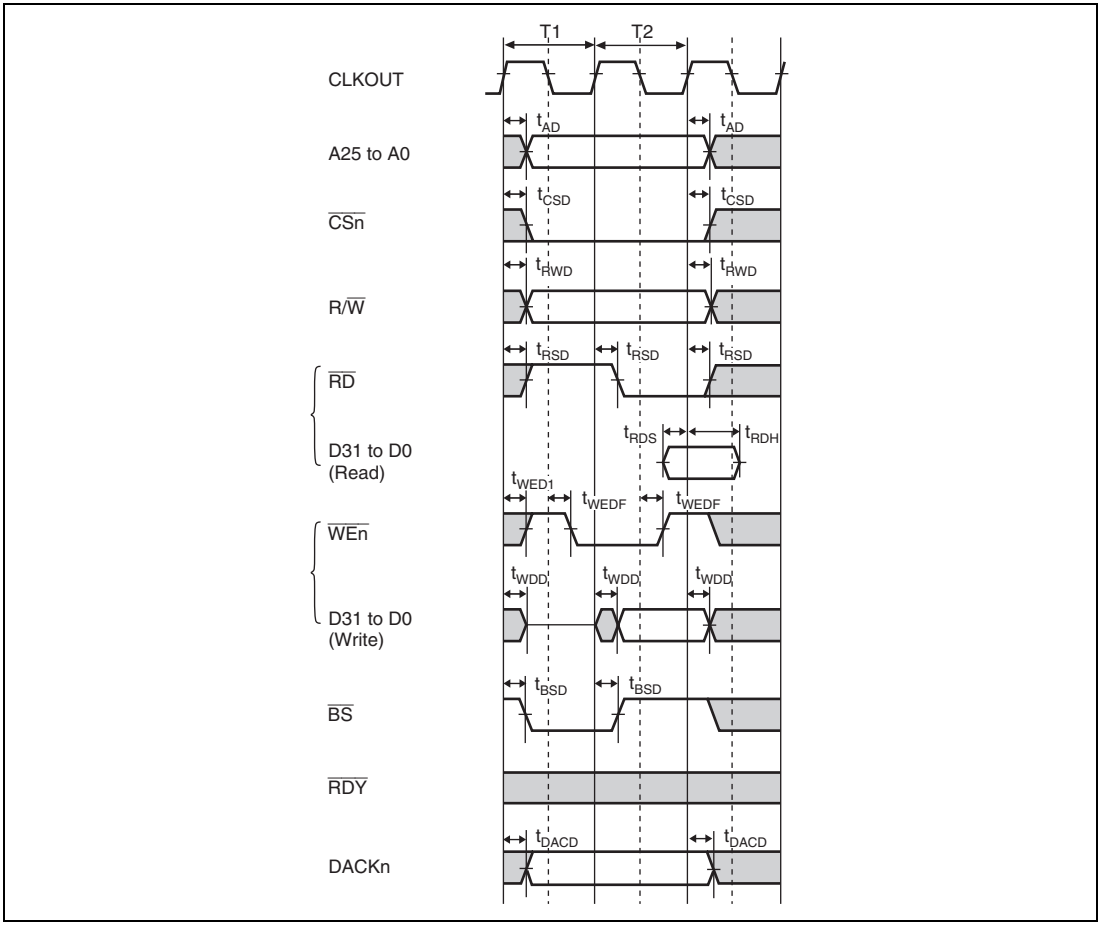


Figure 34.9 SRAM Bus Cycle: Basic Bus Cycle (No Wait)

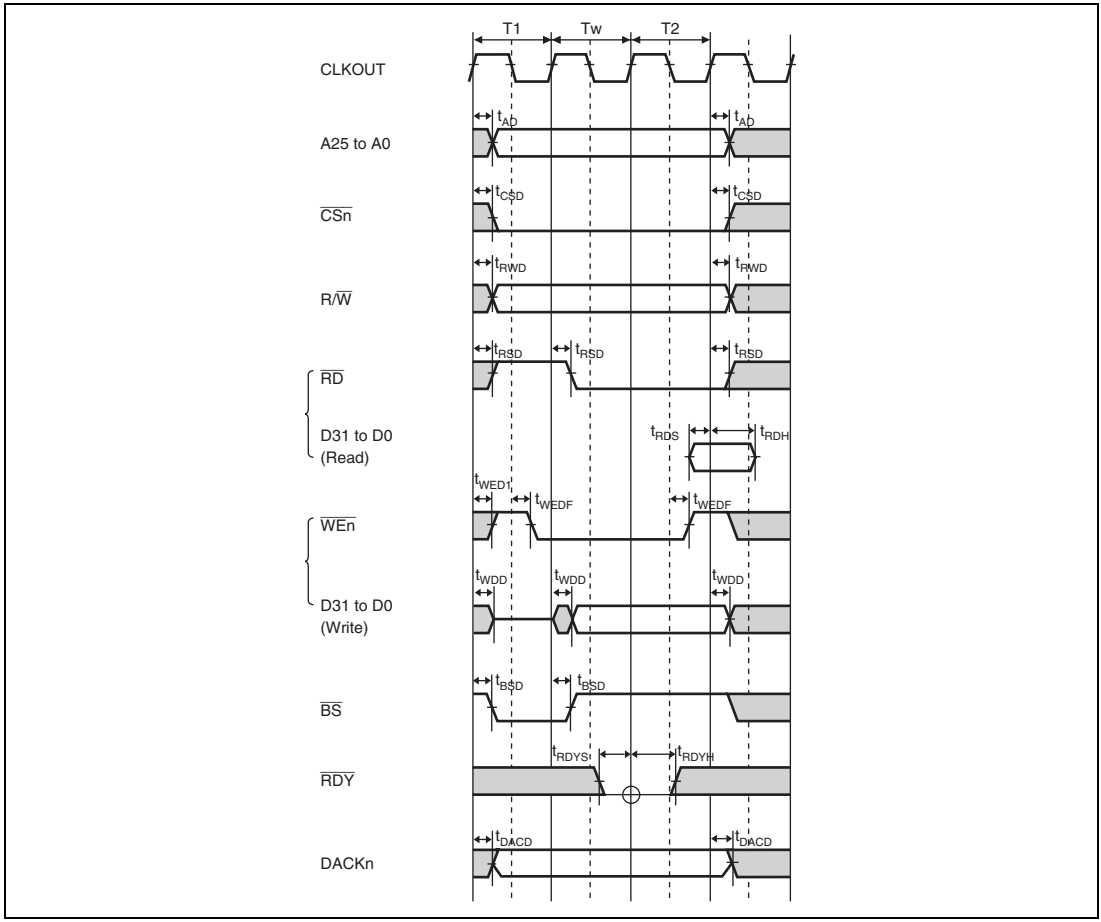


Figure 34.10 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait Cycle)

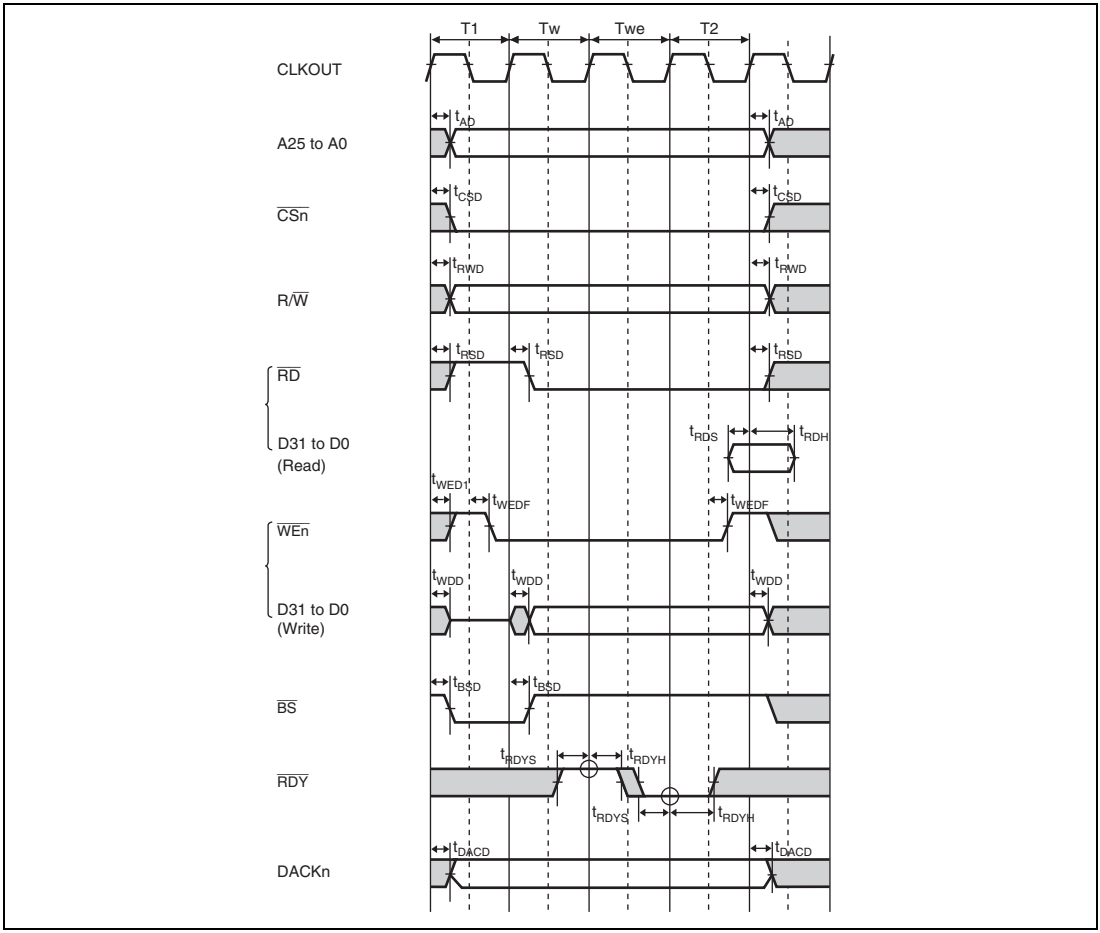


Figure 34.11 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait Cycle + One External Wait Cycle)

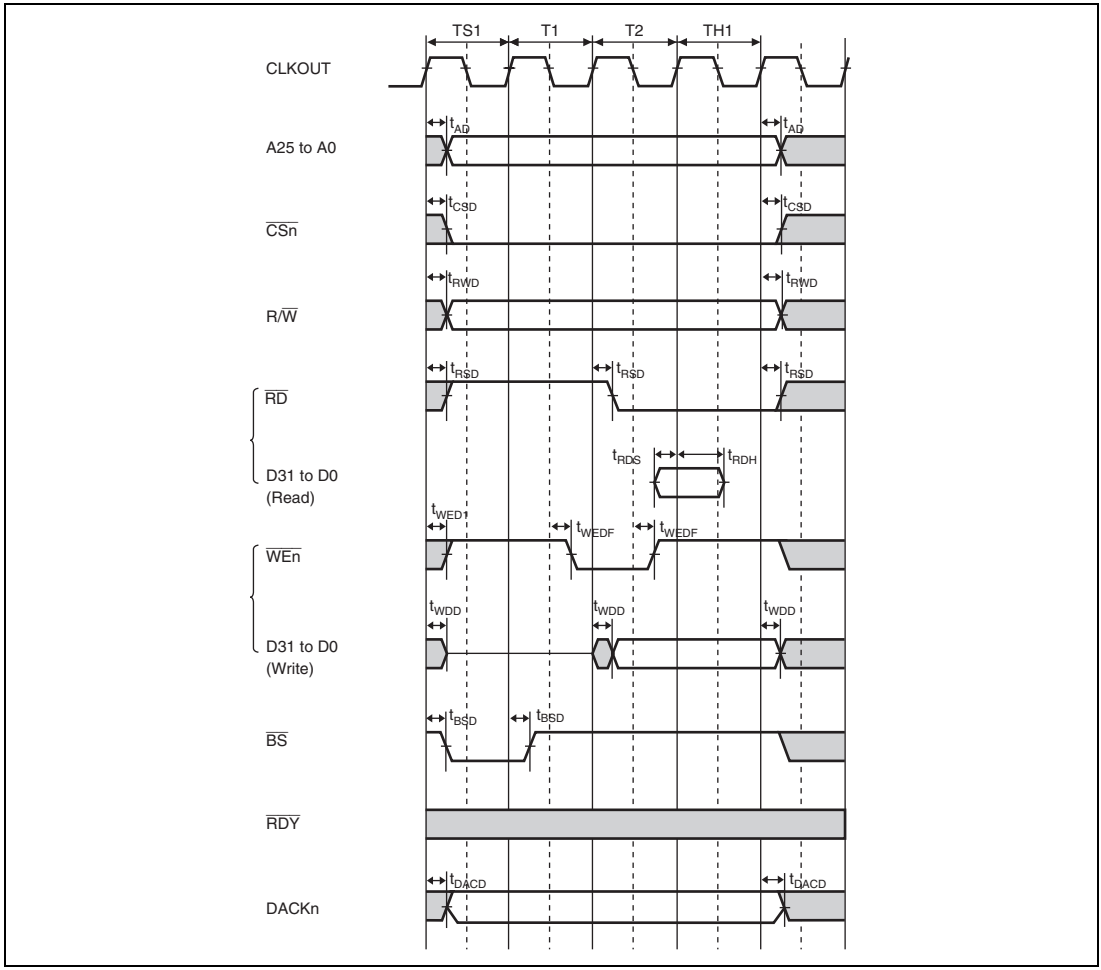


Figure 34.12 SRAM Bus Cycle: Basic Bus Cycle (CSnWCR.IW = 0000, CSnWCR.RDS = 001, CSnWCR.WTS = 001, CSnWCR.RDH = 001, CSnWCR.WTH = 001)

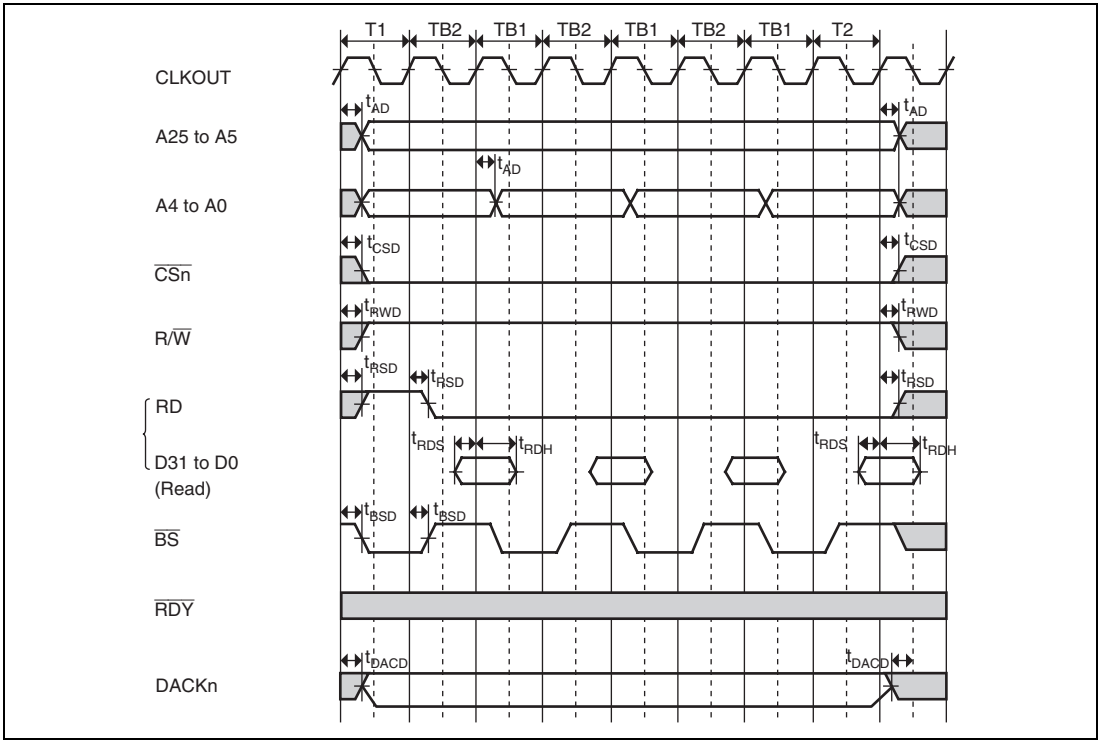


Figure 34.13 Burst ROM Bus Cycle (No Wait)

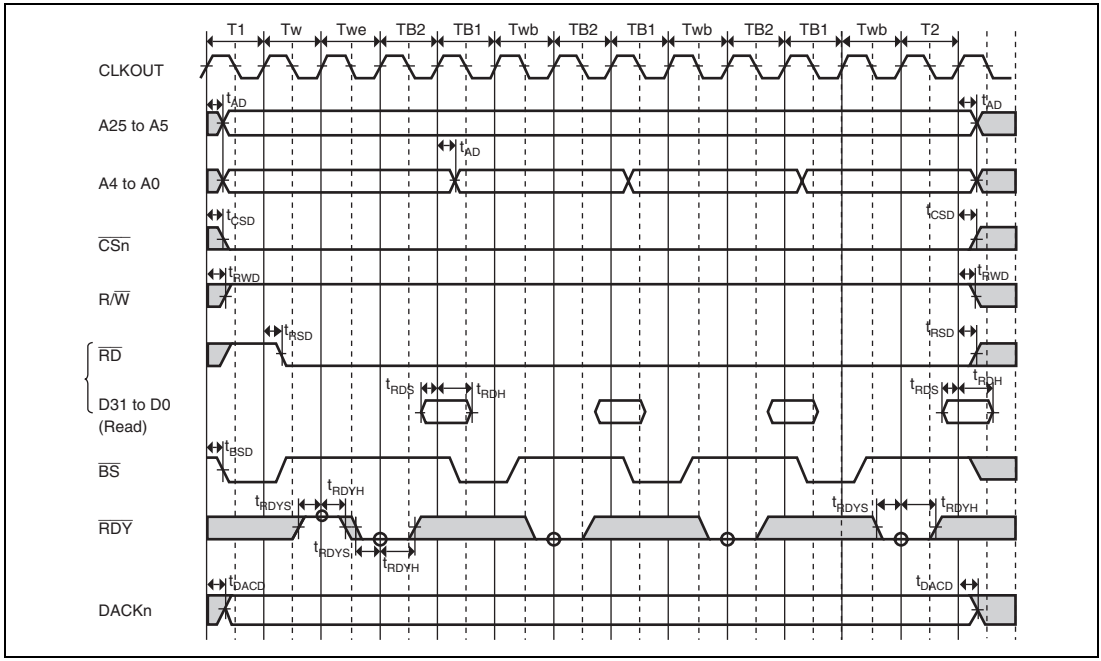


Figure 34.14 Burst ROM Bus Cycle (One Internal Wait Cycle + One External Wait Cycle for the 1st Datum; One Internal Wait Cycle for the 2nd, 3rd, and 4th Data)

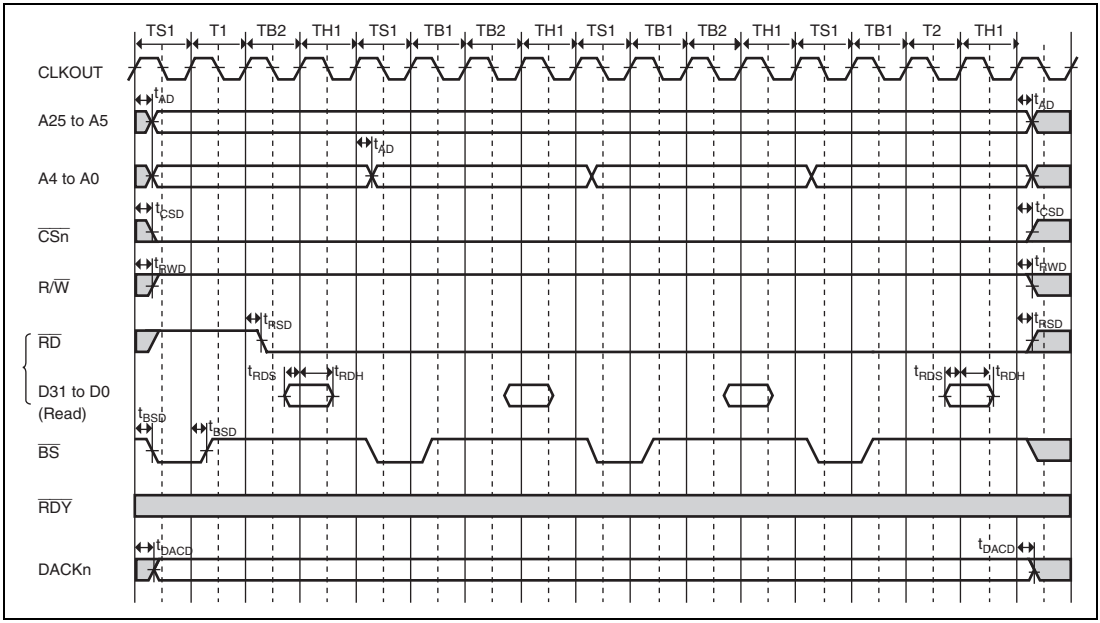


Figure 34.15 Burst ROM Bus Cycle (CSnWCR.IW = 0000, CSnWCR.RDS = 001, CSnWCR.WTS = 001, CSnWCR.RDH = 001, CSnWCR.WTH = 001)

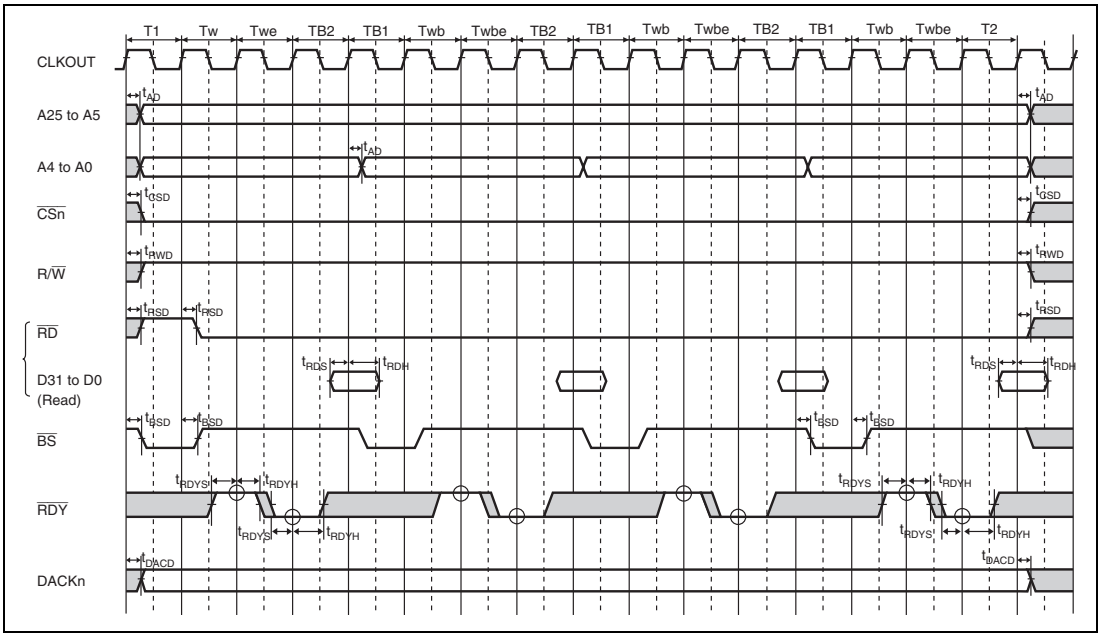


Figure 34.16 Burst ROM Bus Cycle (One Internal Wait Cycle + One External Wait Cycle)

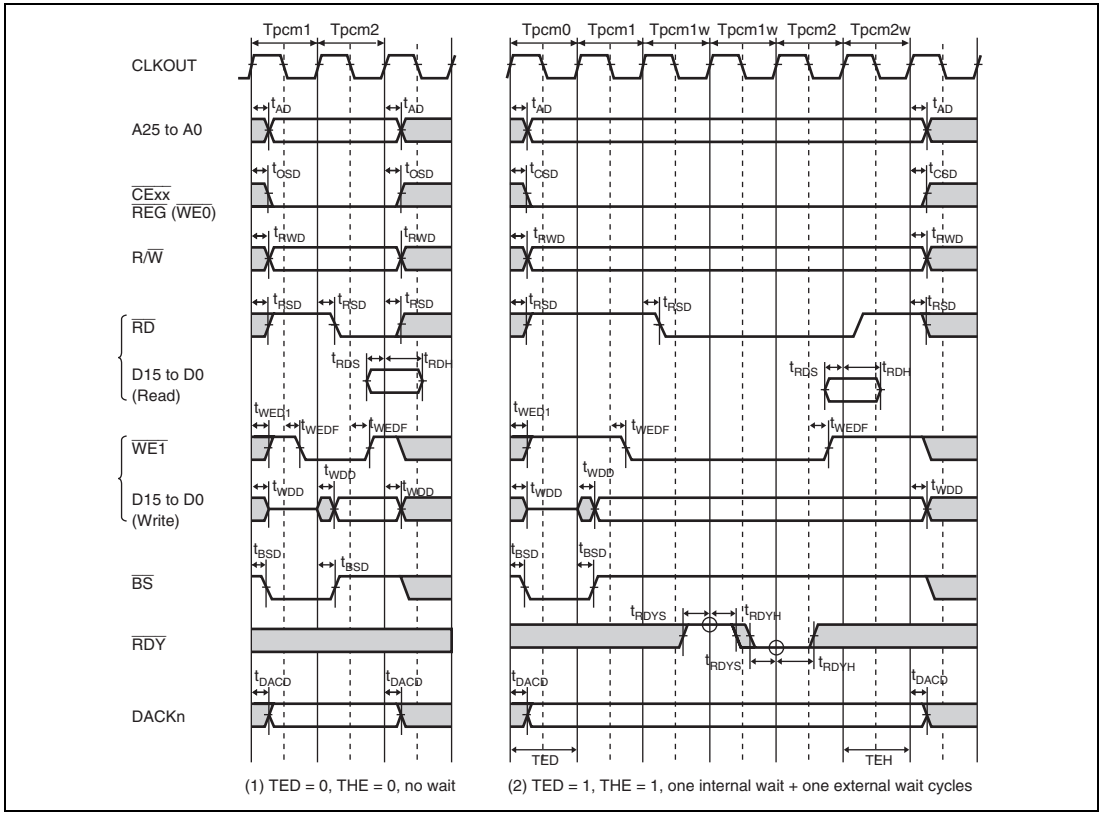


Figure 34.17 PCMCIA Memory Bus Cycle

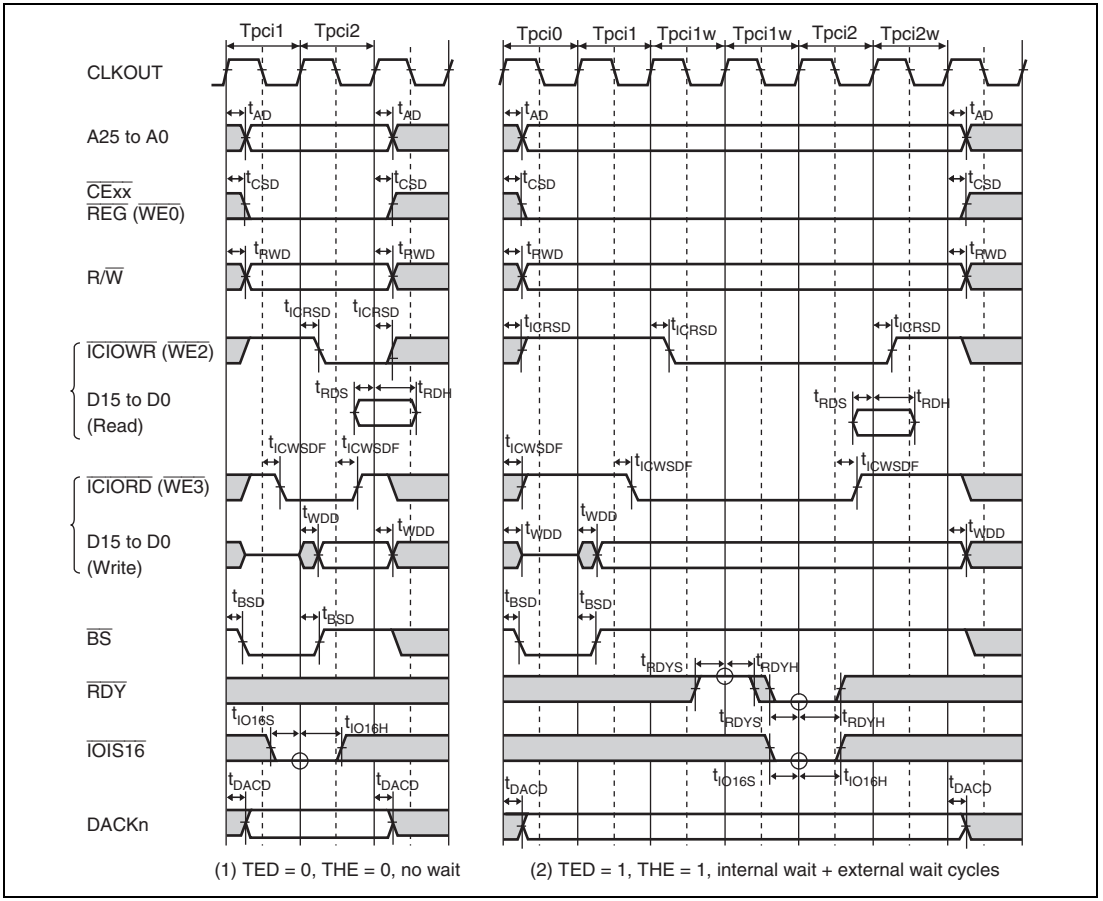


Figure 34.18 PCMCIA I/O Bus Cycle

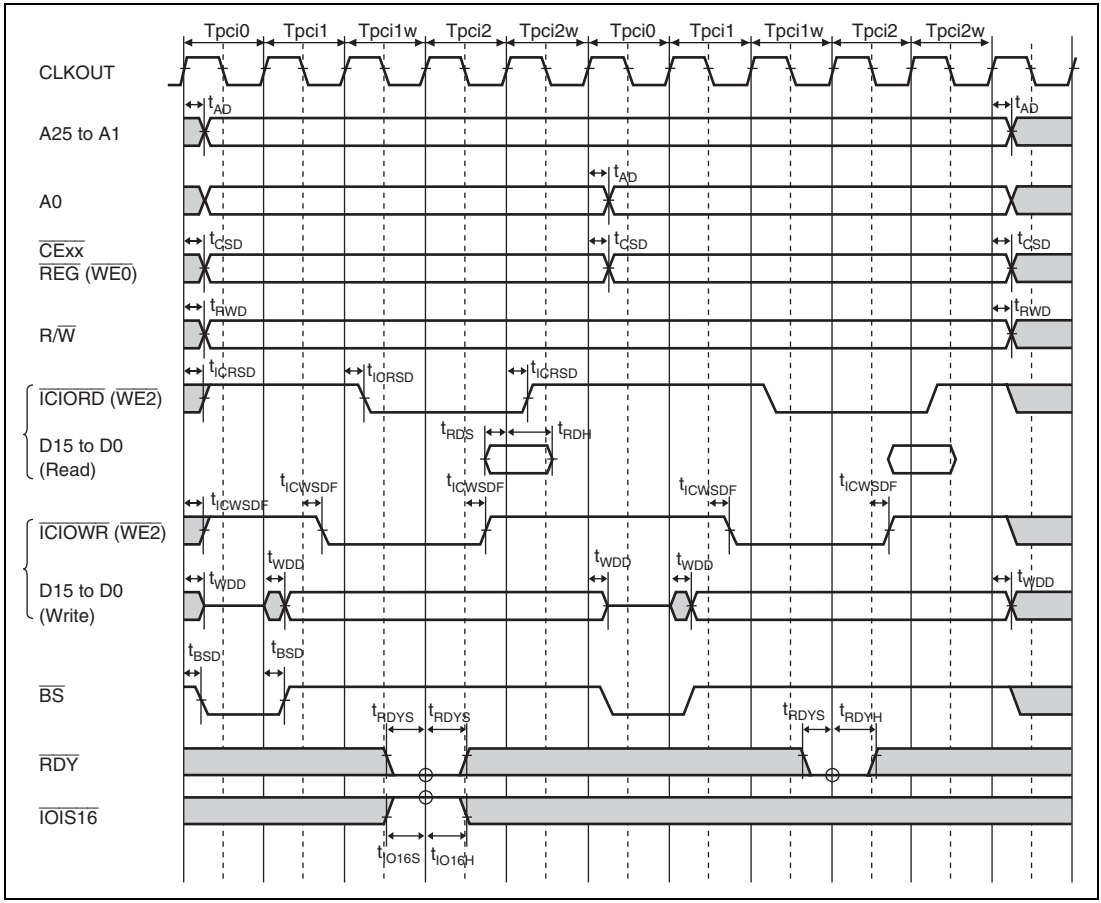
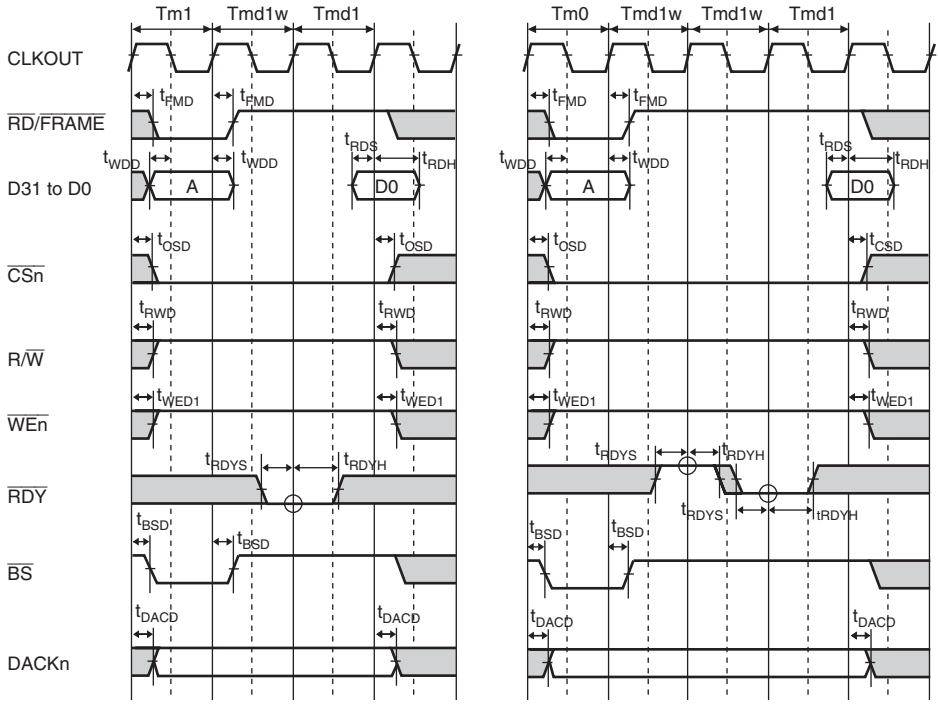


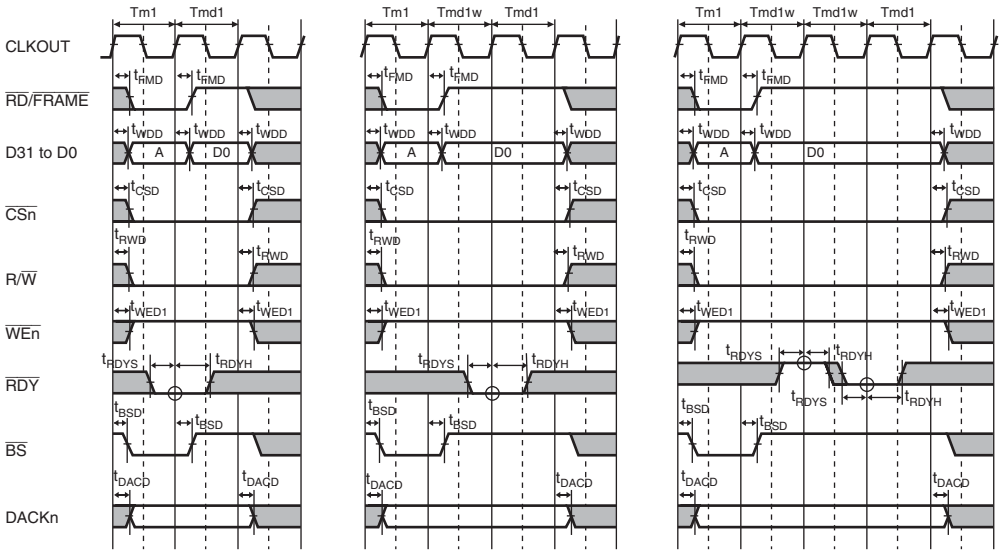
Figure 34.19 PCMCIA I/O Bus Cycle
(TED = 1, TEH = 1, One Internal Wait Cycle, with Bus Sizing)



(1) 1st data: One internal wait cycle
Information in the first data bus cycle
D31 to D29: Access size
000: Byte
001: Word (2 bytes)
010: Longword (4 bytes)
011: Quadword (8 bytes)
1xx: Burst (32 bytes)
D25 to D0: Address

(2) 1st data: One internal wait + one external wait cycles
Information in the first data bus cycle
D31 to D29: Access size
000: Byte
001: Word (2 bytes)
010: Longword (4 bytes)
011: Quadword (8 bytes)
1xx: Burst (32 bytes)
D25 to D0: Address

Figure 34.20 MPX Basic Bus Cycle (Read)

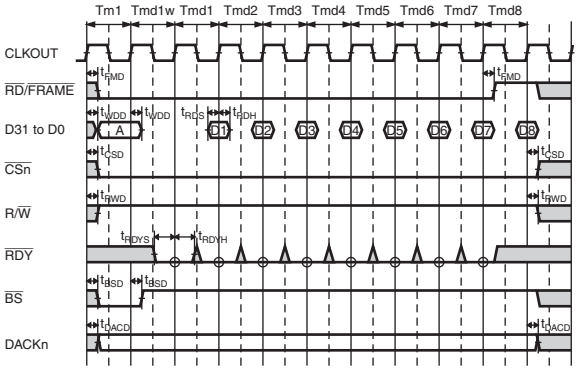


(1) 1st data: No wait
Information in the first data bus cycle
D31 to D29: Access size
000: Byte
001: Word (2 bytes)
010: Longword (4 bytes)
011: Quadword (8 bytes)
1xx: Burst (32 bytes)
D25 to D0: Address

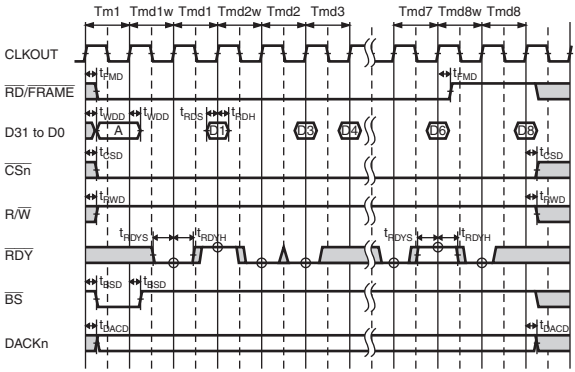
(2) 1st data: One internal wait cycle
Information in the first data bus cycle
D31 to D29: Access size
000: Byte
001: Word (2 bytes)
010: Longword (4 bytes)
011: Quadword (8 bytes)
1xx: Burst (32 bytes)
D25 to D0: Address

(3) 1st data: One internal wait + one external wait cycles
Information in the first data bus cycle
D31 to D29: Access size
000: Byte
001: Word (2 bytes)
010: Longword (4 bytes)
011: Quadword (8 bytes)
1xx: Burst (32 bytes)
D25 to D0: Address

Figure 34.21 MPX Basic Bus Cycle (Write)

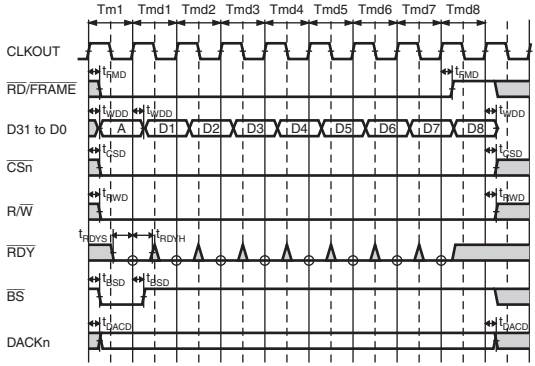


(1) 1st data: One internal wait cycle, 2nd to 8th data: No waitl
Information in the first data bus cycle
D31 to D29: Access size
000: Byte
001: Word (2 bytes)
010: Longword (4 bytes)
011: Quadword (8 bytes)
1xx: Burst (32 bytes)
D25 to D0: Address

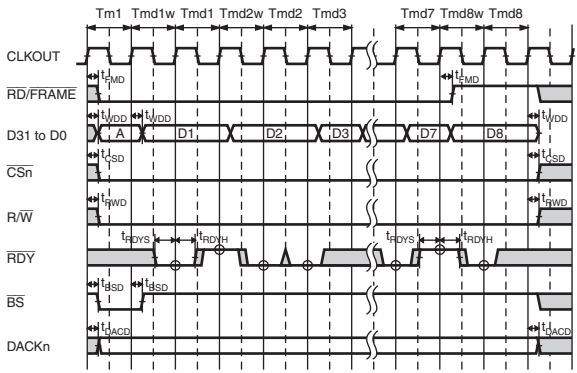


(2) 1st data: No internal wait, 2nd to 8th data: No internal wait + external wait control
Information in the first data bus cycle
D31 to D29: Access size
000: Byte
001: Word (2 bytes)
010: Longword (4 bytes)
011: Quadword (8 bytes)
1xx: Burst (32 bytes)
D25 to D0: Address

Figure 34.22 MPX Bus Cycle (Burst Read)



(1) No internal wait
Information in the first data bus cycle
D31 to D29: Access size
000: Byte
001: Word (2 bytes)
010: Longword (4 bytes)
011: Quadword (8 bytes)
1xx: Burst (32 bytes)
D25 to D0: Address



(2) 1st data: One internal wait cycle, 2nd to 8th data: No internal wait + external wait control
Information in the first data bus cycle
D31 to D29: Access size
000: Byte
001: Word (2 bytes)
010: Longword (4 bytes)
011: Quadword (8 bytes)
1xx: Burst (32 bytes)
D25 to D0: Address

Figure 34.23 MPX Bus Cycle (Burst Write)

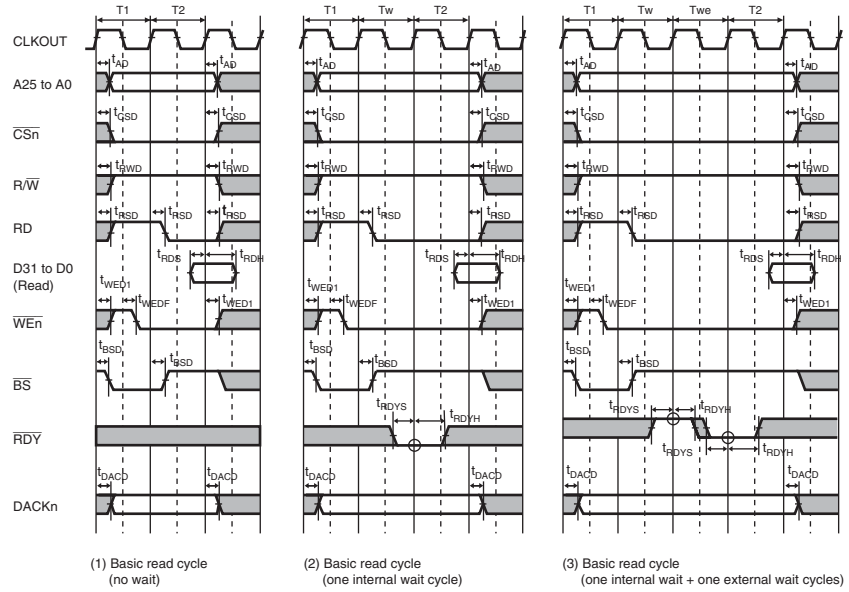


Figure 34.24 Memory Byte Control SRAM Bus Cycle

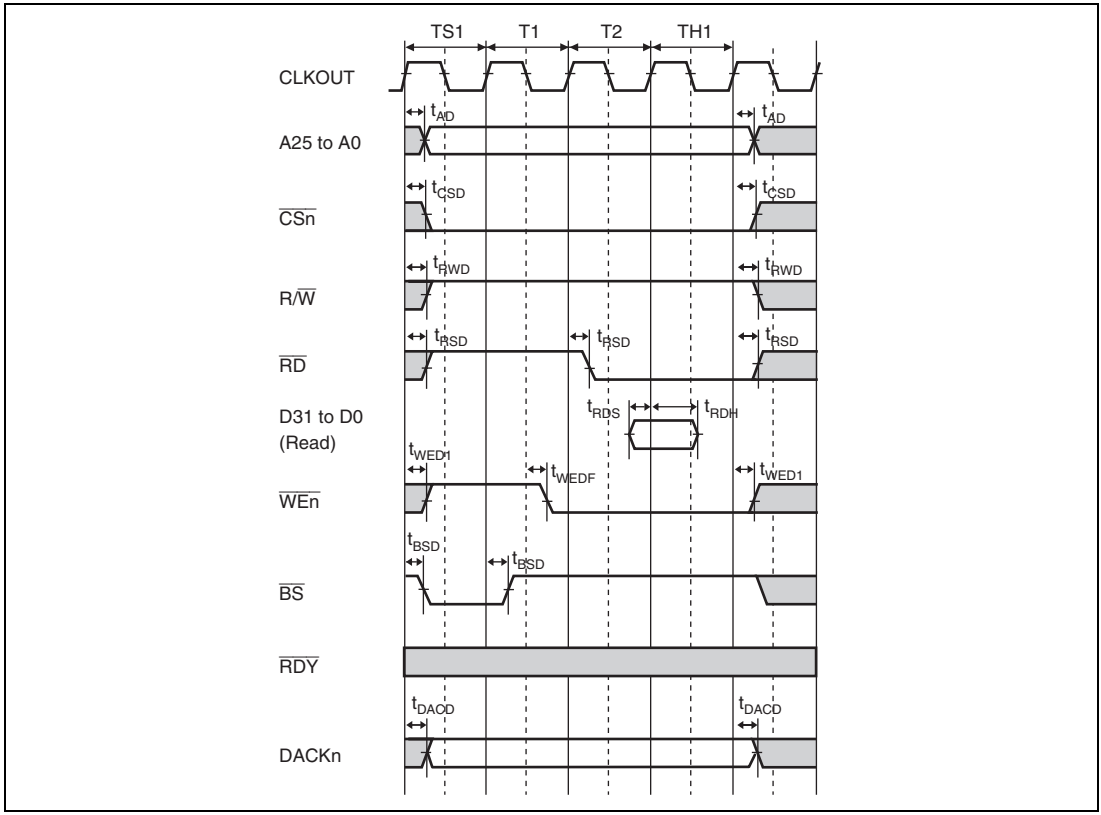


Figure 34.25 Memory Byte Control SRAM Bus Cycle: Basic Read Cycle (CSnWCR.IW = 0000, CSnWCR.RDS = 001, CSnWCR.WTS = 001, CSnWCR.RDH = 001, CSnWCR.WTH = 001)

34.3.4 DDR3-SDRAM Interface Signal Timing

Table 34.14 DDR3-SDRAM interface Signal Timing

Conditions: $V_{DD\text{-}DDR15}$= 1.425 to 1.575 V, MV_{REF}= $V_{DD\text{-}DDR15}$ /2, V_{DD}= 1.25 V,
T_a= –20 to +85°C, ODT on

Item	Symbol	Min.	Max.	Unit	Figure	Notes
MCK output cycle	t_{CK}	1.875	—	ns	34.26	
MCK output high-level pulse width	t_{CH}	0.43	0.57	t_{MCK}	34.26	
MCK output low-level pulse width	t_{CL}	0.43	0.57	t_{MCK}	34.26	
Address and control signal setup time (relative to MCK rising edge)	t_{IS1}	1513	—	ps	34.27	MA, MBA, $\overline{\text{MRAS}}$, $\overline{\text{MCAS}}$, $\overline{\text{MWE}}$
	t_{IS2}	523	—			$\overline{\text{MCS}}$, MODT
Address and control signal hold time (relative to MCK rising edge)	t_{IH1}	978	—	ps	34.27	MA, MBA, $\overline{\text{MRAS}}$, $\overline{\text{MCAS}}$, $\overline{\text{MWE}}$
	t_{IH2}	523	—			$\overline{\text{MCS}}$, MODT
MDQS, $\overline{\text{MDQS}}$ rise time (relative to MCK rising edge)	t_{DQSS}	–0.183	0.183	t_{CK}	34.28	Write
MDQS, $\overline{\text{MDQS}}$ fall setup time (relative to MCK rising edge)	t_{DSS}	0.267	—	t_{CK}	34.28	
MDQS, $\overline{\text{MDQS}}$ fall hold time (relative to MCK rising edge)	t_{DSH}	0.267	—	t_{CK}	34.28	
MDQS, $\overline{\text{MDQS}}$ differential output low-level pulse width	t_{DQSL}	0.45	0.55	t_{CK}	34.29	
MDQS, $\overline{\text{MDQS}}$ differential output high-level pulse width	t_{DQSH}	0.45	0.55	t_{CK}	34.29	
MDQS, $\overline{\text{MDQS}}$ write preamble time	t_{WPRE}	0.9	—	t_{CK}	34.29	
MDQS, $\overline{\text{MDQS}}$ write postamble time	t_{WPST}	0.3	—	t_{CK}	34.29	
MDQ/MDQM setup time (relative to VIH (AC)/VIL (AC) of DQS, $\overline{\text{DQS}}$)	t_{DS}	284	—	ps	34.30	
MDQ/MDQM hold time (relative to VIH (AC)/VIL (AC) of DQS, $\overline{\text{DQS}}$)	t_{DH}	284	—	ps	34.30	

Item	Symbol	Min.	Max.	Unit	Figure	Notes
Skew between MDQS, $\overline{\text{MDQS}}$ and MDQ	t_{DQSQ}	—	225	ps	34.31	Read
MDQ hold time (relative to MDQS, $\overline{\text{MDQS}}$)	t_{QH}	0.319	—	t_{CK}	34.31	
MDQS, $\overline{\text{MDQS}}$ read preamble time	t_{RPRE}	0.9	—	t_{CK}	34.32	
MDQS, $\overline{\text{MDQS}}$ read postamble time	t_{RPST}	0.3	—	t_{CK}	34.32	
MDQS, $\overline{\text{MDQS}}$ high-level pulse width	t_{QSH}	0.38	—	t_{CK}	34.32	
MDQS, $\overline{\text{MDQS}}$ low-level pulse width	t_{QSL}	0.38	—	t_{CK}	34.32	
Skew between MDQS, $\overline{\text{MDQS}}$ and MCK, $\overline{\text{MCK}}$	t_{DQSCK}	–300	1300	ps	34.33	

Note: t_{MCK}: one MCK cycle time

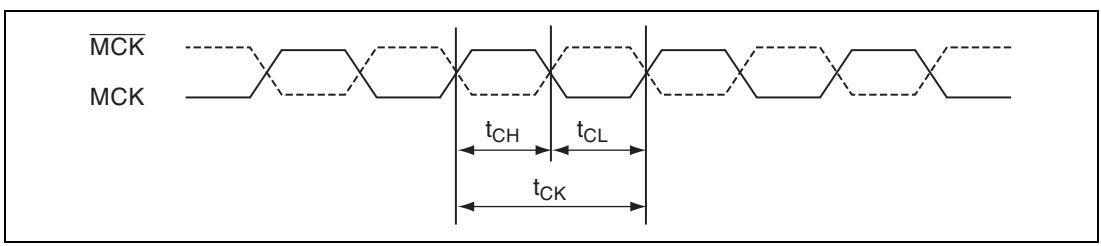


Figure 34.26 MCK Output Clock

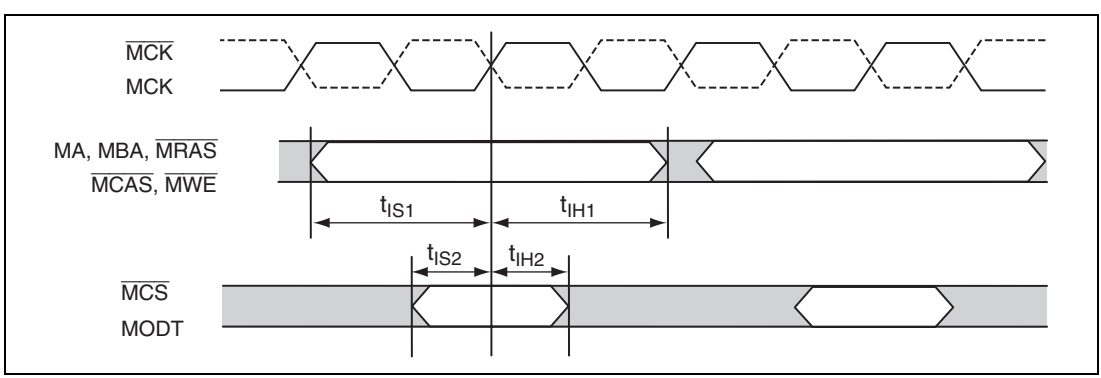


Figure 34.27 Relationship of Address Command Pins and Output Clock

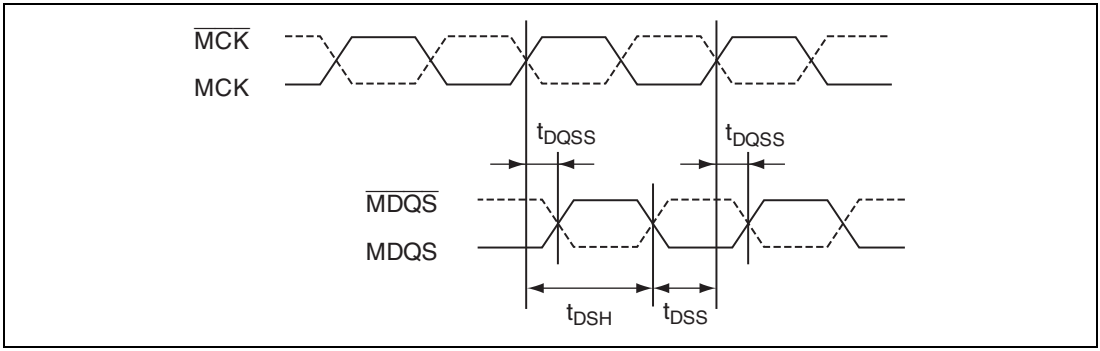


Figure 34.28 Relationship of MDQS Output Waveform to MCK (Write)

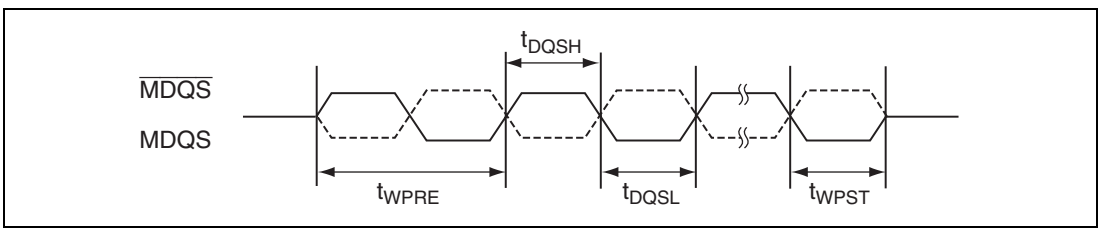


Figure 34.29 MDQS Output Timing (Write)

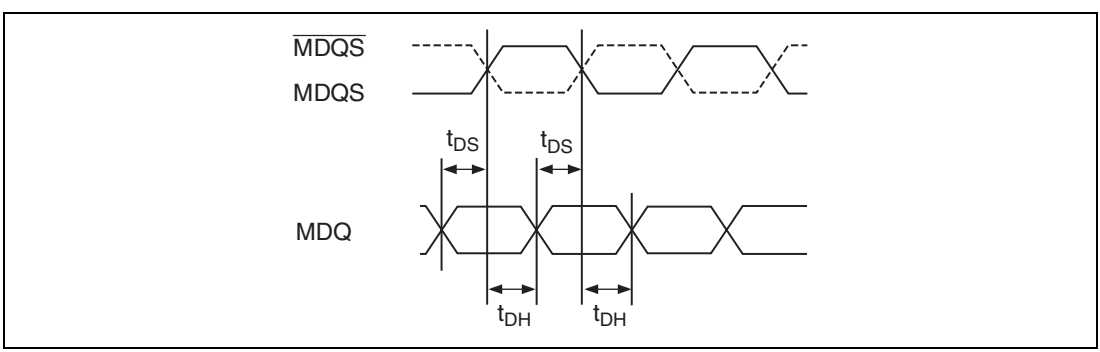


Figure 34.30 Relationship of MDQ Output Waveform to MDQS (Write)

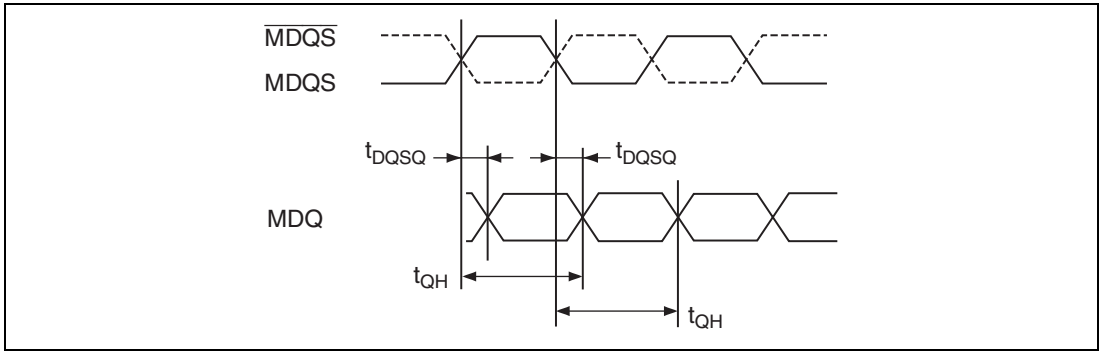


Figure 34.31 Relationship of MDQ Input Waveform to MDQS (Read)

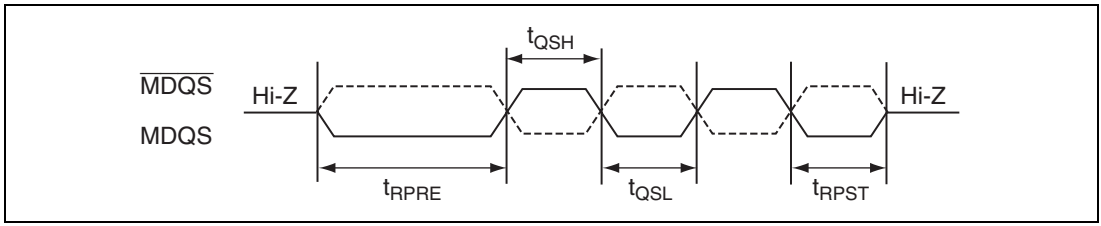


Figure 34.32 MDQS Input Timing (Read)

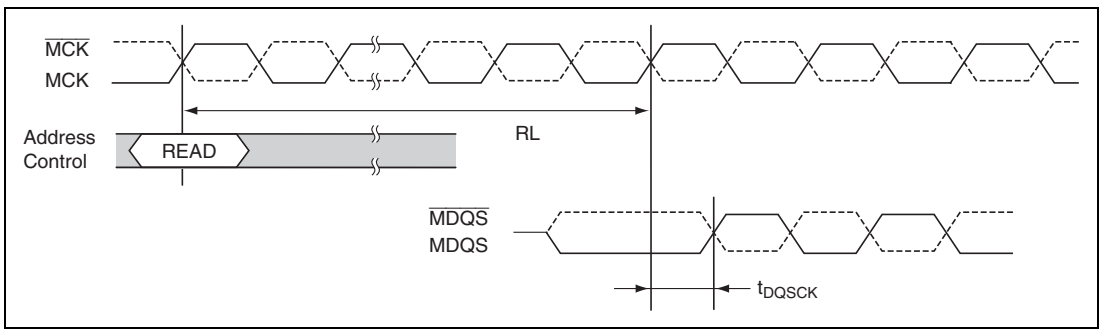


Figure 34.33 MDQS Skew Relative to MCK (Read)

34.3.5 Interrupt Signal Timing

Table 34.15 Interrupt Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
NMI setup time	t_{NMIS}	4	—	—	ns	34.34
NMI hold time	t_{NMIH}	1.5	—	—	ns	34.34
NMI pulse width (high level)	t_{NMIIS}	5	—	—	t_{cyc}*	34.35
NMI pulse width (low level)	t_{NMIIH}	5	—	—	t_{cyc}*	34.35
Edge-sense IRQ pulse width (high level)	t_{IRQIH}	5	—	—	t_{cyc}*	34.35
Edge-sense IRQ pulse width (low level)	t_{IRQIL}	5	—	—	t_{cyc}*	34.35
$\overline{IRL7}$ to $\overline{IRL0}$ setup time	t_{IRLS}	4	—	—	ns	34.34
$\overline{IRL7}$ to $\overline{IRL0}$ hold time	t_{IRLH}	1.5	—	—	ns	34.34
$\overline{IRQOUT}$ delay time	t_{IRQOD}	1.5	—	8	ns	34.36

Note: * t_{cyc} is the period of one CLKOUT cycle.

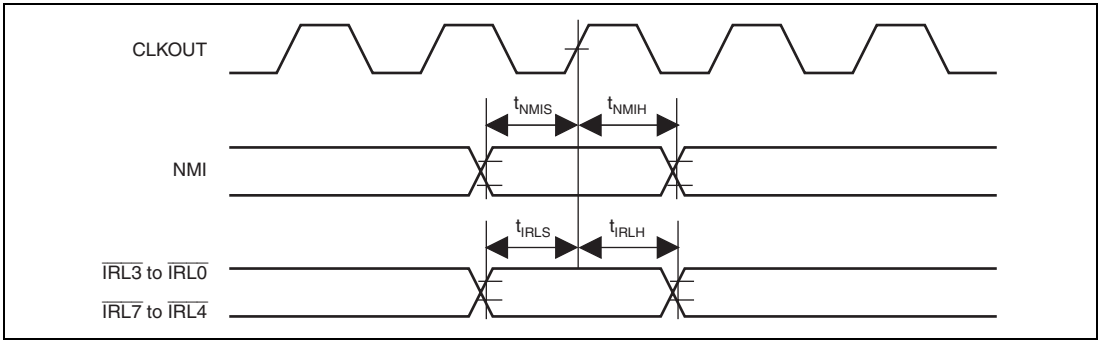


Figure 34.34 Interrupt Signal Input Timing (1)

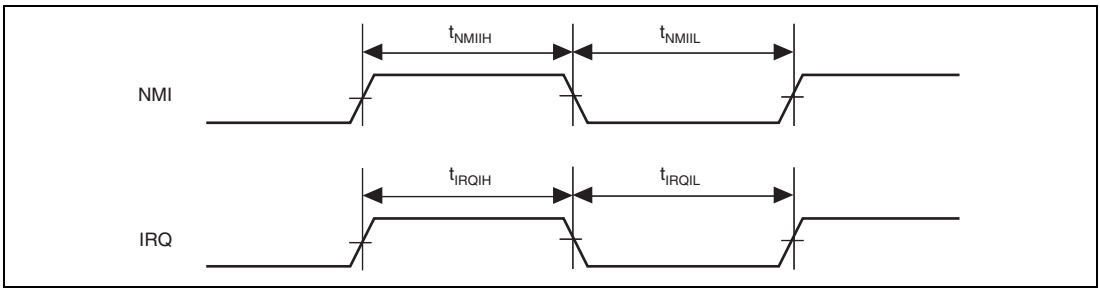


Figure 34.35 Interrupt Signal Input Timing (2)

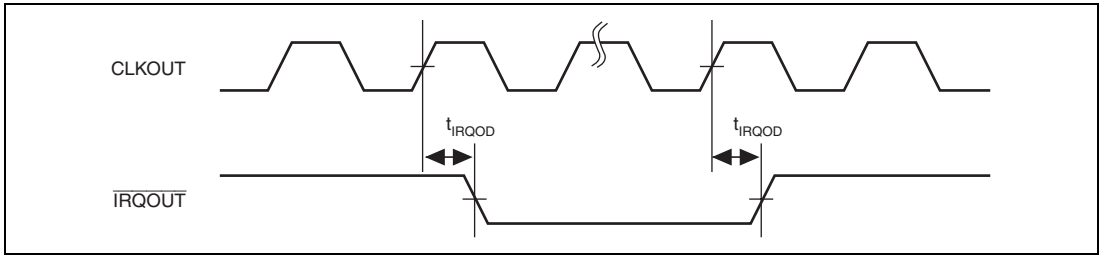


Figure 34.36 $\overline{\text{IRQOUT}}$ Timing

34.3.6 PCIEC Module Signal Timing

Table 34.16 PCIEC Module Signal Timing

Conditions: V_{CCQ} = 3.0 to 3.6 V, V_{DD} = 1.25 V, $V_{DD\text{-}PCIE15}$ = 1.5 V, T_a = –20 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure	Remarks
Unit interval	UI	399.88	—	400.12	ps		

Table 34.17 PCIEC Module Clock Input Timing

Conditions: V_{CCQ} = 3.0 to 3.6 V, V_{DD} = 1.25 V, $V_{DD\text{-}PCIE15}$ = 1.5 V, T_a = –20 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure	Remarks
Input frequency	Tperiod	100 MHz –300 PPM	100	100 MHz +300 PPM	MHz	34.37	

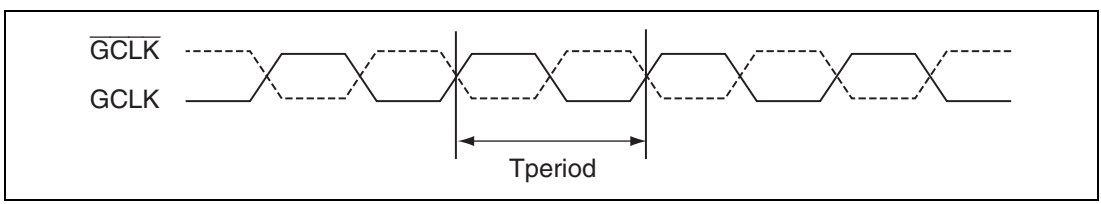


Figure 34.37 Differential REFCLK (GCLK/$\overline{\text{GCLK}}$) Input

34.3.7 USB Module Signal Timing

Table 34.18 USB Module Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Figure	Remarks
External input clock frequency		—	47.981	48.000	48.019	MHz		
High-speed	Data transfer speed	t_{HSDRAT}	479.76	480.00	480.24	Mbps		
Full-speed	DP/DM rise time	t_R	4	—	20	ns	34.38	
	DP/DM fall time	t_F	4	—	20	ns	34.40	
	Rise time/fall time matching	t_{RFM}	90	—	111.1	%		
	DP/DM crossover voltage	t_{CRS}	1.3	—	2.0	V		
Low-speed	DP/DM rise time	t_R	75	—	300	ns	34.39	
	DP/DM fall time	t_F	75	—	300	ns	34.40	
	Rise time/fall time matching	t_{RFM}	80	—	125	%		
	DP/DM crossover voltage	t_{CRS}	1.3	—	2.0	V		

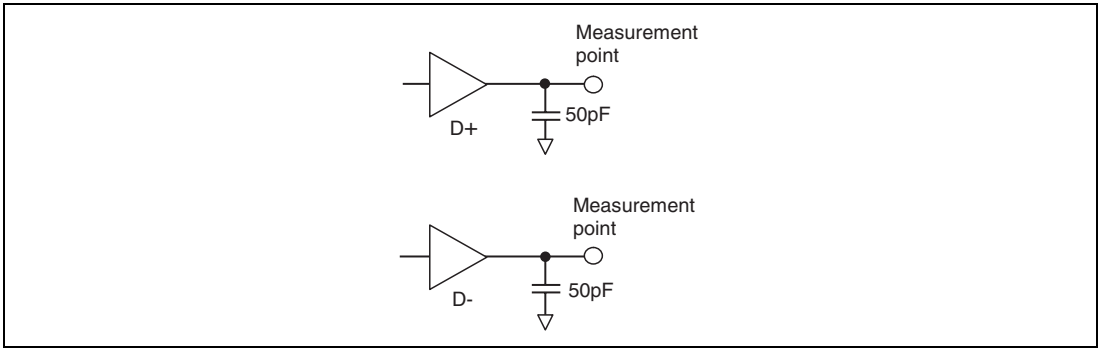


Figure 34.38 USB Full-Speed Mode

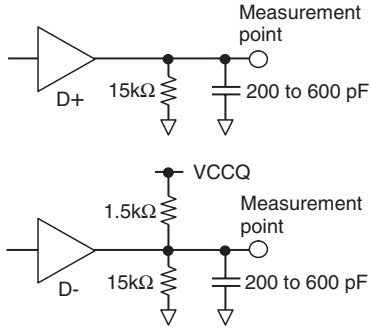


Figure 34.39 USB Low-Speed Mode

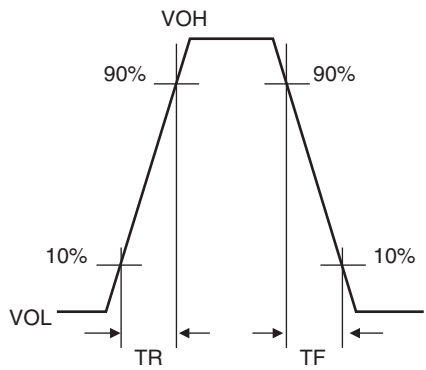


Figure 34.40 USB Signal Waveform

34.3.8 DMAC Module Signal Timing

Table 34.19 DMAC Module Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Max.	Unit	Figure	Remarks
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	2.5	—	ns	34.41	
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	1.5	—			
$\overline{\text{DRAK}}$ delay time	t_{DRAKD}	1.5	7			

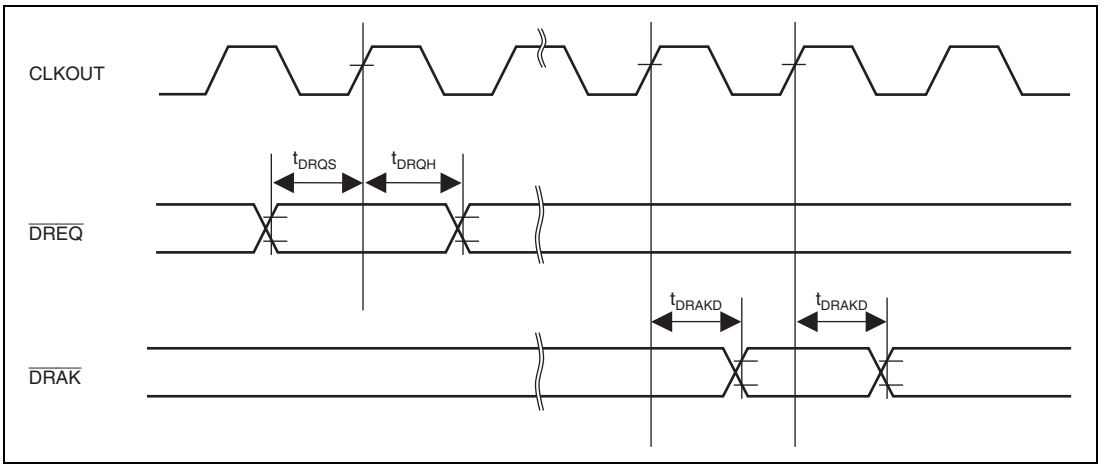


Figure 34.41 $\overline{\text{DREQ}}$/$\overline{\text{DRAK}}$ Signal Timing

34.3.9 TMU Module Signal Timing

Table 34.20 TMU Module Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C

Item	Symbol	Min.	Max.	Unit	Figure	Remarks
Timer clock pulse width (high)	t_{TCLKWH}	4	—	t_{Pcyc}	34.42	
Timer clock pulse width (low)	t_{TCLKWL}	4	—			
Timer clock rise time	t_{TCLKr}	—	0.8			
Timer clock fall time	t_{TCLKf}	—	0.8			

Note: t_{Pcyc} is the period of one peripheral clock (Pck) cycle.

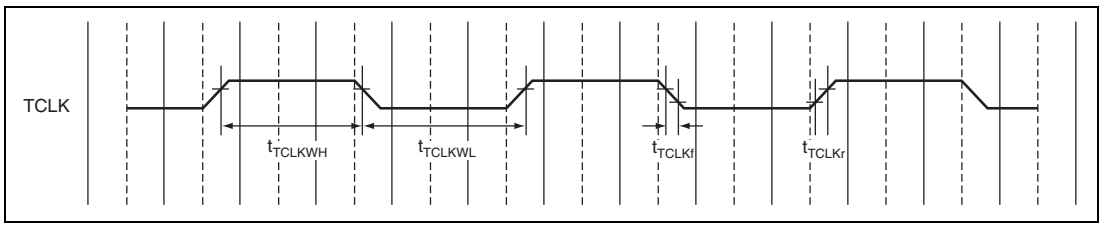


Figure 34.42 TCLK Input Timing

34.3.10 Ether MAC Controller

Table 34.21 Ether MAC Controller Control Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
ETH_TX_CLK period (100 Mbps operation)	t_{cyc}	—	40	—	ns	34.43
ETH_TX_CLK period (10 Mbps operation)	t_{cyc}	—	400	—		
ETH_TXD3 to ETH_TXD0, ETH_TX_EN, ETH_TX_ER output delay time	ttd	0	—	25		
ETH_RX_CLK period (100 Mbps operation)	rcyc	—	40	—	ns	34.44
ETH_RX_CLK period (10 Mbps operation)	rcyc	—	400	—		
ETH_RXD3 to ETH_RXD0, ETH_RX_DV, ETH_RX_ER setup time	trs	10	—	—		
ETH_RXD3 to ETH_RXD0, ETH_RX_EN, ETH_RX_ER hold time	trh	10	—	—		

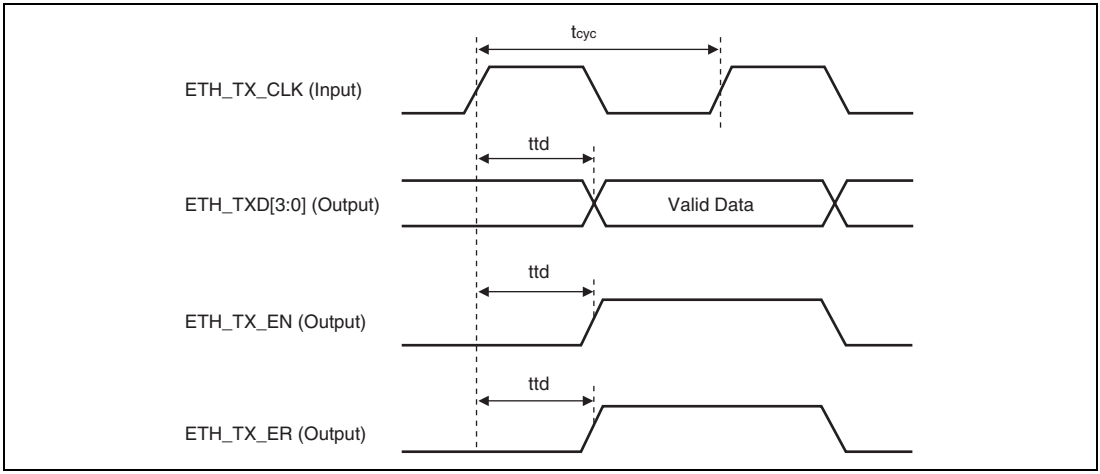


Figure 34.43 MII Interface (Transmission)

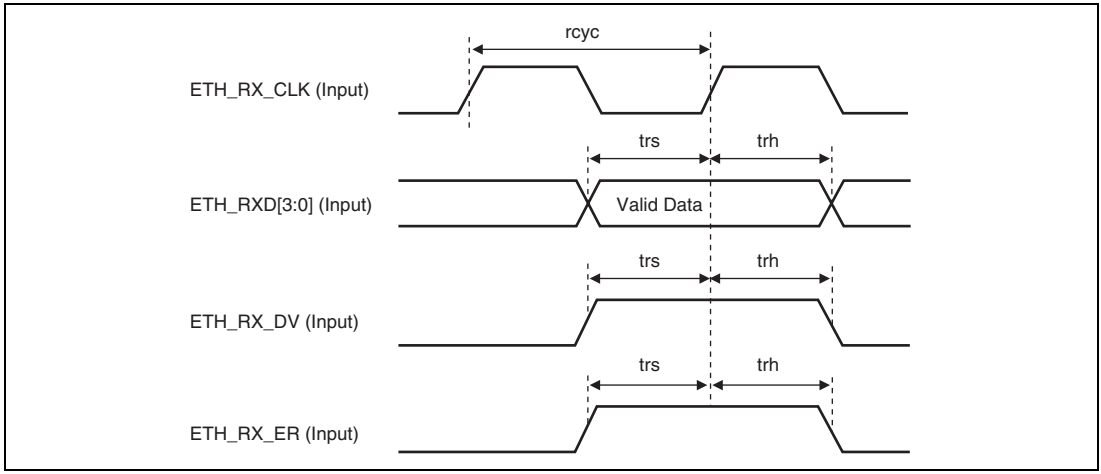


Figure 34.44 MII Interface (Reception)

34.3.11 Display Unit Signal Timing

Table 34.22 DCLKIN Signal Timing

Conditions: V_{CCQ} = 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
DCLKIN cycle time	t_{DICYC}	15	—	—	ns	34.45
DCLKIN high level time	t_{DCKIH}	5.5	—	—	ns	
DCLKIN low level time	t_{DCKIL}	5.5	—	—	ns	

Table 34.23 Display Timing

Conditions: V_{CCQ} = 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Typ.	Max.	Unit	Figures
Display input control signal setup time	t_{DS}	5	—	—	ns	34.46 (with respect to DCLKIN)
Display input control signal hold time	t_{DH}	3	—	—	ns	
DCLKOUT output cycle time	t_{DCYC}	20	—	—	ns	34.47 (with respect to DCLKOUT)
DCLKOUT output high level width	t_{DCKH}	6	—	—	ns	
Delay time of display output control signal output	t_{DD}	–3	—	8	ns	
Display digital data output delay time	t_{DD}	–3	—	8	ns	
$\overline{\text{HSYNC}}$ input low level width	t_{EXHLW}	4 t_{DCYC}	—	—	ns	34.48
$\overline{\text{HSYNC}}$ input high level width	t_{EXHHW}	4 t_{DCYC}	—	—	ns	
$\overline{\text{VSYNC}}$ input low level width	t_{EXVLW}	3 HC	—	—	t_{DCYC}	
ODDF setup time 1	t_{OD1}	(ys + yw) × HC	—	—	t_{DCYC}	
ODDF setup time 2	t_{OD2}	1 × HC	—	—	t_{DCYC}	

Table 34.24 Classification of Pins

Pin Classification	Display Input Control Signal*1	Display Output Control Signal*2	Digital Data for Display*3
Pin Name	$\overline{\text{VSYNC}}$	$\overline{\text{VSYNC}}$	DR0
	$\overline{\text{HSYNC}}$	$\overline{\text{HSYNC}}$	DR1
	ODDF	ODDF	DR2
		DISP	DR3
		CDE	DR4
			DR5
			DG0
			DG1
			DG2
			DG3
			DG4
			DG5
			DB0
			DB1
			DB2
			DB3
			DB4
			DB5

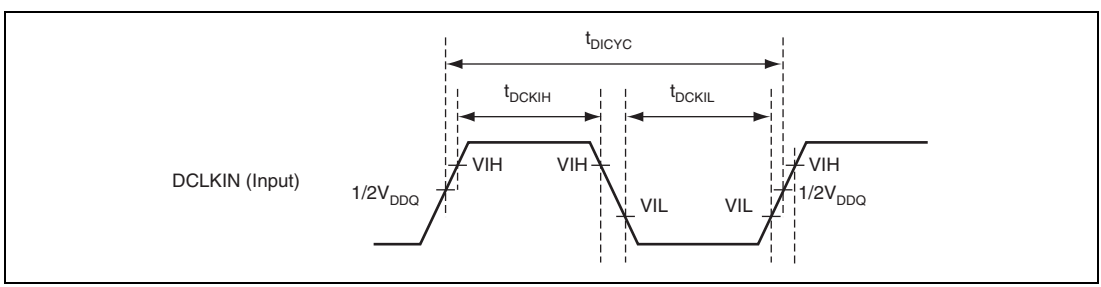


Figure 34.45 DCLKIN Clock Input Timing

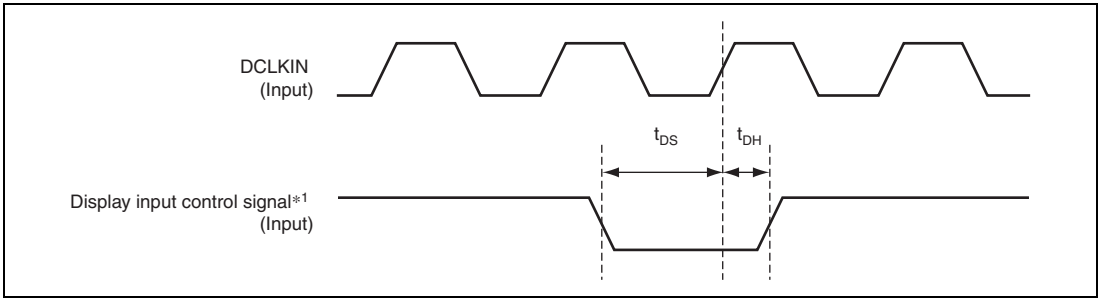


Figure 34.46 Display Timing (with Respect to DCLKIN)

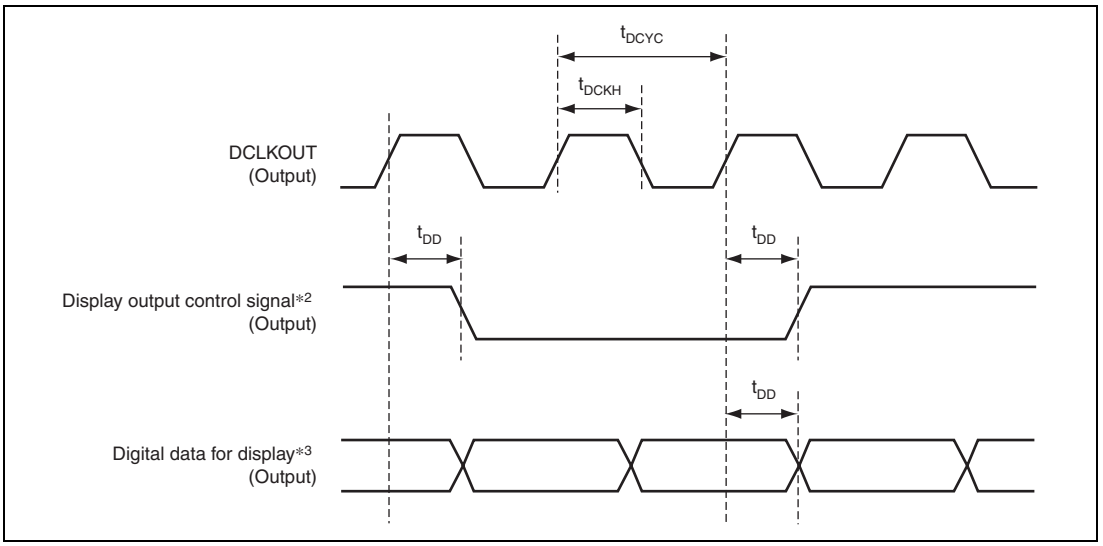


Figure 34.47 Display Timing (with Respect to DCLKOUT)

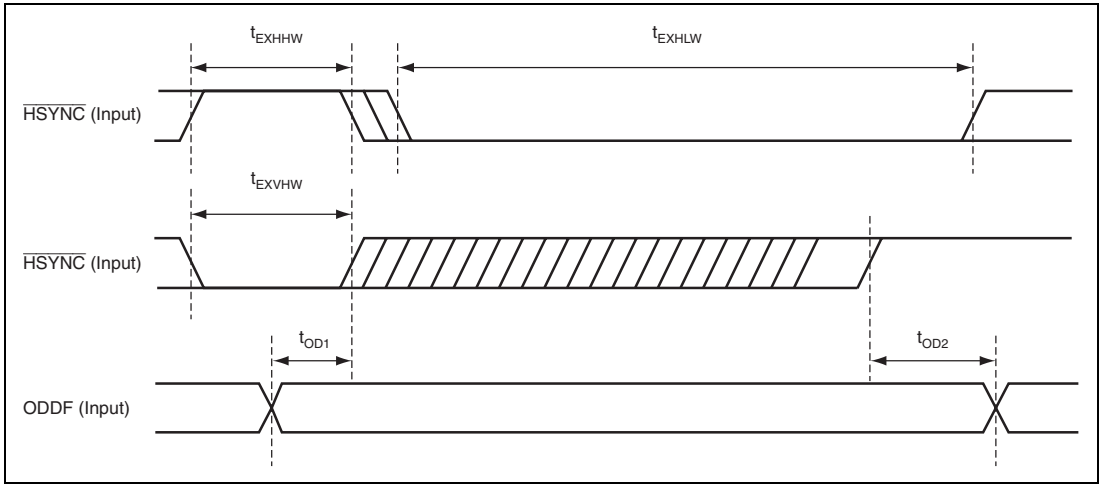


Figure 34.48 Display Timing in TV Synchronous Mode

34.3.12 SCIF Module Signal Timing

Table 34.25 SCIF Module Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Max.	Unit	Figure
Input clock cycle (asynchronous)	t_{Scyc}	4	—	t_{Pcyc}	34.49
Input clock cycle (clock synchronous)		10	—	t_{Pcyc}	
Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	
Input clock rise time	t_{SCKr}	—	0.8	t_{Pcyc}	
Input clock fall time	t_{SCKf}	—	0.8	t_{Pcyc}	
Transfer data delay time	t_{TXD}	—	8	t_{Pcyc}	34.50
Receive data setup time (clock synchronous)	t_{RXS}	t_{Pcyc} +2	—	ns	
Receive data hold time (clock synchronous)	t_{RXH}	16	—	ns	

Note: t_{pcyc} means one cycle time of the peripheral clock (Pck).

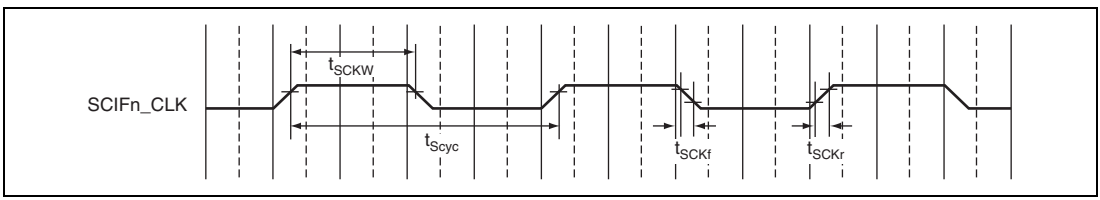


Figure 34.49 SCIFn_CLK Input Clock Timing

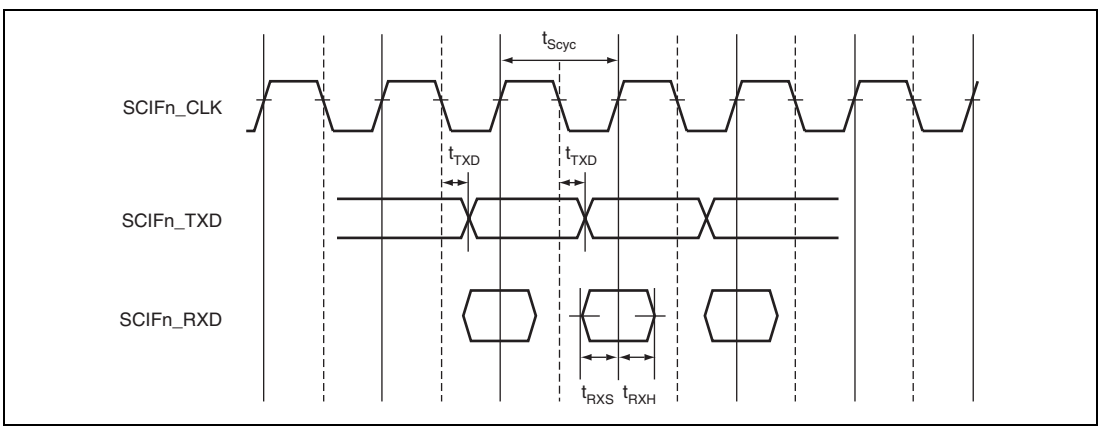


Figure 34.50 Clock Timing in SCIF I/O Synchronous Mode

34.3.13 HSPI Module Signal Timing

Table 34.26 HSPI Module Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Max.	Unit	Figure
HSPI clock frequency	T_{SPICYC}	—	Pck/8	MHz	34.51
HSPI clock high level width	t_{SPIHW}	60	—	ns	
HSPI clock low level width	t_{SPILW}	60	—	ns	
HSPI_TX setup time	$t_{SUSPITX}$	—	20	ns	
HSPI_TX delay time	t_{DSPITX}	—	20	ns	
HSPI_RX setup time	$t_{SUSPIRX}$	20	—	ns	
HSPI_RX hold time	$t_{HLSPIRX}$	20	—	ns	
HSPI_CS lead time	t_{CSLEAD}	100	—	ns	

Note: Pck is the frequency of the peripheral clock.

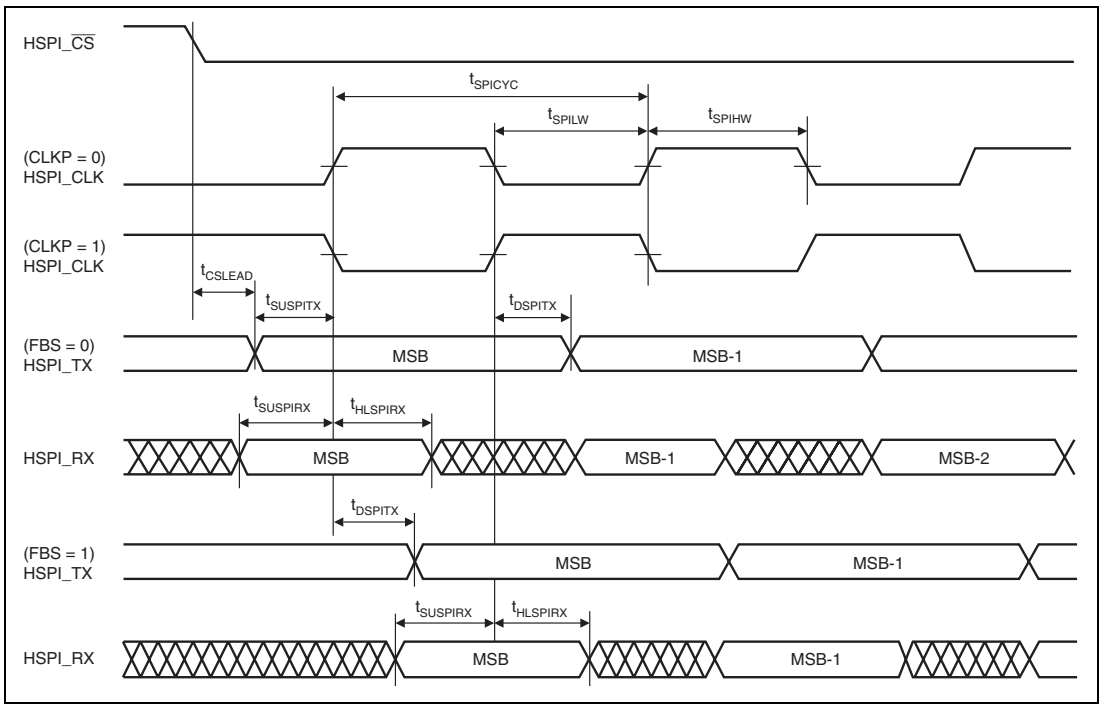


Figure 34.51 HSPI Data Output/Input Timing

34.3.14 NAND-Type Flash Memory Interface Timing

Table 34.27 NAND-Type Flash Memory Interface Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Max.	Unit	Figure
Command issue setup time	t_{NCDS}	2 × t_{fcyc}	—	ns	34.52, 34.56
Command issue hold time	t_{NCDH}	1.5 × t_{fcyc}	—	ns	
Data output setup time	t_{NDOS}	0.5 t_{fcyc}	—	ns	34.52, 34.53, 34.54, 34.56
Data output hold time	t_{NDOH}	0.5 t_{fcyc}	—	ns	
Command to address transition time 1	t_{NCDAD1}	1.5 × t_{fcyc}	—	ns	34.52, 34.53
Command to address transition time 2	t_{NCDAD2}	2 × t_{fcyc}	—	ns	34.53
$\overline{FWE}$ cycle time	t_{NWC}	t_{fcyc}	—	ns	34.53, 34.55
$\overline{FWE}$ low pulse width	t_{NWP}	0.5 t_{fcyc}	—	ns	34.52, 34.53, 34.54, 34.56
$\overline{FWE}$ high pulse width	t_{NWH}	0.5 t_{fcyc}	—	ns	34.53, 34.55
Address to ready/busy transition time	t_{NADRB}	—	32 × t_{pcyc}	ns	34.53, 34.54
Ready/busy to data read transition time 1	t_{NRBDR1}	1.5 × t_{fcyc}	—	ns	34.54
Ready/busy to data read transition time 2	t_{NRBDR2}	32 × t_{pcyc}	—	ns	
FSC cycle time	t_{NSCC}	t_{fcyc}	—	ns	
FSC high pulse width	t_{NSPH}	0.5 × t_{fcyc}	—	ns	
FSC low pulse width	t_{NSP}	0.5 × t_{fcyc}	—	ns	34.54, 34.56
Read data setup time	t_{NRDS}	24	—	ns	
Read data hold time	t_{NRDH}	5	—	ns	
Data write setup time	t_{NDWS}	32 × t_{pcyc}	—	ns	34.55
Command to status read transition time	t_{NCDSR}	4 × t_{fcyc}	—	ns	34.56
Command output off to status read transition time	t_{NCDFSR}	3.5 × t_{fcyc}	—	ns	
Status read setup time	t_{NSTS}	2.5 × t_{fcyc}	—	ns	

Notes: 1. t_{pcyc} is the period of one peripheral clock (Pck) cycle.
2. t_{fcyc} is the period of one FLCTL clock (Fck) cycle.

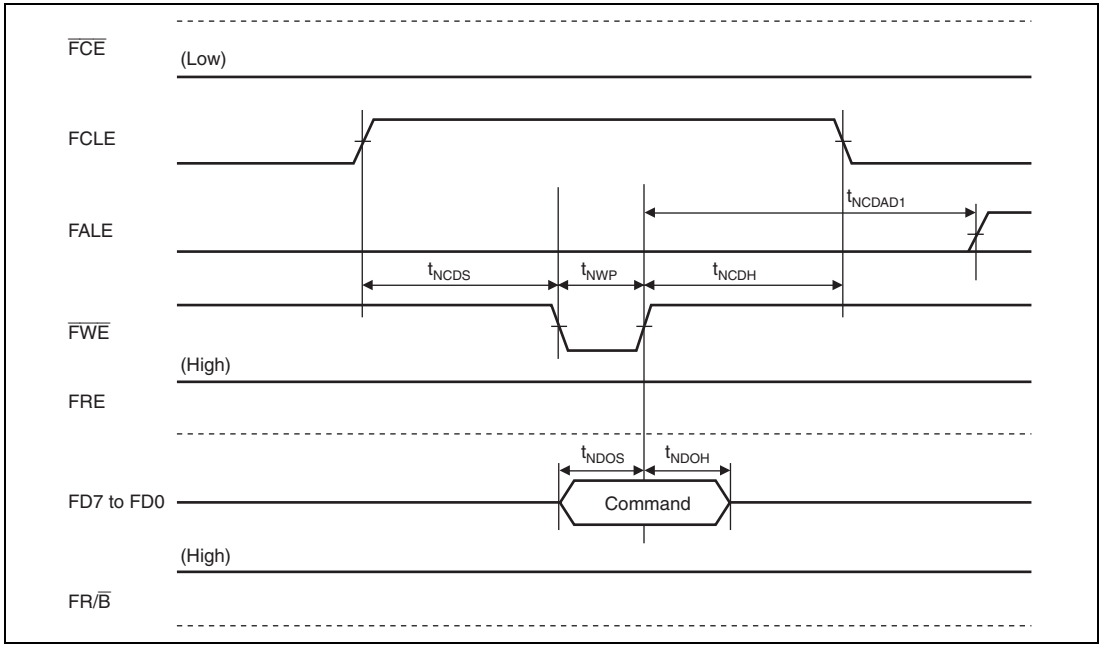


Figure 34.52 Command Issue Timing of NAND-Type Flash Memory

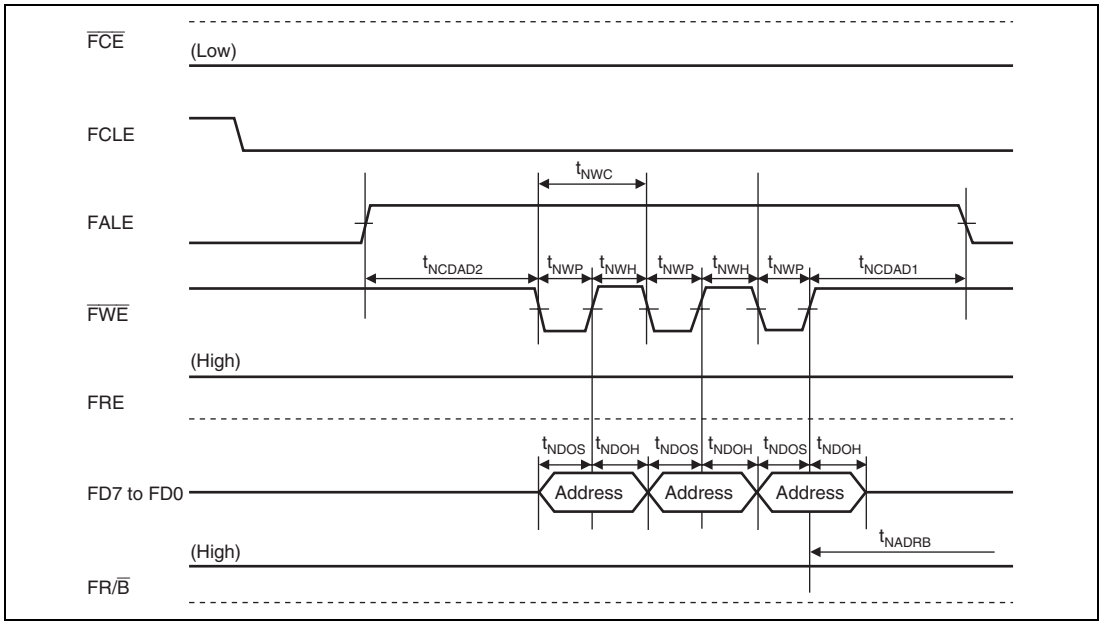


Figure 34.53 Address Issue Timing of NAND-Type Flash Memory

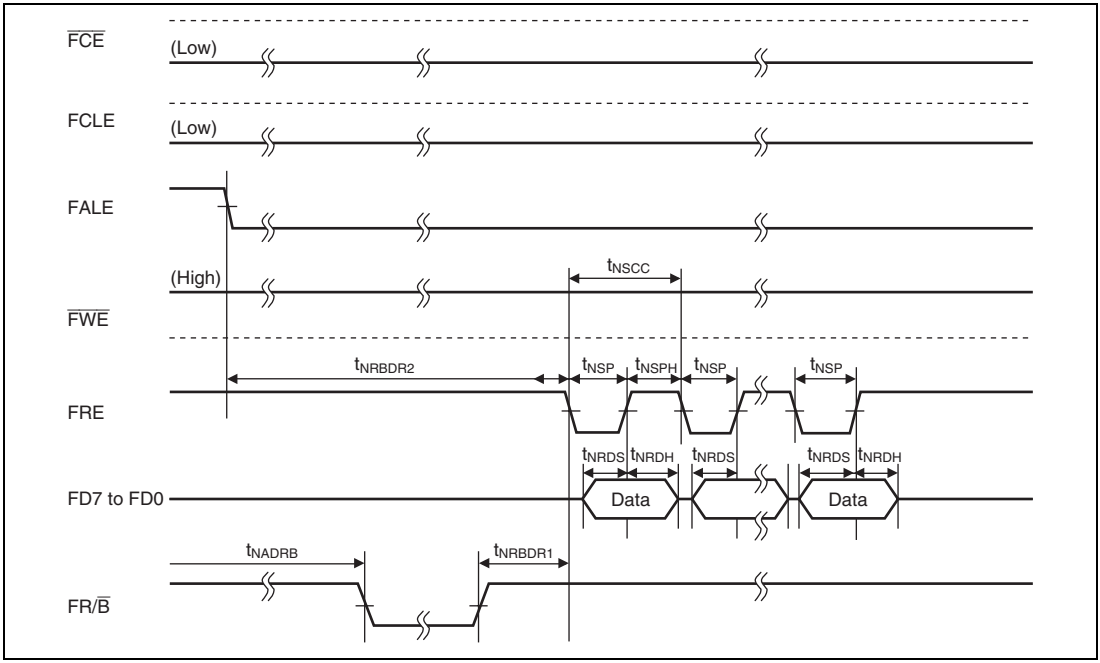


Figure 34.54 Data Read Timing of NAND-Type Flash Memory

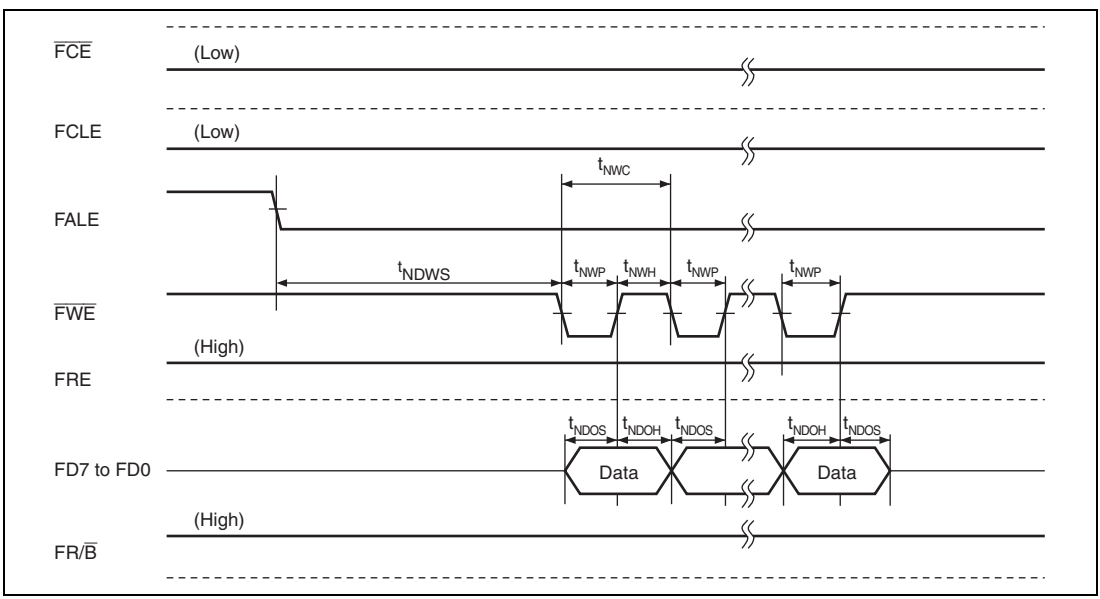


Figure 34.55 Data Write Timing of NAND-Type Flash Memory

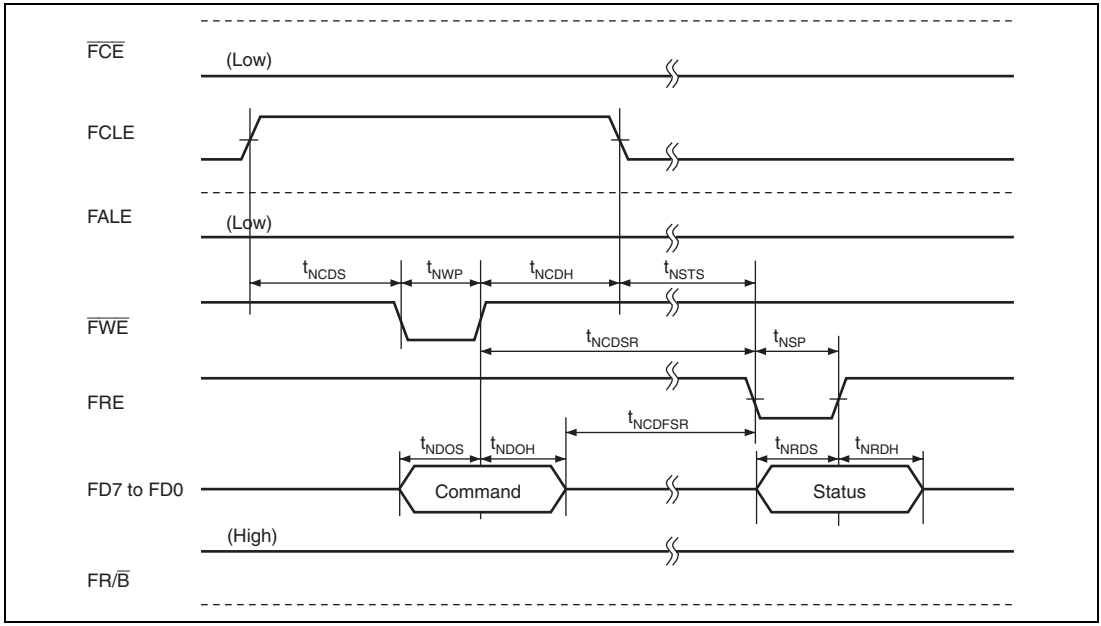


Figure 34.56 Status Read Timing of NAND-Type Flash Memory

34.3.15 HAC Interface Module Signal Timing

Table 34.28 HAC Interface Module Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{HAC\_RES}}$ active low pulse width	t_{RST_LOW}	1000	—	ns	34.57
HAC_SYNC active pulse width	t_{SYN_HIGH}	1000	—	ns	34.58
HAC_SYNC delay time 1	t_{SYNCD1}	—	15	ns	34.60
HAC_SYNC delay time 2	t_{SYNCD2}	—	15	ns	
HAC_SDOUT delay time	t_{SDOUTD}	—	15	ns	
HAC_SDIN setup time	t_{SDIN0S}	10	—	ns	
HAC_SDIN hold time	t_{SDIN0H}	10	—	ns	
HAC_BITCLK input high level width	t_{ICL0_HIGH}	t_{Pcyc}/2	—	ns	34.59
HAC_BITCLK input low level width	t_{ICL0_LOW}	t_{Pcyc}/2	—	ns	

Note: t_{Pcyc} is the period of one peripheral clock (Pck) cycle.

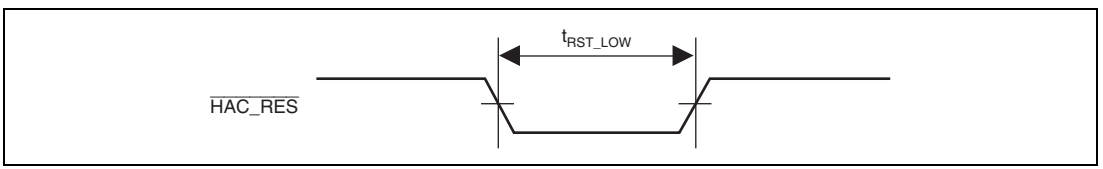


Figure 34.57 HAC Cold Reset Timing

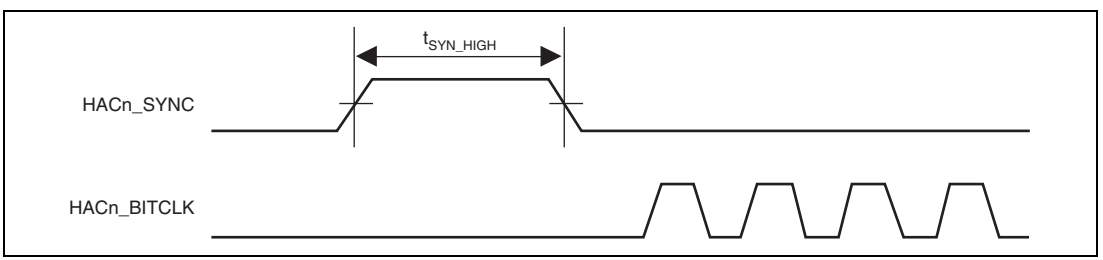


Figure 34.58 HAC Warm Reset Timing

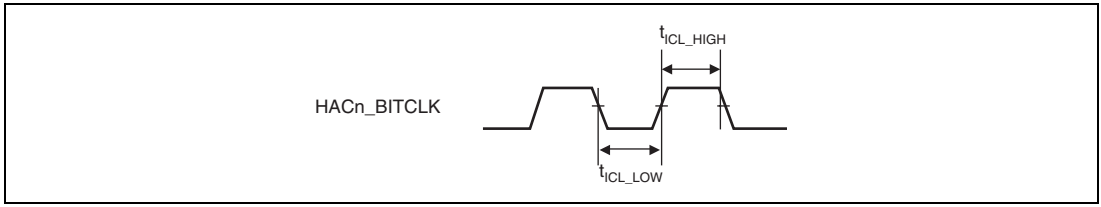


Figure 34.59 HAC Clock Input Timing

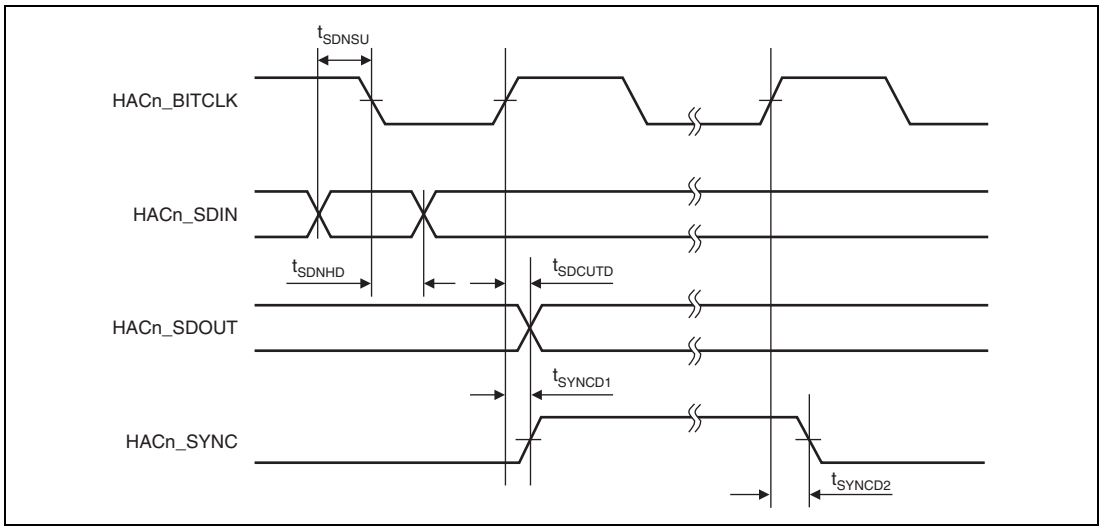


Figure 34.60 HAC Interface Module Signal Timing

34.3.16 SSI Interface Module Signal Timing

Table 34.29 SSI Interface Module Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Max.	Unit	Remarks	Figure
Output cycle time	t_{OSCK}	40	710	ns	Output	
Input cycle time	t_{ISCK}	80	3300	ns	Input	
Input high level width/input low level width	t_{IHC}/t_{ILC}	65	—	ns	Input	34.61
Output high level width/output low level width	T_{OHC}/t_{OLC}	65	—	ns	Output	
SCK output rise time	t_{RC}	—	60	ns	Output	
SDATA output delay time	t_{DTR}	—	50	ns	Transmit	34.62, 34.63
SDATA/WS input setup time	t_{SR}	15	—	ns	Receive	34.64, 34.65
SDATA/WS input hold time	t_{HTR}	5	—	ns	Receive	

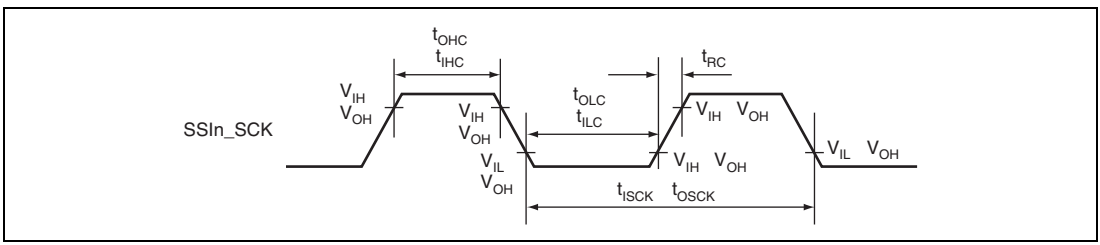


Figure 34.61 SSI Clock Input/Output Timing

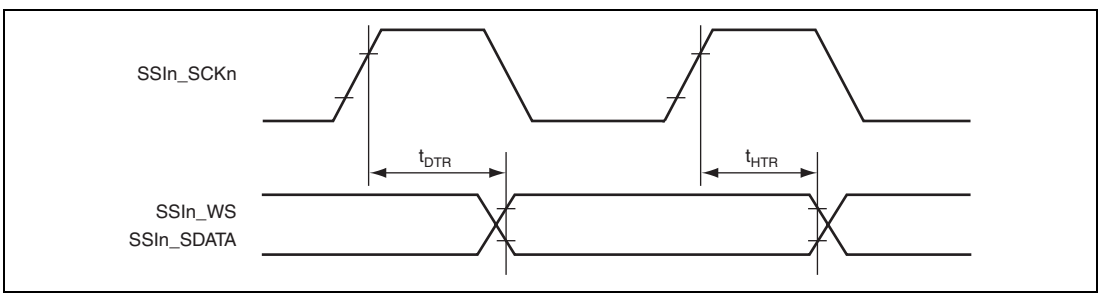


Figure 34.62 SSI Transmission Timing (1)

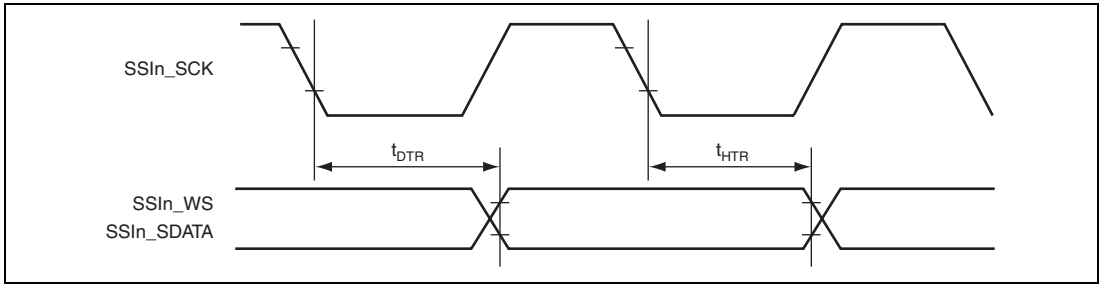


Figure 34.63 SSI Transmission Timing (2)

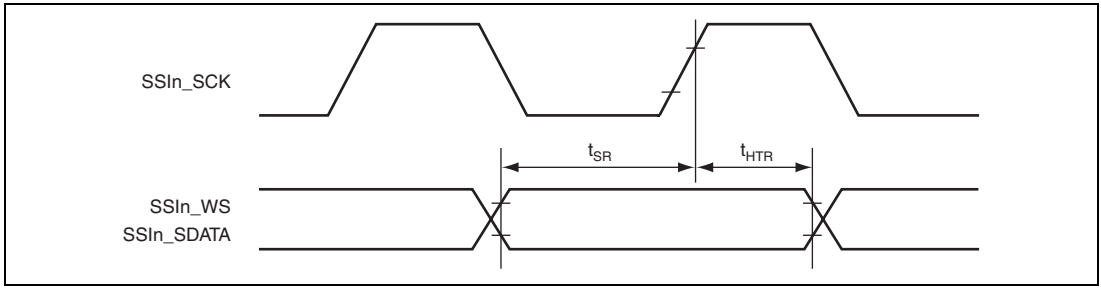


Figure 34.64 SSI Reception Timing (1)

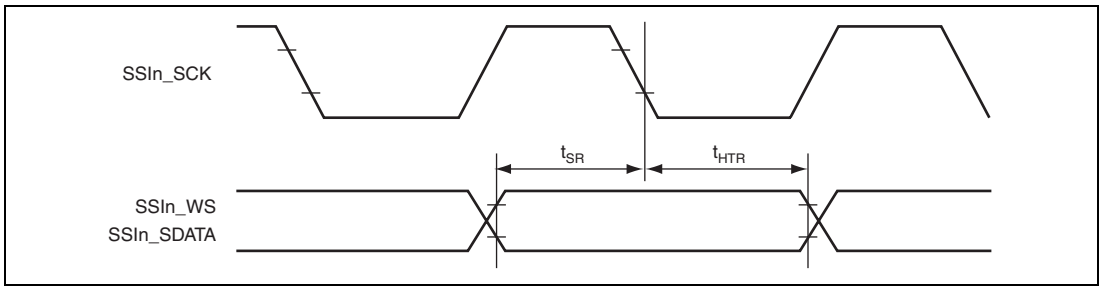


Figure 34.65 SSI Reception Timing (2)

34.3.17 I²C Signal Timing

Table 34.30 I²C Bus Interface Module Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
SCL frequency	t_{ICYC}	—	—	400	kHz	34.66
SCL low level time	t_{LOW}	$1/t_{ICYC}$–100	—	—	ns	
SCL high level time	t_{HIGH}	600	—	—	ns	
SCL/SDA fall time	t_{ICF}	—	—	250	ns	
SDA bus free time	t_{ICBF}	1300	—	—	ns	
SCL start condition hold time	t_{ICH}	600	—	—	ns	
SCL resend start condition setup time	t_{ICS}	600	—	—	ns	
SDA stop condition setup time	t_{ICST}	600	—	—	ns	
SDA setup time	t_{DAS}	100	—	—	ns	
SDA hold time	t_{ICDH}	0	—	900	ns	

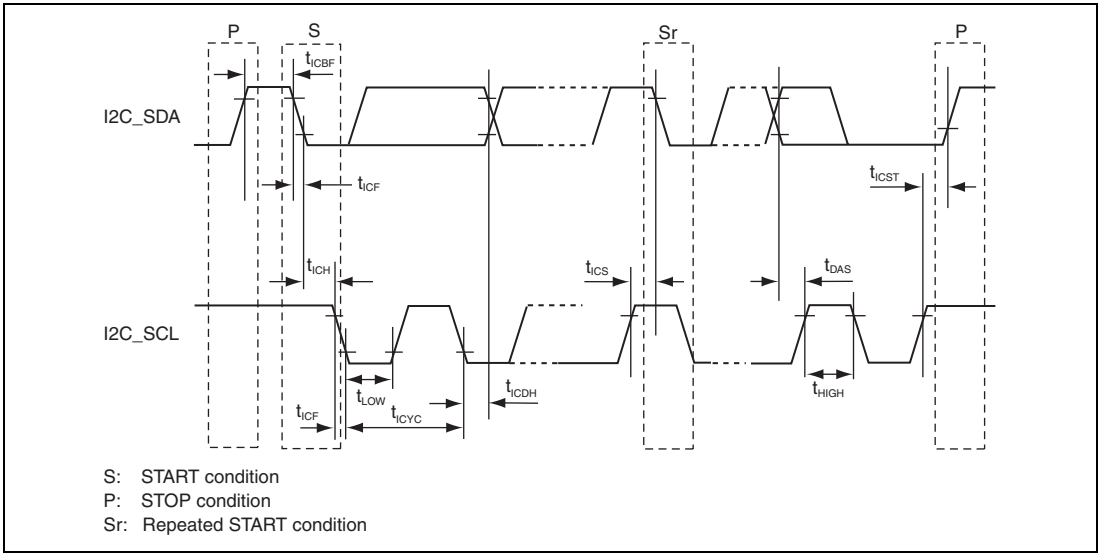


Figure 34.66 I²C Bus Interface Module Signal Timing

34.3.18 GPIO Signal Timing

Table 34.31 GPIO Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Max.	Unit	Figure
GPIO output delay time	t_{IOPD}	—	12	ns	34.67

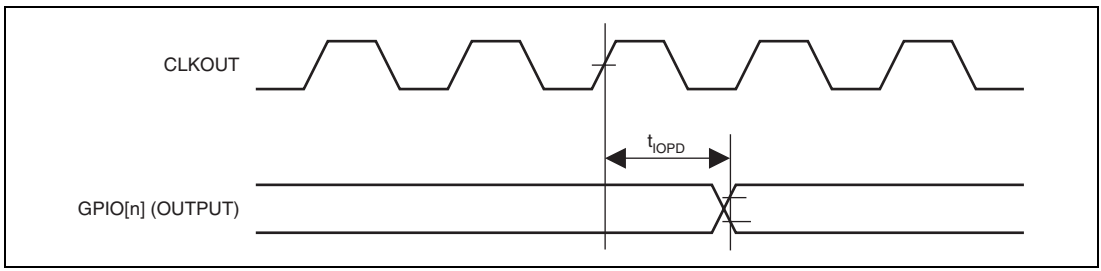


Figure 34.67 GPIO Signal Timing

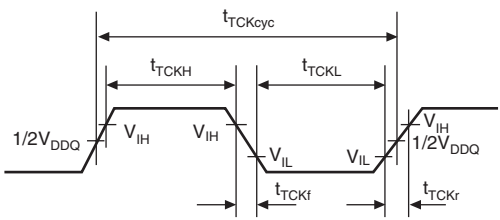
34.3.19 H-UDI Module Signal Timing

Table 34.32 H-UDI Module Signal Timing

Conditions: V_{CCQ}= 3.0 to 3.6 V, V_{DD}= 1.25 V, T_a= –20 to +85°C, C_L= 30 pF

Item	Symbol	Min.	Typ.	Max.	Unit	Figure	Remarks
Input clock cycle	t_{TCKcyc}	50	—	—	ns	34.68, 34.70	
Input clock pulse width (high)	t_{TCKH}	15	—	—	ns	34.68	
Input clock pulse width (low)	t_{TCKL}	15	—	—	ns		
Input clock rise time	t_{TCKr}	—	—	10	ns		
Input clock fall time	t_{TCKf}	—	—	10	ns		
$\overline{\text{ASEBRK}}$ setup time	$t_{ASEBRKS}$	10	—	—	t_{cyc}	34.69	
$\overline{\text{ASEBRK}}$ hold time	$t_{ASEBRKH}$	1	—	—	ms		
TDI/TMS setup time	t_{TDIS}	15	—	—	ns	34.70	
TDI/TMS hold time	t_{TDIH}	15	—	—	ns		
TDO data delay time	t_{TDO}	0	—	12	ns		
ASE-PINBRK pulse width	t_{PINBRK}	2	—	—	t_{Pcyc}	34.71	

Notes: 1. During a boundary scan, t_{TCKcyc} is the period corresponding to a frequency of 10 MHz, i.e. 0.1 μs.
2. t_{cyc} is the period of one CKIO clock cycle.
3. t_{Pcyc} is the period of one peripheral clock (Pck) cycle.



Note: When the clock is input from the TCK pin.

Figure 34.68 TCK Input Timing

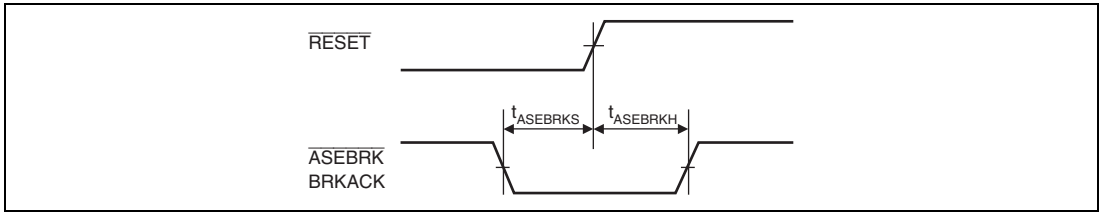


Figure 34.69 $\overline{\text{RESET}}$ Hold Timing

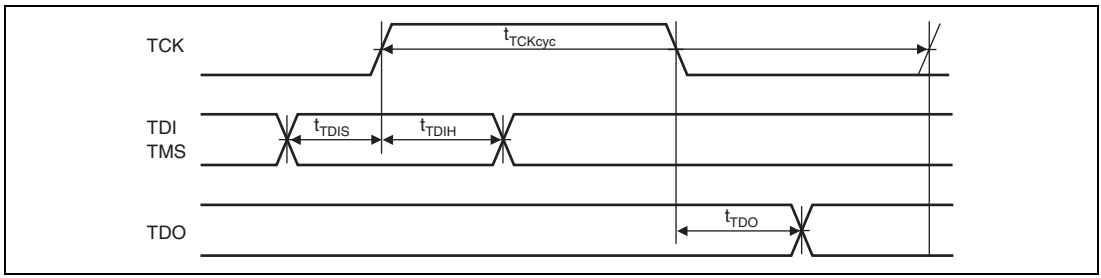


Figure 34.70 H-UDI Data Transfer Timing

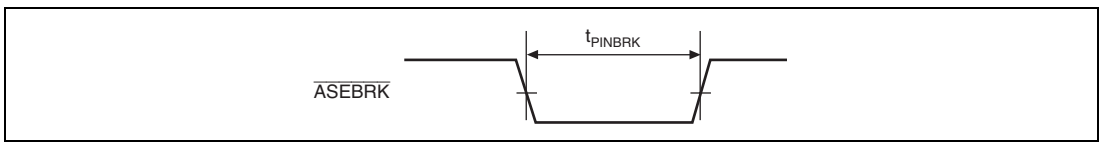


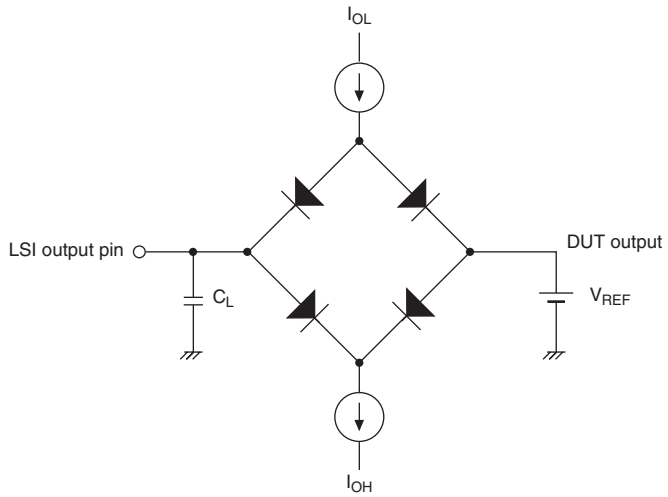
Figure 34.71 Pin Break Timing

34.4 AC Characteristic Test Condition (except DDR, USB, PCI pins)

The AC characteristic test conditions are as follows :

- Input/output signal reference level : V_{CCQ} /2
- Input pulse level : V_{SSQ} to V_{CCQ}
- Input rise/fall time : 1 ns

The output load circuit is shown in figure 34.72



Notes: 1. C_L is the total value, including the capacitance of the test jig, etc.
The capacitance of each pin is set to 30 pF.
2. I_{OL} and I_{OH} values are as shown in table 34.9, Permissible Output Currents.

Figure 34.72 Output Load Circuit

34.5 AC Characteristic Test Condition (DDR pin)

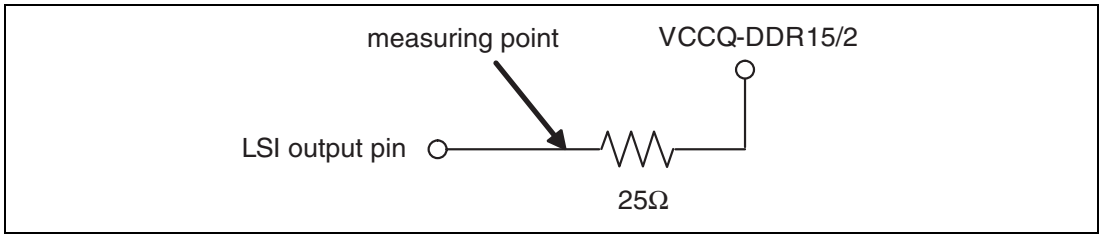


Figure 34.73 Output Load Equivalent Circuit (DDR pin)

34.6 AC Characteristic Test Condition (PCI pin)

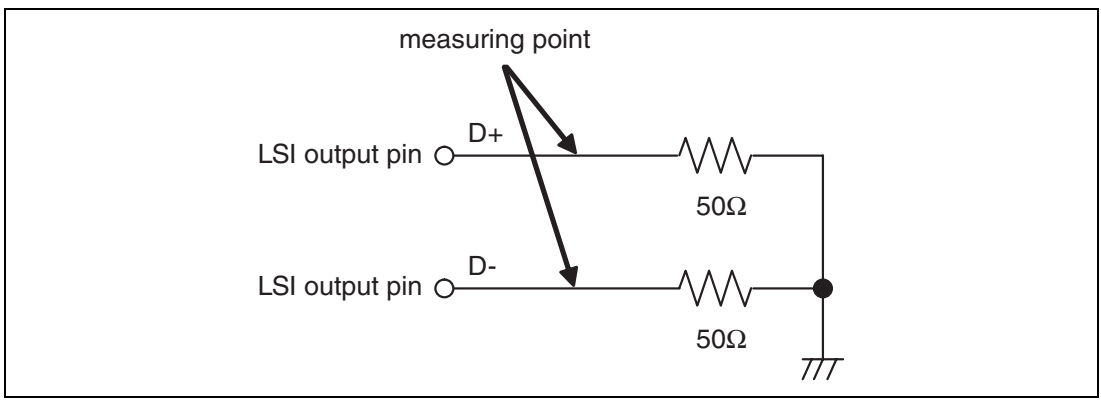


Figure 34.74 Output Load Equivalent Circuit (PCI pin)

34.7 AC Characteristic Test Condition (USB High speed pin)

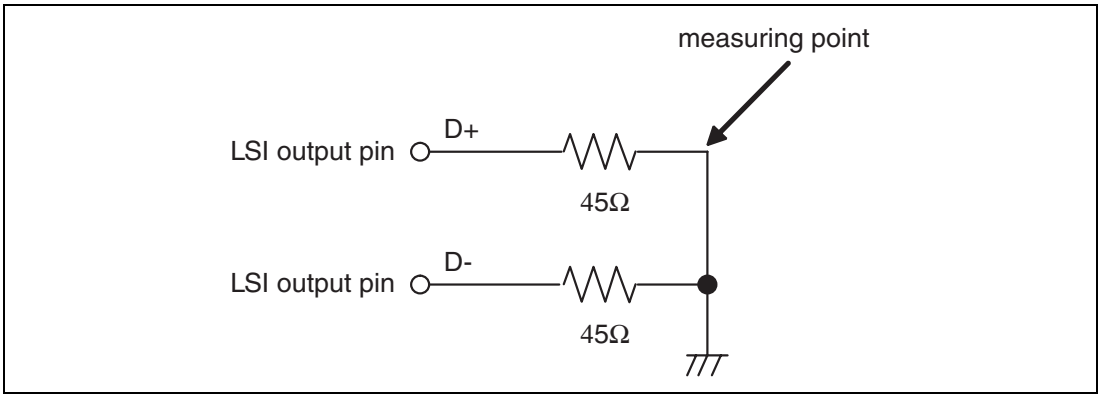


Figure 34.75 Output Load Equivalent Circuit (USB pin)

A. SH-4A Extended Functions (PVR.VER* = H'40)

This section gives an overview of the SH-4A multiprocessor-capable extended functions (PVR.VER = H'40). Details are provided in the subsequent sections. The SH-4A extended functions are available in products for which the value of the VER bits in the processor version register (PVR) is H'40 or higher, but not in products for which the value is H'30 or H'20. Before attempting to use the SH-4A multiprocessor-capable extended functions, confirm the value of the VER bits.

Note: The VER bit is bit 23 to 16 of the PVR.

A.1 Overview

The LSI with the SH-4A extended functions is functionally upward compatible with the SH-4A.

Table A.1 lists the SH-4A extended functions (PVR.VER = H'40).

Table A.1 SH-4A Extended Functions (PVR.VER = H'40)

Item		Features
Memory management unit (MMU) (some material in this section added and changed)		• A mode for avoiding synonym problems has been added for handling of 4-Kbyte pages when they are used. • The conditions for counting up MMUCR.URC have been changed. • The SNM, MCP, and CCD bits have been added to CCR.
Caches (some material in this section added and changed)		• Coherency control for operand caches of multiple CPUs has been added. • The function of invalidating instruction caches of multiple CPUs has been added. • A secondary cache (option) has been added.
On-chip memory (some material in this section added and changed)	IL memory (ILRAM)	• Physical addresses have been changed. • Access from the CPU/FPU via the SuperHyway with a physical address has been added. • Light sleep mode that can be accessed from the SuperHyway bus master to the IL memory while CPU/FPU stops has been added.
	OL memory (OLRAM)	• Physical addresses have been changed. • Access from the CPU/FPU via the SuperHyway with a physical address has been added. • Light sleep mode that can be accessed from the SuperHyway bus master to the OL memory while CPU/FPU stops has been added.
Instruction descriptions (some material in this section changed)		• Operation of the MOVCO data transfer instruction has been changed. • Operation of the MOVLI data transfer instruction has been changed. • Operation of the TAS logic-operation instruction has been changed. • Operation of the ICBI, OCBI, OCBP, and OCBWB instructions has been changed.

A.2 Memory Management Unit (MMU)

A.2.1 Changes to the MMU

The changes to the MMU are as follows:

(1) A mode that uses hardware to avoid synonym problems has been added for handling of 4-Kbyte pages when they are used.
(2) A mode of extending ASID to 16 bits has been added.
(3) The PTEAEX register has been newly added.
(4) The AEX bit has been added to MMUCR.
(5) The SNM, MCP, CCD, and IBE bits have been added to CCR.

A.2.2 Register Descriptions

Functions of the PTEH, MMUCR and CCR registers have been added and changed. Also, the PTEAEX register has been newly added.

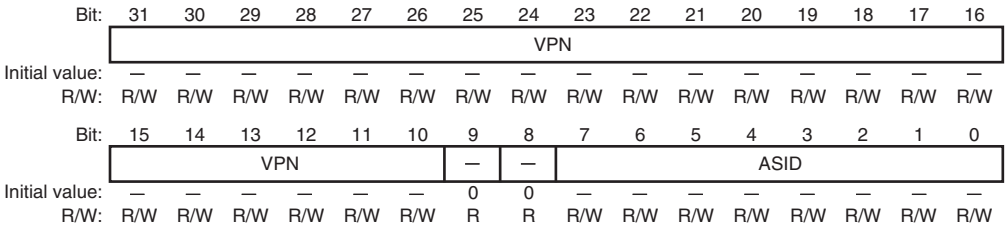
Table A.2 Register Configuration

Register Name	Abbr.	R/W	P4 Address*	Area 7 Address*	Size
Page table entry high register	PTEH	R/W	H'FF00 0000	H'1F00 0000	32
MMU control register	MMUCR	R/W	H'FF00 0010	H'1F00 0010	32
Cache control register	CCR	R/W	H'FF00 001C	H'1F00 001C	32
Page table entry address space identification extension register	PTEAEX	R/W	H'FF00 007C	H'1F00 007C	32

Note: * P4 is the address when virtual address space P4 area is used. Area 7 is the address when physical address space area 7 is accessed by using the TLB.

(1) Page Table Entry High Register (PTEH)

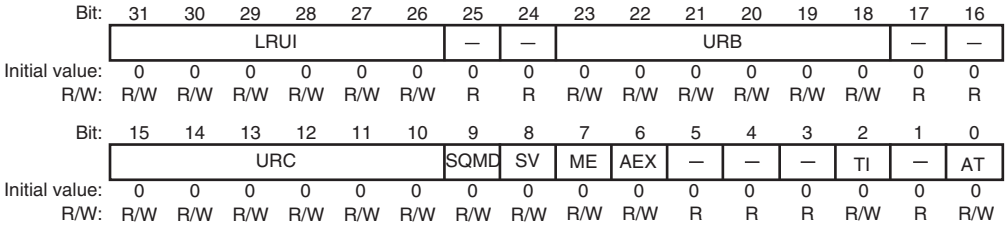
The function corresponding to the ASID bits has been changed.



Bit	Bit Name	Initial Value	R/W	Description
31 to 10	VPN	Undefined	R/W	Virtual page number
9, 8	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
7 to 0	ASID	Undefined	R/W	Address space identifier in 8-bit ASID compatible mode. In 16-bit ASID extended mode, the read value is undefined and writing to this bit is ignored.

(2) MMU Control Register (MMUCR)

The AEX bit has been added to select from 8-bit ASID compatible mode and 16-bit ASID extended mode. To change the value in the AEX bit, set the TI bit to 1 to invalidate the settings in the ITLB and the UTLB.



Bit	Bit Name	Initial Value	R/W	Description
31 to 26	LRUI	All 0	R/W	Least Recently Used ITLB These bits indicate the ITLB entry to be replaced. The LRU (least recently used) method is used to decide the ITLB entry to be replaced in the event of an ITLB miss. The entry to be purged from the ITLB can be confirmed using the LRUI bits. LRUI is updated by means of the algorithm shown below. x means that updating is not performed. 000xxx: ITLB entry 0 is used 1xx00x: ITLB entry 1 is used x1x1x0: ITLB entry 2 is used xx1x11: ITLB entry 3 is used xxxxxx: Other than above When the LRUI bit settings are as shown below, the corresponding ITLB entry is updated by an ITLB miss. Ensure that values for which "Setting prohibited" is indicated below are not set at the discretion of software. After a power-on or manual reset, the LRUI bits are initialized to 0, and therefore a prohibited setting is never made by a hardware update. x means "don't care". 111xxx: ITLB entry 0 is updated 0xx11x: ITLB entry 1 is updated x0x0x1: ITLB entry 2 is updated xx0x00: ITLB entry 3 is updated Other than above: Setting prohibited
25, 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 18	URB	All 0	R/W	These bits indicate the UTLB entry boundary at which replacement is to be performed. They are valid only when URB ≠ 0.
17, 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 10	URC	All 0	R/W	These bits serve as a random counter for indicating the UTLB entry for which replacement is to be performed with an LDTLB instruction.

Bit	Bit Name	Initial Value	R/W	Description
9	SQMD	0	R/W	Store Queue Mode Specifies the right of access to the store queues. 0: User/privileged access is possible 1: Privileged access is possible (address error exception in case of user access)
8	SV	0	R/W	Single Virtual Memory Mode/Multiple Virtual Memory Mode Switching When this bit is changed, ensure that 1 is also written to the TI bit. 0: Multiple virtual memory mode 1: Single virtual memory mode
7	ME	0	R/W	TLB Extended Mode Switching 0: TLB compatible mode 1: TLB extended mode
6	AEX	0	R/W	ASID Extended Mode Switching 0: 8-bit ASID compatible mode 1: 16-bit ASID extended mode
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TI	0	R/W	TLB Invalidate Writing 1 to this bit invalidates (clears to 0) all valid UTLB/ITLB bits. This bit is always read as 0.
1	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	AT	0	R/W	Address Translation Enable This bit enables or disables the MMU. 0: MMU is disabled 1: MMU is enabled MMU exceptions are not generated when the AT bit is 0. In the case of software that does not use the MMU, the AT bit should be cleared to 0.

(3) Cache Control Register (CCR)

The SNM, MCP, and CCD bits have been added to CCR to enable hardware to avoid synonym problems. Also, the IBE bit has been added in order for ICBI instructions to have an effect on other CPUs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	IBE	—	—	—	—	—	SNM	MCP	CCD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	L2IE	ICI	—	—	ICE	—	—	—	L2OE	OCI	CB	WT	OCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	IBE	0	R/W	ICBI Instruction Broadcast Determines whether or not to broadcast ICBI instructions to other CPUs. This bit is enabled only when ICE and CCD are 1 and 0, respectively. 0: The IC of the CPU that executed the ICBI instruction is only invalidated. 1: The ICs of all the CPUs whose CCDs are set to 0 are invalidated
23 to 19	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
18	SNM	0	R/W	Handling of Synonym Problems Specifies whether to take hardware measures for synonym problems. When SNM is set to 0 by a program that might cause synonym problems, operation is not guaranteed. 0: No hardware measures are taken 1: Hardware measures are taken For details of synonym problems, see A.2, Memory Management Unit (MMU).

Bit	Bit Name	Initial Value	R/W	Description
17	MCP	0	R/W	Mixed Coherency Protocols This bit is set when different coherency protocols are to be used for the same physical address space (valid only when CCD = 0). Set this bit to 1 to use different coherency protocols, regardless of whether a single CPU or multiple CPUs use them. When different coherency protocols are used with MCP = 0, operation is not guaranteed. 0: The same coherency protocol is set 1: Different coherency protocols are set
16	CCD	0	R/W	Cache Coherency Disable Enables or disables cache coherency control for multiple CPUs. 0: Cache coherency control is enabled 1: Cache coherency control is disabled
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	L2IE	0	R/W	Instruction Secondary Cache Enable 0: Secondary cache is not used for instruction fetch. 1: Secondary cache is used for instruction fetch.
11	ICI	0	R/W	IC Invalidation When 1 is written to this bit, the V bit of all IC entries is cleared to 0. This bit is always read as 0.
10, 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ICE	0	R/W	IC Enable Specifies whether to use the IC. Note, however, when address translation is performed, the IC cannot be used unless the C bit in the page management information is also 1. 0: IC is not used 1: IC is used
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	L2OE	0	R/W	Operand Secondary Cache Enable 0: Secondary cache is not used for operand access. 1: Secondary cache is used for operand access.
3	OCI	0	R/W	OC Invalidation When 1 is written to this bit, the V bit of all OC entries is cleared to 0. This bit is always read as 0.
2	CB	0	R/W	Cache Coherency Control 1 Indicates the cache coherency protocol for the P1 area in 29-bit address mode. This bit is invalid in 32-bit address mode, and indicates the value of the WT bit of PMB entries. • When cache coherency control is disabled (CCD = 1) 0: EI (write-through) protocol 1: MEI (copy-back) protocol • When cache coherency control is enabled (CCD = 0) 0: ESI (write-through) protocol 1: MESI (copy-back) protocol
1	WT	0	R/W	Cache Coherency Control 0 Indicates the cache coherency protocol for the P0, U0, and P3 areas. When address translation is performed, this bit is invalidated, and the value of the WT bit in the page management information has priority. • When cache coherency control is disabled (CCD = 1) 0: MEI (copy-back) protocol 1: EI (write-through) protocol • When cache coherency control is enabled (CCD = 0) 0: MESI (copy-back) protocol 1: ESI (write-through) protocol

Bit	Bit Name	Initial Value	R/W	Description
0	OCE	0	R/W	OC Enable Specifies whether to use the OC. Note, however, when address translation is performed, the OC cannot be used unless the C bit in the page management information is also 1. 0: OC is not used 1: OC is used

(4) Page Table Entry Address Space Identification Extension Register (PTEAEX)

In 16-bit ASID extended mode, the number of the currently executed process is stored in ASID in the PTEAEX register by software. ASID is not updated by hardware. ASID is recorded in the UTLB by a LDTLB instruction.

PTEAEX rewriting should be performed by a program in the P1 or P2 area. After PTEAEX has been updated, execute one of the following two methods before an access (including an instruction fetch) to the P0, P3, U0, or store queue area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the P0, P3, or U0 area.
2. Execute the ICBI instruction for any address (including non-cacheable area).

Note: Please rewrite PTEAEX in P1 area or P2 area except the last 64 bytes at 32-bit address mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ASID															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
15 to 0	ASID	Undefined	R	Address Space Identifier. Specifies the ASID in 16-bit ASID extended mode. This bit is not accessible in 8-bit ASID compatible mode.

A.2.3 MMU Functions

In 16-bit ASID extended mode (MMUCR.AEX = 1), the address space identifier is extended from 8 bits (ASID[7:0]) to 16 bits (ASID[15:0]).

MMU Instruction (LDTLB)

A TLB load instruction (LDTLB) is used to record a UTLB entry. When a LDTLB instruction is issued, the SH-4A copies the contents of PTEH and PTEL into the UTLB entry specified by the URC bits (the contents of PTEA and PTEAEX are also copied in TLB extended mode and 16-bit ASID extended mode, respectively).

Figures A.1 and A.2 show the operation of a LDTLB instruction in 16-bit ASID extended mode.

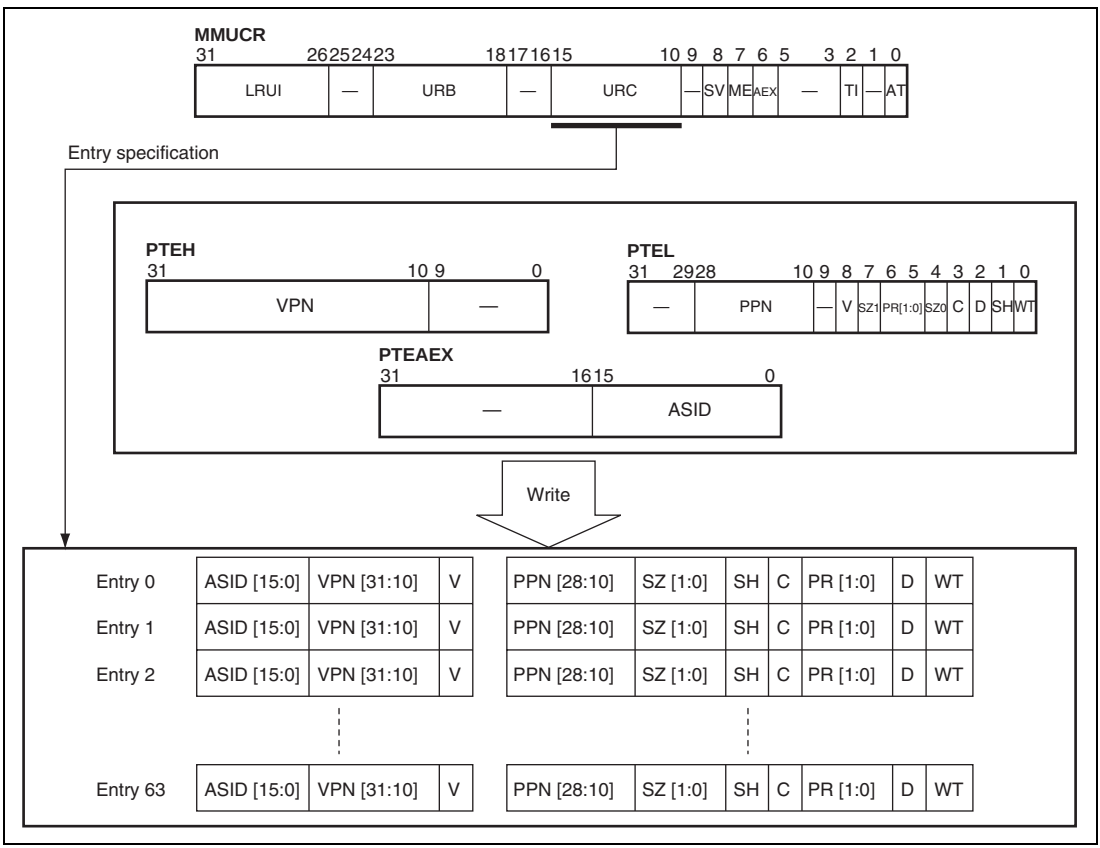


Figure A.1 Operation of LDTLB Instruction (in 16-bit ASID Extended and TLB Compatible Mode)

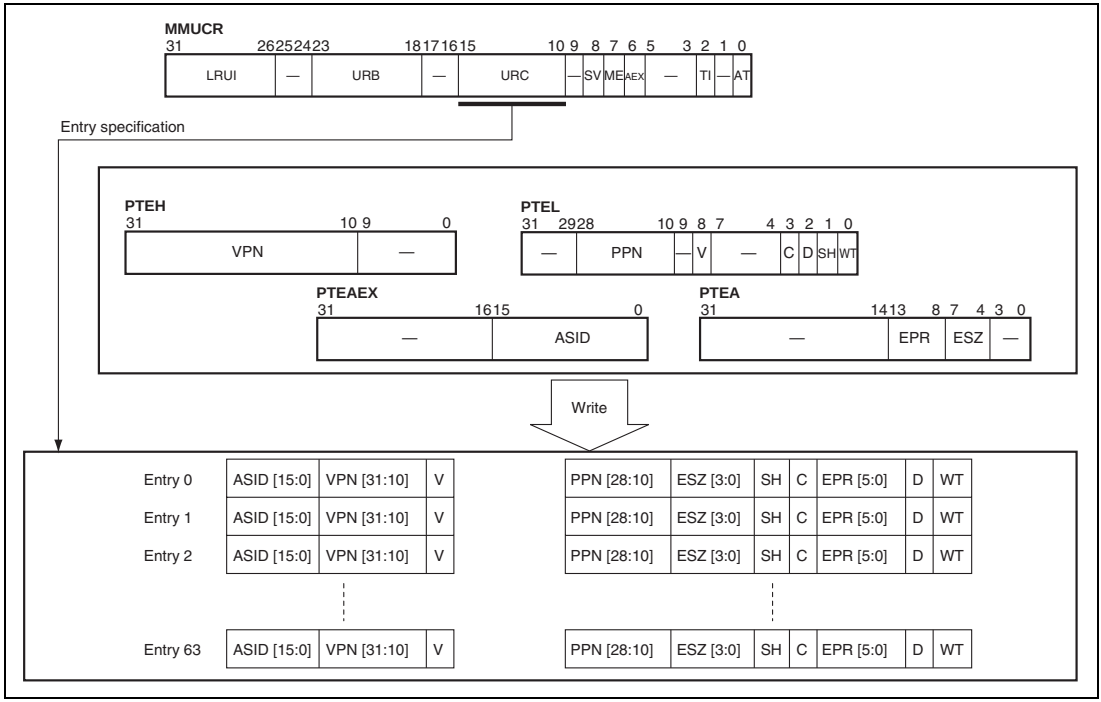


Figure A.2 Operation of LDTLB Instruction (in 16-bit ASID Extended and TLB Extended Mode)

(5) Avoiding Synonym Problems

When 1- or 4-Kbyte pages are recorded in TLB entries, a synonym problem may arise. The problem is that, when a number of virtual addresses are mapped onto a single physical address, the same physical address data is recorded in a number of cache entries, and it becomes impossible to guarantee data integrity. This problem does not occur with the instruction TLB and instruction cache because data is only read in these cases. In the SH-4A, entry specification is performed using bits 12 to 5 of the virtual address in order to achieve fast operand cache operation. However, bits 12 to 10 of the virtual address in the case of a 1-Kbyte page, and bit 12 of the virtual address in the case of a 4-Kbyte page, are subject to address translation. As a result, bits 12 to 10 of the physical address after translation may differ from bits 12 to 10 of the virtual address.

When CCR.CCD = 0 and CCR.SNM = 1, the hardware avoids synonym problems on 4-Kbyte pages. Avoiding synonym problems on 1-Kbyte page or 4-Kbyte page with CCR.CCD = 1 or CCR.SNM = 0 becomes the responsibility of the software.

Consequently, the following restrictions apply to the recording of address translation information in UTLB entries.

1. When address translation information whereby a number of 1-Kbyte page UTLB entries are translated into the same physical address is recorded in the UTLB, ensure that the VPN[12:10] values are the same.
2. When address translation information whereby a number of 4-Kbyte page UTLB entries are translated into the same physical address is recorded in the UTLB, take either of the following actions.
   — Ensure that the VPN[12] value is the same.
   — Set CCR.CCD = 0 and CCR.SNM = 1.
3. Do not use 1-Kbyte page UTLB entry physical addresses with UTLB entries of a different page size.
4. When CCR.CCD = 1 or CCR.SNM = 0, do not use 4-Kbyte page UTLB entry physical addresses with UTLB entries of a different page size. When CCR.CCD = 0 and CCR.SNM = 1, 4-Kbyte page UTLB entry physical addresses can be used with UTLB entries of a different page size.

The above restrictions apply only when performing accesses that use the cache.

Note: When multiple items of address translation information use the same physical memory to provide for future expansion of the SuperH RISC engine family, ensure that the VPN[20:10] values are the same. Also, do not use the same physical address for address translation information of different page sizes.

A.2.4 Memory-Mapped TLB Configuration

ITLB Address Array (16-bit ASID Extended Mode)

In 16-bit ASID extended mode (MMUCR.AEX = 1), the name of the ITLB address array is changed to ITLB address array 1, and ITLB address array 2 is added. Also, ASID[15:0] is accessible. In this mode, the ASID bits in ITLB address array 1 are reserved, and the value to be written should be 0. In 8-bit ASID compatible mode (MMUCR.AEX = 0), ITLB address array 2 is not accessible, and operation is not guaranteed if an attempt to access it is made.

(a) ITLB Address Array 1

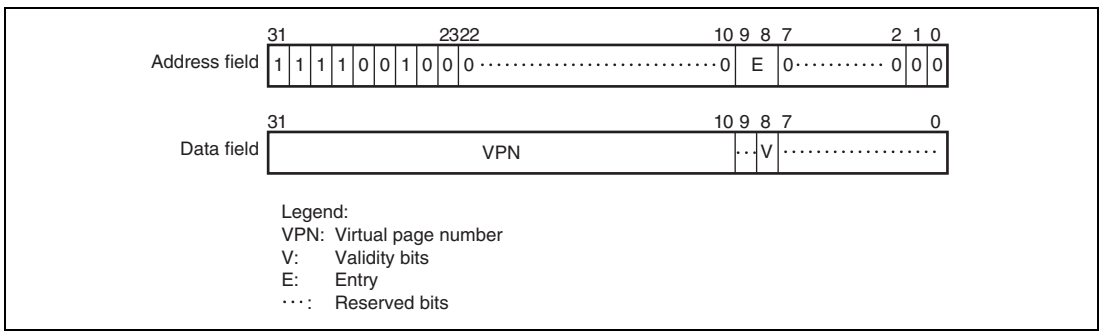


Figure A.3 Memory-Mapped ITLB Address Array 1 (16-bit ASID Extended Mode)

(b) ITLB Address Array 2

ITLB address array 2 is allocated to the addresses H'F280 0000 to H'F2FF FFFF in the P4 area. An access to the address array 2 requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and the ASID to be written to the address array 2 is specified in the data field.

In the address field, bits [31:23] have the value H'F28 indicating ITLB address array 2 and the entry is specified by bits [9:8].

In the data field, bits [15:0] indicate ASID[15:0].

The following two kinds of operation can be used on ITLB address array 2:

1. ITLB address array 2 read

   ASID[15:0] are read into the data field from the ITLB entry corresponding to the entry set in the address field.

2. ITLB address array 2 write

   ASID[15:0] specified in the data field are written to the ITLB entry corresponding to the entry set in the address field.

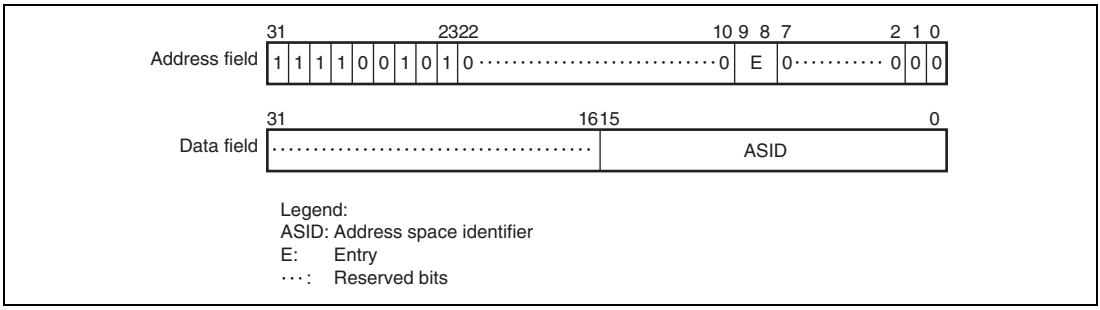


Figure A.4 Memory-Mapped ITLB Address Array 2 (16-bit ASID Extended Mode)

(6) UTLB Address Array (16-bit ASID Extended Mode)

In 16-bit ASID extended mode (MMUCR.AEX = 1), the name of the UTLB address array is changed to UTLB address array 1, and UTLB address array 2 is added. Also, ASID[15:0] is accessible. In this mode, the ASID bits in UTLB address array 1 are reserved, and the value to be written to these bits should be 0. In 8-bit ASID compatible mode (MMUCR.AEX = 0), UTLB address array 2 is not accessible, and operation is not guaranteed if an attempt to access it is made.

(a) UTLB Address Array 1

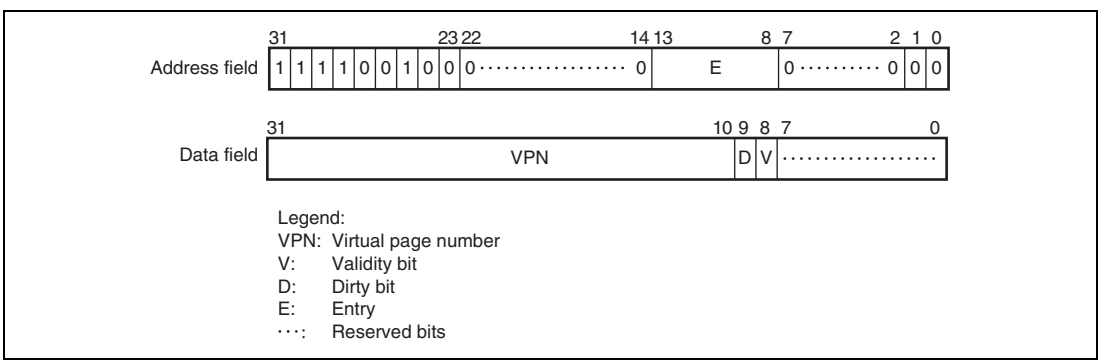


Figure A.5 Memory-Mapped UTLB Address Array 1 (16-bit ASID Extended Mode)

(b) UTLB Address Array 2

UTLB address array 2 is allocated to the addresses H'F680 0000 to H'F6FF FFFF in the P4 area. An access to address array 2 requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and the ASID to be written to address array 2 is specified in the data field.

In the address field, bits [31:23] have the value H'F68 indicating UTLB address array 2 and the entry is specified by bits [13:8].

In the data field, bits [15:0] indicate the ASID.

The following two kinds of operation can be used on UTLB address array 2:

1. UTLB address array 2 read

   ASID[15:0] are read into the data field from the UTLB entry corresponding to the entry set in the address field.
2. UTLB address array 2 write

   ASID[15:0] specified in the data field are written to the UTLB entry corresponding to the entry set in the address field.

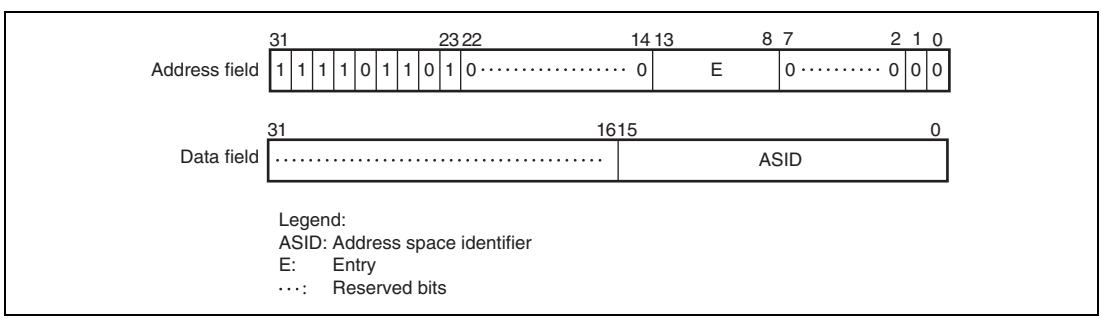


Figure A.6 Memory-Mapped UTLB Address Array 2 (16-bit ASID Extended Mode)

A.3 Caches

The SH-4A has an on-chip 32-Kbyte instruction cache (IC) for instructions and an on-chip 32-Kbyte operand cache (OC) for data.

Note: The bit positions, shown in figures A.7 to A.11, indicating entries and ways vary depending on the cache size. The table below lists the cache sizes and their respective bit positions for ways and entries. The ways are applied in figures A.8 to A.11. The entries are applied in figures A.7 to A.11.

Cache Size	Way	Entry
32 Kbytes	Bit[14:13]	Bit[12:5]

A.3.1 Changes to the Caches

- Operand caches between multiple CPUs are coherently controlled by hardware.
- The ICBI instruction has been changed for coherently controlling instruction caches between multiple CPUs.
- The L2IE bit and L2OE bit have been added to the CCR register to control the secondary cache.

A.3.2 Features

The features of the caches are given in table A.3.

Table A.3 Cache Features

Item	Instruction Cache	Operand Cache
Capacity	32-Kbyte cache	32-Kbyte cache
Type	4-way set-associative, virtual address index/physical address tag	4-way set-associative, virtual address index/physical address tag
Line size	32 bytes	32 bytes
Entries	256 entries/way	256 entries/way
Write method	—	Copy-back/write-through selectable
Replacement method	LRU (least-recently-used) algorithm	LRU (least-recently-used) algorithm
Cache coherency method between CPUs	Coherency is ensured by software	Coherency enable or disable is selectable for each CPU. • MESI (copy-back) protocol • ESI (write-through) protocol

With the multiprocessor-capable SH-4A, a sharing bit (S bit) for indicating that a cache line is shared between multiple CPUs is added to the operand cache address array. Figure A.7 shows the configuration of the operand cache.

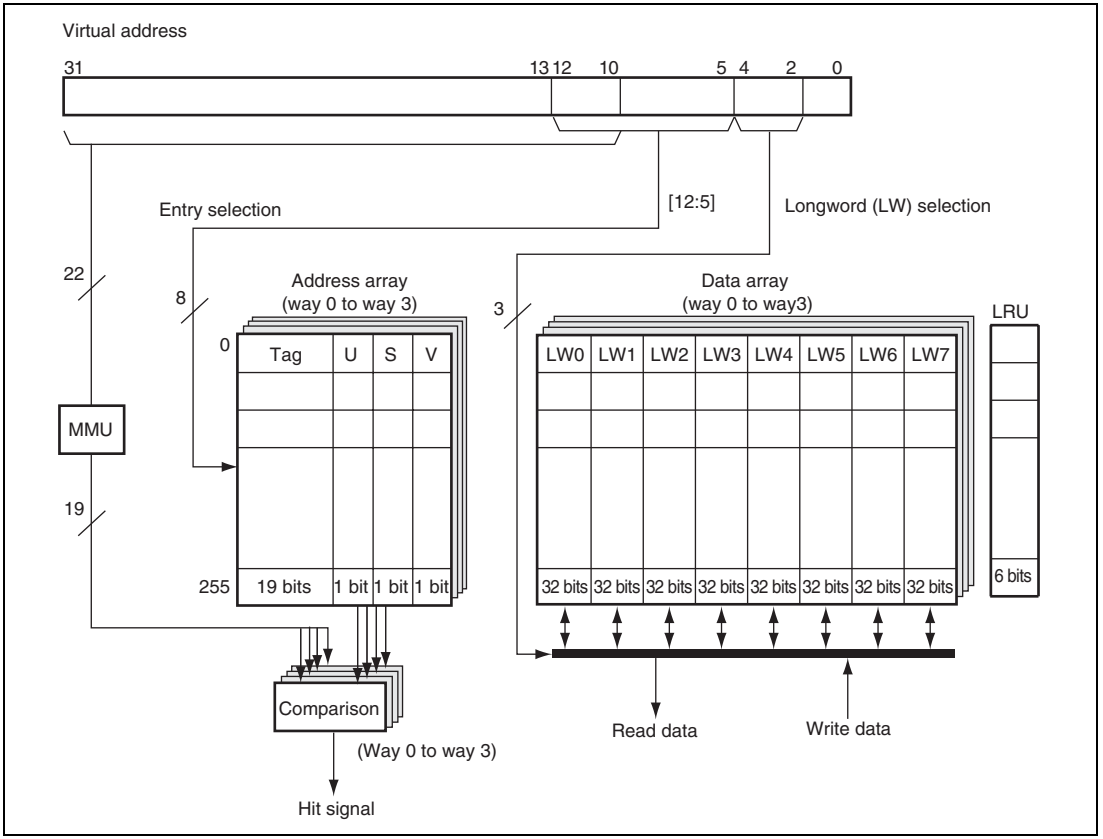


Figure A.7 Configuration of Operand Cache (OC)

Tag

V bit (validity bit)

U bit (dirty bit)

S bit (sharing bit)

This bit is set to 1 when cached data is also cached in the operand cache of another CPU for which coherency control has been enabled. That is, the S bit indicates the sharing of external memory data cached between CPUs. This bit is not set to 1 when coherency control has been disabled. In addition, unless the S bit or U bit is modified by a memory allocation access, neither of the following conditions ever occurs:
"V bit = 1, S bit = 1 and U bit = 1" or "V bit = 0 and S bit = 1 (Ubit: don't care)"

(c) Data field

(d) LRU field

A.3.3 Register Descriptions

The cache coherency disable (CCD) bit, the mixed coherency protocols (MCP) bit, and the handling of synonym problems (SNM) bit have been added to the CCR register. Also, the IBE bit has been added to allow broadcasting ICBI instructions to other CPUs. The L2FC bit and the L2E bit have been deleted from the RAMCR register.

Table A.4 Register Configuration

Register Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Size
Cache control register	CCR	R/W	H'FF00 001C	H'1F00 001C	32
On-chip memory control register	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32

Note: * The P4 addresses are for the P4 area in the virtual address space. The area 7 addresses are accessed from area 7 in the physical address space by means of the TLB.

Table A.5 Register States in Each Processing State

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sleep
Cache control register	CCR	H'0000 0000	H'0000 0000	Retained
On-chip memory control register	RAMCR	H'0000 0000	H'0000 0000	Retained

(1) Cache Control Register (CCR)

CCR controls the cache operating mode, the cache write mode, and invalidation of all cache entries.

CCR modifications must be made by a program in the non-cacheable P2 area. After CCR has been updated, execute one of the following three methods before an access (including an instruction fetch) to the cacheable area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the cacheable area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating CCR, the specific instruction sequence is not needed. However, note that the CPU processing performance will be lowered

because the instruction fetch is performed again for the next instruction after CCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Family. Therefore, it is recommended that the method 1 or 2 be used for compatibility with the future SuperH Family.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	IBE	—	—	—	—	—	SNM	MCP	CCD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	L2IE	ICI	—	—	ICE	—	—	—	L2OE	OCI	CB	WT	OCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	IBE	0	R/W	ICBI Instruction Broadcast Determines whether or not to broadcast ICBI instructions to other CPUs. This bit is enabled only when ICE and CCD are 1 and 0, respectively. 0: The IC of the CPU that executed the ICBI instruction is only invalidated. 1: The ICs of all the CPUs whose CCDs are set to 0 are invalidated
23 to 19	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
18	SNM	0	R/W	Handling of Synonym Problems Specifies whether to take hardware measures for synonym problems. When SNM is set to 0 by a program that might cause synonym problems, operation is not guaranteed. 0: No hardware measures are taken 1: Hardware measures are taken For details of synonym problems, see A.2, Memory Management Unit (MMU).

Bit	Bit Name	Initial Value	R/W	Description
17	MCP	0	R/W	Mixed Coherency Protocols This bit is set when different coherency protocols are to be used for the same physical address space (valid only when CCD = 0). Set this bit to 1 to use different coherency protocols, regardless of whether a single CPU or multiple CPUs use them. When different coherency protocols are used with MCP = 0, operation is not guaranteed. 0: The same coherency protocol is set 1: Different coherency protocols are set
16	CCD	0	R/W	Cache Coherency Disable Enables or disables cache coherency control for multiple CPUs. 0: Cache coherency control is enabled 1: Cache coherency control is disabled
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	L2IE	0	R/W	Instruction Secondary Cache Enable 0: Secondary cache is not used for instruction fetch. 1: Secondary cache is used for instruction fetch.
11	ICI	0	R/W	IC Invalidation When 1 is written to this bit, the V bit of all IC entries is cleared to 0. This bit is always read as 0.
10, 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ICE	0	R/W	IC Enable Specifies whether to use the IC. Note, however, when address translation is performed, the IC cannot be used unless the C bit in the page management information is also 1. 0: IC is not used 1: IC is used
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	L2OE	0	R/W	Operand Secondary Cache Enable 0: Secondary cache is not used for operand access. 1: Secondary cache is used for operand access.
3	OCI	0	R/W	OC Invalidation When 1 is written to this bit, the V bit of all OC entries is cleared to 0. This bit is always read as 0.
2	CB	0	R/W	Cache Coherency Control 1 Indicates the cache coherency protocol for the P1 area in 29-bit address mode. This bit is invalid in 32-bit address mode, and indicates the value of the WT bit of PMB entries. • When cache coherency control is disabled (CCD = 1) 0: EI (write-through) protocol 1: MEI (copy-back) protocol • When cache coherency control is enabled (CCD = 0) 0: ESI (write-through) protocol 1: MESI (copy-back) protocol
1	WT	0	R/W	Cache Coherency Control 0 Indicates the cache coherency protocol for the P0, U0, and P3 areas. When address translation is performed, this bit is invalidated, and the value of the WT bit in the page management information has priority. • When cache coherency control is disabled (CCD = 1) 0: MEI (copy-back) protocol 1: EI (write-through) protocol • When cache coherency control is enabled (CCD = 0) 0: MESI (copy-back) protocol 1: ESI (write-through) protocol

Bit	Bit Name	Initial Value	R/W	Description
0	OCE	0	R/W	OC Enable Specifies whether to use the OC. Note, however, when address translation is performed, the OC cannot be used unless the C bit in the page management information is also 1. 0: OC is not used 1: OC is used

(2) On-Chip Memory Control Register (RAMCR)

RAMCR controls the number of ways in the IC and OC, and also controls the prediction of IC ways.

RAMCR modifications must be made by a program in the non-cacheable P2 area. After RAMCR has been updated, execute one of the following two methods before an access (including an instruction fetch) to the cacheable area, IL memory area, or OL memory area.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the non-cacheable area, IL memory area, or OL memory area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating RAMCR, the specific instruction sequence is not needed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after RAMCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Family. Therefore, it is recommended that the method 1 or 2 be used for compatibility with the future SuperH Family.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RMD	RP	IC2W	OC2W	ICWPD	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	RMD	0	R/W	On-Chip Memory Access Mode For details, see section 9.4, On-Chip Memory Protective Functions in the SH-4A Extended Function Software Manual.
8	RP	0	R/W	On-Chip Memory Protection Enable For details, see section 9.4, On-Chip Memory Protective Functions in the SH-4A Extended Function Software Manual.
7	IC2W	0	R/W	IC Two-Way Mode 0: IC is a four-way operation 1: IC is a two-way operation
6	OC2W	0	R/W	OC Two-Way Mode 0: OC is a four-way operation 1: OC is a two-way operation
5	ICWPD	0	R/W	IC Way Prediction Inhibit Specifies whether to use prediction of IC ways. 0: The instruction cache predicts IC ways 1: The instruction cache does not predict IC ways
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

A.3.4 Operand Cache Operation

(1) Read Operation (Coherency Control Disabled)

When the operand cache (OC) is enabled (OCE = 1 in CCR), coherency control is disabled (CCD = 1 in CCR), and data is read from a cacheable area, the cache operates as follows:

1. The tag, V bit, U bit, S bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU.
   - If there is a way whose tag matches and its V bit is 1, see No. 3.
   - If there is no way whose tag matches and its V bit is 1 and the U bit of the way that is selected to replace using the LRU bits is 0, see No. 4.
   - If there is no way whose tag matches and its V bit is 1 and the U bit of the way that is selected to replace using the LRU bits is 1, see No. 5.
3. Cache hit

   The data indexed by virtual address bits [4:0] is read from the data field of the cache line on the hit way in accordance with the access size. Then the LRU bits are updated to indicate the hit way is the latest one.
4. Cache miss (no write-back)

   Data is read into the cache line on the way, which is selected to replace, from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. When the corresponding data arrives in the cache, the read data is returned to the CPU. While the remaining data on one cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 is written to the V bit, and 0 is written to the S and U bits. Then the LRU bits are updated to indicate the way is the latest one.

5. Cache miss (with write-back)

   The tag and data field of the cache line on the way that is selected to replace are saved in the write-back buffer. Then data is read into the cache line on the way that is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. When the corresponding data arrives in the cache, the read data is returned to the CPU. While the remaining data on one cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 is written to the V bit, and 0 is written to the S and U bits. The LRU bits are also updated to indicate the way is the latest one. The data in the write-back buffer is then written back to external memory.

(2) Read Operation (Coherency Control Enabled)

When the operand cache (OC) is enabled (OCE = 1 in CCR), coherency control is enabled (CCD = 0 in CCR), and data is read from a cacheable area, the cache operates as follows.

1. The tag, V bit, U bit, S bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU.
   - If there is a way whose tag matches and its V bit is 1, see No. 3.
   - If there is no way whose tag matches and its V bit is 1 and the U bit of the way that is selected to replace using the LRU bits is 0, see No. 4.
   - If there is no way whose tag matches and its V bit is 1 and the U bit of the way that is selected to replace using the LRU bits is 1, see No. 5.
3. Cache hit

   The data indexed by virtual address bits [4:0] is read from the data field of the cache line on the hit way in accordance with the access size. Then the LRU bits are updated to indicate the hit way is the latest one.
4. Cache miss (no write-back)

   The cache of a CPU with coherency control enabled (CCD = 0 in CCR) is searched.
   - If there is a CPU that is caching data and the U bit is 1, see No. 6.
   - If there is a CPU that is caching data and the U bit is 0 or there is no such CPU, see No. 7.
5. Cache miss (with write-back)

   The data on the cache line is written back to external memory. At the same time, the cache of a CPU with coherency control enabled (CCD = 0 in CCR) is searched.
   - If there is a CPU that is caching data and the U bit is 1, see No. 6.

— If there is a CPU that is caching data and the U bit is 0 or there is no such CPU, see No. 7.

6. Snoop operation (when another CPU has dirty data)

   Zero is written to the U bit and 1 is written to the S bit of the corresponding line on the corresponding way of the CPU that is caching data. Data is read from the CPU, and is then written to the cache line on the way to be replaced while written back to external memory. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. When the corresponding data arrives in the cache, the read data is returned to the CPU. While the remaining data on one cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 is written to the S and V bits, and 0 is written to the U bit. Then the LRU bits are updated to indicate the way is the latest one.

7. Snoop operation (when no other CPU has dirty data)

   When there is a CPU that is caching data, 1 is written to the S bit of the corresponding line on the corresponding way of the CPU. Then data is read into the cache line on the way, which is selected to replace, from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. When the corresponding data arrives in the cache, the read data is returned to the CPU. While the remaining data on one cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 and 0 are written to the V bit and the U bit, respectively. When there is a CPU caching data, 1 is written to the S bit. Otherwise, 0 is written. Then the LRU bits are updated to indicate the way is the latest one.

(3) Prefetch Operation (Coherency Control Disabled)

When the operand cache (OC) is enabled (OCE = 1 in CCR), coherency control is disabled (CCD = 1 in CCR), and data is prefetched from a cacheable area, the cache operates as follows.

1. The tag, V bit, U bit, S bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU.
   — If there is a way whose tag matches and its V bit is 1, see No. 3.
   — If there is no way whose tag matches and the V bit is 1, and the U bit of the way that is selected to replace using the LRU bits is 0, see No. 4.
   — If there is no way whose tag matches and the V bit is 1, and the U bit of the way that is selected to replace using the LRU bits is 1, see No. 5.
3. Cache hit

   The LRU bits are updated to indicate the hit way is the latest one.

4. Cache miss (no write-back)

   Data is read into the cache line on the way, which is selected to replace, from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In a prefetch operation, the CPU does not wait for the data to arrive. Instead, it can execute the next processing while one cache line of data is being read. When reading of one line of data is completed, the tag is recorded at the physical address, 1 is written to the V bit, and 0 is written to the S and U bits on the way. Then the LRU bits are updated to indicate the way is the latest one.

5. Cache miss (with write-back)

   The tag and data field of the cache line on the way that is selected to replace are saved in the write-back buffer. Then data is read into the cache line on the way that is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In the prefetch operation the CPU does not wait the data arrives. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 is written to the V bit, and 0 is written to the S and U bits on the way. The LRU bits are also updated to indicate the way is the latest one. The data in the write-back buffer is then written back to external memory.

(4) Prefetch Operation (Coherency Control Enabled)

When the operand cache (OC) is enabled (OCE = 1 in CCR), coherency control is enabled (CCD = 0 in CCR), and data is prefetched from a cacheable area, the cache operates as follows.

1. The tag, V bit, U bit, S bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU.
   - If there is a way whose tag matches and its V bit is 1, see No. 3.
   - If there is no way whose tag matches and the V bit is 1, and the U bit of the way that is selected to replace using the LRU bits is 0, see No. 4.
   - If there is no way whose tag matches and the V bit is 1, and the U bit of the way that is selected to replace using the LRU bits is 1, see No. 5.
3. Cache hit

   The LRU bits are updated to indicate the hit way is the latest one.

4. Cache miss (no write-back)

   The cache of a CPU with coherency control enabled (CCD = 0 in CCR) is searched.

   — If there is a CPU that is caching data and the U bit is 1, see No. 6.

   — If there is a CPU that is caching data and the U bit is 0 or there is no such CPU, see No. 7.

5. Cache miss (with write-back)

   The data on the cache line is written back to the external memory. At the same time, the cache of a CPU with coherency control enabled (CCD = 0 in CCR) is searched.

   — If there is a CPU that is caching data and the U bit is 1, see No. 6.

   — If there is a CPU that is caching data and the U bit is 0 or there is no such CPU, see No. 7.

6. Snoop operation (when another CPU has dirty data)

   Zero is written to the U bit and 1 is written to the S bit of the corresponding line on the corresponding way of the CPU that is caching data. Data is read from the CPU, and is then written to the cache line on the way to be replaced while written back to external memory. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In the prefetch operation the CPU does not wait the data arrives. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 is written to the S and V bits, and 0 is written to the U bit. Then the LRU bits are updated to indicate the way is the latest one.

7. Snoop operation (when no other CPU has dirty data)

   When there is a CPU that is caching data, 1 is written to the S bit of the corresponding line on the corresponding way of the CPU. Then data is read into the cache line on the way, which is selected to replace, from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In the prefetch operation the CPU does not wait the data arrives. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 and 0 are written to the V bit and the U bit, respectively. When there is a CPU caching data, 1 is written to the S bit. Otherwise, 0 is written. Then the LRU bits are updated to indicate the way is the latest one.

(5) Write Operation (Coherency Control Disabled)

When the operand cache (OC) is enabled (OCE = 1 in CCR), coherency control is disabled (CCD = 1 in CCR), and data is written to a cacheable area, the cache operates as follows.

1. The tag, V bit, U bit, S bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU, and according to the attributes of the area.
   - If there is a way whose tag matches and its V bit is 1, see No. 3 for copy-back and No. 4 for write-through.
   - If there is no way whose tag matches and its V bit is 1 and the U bit of the way that is selected to replace using the LRU bits is 0, see No. 5 for copy-back and No. 7 for write-through.
   - If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 1, see No. 6 for copy-back and No. 7 for write-through.
3. Cache hit (copy-back)

   A data write in accordance with the access size is performed for the data indexed by virtual address bits [4:0] in the data field on the hit way. Then 1 is written to the U bit. The LRU bits are updated to indicate the way is the latest one.
4. Cache hit (write-through)

   A data write in accordance with the access size is performed for the data indexed by virtual address bits [4:0] in the data field on the hit way. A write is also performed to external memory corresponding to the virtual address. Then the LRU bits are updated to indicate the way is the latest one. In this case, the U bit is not updated.
5. Cache miss (copy-back, no write-back)

   A data write in accordance with the access size is performed for the data indexed by virtual address bits [4:0] in the data field on the way that is selected to replace. Then the data, excluding the cache-missed data that is written already, is read into the cache line on the way that is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 is written to the V and U bits, and 0 is written to the S bit on the way. Then the LRU bits are updated to indicate the way is the latest one.

6. Cache miss (copy-back, with write-back)

   The tag and data field of the cache line on the way that is selected to replace are saved in the write-back buffer. Then a data write in accordance with the access size is performed for the data indexed by virtual address bits [4:0] in the data field on the way that is selected to replace. Then the data, excluding the cache-missed data that is written already, is read into the cache line on the way that is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 is written to the V and U bits, and 0 is written to the S bit on the way. The LRU bits are also updated to indicate the way is the latest one. The data in the write-back buffer is then written back to external memory.

7. Cache miss (write-through)

   A write of the specified access size is performed to the external memory corresponding to the virtual address. In this case, a write to the cache is not performed. None of the tag, V bit, U bit, S bit, and LRU bits is updated.

(6) Write Operation (Coherency Control Enabled)

When the operand cache (OC) is enabled (OCE = 1 in CCR), coherency control is enabled (CCD = 0 in CCR), and data is written to a cacheable area, the cache operates as follows.

1. The tag, V bit, U bit, S bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU, and according to the attributes of the area.
   - If there is a way whose tag matches and its V bit is 1 and S bit is 0, see No. 3 for MESI protocol and No. 4 for ESI protocol.
   - If there is a way whose tag matches and its V bit is 1 and S bit is 1, see No. 5 for MESI protocol and No. 6 for ESI protocol.
   - If there is no way whose tag matches and its V bit is 1 and the U bit of the way that is selected to replace using the LRU bits is 0, see No. 7 for MESI protocol and No. 9 for ESI protocol.
   - If there is no way whose tag matches and its V bit is 1 and the U bit of the way that is selected to replace using the LRU bits is 1, see No. 8 for MESI protocol and No. 9 for ESI protocol.

3. Cache hit (S bit = 0) (MESI protocol)

   A data write in accordance with the access size is performed for the data indexed by virtual address bits [4:0] in the data field on the hit way. Then 1 is written to the U bit. The LRU bits are updated to indicate the way is the latest one.

4. Cache hit (S bit = 0) (ESI protocol)

   A data write in accordance with the access size is performed for the data indexed by virtual address bits [4:0] in the data field on the hit way. A write is also performed to external memory corresponding to the virtual address. Then the LRU bits are updated to indicate the way is the latest one. In this case, the U bit is not updated.

5. Cache hit (S bit = 1) (MESI protocol)

   A data write in accordance with the access size is performed for the data indexed by virtual address bits [4:0] in the data field on the hit way. Then 1 is written to the U bit and 0 is witten to the S bit. The LRU bits are updated to indicate the way is the latest one. If another CPU for which coherency control is enabled (CCD = 0 in CCR) is caching data, 0 is written to the V and S bits of the corresponding line on the corresponding way of the CPU.

6. Cache hit (S bit = 1) (ESI protocol)

   A data write in accordance with the access size is performed for the data indexed by virtual address bits [4:0] in the data field on the hit way. A write is also performed to external memory corresponding to the virtual address. Also, 0 is written to the S bit to update the LRU bit so that the way is the latest one. In this case, the U bit is not updated. If another CPU for which coherency control is enabled (CCD = 0 in CCR) is caching data, 0 is written to the V and S bits of the corresponding line on the corresponding way of the CPU.

7. Cache miss (MESI protocol, no write-back)

   The cache of a CPU with coherency control enabled (CCD = 0 in CCR) is searched.

   — If there is a CPU that is caching data and the U bit is 1, see No. 10.

   — If there is a CPU that is caching data and the U bit is 0 or there is no such CPU, see No. 11.

8. Cache miss (MESI protocol, with write-back)

   The data on the cache line is written back to external memory. At the same time, the cache of a CPU with coherency control enabled (CCD = 0 in CCR) is searched.

   — If there is a CPU that is caching data and the U bit is 1, see No. 10.

   — If there is a CPU that is caching data and the U bit is 0 or there is no such CPU, see No. 11.

9. Cache miss (ESI protocol)

   A write of the specified access size is performed to the external memory corresponding to the virtual address. In this case, a write to the cache is not performed. None of the tag, V bit, U bit, S bit, and LRU bits is updated. If another CPU for which coherency control is enabled (CCD=0 in CCR) is caching data, 0 is written to the V and S bits of the corresponding line on the corresponding way of the CPU. (When different coherency protocols are used (MCP = 1 in CCR), another CPU may have dirty data. In this case, the dirty data is written back to external memory.

10. Snoop operation (when another CPU has dirty data)

    Zero is written to the V and U bits of the corresponding line on the corresponding way of the CPU that is caching data. Data is read from the CPU, and is then written to the cache line on the way to be replaced while written back to external memoryA data write in accordance with the access size is performed for the data indexed by virtual address bits [4:0] in the data field on the way that is selected to replace. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 is written to the V and U bits, and 0 is written to the S bit. Then the LRU bits are updated to indicate the way is the latest one.

11. Snoop operation (when no other CPU has dirty data)

    When there is a CPU that is caching data, 0 is written to the V bit of the corresponding line on the corresponding way of the CPU. A data write in accordance with the access size is performed for the data indexed by virtual address bits [4:0] in the data field on the way that is selected to replace. Then the data, excluding the cache-missed data that is written already, is read into the cache line on the way that is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. When the corresponding data arrives in the cache, the read data is returned to the CPU. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag is recorded at the physical address, 1 is written to the V and U bits, and 0 is written to the S bit. The LRU bits are also updated to indicate the way is the latest one.

(7) Cache Operation Instructions

Coherency between OCs of a CPU with coherency control enabled (CCD = 0 in CCR) is assured by hardware. However, coherency between OCs of a CPU with coherency control disabled (CCD = 1 in CCR) and between ICs of all CPUs should be assured by software. In addition, coherency between cache and external memory should be assured by software.

In the SH-4A, the following six instructions are supported for cache operations. For details of these instructions, see section 8, Instruction Descriptions, and the SH-4A Extended Functions Software Manual.

- Operand cache invalidate instruction: OCBI @Rn (Operand cache invalidation (no write-back))

  When coherency control is disabled (CCD = 1 in CCR) for the CPU that has issued the OCBI instruction, cache operation is performed for only the OC of the CPU. When coherency control is enabled (CCD = 0 in CCR) for the CPU that has issued the OCBI instruction, cache operation is performed for the OC of all CPUs for which coherency control is enabled.

  When a CPU issues the OCBI instruction to the OC address array area, cache operation is performed for the OC of the CPU regardless of coherency control.
- Operand cache purge instruction: OCBP @Rn (Operand cache invalidation (with write-back))

  When coherency control is disabled (CCD = 1 in CCR) for the CPU that has issued the OCBP instruction, cache operation is performed for only the OC of the CPU. When coherency control is enabled (CCD = 0 in CCR) for the CPU that has issued the OCBP instruction, cache operation is performed for the OC of all CPUs for which coherency control is enabled.

  When a CPU issues the OCBP instruction to the OC address array area, cache operation is performed for the OC of the CPU regardless of coherency control.
- Operand cache write-back instruction: OCBWB @Rn (Operand cache write-back)

  When coherency control is disabled (CCD = 1 in CCR) for the CPU that has issued the OCBWB instruction, cache operation is performed for only the OC of the CPU. When coherency control is enabled (CCD = 0 in CCR) for the CPU that has issued the OCBWB instruction, cache operation is performed for the OC of all CPUs for which coherency control is enabled.

  When a CPU issues the OCBWB instruction to the OC address array area, cache operation is performed for the OC of the CPU regardless of coherency control.

- Operand cache allocate instruction: MOVCA.L R0, @Rn (Operand cache allocation)

  When coherency control is disabled (CCD = 1 in CCR) for the CPU that has issued the MOVCA.L instruction, cache operation is performed for only the OC of the CPU. When coherency control is enabled (CCD = 0 in CCR) for the CPU that has issued the MOVCA.L instruction, the OC of all CPUs with coherency control enabled is searched. If the OC of a CPU other than the CPU that has issued the MOVCA.L instruction has a cache line, the line is invalidated and the OC of the CPU that has issued the MOVCA.L instruction is allocated.
- Instruction cache invalidate instruction: ICBI @Rn (Instruction cache invalidation)

  When ICBI instruction broadcast is enabled (IBE = 1 in CCR) and cache coherency control is enabled (CCD = 0 in CCR) for the CPU that has issued the ICBI instruction, cache operation is performed for the ICs of all the CPUs for which coherency control is enabled (CCD = 0 in CCR).
- Operand access synchronization instruction: SYNCO (Wait for data transfer completion)

  Cache operation is performed for only the bus access of the CPU that has issued the SYNCO instruction regardless of whether coherency control is disabled (CCD = 1 in CCR) or enabled (CCD = 0 in CCR) for the CPU.

Only the CPU with coherency control enabled (CCD = 0 in CCR) can receive PURGE and FLUSH transactions from the SuperHyway bus to control the operand cache coherency. Note that no CPU with coherency control disabled (CCD = 1 in CCR) can receive these transactions. These transactions from the SuperHyway bus are received simultaneously by all CPUs with coherency control enabled.

A.3.5 Memory-Mapped Cache Configuration

(1) IC Address Array

The IC address array is allocated to the addresses H'F000 0000 to H'F0FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the write tag and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F0 indicating the IC address array, and the way and the entry are specified by bits [14:13] and bits [12:5], respectively. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the IC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag and the V bit are indicated by bits [31:10] and bit [0], respectively.

The following three kinds of operation can be used on the IC address array:

1. IC address array read

   The tag and V bit are read into the data field from the IC entry corresponding to the way and entry set in the address field. When they are read, set the A bit in the address field to 0.
2. IC address array write (non-associative)

   The tag and V bit specified in the data field are written to the IC entry corresponding to the way and entry set in the address field. Set the A bit in the address field to 0.
3. IC address array write (associative)

   When 1 is written to the A bit in the address field, the tag in each way stored in the entry specified in the address field is compared with the tag specified in the data field. The way numbers of bits [14:13] are not used. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the ITLB. If the addresses match and the V bit in the way is 1, the V bit specified in the data field is written into the IC entry. In other cases, no operation is performed. This operation is used to invalidate a specific IC entry. If an ITLB miss occurs during address translation, or the comparison shows a mismatch, an exception is not generated, no operation is performed, and the write is not executed.

Note: This function may not be supported in the future SuperH Family. Therefore, it is recommended that the ICBI instruction be used to surely operate the IC by handling ITLB miss and reporting instruction TLB miss exception.

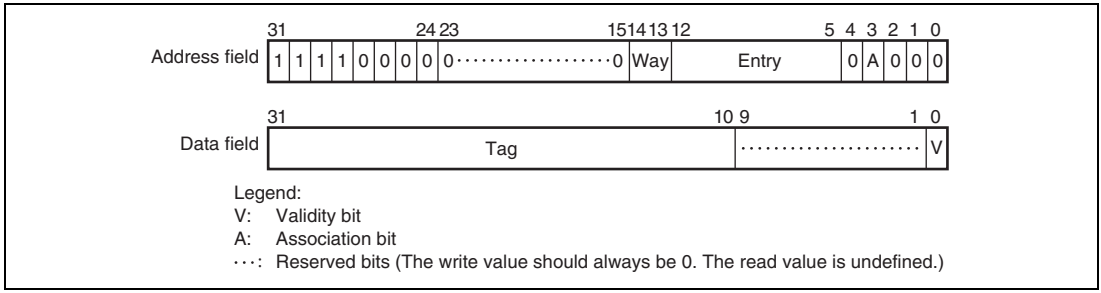


Figure A.8 Memory-Mapped IC Address Array

(2) IC Data Array

The IC data array is allocated to addresses H'F100 0000 to H'F1FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F1 indicating the IC data array, and the way and the entry are specified by bits [14:13] and bits [12:5], respectively. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

The data field is used for the longword data specification.

The following two kinds of operation can be used on the IC data array:

1. IC data array read

   Longword data is read into the data field from the data specified by the longword specification bits in the address field in the IC entry corresponding to the way and entry set in the address field.
2. IC data array write

   The longword data specified in the data field is written for the data specified by the longword specification bits in the address field in the IC entry corresponding to the way and entry set in the address field.

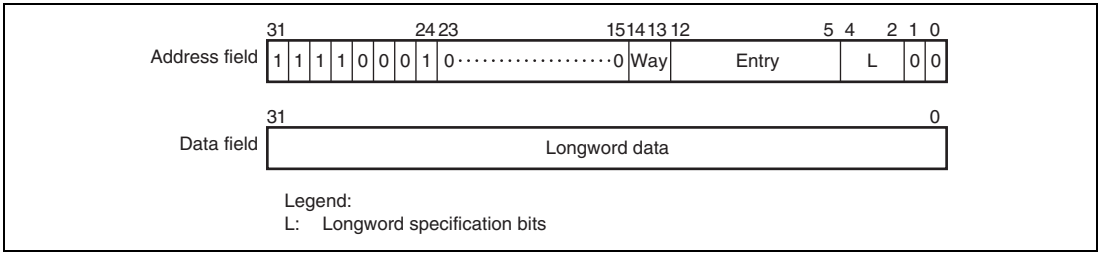


Figure A.9 Memory-Mapped IC Data Array

(3) OC Address Array

The OC address array is allocated to addresses H'F400 0000 to H'F4FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the write tag, S bit, U bit, and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F4 indicating the OC data array, and the way and the entry are specified by bits [14:13] and bits [12:5], respectively. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the OC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, [31:10] indicates the tag, and [2], [1], [0] indicates the S bit, U bit, and V bit respectively.

The following three kinds of operation can be used for the OC address array.

1. OC address array read

   The tag, S bit, U bit, and V bit are read into the data field from the OC entry corresponding to the way and entry set in the address field. Before reading these, clear the A bit in the address field to 0.

2. OC address array write (non-associative)

   The tag, S bit, U bit, and V bit specified in the data field are written to the OC entry corresponding to the way and entry set in the address field. Before writing these, set the A bit in the address field to 0. When a write is performed to a cache line whose U and V bits are 1, the cache line is written back and then the tag, S bit, U bit, and V bit specified in the data field are written.

3. OC address array write (associative)

   When a write is performed with the A bit in the address field set to 1, the tag in each way stored in the entry specified in the address field is compared with the tag specified in the data field. The way numbers of bits [14:13] in the address field are not used. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the UTLB. If the addresses match and the V bit in the way is 1, the S, U, and V bits specified in the data field are written to the OC entry. In other cases, no operation is performed. This operation is used to invalidate a specific OC entry. If the OC entry U bit is 1, and 0 is written to the V bit or to the U bit, write-back is performed. If a UTLB miss occurs during address translation, or the comparison shows a mismatch, an exception is not generated, no operation is performed, and the write is not executed.

Note: This function may not be supported in the future SuperH Series. Therefore, it is recommended that the OCBI, OCBP, or OCBWB instruction should be used to operate the OC definitely by reporting data TLB miss exception.

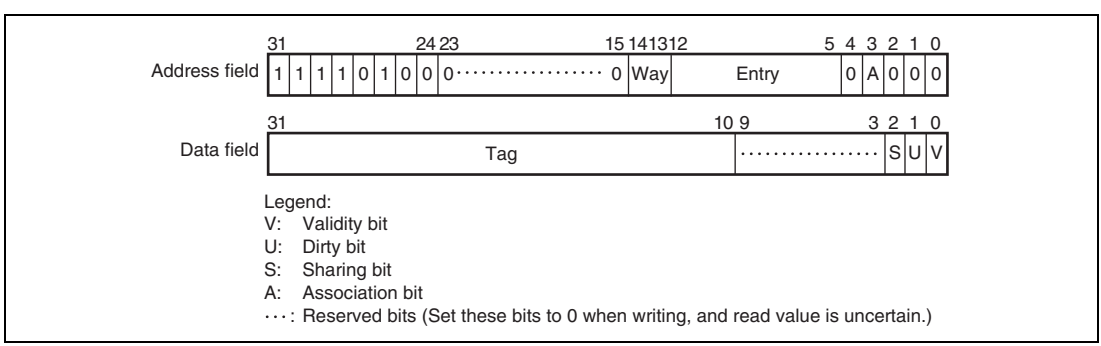


Figure A.10 Memory-Mapped OC Address Array

(4) OC Data Array

The OC data array is allocated to addresses H'F500 0000 to H'F5FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F5 indicating the OC data array, and the way and the entry are specified by bits [14:13] and bits [12:5], respectively. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

The data field is used for the longword data specification.

The following two kinds of operation can be used on the OC data array:

1. OC data array read

   Longword data is read into the data field from the data specified by the longword specification bits in the address field in the OC entry corresponding to the way and entry set in the address field.
2. OC data array write

   The longword data specified in the data field is written for the data specified by the longword specification bits in the address field in the OC entry corresponding to the way and entry set in the address field. This write does not set the U bit to 1 on the address array side.

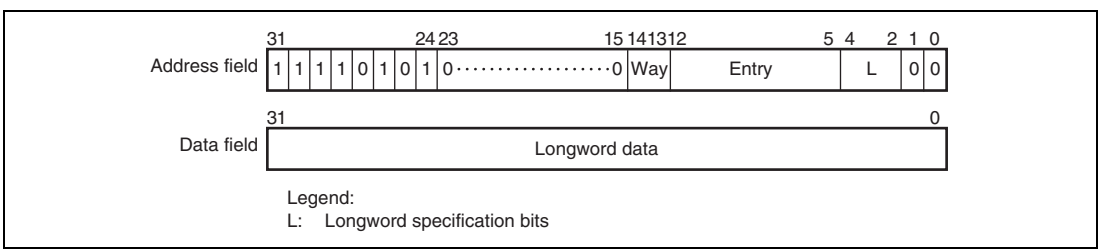


Figure A.11 Memory-Mapped OC Data Array

A.3.6 Usage Notes

(1) Sleep Mode

Before a CPU enters the sleep mode while coherency control is enabled for the CPU, all the entries of the OC and IC must be purged and invalidated, respectively. When a CPU enters the light sleep mode while coherency control is enabled for the CPU, it is not necessary for all the entries of the OC and IC to be purged and invalidated. In the light sleep mode, the coherency of the OC between the CPU and the other CPUs is maintained. When the ICBI instruction is executed while CCR.IBE is 1, IC invalidation is performed for CPUs that are in the light sleep mode.

(2) Share Data

In case of sharing data between the CPUs both of coherency control are disabled, or between the CPU which coherency control is enabled and the CPU which coherency control is disabled, a non-cacheable area must be used.

A.4 On-Chip Memory

A.4.1 Changes to the On-Chip Memory

- Physical addresses have been changed.
- Access from the CPU/FPU via the SuperHyway with a physical address has been added.
- A light sleep mode accessible from the SuperHyway bus master of the on-chip memory has been added.

A.4.2 Features

- Memory map:

  OL memory and IL memory are allocated to both the virtual address space and the physical address space shown in tables A.6 to A.7.

Note: The physical address space allocation is not uniform as listed below. It is according to the product.

An address in the virtual address space can be accessed from the P4 area (when MD = 1 in SR) or from the on-chip area (when MD = 0 in SR and RMD = 1 in RAMCR) according to CPU operating mode. When this address is used, the on-chip memory is always accessed through non-cacheable access. Only the on-chip memory of the self-CPU is accessible.

An address in the physical address space can be accessed from area U0, P0, P1, P2, or P3. When the address is used, whether the on-chip memory is accessed through cacheable access or non-cacheable access depends on the settings of the CCR register, MMUCR register, TLB, and PMB. The same address of the same on-chip memory can be accessed from all CPUs.

Table A.6 OL Memory Addresses

Address Space		Page	Memory Size 16 Kbytes
Virtual address		Page 0	H'E500E000 to H'E500FFFF
		Page 1	H'E5010000 to H'E5011FFF
Physical address	CPU 0	Page 0	H'1400E000 to H'1400FFFF
		Page 1	H'14010000 to H'14011FFF
	CPU 1	Page 0	H'1480E000 to H'1480FFFF
		Page 1	H'14810000 to H'14811FFF

Table A.7 IL Memory Addresses

Address Space		Memory Size
		8 Kbytes
Virtual address	Page 0	H'E5200000 to H'E5200FFF
	Page 1	H'E5201000 to H'E5201FFF
Physical address	CPU 0	H'14200000 to H'14201FFF
	CPU 1	H'14A00000 to H'14A01FFF

A.4.3 Operation

(1) Operand Access from the CPU or Access from the FPU

(a) OL memory

Operand access from the CPU and access from the FPU can be made from a virtual address directly via the operand bus or from a physical address via the SuperHyway bus.

As long as a conflict on the page does not occur, the OL memory is accessed via the operand bus in one cycle. Access to the OL memory of other CPUs is not possible.

Access via the SuperHyway bus includes both cacheable access, in which the OL memory data is cached into the OC, and non-cacheable access. In cacheable access, the coherency between the OC and OL memory should be guaranteed by software. For example, to read the area using the DMAC, which is accessed through cacheable write access, the coherency should be guaranteed by performing write-back with the OCBP or OCBWB instruction in advance or by issuing a PURGE or FLUSH transaction from the DMAC.

Whether operand access by the CPU is non-cacheable access or cacheable access is the same as in the case of external memory. For details, see A.2, Memory Management Unit (MMU).

(b) IL memory

Operand access from the CPU and access from the FPU can be made from a virtual address via the cache/RAM internal bus or from a physical address via the SuperHyway bus.

Access via the cache/RAM internal bus requires several cycles. Access to the IL memory of other CPUs is not possible.

Access via the SuperHyway bus includes both cacheable access, in which the IL memory data is cached into the OC, and non-cacheable access. In cacheable access, the coherency between the OC

and IL memory should be guaranteed by software. For example, to read the area using the DMAC, which is accessed through cacheable write access, the coherency should be guaranteed by performing write-back with the OCBP or OCBWB instruction in advance or by issuing a PURGE or FLUSH transaction from the DMAC.

Whether operand access by the CPU is non-cacheable access or cacheable access is the same as in the case of external memory. For details, see A.2, Memory Management Unit (MMU).

(2) Instruction Fetch Access from the CPU

(a) OL memory

The CPU is able to fetch instructions using either of the following methods: access to the OL memory from a virtual address via the cache/RAM internal bus or access to the OL memory from a physical address via the SuperHyway bus.

An instruction fetch via the cache/RAM internal bus requires several cycles. Access to the OL memory of other CPUs is not possible.

Access via the SuperHyway bus includes both cacheable access, in which the OL memory data is cached into the IC, and non-cacheable access. In cacheable access, the coherency between the IC and OL memory should be guaranteed by software. For example, when modifying a program running with cacheable access, invalidate the applicable part by using the ICBI instruction or set the ICI bit in CCR to 1 to completely invalidate the IC after modifying the program, and then branch to the corresponding program.

Whether an instruction fetch by the CPU is non-cacheable access or cacheable access is the same as in the case of external memory. For details, see A.2, Memory Management Unit (MMU).

(b) IL memory

The CPU is able to fetch instructions using either of the following methods: direct access to the IL memory from a virtual address via the instruction bus or access to the IL memory from a physical address via the SuperHyway bus.

Access via the instruction bus takes a single cycle as long as a conflict on the page does not occur. However, access to the IL memory of other CPUs is not possible.

Access via the SuperHyway bus includes both cacheable access, in which the IL memory data is cached into the IC, and non-cacheable access. In cacheable access, the coherency between the IC and IL memory should be guaranteed by software. For example, when modifying a program running with cacheable access, invalidate the applicable part by using the ICBI instruction or set

the ICI bit in CCR to 1 to completely invalidate the IC after modifying the program, and then branch to the corresponding program.

Whether an instruction fetch by the CPU is non-cacheable access or cacheable access is the same as in the case of external memory. For details, see A.2, Memory Management Unit (MMU).

A.4.4 Usage Notes

(1) On-Chip Memory Coherency

In order to allocate instructions in the on-chip memory, write an instruction to the on-chip memory, execute the following sequence, and then branch to the rewritten instruction.

- SYNCO
- ICBI @Rn

In this case, the target for the ICBI instruction can be any address (including an on-chip memory address) within the range where no address error exception occurs, and cache hit/miss is possible.

Do not access the same address directly from the cache/RAM internal bus and indirectly via the SuperHyway bus at the same time. If simulaneous accesses such as these are made, the on-chip memory coherency is not guaranteed.

(2) Sleep Mode

(a) OL memory, IL memory

The SuperHyway bus master module, such as DMAC, cannot access the OL memory or IL memory in sleep mode, but can access either in light sleep mode.

(3) Note on Using 32-Bit Address Extended Mode

Do not set any virtual address of the on-chip memory in the PPN field of UTLB, ITLB, or PMB. To access the on-chip memory using the address translation of MMU or PMB, be sure to set physical addresses of the on-chip memory in the PPN field. If a virtual address of the on-chip memory is set in the PPN field, operation is not guaranteed.

A.5 Secondary Cache

The multiprocessor-capable SH-4A incorporates a 256-Kbyte secondary cache that stores both instruction and data.

Note: The tag size and the bit positions, shown in figure A.12, indicating entries and ways vary depending on the cache size. The table below lists the cache sizes and their respective bit positions for ways, entries, and tag sizes.

Table A.8 Cache Size Respective Bit Position for Way and Entry, and Tag Size

Cache Size	Way	Entry	Tag Size
128 Kbytes	Bit[16:15]	Bit[14:5]	18 Bit (Bit[31:14])
256 Kbytes	Bit[17:16]	Bit[15:5]	17 Bit (Bit[31:15])

A.5.1 Features

The features of the secondary cache are listed in table A.9.

Table A.9 Secondary Cache Features

Item	Description
Capacity	256 Kbytes
Method	4-way set associative, physical address index/physical address tag
Line Size	32 bytes
Number of Entries	2048 entries per way
Write Method	Write-through
Replace Method	LRU (least recently used) algorithm
Operation Clock	System clock
Others	Shared between CPUs for which coherency control is enabled. Exclusively used by CPUs for which coherency control is disabled, or by CPUs for either of which coherency control is disabled. Selectable from IC/OC, IC only, and OC only. A half or all of the capacity can be used as a shared memory*.

Note: * Shared memory is allocated to both virtual address and physical address H'E4000000 to H'E403FFFF(256KB), H'E4000000 to H'E401FFFF(128KB) and can be accessed only that address as a RAM.

A.5.2 Register Descriptions

Bits for enabling/disabling the secondary cache have been added to the CCR register. Also, the L2CR register has been added to control the secondary cache. Each CPU has a CCR register. The L2CR register is shared by all the CPUs.

Table A.10 Register Configuration

Name	Abbr.	R/W	P4 Area Address*	Area 7 Address*	Size
Cache Control Register	CCR	R/W	H'FF00 001C	H'1F00 001C	32
Secondary Cache Control Register	L2CR	R/W	H'FBF0 0000	—	32

Note: * P4 area address is for using the P4 area in the virtual address space. Area 7 address is for an access from the area 7 in the physical address space by using the TLB.

Table A.11 Register State in Each Processing Mode

Name	Abbr.	Power-on Reset	Manual Reset	Sleep
Cache Control Register	CCR	H'0000 0000	H'0000 0000	Retained
Secondary Cache Control Register	L2CR	H'0000 0000	—	—

(1) Cache Control Register (CCR)

CCR controls the cache operating mode, the cache write mode, and invalidation of all cache entries.

CCR modifications must only be made by a program in the non-cacheable P2 area. After CCR has been updated, execute one of the following three methods before an access (including an instruction fetch) to the cacheable area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the cacheable area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating CCR, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after CCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH family. Therefore, it is recommended that the method 1 or 2 should be used for being compatible with the future SuperH family.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	IBE	—	—	—	—	—	SNM	MCP	CCD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	L2IE	ICI	—	—	ICE	—	—	—	L2OE	OCI	CB	WT	OCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
24	IBE	0	R/W	ICBI Instruction Broadcast Determines whether or not to broadcast ICBI instructions to other CPUs. This bit is enabled only when ICE and CCD are 1 and 0, respectively. 0: The IC of the CPU that executed the ICBI instruction is only invalidated. 1: The L2C and the ICs of all the CPUs whose CCDs are set to 0 are invalidated
23 to 19	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
18	SNM	0	R/W	Measures for Synonym Problems Specifies whether to take hardware measures for synonym problems and address array mismatch problems. When SNM is set to 0 by a program that might cause such problems, operation is not guaranteed. 0: No hardware measures are taken 1: Hardware measures are taken For details on synonym problems, see A.2, Memory Management Unit (MMU).

Bit	Bit Name	Initial Value	R/W	Description
17	MCP	0	R/W	Mixed Coherency Protocols This bit is set when different coherency protocols are to be used for the same physical address space (valid only when CCD = 0). Set this bit to 1 to use different coherency protocols, regardless of whether a single CPU or multiple CPUs use them. When different coherency protocols are used with MCP = 0, operation is not guaranteed. 0: The same coherency protocol is set 1: Different coherency protocols are set
16	CCD	0	R/W	Cache Coherency Control Enables or disables cache coherency control for multiple CPUs. 0: Cache coherency control is enabled 1: Cache coherency control is disabled
15 to 13	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
12	L2IE	0	R/W	Instruction Secondary Cache Enable 0: Secondary cache is not used for instruction fetch. 1: Secondary cache is used for instruction fetch.
11	ICI	0	R/W	IC Invalidation When 1 is written to this bit, the V bits of all the IC entries are cleared to 0. This bit is always read as 0.
10, 9	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
8	ICE	0	R/W	ICE Enable Specifies whether to use the IC. Note, however, when address translation is performed, the IC cannot be used unless the C bit in the page management information is also 1. 0: IC is not used 1: IC is used
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	L2OE	0	R/W	Operand Secondary Cache Enable 0: Secondary cache is not used for operand access. 1: Secondary cache is used for operand access.
3	OCI	0	R/W	OC Invalidation When 1 is written to this bit, the V bits of all the OC entries are cleared to 0. This bit is always read as 0.
2	CB	0	R/W	Cache Coherency Control 1 Indicates the cache coherency protocol for the P1 area in 29-bit address mode. This bit is invalid in 32-bit address mode, and indicates the value of the WT bit of PMB entries. • When cache coherency control is disabled (CCD = 1) 0: EI (write-through) protocol 1: MEI (write-back) protocol • When cache coherency control is enabled (CCD = 0) 0: ESI (write-through) protocol 1: MESI (write-back) protocol
1	WT	0	R/W	Cache Coherency Control 0 Indicates the cache coherency protocol for the P0, U0 and P3 areas. When address translation is performed, this bit is invalidated, and the value of the WT bit in the page management information is referenced. • When cache coherency control is disabled (CCD = 1) 0: MEI (write-back) protocol 1: EI (write-through) protocol • When cache coherency control is enabled (CCD = 0) 0: MESI (write-back) protocol 1: ESI (write-through) protocol
0	OCE	0	R/W	OC Enable Specifies whether to use the OC. Note, however, when address translation is performed, the OC cannot be used unless the C bit in the page management information is also 1. 0: OC is not used 1: OC is used

(2) Secondary Cache Control Register (L2CR)

L2CR selects the secondary cache operation mode and invalidates all the entries of the secondary cache. Immediately after writing 1 to L2CI to disable the L2C, read L2CR once.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	L2CI	—	SMS	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	W	R	R/W	R/W

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	L2CI	0	W	Secondary Cache Invalidation Writing 1 to this bit clears the V bits of all the secondary cache entries. This bit is always read as 0.
2	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
1, 0	SMS	All 0	R/W	Shared Memory Select Select whether to use the entries as the secondary cache or the shared memory. 00: All the entries are used as the secondary cache 01: Setting prohibited 10: All the entries are used as the shared memory 11: Entries 1024 to 2047 are used as the shared memory When writing to these bits, make sure that the ICs and OCs of all the CPUs are invalid, and set the L2CI bit to 1 to invalidate all the secondary cache entries. After setting 10 or 11, SYNCO instruction must be executed before accessing the shared memory.

A.5.3 Configuration and Operation of Secondary Cache

(1) Configuration

Figure A.12 shows the configuration of the secondary cache.

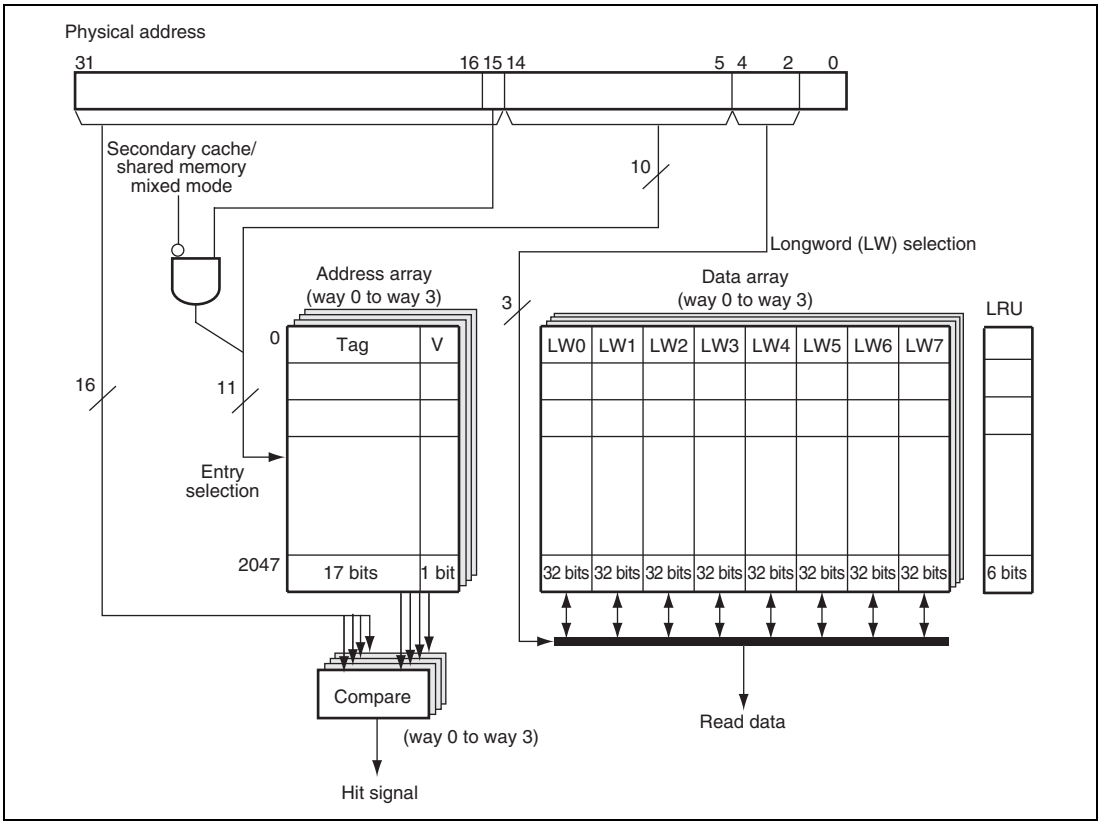


Figure A.12 Configuration of secondary Cache

The secondary cache has 2048 cache lines, and each of them consists of a 17-bit tag, a V bit, a 6-bit LRU, and 32-byte data.

(a) Tag

Stores the upper 17 bits of the 32-bit physical address of a data line to be cached.

(b) V Bit (Validity Bit)

Indicates whether the cache line stores valid data. When this bit is 1, the cache line is valid. A power-on reset or writing 1 to L2CR.L2CI initializes the V bit to 0.

(c) LRU Field

In 4-way associative method, up to four pieces of data that have the same entry address can be recorded in the cache. When an entry is recorded, the LRU bits select from four ways on which the entry is recorded. The LRU bits consist of six bits, and are controlled by the hardware. The LRU (Least Recently Used) algorithm, which selects the least recently accessed way, is used as the way selection algorithm. A power-on reset or writing 1 to L2CR.L2CI initializes the LRU bits to 0.

(d) Data Field

A single cache line of 32-byte (256-bit) data is stored in the data field. A power-on reset does not initialize the data field.

(2) Operation of Secondary Cache

The multiprocessor-capable SH-4A can use the secondary cache when the instruction cache (IC) or the operand cache (OC) is enabled. The V bit and LRU bits are initialized by the hardware when a power-on reset is performed or 1 is written to L2CR.L2CI. Invalidating the IC or OC does not initialize the V bit and the LRU bits in the secondary cache. It takes clock cycles of a quarter of the entries to initialize the V bit and the LRU bits. During the initialization of these bits, the secondary cache cannot be accessed through the CPU memory access. In this case, it is necessary to wait until the initialization is completed.

The secondary cache is accessed by means of the following operations:

- Operation via the IC by instruction fetch from the CPU -> (a)
- Operation via the OC by operand access from the CPU -> (b)
- Operation by cache operation instruction from the CPU -> (c)
- Coherency access from the SuperHyway -> (d)

(a) Operation via IC by Instruction Fetch from CPU

Instruction fetch from the CPU operates the secondary cache via the IC when CCR.ICE and CCR.L2IE are 1. The operation of the secondary cache is the following:

- Read by the IC

Read by IC:

A block read operation for an IC cache miss. Data transfer for this operation is always of a single line (32 bytes). The tag and the V bit are read from the secondary cache line of each way indexed by the physical address bits [15:5] ([14:5] in secondary cache/shared memory mixed mode), which indicates the IC cache miss address transmitted from the IC. Then, the physical address bits [31:15] and the tag are compared.

- When they matches and the V bit is 1, -> 1.
- Otherwise, -> 2.

1. Secondary Cache Hit

   32-byte data is read from the data field in the secondary cache line on the hit way indexed by the physical bit [15:5] ([14:5] in secondary cache/shared memory mixed mode), and the IC is filled with the data.
2. Secondary Cache Miss

   Data is read into the secondary cache line from the external memory corresponding to the physical address. The data read operation begins with the data corresponding to the physical address at which the cache miss occurred, and then is performed in the order of data by means of wrap around method. When the first data arrives in the secondary cache, the read data is also returned to the IC. When writing of one line data is completed, the secondary cache records the tag corresponding to the physical address, and writes 1 to the V bit.

(b) Operation via OC by Operand Access from CPU

Operand access from the CPU operates the secondary cache via the IC when CCR.OCE and CCR.L2OE are 1. The operation of the secondary cache is one of the following:

- Read by OC
- Write by OC write-through
- Write by OC Write-back

Read by OC:

A block read operation for an OC cache miss. Data transfer for this operation is always of a single line (32 bytes). The tag and the V bit are read from the secondary cache line of each way indexed by the physical address bits [15:5] ([14:5] in secondary cache/shared memory mixed mode), which indicates the OC cache miss address transmitted from the OC. The physical address bits [31:15] and the tag are compared.

- When they matches and the V bit is 1, -> 1.
- Otherwise, -> 2.

1. Secondary Cache Hit

   32-byte data is read from the data field in the secondary cache line on the hit way indexed by the physical bit [15:5] ([14:5] in secondary cache/shared memory mixed mode), and the OC is filled with the data.
2. Secondary Cache Miss

   Data is read into the secondary cache line from the external memory corresponding to the physical address. The data read operation begins with the data corresponding to the physical address at which the cache miss occurred, and then is performed in the order of data by means of wrap around method. When the first data arrives in the secondary cache, the read data is also returned to the IC. When writing of one line data is completed, the secondary cache records the tag corresponding to the physical address, and writes 1 to the V bit.

Write by OC write-through:

A single write operation (quadword/longword/word/byte) for an OC write-through. The tag and the V bit are read from the secondary cache line of each way indexed by the physical address bits [15:5] ([14:5] in secondary cache/shared memory mixed mode), which are transmitted from the OC. The physical address bits [31:15] and the tag are compared.

- When they matches and the V bit is 1, -> 1.
- Otherwise, -> 2.

1. Secondary Cache Hit

   Data is written in the specified access size (quadword/longword/word/byte) to the data position specified by the physical address bits [4:0] in the data field in the secondary cache line on the hit way indexed by the physical address bits [15:5] ([14:5] in the secondary cache/shared memory mixed mode). The write operation is also performed for the external memory in the specified access size.

2. Secondary Cache Miss

   Data is written to the external memory corresponding to the physical address in the specified access size. In this case, data is not written to the secondary cache.

Write by OC write-back:

A block write operation for an OC write-back. Data transfer for this operation is always of a single line (32 bytes). The tag and the V bit are read from the secondary cache line of each way indexed by the physical address bits [15:5] ([14:5] in secondary cache/shared memory mixed mode), which are transmitted from the OC. The physical address bits [31:15] and the tag are compared.

- When they matches and the V bit is 1, -> 1.
- Otherwise, -> 2.

1. Secondary Cache Hit

   32-byte data is written to the data in the cache line on the hit way indexed by the physical address bits [15:5] ([14:5] in secondary cache/shared memory mixed mode). 32-byte data is also written to external memory.
2. Secondary Cache Miss

   32-byte data is written to the external memory corresponding to the physical address. In this case, no data is written to the secondary cache.

(c) Operation by Cache Control Instruction from CPU

There are the following CPU cache control instructions:

- ICBI @Rn
- OCBI @Rn and OCBP @Rn
- OCBWB @Rn

ICBI @Rn:

When CCR.IBE, CCR.CCD, and CCR.L2IE are 1, 0, and 1, respectively, and the virtual address space specified by Rn is cacheable, the secondary cache line including the physical address corresponding to the virtual address is invalidated. Since the secondary cache uses write-through method and does not become dirty, write-back operation is not performed.

OCBI @Rn and OCBP @Rn:

When CCR.L2OE is 1 and the virtual address space specified by Rn is cacheable, the secondary cache line including the physical address corresponding to the virtual address is invalidated. Since

the secondary cache uses write-through method and does not become dirty, write-back operation is not performed.

When the address bit [31:24] specified by Rn is H'F8, the way specified by the address [17:16], and the index specified by the address [15:5] (the way specified by the address[16:15], and the index specified by the address [14:5] in secondary cache/shared memory mixed mode) are invalidated. Since the secondary cache uses write-through method and does not become dirty, write-back operation is not performed.

OCBWB @Rn:

Since the secondary cache uses write-through method and does not become dirty, no operation is performed.

(d) Coherency Access from SuperHyway Bus

Coherency access is made from SuperHyway bus when

- PURGE transaction is received;
- FLUSH transaction is received.

For PURGE Transaction:

Secondary cache lines that is used by coherency-enabled CPUs and that include the specified physical address are invalidated.

For FLUSH Transaction:

Since the secondary cache uses write-through method and does not become dirty, no operation is performed.

A.6 Instruction Descriptions

A.6.1 Changes to CPU Instructions

(1) MOVCO (MOVe Conditional): Data Transfer Instruction

Conditional store for storing data when a read-modify-write operation is atomically completed:

Format	Operation	Instruction Code	Cycle	T Bit
MOVCO.L R0,@Rn	LDST → T if (T = = 1) R0 → (Rn) 0 → LDST	0000nnnn01110011	3	LDST

(a) Description

MOVCO is used in combination with the MOVLI instruction to realize an atomic read-modify-write operation in a multiprocessor system that maintains cache coherency.

This instruction copies the value of the LDST flag to the T bit. The value of R0 is stored at address indicated by Rn only when the T bit is 1. After the store operation, the LDST flag is cleared to 0. When coherency control is enabled, after MOVLI instruction is executed, if an interrupt or exception occurs, if cache lines including the address specified by MOVLI instruction are invalidated via snoop bus, if the address specified by a MOVCO instruction is not included in the cache line that includes the address specified by MOVLI instruction, or if a cache line secured by MOVLI instruction does not exist due to replacing caches, the LDST flag is cleared prior to the operation above.

When coherency control is disabled, if an interrupt or exception occurs after MOVLI instruction is executed, the LDST flag is cleared prior to the operation above.

(b) Notes

To realize an atomic read-modify-write operation in the non-cacheable area, use the TAS.B instruction. In the non-cacheable area, atomic operation is not guaranteed by the MOVLI and MOVCO instructions. When coherency control is disabled, the LDST flag is not cleared even if another processor handles the address specified during execution of the MOVLI instruction.

(c) Operation

```
MOVCO(long n) /* MOVCO Rn,@Rn */
{
    T = LDST;
    if(T = = 1)
        Write_Long(R[n],R[0]);
    LDST = 0;
    PC + = 2
}
```

(d) Example

```
Retry:   MOVLI.L @Rn,R0   ;Atomic incrementation
         ADD #1,R0
         MOVCO.L R0,@Rn
         BF Retry         ;Reexecute if an event that clears
                           the LDST flag occurs between
                           the MOVLI and MOVCO instructions.
         NOP
```

(e) Possible exceptions

- Data TLB multiple-hit exception
- Data TLB miss exception
- Data TLB protection violation exception
- Initial page write exception
- Data address error

(2) MOVLI (MOVe Linked): Data Transfer Instruction

Load for starting an atomic read-modify-write operation:

Format	Operation	Instruction Code	Cycle	T Bit
MOVLI.L @Rm,R0	1 → LDST, (Rm) → R0 If an interrupt, an exception, or storage in the cache line including (Rm) has occurred, 0 → LDST	0000mmmm01100011	1	—

(a) Description

MOVLI is used in combination with the MOVCO instruction to realize an atomic read-modify-write operation in a multiprocessor system that maintains cache coherency.

This instruction sets the LDST flag to 1 and reads the four-byte data at address indicated by Rm into R0. When coherency control is enabled, after MOVLI instruction is executed, if an interrupt or exception occurs, if cache lines including the address specified by MOVLI instruction are invalidated via snoop bus, if the address specified by a MOVCO instruction is not included in the cache line that includes the address specified by MOVLI instruction, or if a cache line secured by MOVLI instruction does not exist due to replacing caches, the LDST flag is cleared prior to the operation above.

When coherency control is disabled, if an interrupt or exception occurs after MOVLI instruction is executed, the LDST flag is cleared prior to the operation above.

When the MOVCO instruction is executed while the LDST flag is 1, the instruction performs a store operation, and the T bit is set to 1. When the MOVCO instruction is executed while the LDST flag is 0, the instruction does not perform a store operation, and the T bit is cleared to 0.

(b) Notes

To realize an atomic read-modify-write operation in the non-cacheable area, use the TAS.B instruction. In the non-cacheable area, atomic operation is not guaranteed by the MOVLI and MOVCO instructions. When coherency control is disabled, the LDST flag is not cleared even if another processor handles the address specified during execution of the MOVLI instruction.

(c) Operation

```
MOVLINK(long m) /* MOVLI Rm,@Rn */
{
    LDST = 1;
    R[0] = Read_Long(R[m]);
    PC + = 2
}
```

(d) Example

See the exmple of the MOVCO instruction.

(e) Possible exceptions

- Data TLB multiple-hit exception
- Data TLB miss exception

- Data TLB protection violation exception
- Data address error

(3) TAS (Test And Set): Logical Instruction

Test memory and set bit:

Format	Operation	Instruction Code	Cycle	T Bit
TAS.B @Rn	When (Rn) = 0, 1 → T, else 0 → T 1 → MSB of (Rn)	0100nnnn00011011	4	Test result

(a) Description

This instruction reads the byte data at the address indicated by Rn from the memory, and sets the T bit to 1 if the data is zero or clears the T bit to 0 if the data is nonzero. The instruction then sets bit 7 to 1 and writes to the same address. The bus is not released during this period.

The consecutive TAS.B memory accesses are performed automatically. Since the bus is not released during these accesses, there is no other memory access by other modules.

(b) Notes

From the SH-4A extended functions (VER = H'40 in PVR), even when the cache block includes the memory area specified by general register Rn, the memory area is not purged before the byte data is read, and the data in the cache is read or written. Accordingly, if a cacheable area is specified in general register Rn, operation is not guaranteed. Be sure to specify a non-cacheable area in the register. To guarantee the operation in a cacheable area, use the MOVLI and MOVCO instructions.

(c) Operation

```
TAS(int n) /* TAS.B @Rn */
{
    int temp;

    temp = (int)Read_Byte(R[n]); /* Bus Lock */
    if(temp = = 0) T = 1;
    else T = 0;
    temp | = 0x00000080;
    Write_Byte(R[n],temp); /* Bus unlock */
    PC + = 2;
}
```

(d) Possible exceptions

- Data TLB multiple-hit exception
- Data TLB miss exception
- Data TLB protection violation exception
- Initial page write exception
- Data address error

Exceptions are checked by handling a data access by this instruction as a byte load and a byte store.

(4) OCBI (Operand Cache Block Invalidate): Data Transfer Instruction

Invalidate cache block:

Format	Operation	Instruction Code	Cycle	T Bit
OCBI @Rn	Invalidates operand cache block.	0000nnnn10010011	1	—

(a) Description

This instruction accesses data using the value indicated by effective address Rn. If the cache is hit, the corresponding cache block is invalidated (the V bit is cleared to 0). If there is unwritten information (U bit = 1), write-back is not performed even though write-back mode is selected. This instruction is executed via the snoop bus for all CPUs for which coherency control is enabled.

When CCR.L2OE = 1, the corresponding secondary cache block is invalidated (V bit = 0).

If the address indicated by Rn is the non-cacheable area, the operand cache line indicated by way = Rn[14:13] and entry = Rn[12:5] is invalidated only when Rn[31:24] is H'F4 (OC address array area). This operation can be executed in the non-cacheable area only in privileged mode. In user mode, an address error exception occurs. A TLB-related exception does not occur.

When the address indicated by Rn is non-cacheable area, the secondary cache line indicated by way = Rn[17:16], and entry = Rn[15:5] is invalidated only when Rn[31:24] = H'F8 (L2C address array area). Operation for this non-cacheable area is only allowed in privilege mode.

Do not execute this instruction for the memory allocation array area unless Rn[31:24] = H'F4, H'F8 or for the control register area or its reserved area (H'F0 to H'F3, H'F5 to H'F7, H'F9 to H'FF). When the access is to a non-cacheable area other than the memory allocation array area or control register area, no operation is performed.

(b) Notes

From the SH-4A extended functions (VER = H'40 in PVR), when this instruction is executed for the cacheable area, invalidation processing is performed via the snoop bus for all CPUs for which coherency control is enabled. This extended function ensures that cache operation is performed properly with this instruction in a multiprocessor system that maintains cache coherency even when processes migrate between CPUs.

(c) Operation

```
OCBI(int n) /* OCBI @Rn */
{
    int temp;

    invalidate_operand_cache_block(R[n]);
    PC + = 2;
}
```

(d) Possible exceptions

- Data TLB multiple-hit exception
- Data TLB miss exception
- Data TLB protection violation exception
- Initial page write exception
- Data address error

Note that these exceptions can occur even when the cache is not hit.

(5) OCBP (Operand Cache Block Purge): Data Transfer Instruction

Purge cache block:

Format	Operation	Instruction Code	Cycle	T Bit
OCBP @Rn	Writes back operand cache block and invalidates it.	0000nnnn10100011	1	—

(a) Description

This instruction accesses data using the value indicated by effective address Rn. If the cache is hit and there is unwritten information (U bit = 1), the corresponding cache block is written back to the external memory and is then invalidated (the V bit is cleared to 0). However, if there is no unwritten information (U bit = 0), the block is only invalidated. This instruction is executed via the snoop bus for all CPUs for which coherency control is enabled.

When CCR.L2OE = 1, the corresponding secondary cache block is invalidated (V bit = 0).

If the address indicated by Rn is the non-cacheable area, the operand cache line indicated by way = Rn[14:13] and entry = Rn[12:5] is invalidated only when Rn[31:24] is H'F4 (OC address array area). If the line to be invalidated is dirty, it is written back. This operation can be executed in the non-cacheable area only in privileged mode. In user mode, an address error exception occurs. A TLB-related exception does not occur.

When the address indicated by Rn is non-cacheable area, the secondary cache line indicated by way = Rn[17:16], and entry = Rn[15:5] is invalidated only when Rn[31:24] = H'F8 (L2C address array area). Operation for this non-cacheable area is only allowed in privilege mode.

Do not execute this instruction for the memory allocation array area unless Rn[31:24] = H'F4, H'F8 or for the control register area or its reserved area (H'F0 to H'F3, H'F5 to H'F7, H'F9 to H'FF). When the access is to a non-cacheable area other than the memory allocation array area, on-chip memory area or control register area, no operation is performed.

(b) Notes

From the SH-4A extended functions (VER = H'40 in PVR), when this instruction is executed for the cacheable area, write back and invalidation processing is performed via the snoop bus for all CPUs for which coherency control is enabled. This extended function ensures that cache operation is performed properly with this instruction in a multiprocessor system that maintains cache coherency even when processes migrate between CPUs.

(c) Operation

```
OCBP(int n) /* OCBP @Rn */
{
    int temp;

    if(is_dirty_block(R[n]))  write_back(R[n]);
    invalidate_operand_cache_block(R[n]);
    PC + = 2;
}
```

(d) Possible exceptions

- Data TLB multiple-hit exception
- Data TLB miss exception
- Data TLB protection violation exception
- Initial page write exception
- Data address error

Note that these exceptions can occur even when the cache is not hit.

(6) OCBWB (Operand Cache Block Write Back): Data Transfer Instruction

Write back cache block:

Format	Operation	Instruction Code	Cycle	T Bit
OCBWB @Rn	Writes back operand cache block.	0000nnnn10110011	1	—

(a) Description

This instruction accesses data using the value indicated by effective address Rn. If the cache is hit and there is unwritten information (U bit = 1), the corresponding cache block is written back to the external memory and then cleaned (the U bit is cleared to 0). If the block is already clean or if a cache miss occurred, nothing is done to the block. This instruction is executed via the snoop bus for all CPUs for which coherency control is enabled.

This instruction does not handle the secondary cache even when CCR.L2OE = 1 because the secondary cache supports write throug only.

If the address indicated by Rn is the non-cacheable area, and if the operand cache line indicated by way = Rn[14:13] and entry = Rn[12:5] is dirty when Rn[31:24] is H'F4 (OC address array area), the cache line is written back, and the dirty bit cleared to 0. This operation can be executed in the non-cacheable area only in privileged mode. In user mode, an address error exception is generated. A TLB-related exception does not occur.

When the address space indicated by Rn is non-cacheable, If Rn[31:24] = H'F8 (L2C address array area), no operation is performed.Operation for this non-cacheable area is only allowed in privilege mode.

Do not execute this instruction for the memory allocation array area unless Rn[31:24] = H'F4, H'F8 or for the control register area or its reserved area (H'F0 to H'F3, H'F5 to H'F7, H'F9 to H'FF). When the access is to a non-cacheable area other than the memory allocation array area, on-chip memory area or control register area, no operation is performed.

(b) Notes

From the SH-4A extended functions (VER = H'40 in PVR), when this instruction is executed for the cacheable area, write-back processing is performed via the snoop bus for all CPUs for which coherency control is enabled. This extended function ensures that cache operation is performed

properly with this instruction in a multiprocessor system that maintains cache coherency even when processes migrate between CPUs.

(c) Operation

```
OCBWB(int n) /* OCBWB @Rn */
{
    int temp;

    if(is_dirty_block(R[n]))  write_back(R[n]);
    PC + = 2;
}
```

(d) Possible exceptions

- Data TLB multiple-hit exception
- Data TLB miss exception
- Data TLB protection violation exception
- Initial page write exception
- Data address error

Note that these exceptions can occur even when the cache is not hit.

(7) ICBI (Instruction Cache Block Invalidate): Data Transfer Instruction

Instruction cache block invalidate

Format	Operation	Instruction Code	Cycle	T Bit
ICBI @Rn	Invalidates a instruction cache block	0000nnnn11100011	13	—

(a) Description

This instruction accesses the instruction cache at the effective address indicated by the contents of Rn. When the cache is hit, the corresponding cache block is invalidated (the V bit is cleared to 0). At this time, write-back is not performed. No operation is performed in the case of a cache miss or access to a non-cache area.

(b) Note

From the SH-4A extended function (PVR.VER = H'40), when ICBI instruction broadcast is enabled (CCR.IBE = 1 and CCR.CCD = 0), the ICs of all the CPUs for which coherency control is enabled (CCR.CCD = 0) are invalidated via snoop bus. The extended function is intended for

properly performing cache operation even during process migration in a multi-processor system that maintains its cache coherency.

When ICBI instruction broadcast is enabled (CCR.IBE = 1 and CCR.CCD = 0) and the instruction secondary cache is enabled (CCR.L2IE = 1), the secondary cache is invalidated as well.

(c) Operation

```
ICBI(int n) /* ICBI @Rn */
{
    invalidate_instruction_cache_block(R[n]);
    PC + = 2;
}
```

(d) Example

When a program is overwriting RAM to modify its own execution, the corresponding block of the instruction cache should be invalidated by the ICBI instruction. This prevents execution of the program from the instruction cache, where the non-overwritten instructions are stored.

(e) Possible Exceptions

- Instruction TLB multiple-hit exception
- Instruction TLB miss exception
- Instruction TLB protection violation exception
- Instruction address error
- Slot illegal instruction exception

Note that these exceptions can occur even when the cache is not hit.

A.7 List of Registers (Changes)

The following registers have been changed or added to the multiprocessor-capable SH-4A.

A.7.1 List of Registers

Module	Register Name	Abbr.	R/W	P4 Address*	Area 7 Address*	Size
MMU	Page table entry high register	PTEH	R/W	H'FF00 0000	H'1F00 0000	32
	MMU control register	MMUCR	R/W	H'FF00 0010	H'1F00 0010	32
	Page table entry address space identifier register	PTEAEX	R/W	H'FF00 007C	H'1F00 007C	32
Cache	Cache control register	CCR	R/W	H'FF00 001C	H'1F00 001C	32
Secondary cache	Secondary cache control register	L2CR	R/W	H'FBF0 0000	—	32

Note * P4 is the address when virtual address space P4 area is used. Area 7 is the address when physical address space area 7 is accessed by using the TLB.

A.7.2 Register States in Each Operating Mode

Module	Register Name	Abbr.	Power-on Reset	Manual Reset	Sleep
MMU	Page table entry high register	PTEH	Undefined	Undefined	Retained
	MMU control register	MMUCR	H'0000 0000	H'0000 0000	Retained
	Page table entry address space identifier register	PTEAEX	Undefined	Undefined	Retained
Cache	Cache control register	CCR	H'0000 0000	H'0000 0000	Retained
Secondary cache	Secondary cache control register	L2CR	H'0000 0000	—	—

A.8 CPU Core ID Register (CPIDR)

The CPU core ID register (CPIDR) is used to discriminate between the CPU cores in a multi-CPU configuration.

Register Name	Abbr.	R/W	P4 Address*	Area 7 Address*	Size
CPU core ID register	CPIDR	R	H'FF00 0048	H'1F00 0048	32

Note * P4 is the address when virtual address space P4 area is used. Area 7 is the address when physical address space area 7 is accessed by using the TLB.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CPUID							
Initial value:	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
7 to 0	CPUID	Undefined	R	CPU Core ID This register is used to discriminate between the CPU cores in a multi-CPU configuration. Its value differs for each CPU. CPU0: 00000000 CPU1: 00000001

A.9 Barrier Synchronization Register

The barrier synchronization register is used for synchronization among multiple CPU cores.

Register Name	Abbr.	R/W	P4 Address*	Area 7 Address*	Size
Barrier write register 0	BARW0	W	H'FF00 0100	H'1F00 0100	32
Barrier read register 0	BARR0	R	H'FF00 0104	H'1F00 0104	32
Barrier write register 1	BARW1	W	H'FF00 0110	H'1F00 0110	32
Barrier read register 1	BARR1	R	H'FF00 0114	H'1F00 0114	32
Barrier write register 2	BARW2	W	H'FF00 0120	H'1F00 0120	32
Barrier read register 2	BARR2	R	H'FF00 0124	H'1F00 0124	32

Note * P4 is the address when virtual address space P4 area is used. Area 7 is the address when physical address space area 7 is accessed by using the TLB.

- BARW0, BARW1 and BARW2

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
0	BR	0	W	Barrier Bit Either 0 or 1 can be written to the barrier bit [012]. Its value is always 0 when read.

- BARR0, BARR1 and BARR2

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	BR3	BR2	BR1	BR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
3	BR3	0	R	CPU3 Barrier Bit (Unsupported) The CPU3 barrier bit [012] value is read from this bit.
2	BR2	0	R	CPU2 Barrier Bit (Unsupported) The CPU2 barrier bit [012] value is read from this bit.
1	BR1	0	R	CPU1 Barrier Bit The CPU1 barrier bit [012] value is read from this bit.
0	BR0	0	R	CPU0 Barrier Bit The CPU0 barrier bit [012] value is read from this bit.

Note: The CPU's BR bits are undefined in sleep mode.

B. Mode Pin Settings

The MODE14–MODE0 pin values are input in the event of a power-on reset via the $\overline{\text{PRESET}}$ pin.

Note: Setting prohibited other than the value of the following tables.

Legend:

H: High level input

L: Low level input

Table B.1 Clock Operating Modes with External Pin Combination

Clock Operating Mode	Pin Value MODE [3:0] Pin Number				OSC/ External input Frequency [MHz]			Frequency (vs. Input Clock)					
	3	2	1	0	Typ.	PLL1	PLL2	Ick	SHck	DUck	Bck	Pck	DDRck
0	L	L	L	L	16.7	× 64	×1	× 32	× 16	× 8	× 16/3	× 3	× 32
1	L	L	L	H				× 32	× 16	× 8	× 4	× 4	× 32
2	L	L	H	L				× 32	× 16	× 8	× 2	× 2	× 32
3	L	L	H	H	33.3	× 32		× 16	× 8	× 4	× 8/3	× 4/3	× 16
4	L	H	L	L				× 16	× 8	× 4	× 2	× 2	× 16
5	L	H	L	H				× 16	× 8	× 4	× 1	× 1	× 16

Table B.2 Area 0 Memory Type and Bus Width

Pin Value MODE6	MODE5*	MODE4	Memory Interface	Bus Width
L	L	L	MPX interface	32 bits
		H	SRAM interface	8 bits
	H	L	SRAM interface	16 bits
		H	SRAM interface	32 bits

Table B.3 Area 0 Space Extended Mode

Pin Value	
MODE7	**Area 0 Space**
L	CS0 area is 128-Mbytes space (CS1 is used as A25)
H	Normal mode (CS0 area is 64-Kbytes)

Table B.4 Endian

Pin Value	
MODE8	**Endian**
L	Big endian
H	Little endian

Table B.5 Clock Input

Pin Value	
MODE9	**Clock Input**
L	External input clock
H	Crystal resonator

Table B.6 Address Extended Mode

Pin Value	
MODE10	**Address Extended Mode**
L	29-bit address mode
H	32-bit address extended mode

Table B.7 PCIexpress Mode

Pin Value	
MODE11	**PCIexpress PHY Mode**
L	Root point mode
H	End point mode

Table B.8 PCIexpress PHY Mode

Pin Value	
MODE12	**PCIexpress PHY Mode**
L	4 lanes+1 lane
H	2 lanes+1 lane+1 lane

Table B.9 AUD Mode

Pin Value	
MODE13	**EXAUD Mode**
L	AUD enable
H	AUD disable (AUD pins cannot be used)

Table B.10 Operation Mode

Pin Value	
MODE14	**Operation Mode**
L	Normal operation
H	Setting prohibited

Table B.11 Mode Control

Pin Value	
MPMD	**Mode**
L	Emulation support mode
H	LSI operation mode

C. Version Registers (PVR, PRR)

The SH7786 has the read-only registers which show the version of a processor core, and the version of a product. By using the value of these registers, it becomes possible to be able to distinguish the version and product of a processor from software, and to realize the scalability of the high system.

Note: The bits 7 to 0 of PVR register and the bits 3 to 0 of PRR register should be masked by the software.

Table C.1 Register Configuration

Register Name	Abbr.	R/W	Initial Value	P4 Address	Area 7 Address	Access Size
Processor Version Register	PVR	R	H'1040 05xx	H'FF00 0030	H'1F00 0030	32
Product Register	PRR	R	H'0000 04xx	H'FF00 0044	H'1F00 0044	32

Legend:
x: Undefined

- Processor Version Register (PVR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	version information															
Initial value	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	version information								—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	1	0	1	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	—	—	—	—	—	—	—	—

- Product Register (PRR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	version information															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	version information								—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	1	0	0	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	—	—	—	—	—	—	—	—

D. Power-On and Power-Off Sequence

D.1 Power-On and Power-Off Sequence for Power Supplies at the Same and at Different Potentials

The following stipulations apply to the power-on and power-off sequences of the 1.25 V power supplies (pins VDD, VDD-PCIE, VDDAI, and AV12), 1.5 V power supplies (pins VCCQ-DDR15 and VCCQ-PCI15), and 3.3 V power supplies (pins VCCQ, VCCQ-PCI, and AV33).

(1) Power supply power-on sequence

There are no restrictions on the power-on sequence. All power supplies should rise from the VSS** level within 300 ms of the first.

(2) Power supply power-off sequence

There are no restrictions on the power-off sequence. All power supplies should fall within 300 ms of the first.

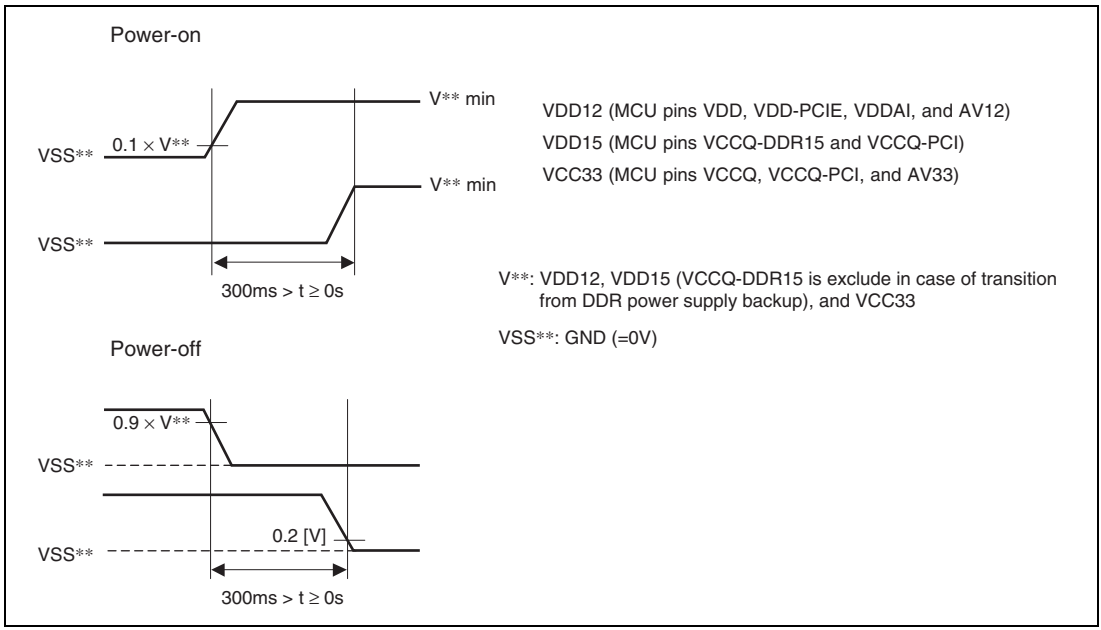


Figure D.1 Power-On/Power-Off Sequence for Power Supplies at the Same and at Different Potentials

D.2 Power-On and Power-Off Sequence in DDR3-SDRAM Power Supply Backup Mode for Power Supplies at the Same and at Different Potentials

In DDR3-SDRAM power supply backup mode, the following stipulations apply to the power-on and power-off sequences of the 1.25 V power supplies (pins VDD, VDD-PCIE, VDDAI, and AV12), 1.5 V power supplies (pins VCCQ-DDR15 and VCCQ-PCI), and 3.3 V power supplies (pins VCCQ, VCCQ-PCI, and AV33).

(1) Power supply power-on sequence

There are no restrictions on the power-on sequence. All power supplies should rise from the VSS** level within 300 ms of the first power supply other than VCCQ-DDR15.

(2) Power supply power-off sequence

There are no restrictions on the power-off sequence. All power supplies should fall within 300 ms of the first power supply other than VCCQ-DDR15.

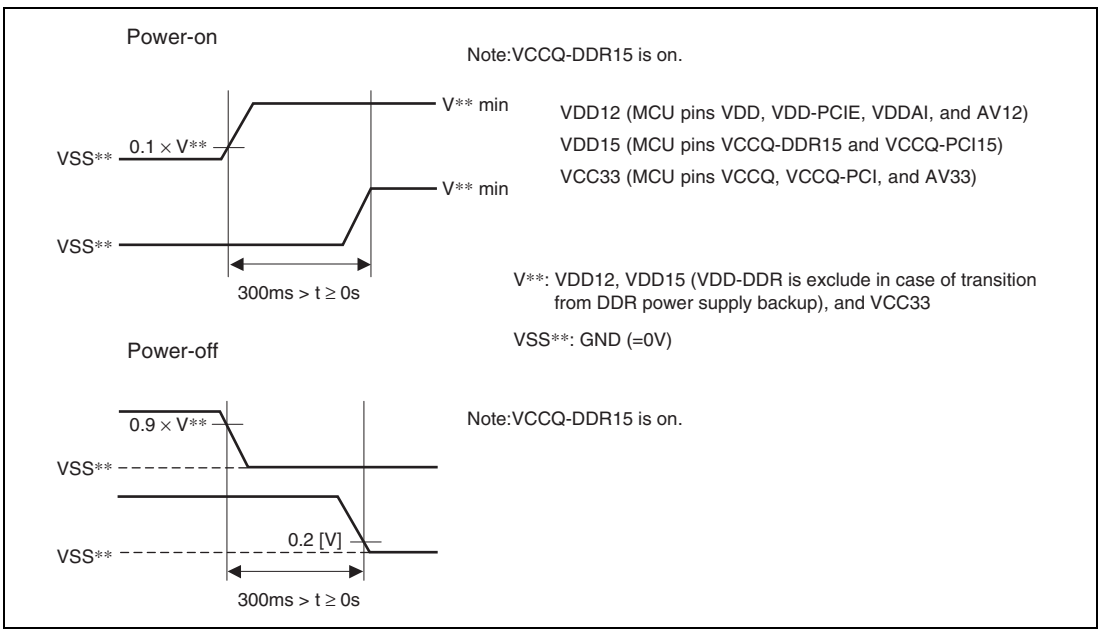
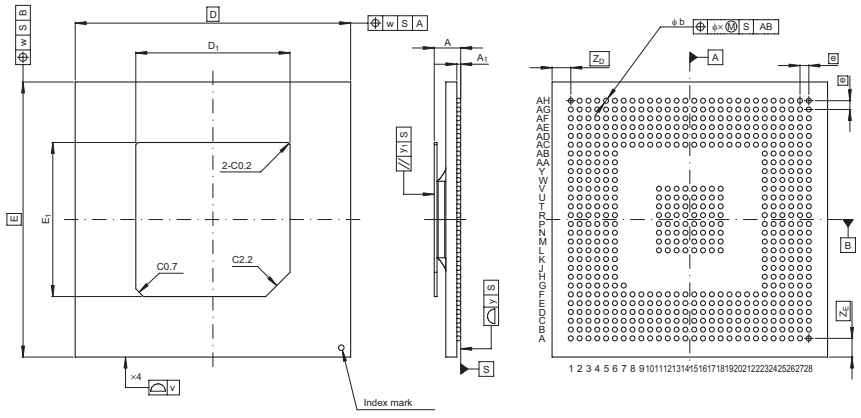


Figure D.2 Power-On/Power-Off Sequence in DDR3-SDRAM Power Supply Backup Mode for Power Supplies at the Same and at Different Potentials

E. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-HFBGA593-25x25-0.80	PRBG0593GA-A	593F7S-A	2.3g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	25.0	—
E	—	25.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	2.6
A_1	0.3	0.35	0.4
e	—	0.8	—
b	0.4	0.45	0.5
x	—	—	0.08
y	—	—	0.12
y_1	—	—	0.20
Z_D	—	1.7	—
Z_E	—	1.7	—
D_1	13.9	14.0	14.1
E_1	13.9	14.0	14.1

F. Product Lineup

Product Name	Voltage	Operating Frequency	Product Code	Operating Temperature	Package
SH7786	1.2V	533MHz	R8A77860NBGV	-20°C to +85°C	593 pin BGA(lead-free)

Renesas 32-Bit RISC Microcomputer
SH7786 Group
User's Manual: Hardware

Publication Date: Rev.1.00, November 30, 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

© 2010 Renesas Electronics Corporation. All rights reserved.

Colophon 1.0

SH7786 Group
User's Manual: Hardware

Renesas Electronics Corporation

REJ09B0501-0100