

# 32

## SH7764 Group Hardware Manual

Renesas 32-Bit RISC Microcomputer  
SuperH™ RISC Engine Family  
SH-4A Series

SH77641

R5S77641

SH77640

R5S77640

Hardware Manual

Rev.1.00

Revision Date: Nov. 22, 2007

Renesas Technology  
[www.renesas.com](http://www.renesas.com)



## Notes regarding these materials

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

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

## General Precautions on Handling of Product

### 1. Treatment of NC Pins

**Note:** Do not connect anything to the NC pins.

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

### 2. Treatment of Unused Input Pins

**Note:** Fix all unused input pins to high or low level.

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

### 3. Processing before Initialization

**Note:** When power is first supplied, the product's state is undefined.

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

### 4. Prohibition of Access to Undefined or Reserved Addresses

**Note:** Access to undefined or reserved addresses is prohibited.

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

### 5. Reading from/Writing to Reserved Bit of Each Register

**Note:** Treat the reserved bit of register used in each module as follows except in cases where the specifications for values which are read from or written to the bit are provided in the description.

The bit is always read as 0. The write value should be 0 or one, which has been read immediately before writing.

Writing the value, which has been read immediately before writing has the advantage of preventing the bit from being affected on its extended function when the function is assigned.



# Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules

- CPU and System-Control Modules
- On-Chip Peripheral Modules

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

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

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

7. Electrical Characteristics
8. Appendix
9. Index

# Preface

This LSI is a RISC (Reduced Instruction Set Computer) microcomputer which includes a Renesas Technology-original RISC CPU as its core, and the peripheral functions required to configure a system.

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

**Objective:** This manual was written to explain the hardware functions and electrical characteristics of this LSI to the above users.

**Notes on reading this manual:**

- In order to understand the overall functions of the chip  
Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details on FPU functions and each instructions  
Read the additional volume, SH-4A Extended Functions 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 a low-active signal: $\overline{\text{xxx}}$

## Abbreviations

ATAPI	ATAPI Controller
CPG	Clock Pulse Generator
DMAC	Direct Memory Access Controller
E-DMAC	Ethernet Controller Direct Memory Access Controller
EtherC	Ethernet Controller
FLCTL	NAND Flash Memory Controller
G2D	2D Graphics Engine
GPIO	General Purpose I/O
H-UDI	User Debugging Interface
IIC	I2C Bus Interface
INTC	Interrupt Controller
MCU	Memory Controller Unit
MMU	Memory Management Unit
SCIF	Serial Communication Interface with FIFO
TMU	Timer Unit
UBC	User Break Controller
USB	USB Host/Function Interface
VDC2	Video Display Controller 2
WDT	Watchdog Timer and Reset
SSI	Serial Sound Interface
LCDC	LCD Controller
SRC	Sampling Rate Converter

bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
Hi-Z	High Impedance
I/O	Input/Output
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connection
PLL	Phase Locked Loop

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

# Contents

Section 1 Overview.....	1
1.1 SH7764 Features.....	1
1.2 Block Diagram.....	12
1.3 Pin Arrangement.....	14
1.4 Pin Functions .....	15
1.5 Address Map.....	28
Section 2 Programming Model.....	31
2.1 Data Formats.....	31
2.2 Register Descriptions.....	32
2.2.1 Privileged Mode and Banks.....	32
2.2.2 General Registers.....	36
2.2.3 Floating-Point Registers.....	37
2.2.4 Control Registers .....	39
2.2.5 System Registers.....	41
2.3 Memory-Mapped Registers.....	45
2.4 Data Formats in Registers.....	46
2.5 Data Formats in Memory .....	46
2.6 Processing States.....	47
2.7 Usage Notes .....	49
2.7.1 Notes on Self-Modifying Code.....	49
Section 3 Instruction Set.....	51
3.1 Execution Environment .....	51
3.2 Addressing Modes .....	53
3.3 Instruction Set.....	58
Section 4 Pipelining.....	71
4.1 Pipelines.....	71
4.2 Parallel-Executability.....	82
4.3 Issue Rates and Execution Cycles.....	85
Section 5 Exception Handling .....	95
5.1 Summary of Exception Handling.....	95
5.2 Register Descriptions.....	95
5.2.1 TRAPA Exception Register (TRA) .....	96
5.2.2 Exception Event Register (EXPEVT).....	97

5.2.3	Interrupt Event Register (INTEVT).....	98
5.2.4	Non-Support Detection Exception Register (EXPMASK) .....	99
5.3	Exception Handling Functions.....	101
5.3.1	Exception Handling Flow .....	101
5.3.2	Exception Handling Vector Addresses .....	101
5.4	Exception Types and Priorities .....	102
5.5	Exception Flow .....	104
5.5.1	Exception Flow .....	104
5.5.2	Exception Source Acceptance.....	106
5.5.3	Exception Requests and BL Bit .....	107
5.5.4	Return from Exception Handling.....	107
5.6	Description of Exceptions.....	108
5.6.1	Resets.....	108
5.6.2	General Exceptions .....	110
5.6.3	Interrupts.....	126
5.6.4	Priority Order with Multiple Exceptions .....	127
5.7	Usage Notes .....	129
Section 6 Floating-Point Unit (FPU).....		131
6.1	Features.....	131
6.2	Data Formats.....	132
6.2.1	Floating-Point Format.....	132
6.2.2	Non-Numbers (NaN) .....	135
6.2.3	Denormalized Numbers .....	136
6.3	Register Descriptions.....	137
6.3.1	Floating-Point Registers .....	137
6.3.2	Floating-Point Status/Control Register (FPSCR) .....	139
6.3.3	Floating-Point Communication Register (FPUL) .....	142
6.4	Rounding.....	143
6.5	Floating-Point Exceptions.....	144
6.5.1	General FPU Disable Exceptions and Slot FPU Disable Exceptions .....	144
6.5.2	FPU Exception Sources .....	144
6.5.3	FPU Exception Handling.....	145
6.6	Graphics Support Functions.....	146
6.6.1	Geometric Operation Instructions.....	146
6.6.2	Pair Single-Precision Data Transfer.....	147
Section 7 Memory Management Unit (MMU).....		149
7.1	Overview of MMU .....	150
7.1.1	Address Spaces .....	152

7.2	Register Descriptions .....	158
7.2.1	Page Table Entry High Register (PTEH) .....	159
7.2.2	Page Table Entry Low Register (PTEL) .....	160
7.2.3	Translation Table Base Register (TTB) .....	161
7.2.4	TLB Exception Address Register (TEA) .....	162
7.2.5	MMU Control Register (MMUCR) .....	162
7.2.6	Page Table Entry Assistance Register (PTEA) .....	166
7.2.7	Physical Address Space Control Register (PASCR) .....	167
7.2.8	Instruction Re-Fetch Inhibit Control Register (IRMCR) .....	168
7.3	TLB Functions (TLB Compatible Mode; MMUCR.ME = 0) .....	170
7.3.1	Unified TLB (UTLB) Configuration .....	170
7.3.2	Instruction TLB (ITLB) Configuration .....	173
7.3.3	Address Translation Method .....	173
7.4	TLB Functions (TLB Extended Mode; MMUCR.ME = 1) .....	176
7.4.1	Unified TLB (UTLB) Configuration .....	176
7.4.2	Instruction TLB (ITLB) Configuration .....	179
7.4.3	Address Translation Method .....	180
7.5	MMU Functions .....	183
7.5.1	MMU Hardware Management .....	183
7.5.2	MMU Software Management .....	183
7.5.3	MMU Instruction (LDTLB) .....	184
7.5.4	Hardware ITLB Miss Handling .....	186
7.5.5	Avoiding Synonym Problems .....	187
7.6	MMU Exceptions .....	189
7.6.1	Instruction TLB Multiple Hit Exception .....	189
7.6.2	Instruction TLB Miss Exception .....	190
7.6.3	Instruction TLB Protection Violation Exception .....	191
7.6.4	Data TLB Multiple Hit Exception .....	192
7.6.5	Data TLB Miss Exception .....	192
7.6.6	Data TLB Protection Violation Exception .....	194
7.6.7	Initial Page Write Exception .....	195
7.7	Memory-Mapped TLB Configuration .....	197
7.7.1	ITLB Address Array .....	198
7.7.2	ITLB Data Array (TLB Compatible Mode) .....	199
7.7.3	ITLB Data Array (TLB Extended Mode) .....	200
7.7.4	UTLB Address Array .....	202
7.7.5	UTLB Data Array (TLB Compatible Mode) .....	203
7.7.6	UTLB Data Array (TLB Extended Mode) .....	204
7.8	Usage Notes .....	206
7.8.1	Note on Using LDTLB Instruction .....	206

Section 8	Caches.....	207
8.1	Features.....	207
8.2	Register Descriptions.....	211
8.2.1	Cache Control Register (CCR) .....	212
8.2.2	Queue Address Control Register 0 (QACR0).....	214
8.2.3	Queue Address Control Register 1 (QACR1).....	215
8.2.4	On-Chip Memory Control Register (RAMCR) .....	216
8.3	Operand Cache Operation.....	218
8.3.1	Read Operation .....	218
8.3.2	Prefetch Operation .....	219
8.3.3	Write Operation .....	220
8.3.4	Write-Back Buffer .....	221
8.3.5	Write-Through Buffer.....	221
8.3.6	OC Two-Way Mode .....	222
8.4	Instruction Cache Operation .....	223
8.4.1	Read Operation .....	223
8.4.2	Prefetch Operation .....	223
8.4.3	IC Two-Way Mode.....	224
8.4.4	Instruction Cache Way Prediction Operation .....	224
8.5	Cache Operation Instruction .....	225
8.5.1	Coherency between Cache and External Memory .....	225
8.5.2	Prefetch Operation .....	227
8.6	Memory-Mapped Cache Configuration .....	228
8.6.1	IC Address Array.....	228
8.6.2	IC Data Array .....	230
8.6.3	OC Address Array .....	230
8.6.4	OC Data Array .....	232
8.6.5	Memory-Mapped Cache Associative Write Operation.....	233
8.7	Store Queues .....	234
8.7.1	SQ Configuration.....	234
8.7.2	Writing to SQ.....	234
8.7.3	Transfer to External Memory .....	235
8.7.4	Determination of SQ Access Exception.....	236
8.7.5	Reading from SQ.....	236
Section 9	On-Chip Memory .....	237
9.1	Features.....	237
9.2	Register Descriptions.....	238
9.2.1	On-Chip Memory Control Register (RAMCR) .....	239
9.3	Operation .....	241



9.3.1	Instruction Fetch Access from the CPU .....	241
9.3.2	Operand Access from the CPU and Access from the FPU .....	241
9.3.3	Access from the SuperHyway Bus Master Module .....	241
9.4	On-Chip Memory Protective Functions .....	242
9.5	Usage Notes .....	243
9.5.1	Page Conflict .....	243
9.5.2	Access Across Different Pages .....	243
9.5.3	On-Chip Memory Coherency .....	243
9.5.4	Sleep Mode .....	243
Section 10 Clock Pulse Generator (CPG) .....		245
10.1	Features .....	245
10.2	Input/Output Pins .....	248
10.3	Clock Operating Mode .....	249
10.4	Register Descriptions .....	250
10.4.1	Frequency Control Register (FRQCR) .....	251
10.4.2	PLL Control Register (PLLCR) .....	253
10.4.3	VDC2 Clock Control Register (VDC2CLKCR) .....	254
Section 11 Memory Controller Unit (MCU) .....		255
11.1	Features .....	255
11.2	Input/Output Pins .....	258
11.3	Area Overview .....	260
11.3.1	Space Divisions .....	260
11.3.2	Memory Bus Width .....	261
11.3.3	Endian Setting .....	262
11.4	Register Description .....	263
11.4.1	Version Control Register (VCR) .....	266
11.4.2	Memory Interface Mode Register (MIM) .....	267
11.4.3	SDRAM Control Register (SCR) .....	271
11.4.4	SDRAM Timing Register (STR) .....	273
11.4.5	SDRAM Row Attribute Register (SDRA) .....	276
11.4.6	SDRAM Mode Register (SDMR) .....	278
11.4.7	Arbitration Mode Register (AMR) .....	279
11.4.8	Linear-to-Tiled Memory Address Translation Control Register (LTCn) .....	280
11.4.9	Linear-to-Tiled Memory Address Translation Area Start Address Register (LTADn) .....	282
11.4.10	Linear-to-Tiled Memory Address Translation Area Start Address Mask Register (LTAMn) .....	283
11.4.11	Request Mask Setting Register (RQM) .....	284

11.4.12	Bus Control Register (BCR) .....	287
11.4.13	CS0 Bus Control Register (CS0BCR) .....	290
11.4.14	$\overline{CSn}$ Wait Control Register (CSnWCR) .....	295
11.4.15	CS3 Bus Control Register (CS3BCR) .....	301
11.5	Operation .....	307
11.5.1	Endian/Access Size and Data Alignment.....	307
11.5.2	Data Alignment in Various Modules .....	316
11.6	SRAM Interface .....	316
11.6.1	Basic Timing.....	316
11.6.2	Wait Cycle Control.....	320
11.6.3	Read-Strobe Negate Timing .....	322
11.7	SDRAM Interface .....	323
11.7.1	SDRAM Direct Connection.....	323
11.7.2	Address Multiplexing .....	326
11.7.3	Burst Read .....	328
11.7.4	Burst Write .....	329
11.7.5	Single Read.....	330
11.7.6	Single Write.....	331
11.7.7	Bank Open Mode.....	332
11.7.8	Refresh.....	337
11.7.9	SDRAM Initialization Sequence.....	338
11.8	Wait Cycles between Accesses .....	339
11.8.1	Wait Cycles between Accesses to Area 0 or 3 .....	339
11.8.2	Wait Cycles between Accesses to Area 1 or 2 .....	339
11.8.3	Wait Cycles between Access to Area 1 or 2 and the Subsequent Access to Area 0 or 3 .....	339
11.9	Bus Arbitration .....	340
11.9.1	Arbitration of Accesses between Internal Modules .....	340
11.9.2	Multi-Step Arbitration .....	343
11.9.3	Bus Requests from External Devices.....	346
11.9.4	Bus Release and Recovery Sequences .....	347
11.9.5	Cooperation between Master and Slave.....	349
11.10	Data Coherency.....	349
11.11	Linear-to-Tiled Memory Address Translation .....	352
11.12	Usage Notes .....	357
11.12.1	Refresh.....	357
11.12.2	External Bus Arbitration.....	357

Section 12	Direct Memory Access Controller (DMAC)	359
12.1	Features	359
12.2	Input/Output Pins	361
12.3	Register Descriptions	362
12.3.1	DMA Source Address Registers (SAR0 to SAR5)	366
12.3.2	DMA Source Address Registers (SARB0 to SARB3)	367
12.3.3	DMA Destination Address Registers (DAR0 to DAR5)	367
12.3.4	DMA Destination Address Registers (DARB0 to DARB3)	368
12.3.5	DMA Transfer Count Registers (TCR0 to TCR5)	368
12.3.6	DMA Transfer Count Registers (TCRB0 to TCRB3)	369
12.3.7	DMA Channel Control Registers (CHCR0 to CHCR5)	370
12.3.8	DMA Operation Register 0 (DMAOR0)	378
12.3.9	DMA Extended Resource Selectors (DMARS0 to DMARS2)	381
12.4	Operation	384
12.4.1	DMA Transfer Requests	384
12.4.2	Channel Priority	388
12.4.3	DMA Transfer Types	391
12.4.4	DMA Transfer Flow	398
12.4.5	Repeat Mode Transfer	400
12.4.6	Reload Mode Transfer	401
12.4.7	DREQ Pin Sampling Timing	402
12.5	Usage Notes	406
12.5.1	Module Stop	406
12.5.2	Address Error	406
12.5.3	Notes on Burst Mode Transfer	406
12.5.4	DACK Output Division and External Request	407
12.5.5	DMA Transfer to DMAC Prohibited	407
12.5.6	NMI Interrupt	407
Section 13	Interrupt Controller (INTC)	409
13.1	Features	409
13.1.1	Interrupt Method	411
13.1.2	Interrupt Types in INTC	412
13.2	Input/Output Pins	414
13.3	Register Descriptions	415
13.3.1	Interrupt Control Register 0 (ICR0)	419
13.3.2	Interrupt Control Register 1 (ICR1)	421
13.3.3	Interrupt Priority Register (INTPRI)	422
13.3.4	Interrupt Source Register (INTREQ)	423

13.3.5	Interrupt Mask Register (INTMSK) .....	424
13.3.6	Interrupt Mask Clear Register (INTMSKCLR) .....	425
13.3.7	NMI Flag Control Register (NMIFCR) .....	426
13.3.8	User Interrupt Mask Level Register (USERIMASK) .....	428
13.3.9	On-Chip Module Interrupt Priority Registers (INT2PRI0 to INT2PRI12) .....	430
13.3.10	Interrupt Source Register 0 (Mask State is not affected) (INT2A0) .....	432
13.3.11	Interrupt Source Register 01 (Mask State is not affected) (INT2A01) .....	434
13.3.12	Interrupt Source Register (Mask State is affected) (INT2A1) .....	436
13.3.13	Interrupt Source Register 11 (Mask State is affected) (INT2A11) .....	438
13.3.14	Interrupt Mask Register (INT2MSKR).....	440
13.3.15	Interrupt Mask Register 1 (INT2MSKR1).....	442
13.3.16	Interrupt Mask Clear Register (INT2MSKCR) .....	444
13.3.17	Interrupt Mask Clear Register 1 (INT2MSKCR1).....	446
13.3.18	On-Chip Module Interrupt Source Registers (INT2B0 and INT2B2 to INT2B7) .....	448
13.3.19	GPIO Interrupt Set Register (INT2GPIC) .....	453
13.4	Interrupt Sources.....	455
13.4.1	NMI Interrupt.....	455
13.4.2	IRQ Interrupts .....	455
13.4.3	On-Chip Module Interrupts .....	456
13.4.4	Interrupt Priority Level of On-Chip Module Interrupts .....	456
13.4.5	Interrupt Exception Handling and Priority.....	457
13.5	Operation .....	462
13.5.1	Interrupt Sequence .....	462
13.5.2	Multiple Interrupts .....	464
13.5.3	Interrupt Masking by MAI Bit.....	464
13.6	Interrupt Response Time.....	465
13.7	Usage Notes .....	466
13.7.1	To Clear Interrupt Request When Holding Function Selected .....	466
13.7.2	Notes on Setting IRQ1 and IRQ0 Pin Function .....	467
13.7.3	To Clear IRQ Interrupt Requests .....	467
Section 14	Timer Unit (TMU).....	469
14.1	Features.....	469
14.2	Input/Output Pins.....	471
14.3	Register Descriptions.....	472
14.3.1	Timer Output Control Register (TOCR).....	474
14.3.2	Timer Start Register (TSTR0, TSTR1).....	475
14.3.3	Timer Constant Register (TCORn) (n = 0 to 5) .....	477
14.3.4	Timer Counter (TCNTn) (n = 0 to 5).....	477

14.3.5	Timer Control Registers (TCRn) (n = 0 to 5)	478
14.3.6	Input Capture Register 2 (TCPR2)	480
14.4	Operation	481
14.4.1	Counter Operation	481
14.4.2	Input Capture Function	484
14.5	Interrupts	486
14.6	Usage Notes	487
14.6.1	Register Writes	487
14.6.2	Reading from TCNT	487
14.6.3	External Clock Frequency	487
Section 15 Serial Communication Interface with FIFO (SCIF)		489
15.1	Features	489
15.2	Input/Output Pins	492
15.3	Register Descriptions	493
15.3.1	Receive Shift Register (SCRSR)	496
15.3.2	Receive FIFO Data Register (SCFRDR)	496
15.3.3	Transmit Shift Register (SCTSR)	497
15.3.4	Transmit FIFO Data Register (SCFTDR)	497
15.3.5	Serial Mode Register (SCSMR)	498
15.3.6	Serial Control Register (SCSCR)	501
15.3.7	Serial Status Register (SCFSR)	505
15.3.8	Bit Rate Register (SCBRR)	513
15.3.9	FIFO Control Register (SCFCR)	518
15.3.10	FIFO Data Count Set Register (SCFDR)	521
15.3.11	Serial Port Register (SCSPTR)	522
15.3.12	Line Status Register (SCLSR)	525
15.3.13	Serial Extension Mode Register (SCEMR)	526
15.4	Operation	527
15.4.1	Overview	527
15.4.2	Operation in Asynchronous Mode	530
15.4.3	Operation in Clock Synchronous Mode	541
15.5	SCIF Interrupts	549
15.6	Usage Notes	550
15.6.1	SCFTDR Writing and TDFE Flag	550
15.6.2	SCFRDR Reading and RDF Flag	550
15.6.3	Break Detection and Processing	551
15.6.4	Sending a Break Signal	551
15.6.5	Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)	551
15.6.6	Selection of Base Clock in Asynchronous Mode	553

Section 16	I <sup>2</sup> C Bus Interface (IIC)	555
16.1	Features	555
16.2	Input/Output Pins	556
16.3	Register Descriptions	556
16.3.1	Slave Control Register (ICSCR)	558
16.3.2	Slave Status Register (ICSSR)	559
16.3.3	Slave Interrupt Enable Register (ICSIER)	562
16.3.4	Slave Address Register (ICSAR)	563
16.3.5	Master Control Register (ICMCR)	564
16.3.6	Master Status Register (ICMSR)	566
16.3.7	Master Interrupt Enable Register (ICMIER)	568
16.3.8	Master Address Register (ICMAR)	569
16.3.9	Clock Control Register (ICCCR)	569
16.3.10	Receive and Transmit Data Registers (ICRXD and ICTXD)	571
16.4	Operations	572
16.4.1	Data and Clock Filters	572
16.4.2	Clock Generator	572
16.4.3	Master/Slave Interfaces	572
16.4.4	Software Status Interlocking	572
16.4.5	I <sup>2</sup> C Bus Data Format	574
16.4.6	7-Bit Address Format	575
16.4.7	10-Bit Address Format	576
16.4.8	Master Transmit Operation	578
16.4.9	Master Receive Operation	580
16.5	Programming Examples	582
16.5.1	Master Transmitter	582
16.5.2	Master Receiver	583
16.5.3	Master Transmitter - Restart - Master Receiver	584
Section 17	ATAPI	587
17.1	Features	587
17.2	Input/Output Pins	588
17.3	Register Description	589
17.3.1	ATAPI Control (ATAPI_CONTROL)	592
17.3.2	ATAPI Status (ATAPI_STATUS)	594
17.3.3	Interrupt Enable (ATAPI_INT_ENABLE)	596
17.3.4	PIO Timing Register (ATAPI_PIO_TIMING)	597
17.3.5	Multiword DMA Timing Register (ATAPI_MULTI_TIMING)	598
17.3.6	Ultra DMA Timing Register (ATAPI_ULTRA_TIMING)	600
17.3.7	Descriptor Table Base Address Register (ATAPI_DTB_ADR)	601

17.3.8	Descriptor Table .....	602
17.3.9	Termination Flag and Descriptor DMA Start Address .....	603
17.3.10	Descriptor DMA Transfer Count .....	604
17.3.11	DMA Start Address Register (ATAPI_DMA_START_ADR).....	605
17.3.12	DMA Transfer Count Register (ATAPI_DMA_TRANS_CNT) .....	606
17.3.13	ATAPI Control 2 (ATAPI_CONTROL2) .....	607
17.3.14	ATAPI Signal Status Register (ATAPI_SIG_ST) .....	608
17.3.15	Byteswap (ATAPI_BYTE_SWAP).....	609
17.3.16	ATAPI Data Bus Alignment.....	610
17.4	Functional Description .....	611
17.4.1	Data Transfer Modes .....	611
17.4.2	Descriptor Function .....	611
17.5	Operating Procedure .....	612
17.5.1	Initialization .....	612
17.5.2	Procedure in PIO Transfer Mode .....	612
17.5.3	Procedure in Multiword DMA Transfer Mode .....	613
17.5.4	Procedure in Ultra DMA Transfer Mode .....	616
17.5.5	Procedure in Hardware Reset for ATAPI Device .....	617
Section 18	Serial Sound Interface (SSI) .....	619
18.1	Features .....	619
18.1.1	SSI Module Configuration .....	619
18.1.2	SSI Features .....	619
18.2	Input/Output Pins .....	621
18.3	Register Descriptions .....	621
18.3.1	DMA Mode Registers 0 to 5 (SSIDMMR0 to SSIDMMR5).....	633
18.3.2	RDMA Transfer Source Address Registers 0 to 5 (SSIRDMADR0 to SSIRDMADR5) .....	634
18.3.3	RDMA Transfer Word Count Registers 0 to 5 (SSIRDMCNTR0 to SSIRDMCNTR5).....	635
18.3.4	WDMA Transfer Destination Address Registers 0 to 5 (SSIWDMADR0 to SSIWDMADR5) .....	637
18.3.5	WDMA Transfer Word Count Registers 0 to 5 (SSIWDMCNTR0 to SSIWDMCNTR5).....	637
18.3.6	DMA Control Registers 0 to 5 (SSIDMCOR0 to SSIDMCOR5).....	638
18.3.7	Transmit Suspension Block Counters 0 to 5 (SSISTPBLCNT0 to SSISTPBLCNT5) .....	649
18.3.8	Transmit Suspension Transfer Data Registers 0 to 5 (SSISTPDR0 to SSISTPDR5) .....	650
18.3.9	Block Count Source Registers 0 to 5 (SSIBLCNTR0 to SSIBLCNTR5) .....	651

18.3.10	Block Counters 0 to 5 (SSIBLCNT0 to SSIBLCNT5) .....	652
18.3.11	n-Times Block Transfer Interrupt Count Source Registers 0 to 5 (SSIBLNCNTR0 to SSIBLNCNTR5) .....	652
18.3.12	n-Times Block Counters 0 to 5 (SSIBLNCNT0 to SSIBLNCNT5) .....	653
18.3.13	DMA Operation Registers 0 and 1 (SSIDMAOR0 and SSIDMAOR1) .....	654
18.3.14	Interrupt Status Registers 0 and 1 (SSIDMINTSR0 and SSIDMINTSR1) .....	656
18.3.15	Interrupt Mask Registers 0 and 1 (SSIDMINTMR0 and SSIDMINTMR1) .....	661
18.3.16	Control Registers 0 to 5 (SSICR0 to SSICR5) .....	664
18.3.17	Status Registers 0 to 5 (SSISR0 to SSISR5) .....	671
18.3.18	Transmit Data Registers 0 to 5 (SSITDR0 to SSITDR5) .....	677
18.3.19	Receive Data Registers 0 to 5 (SSIRDR0 to SSIRDR5) .....	677
18.4	Operation .....	678
18.4.1	Operation of SSI_CLKSEL .....	678
18.4.2	Operation of SSI_DMAC0 and SSI_DMAC1 .....	678
18.4.3	Operation of SSI_CH0 to SSI_CH5 .....	680
18.5	Usage Note .....	698
18.5.1	Restrictions when an Overflow Occurs during Receive DMA Operation .....	698
18.5.2	Restrictions during Operation in Slave Mode .....	698
18.5.3	Restrictions when Specify Each Register .....	698
Section 19	Ethernet Controller (EtherC) .....	701
19.1	Features .....	701
19.2	Input/Output Pins .....	702
19.3	Register Descriptions .....	703
19.3.1	EtherC Mode Register (ECMR) .....	705
19.3.2	EtherC Status Register (ECSR) .....	709
19.3.3	EtherC Interrupt Permission Register (ECSIPR) .....	711
19.3.4	PHY Interface Register (PIR) .....	712
19.3.5	MAC Address High Register (MAHR) .....	713
19.3.6	MAC Address Low Register (MALR) .....	714
19.3.7	Receive Frame Length Register (RFLR) .....	715
19.3.8	PHY Status Register (PSR) .....	716
19.3.9	Transmit Retry Over Counter Register (TROCR) .....	717
19.3.10	Delayed Collision Detect Counter Register (CDCR) .....	718
19.3.11	Lost Carrier Counter Register (LCCR) .....	719
19.3.12	Carrier Not Detect Counter Register (CNDCR) .....	720
19.3.13	CRC Error Frame Receive Counter Register (CEFCR) .....	721
19.3.14	Frame Receive Error Counter Register (FRECR) .....	722
19.3.15	Too-Short Frame Receive Counter Register (TSFRCR) .....	723
19.3.16	Too-Long Frame Receive Counter Register (TLFRCR) .....	724



19.3.17	Residual-Bit Frame Receive Counter Register (RFCR)	725
19.3.18	Multicast Address Frame Receive Counter Register (MAFCR)	726
19.3.19	IPG Register (IPGR)	727
19.3.20	Automatic PAUSE Frame Register (APR)	728
19.3.21	Manual PAUSE Frame Register (MPR)	729
19.3.22	Automatic PAUSE Frame Retransmit Count Register (TPAUSER)	730
19.3.23	Random Number Generation Counter Upper Limit Setting Register (RDMLR)	731
19.3.24	PAUSE Frame Receive Counter Register (RFCF)	732
19.3.25	PAUSE Frame Retransmit Counter Register (TPAUSECR)	733
19.3.26	Broadcast Frame Receive Count Setting Register (BCFRR)	734
19.4	Operation	735
19.4.1	Transmission	735
19.4.2	Reception	737
19.4.3	MII Frame Timing	739
19.4.4	Accessing MII Registers	741
19.4.5	Magic Packet Detection	744
19.4.6	Operation by IPG Setting	745
19.4.7	Flow Control	745
19.5	Connection to LSI	747
19.6	Usage Notes	748

Section 20	Ethernet Controller Direct Memory Access Controller (E-DMAC)	749
20.1	Features	749
20.2	Register Descriptions	751
20.2.1	E-DMAC Mode Register (EDMR)	753
20.2.2	E-DMAC Transmit Request Register (EDTRR)	754
20.2.3	E-DMAC Receive Request Register (EDRRR)	755
20.2.4	Transmit Descriptor List Start Address Register (TDLAR)	756
20.2.5	Receive Descriptor List Start Address Register (RDLAR)	757
20.2.6	E-MAC/E-DMAC Status Register (EESR)	758
20.2.7	E-MAC/E-DMAC Status Interrupt Permission Register (EESIPR)	763
20.2.8	Transmit/Receive Status Copy Enable Register (TRSCER)	766
20.2.9	Receive Missed-Frame Counter Register (RMFCR)	769
20.2.10	Transmit FIFO Threshold Register (TFTR)	770
20.2.11	FIFO Depth Register (FDR)	772
20.2.12	Receiving Method Control Register (RMCR)	773
20.2.13	Transmit FIFO Underrun Counter (TFUCR)	775
20.2.14	Receive FIFO Overflow Counter (RFOCR)	776

20.2.15	Receive Buffer Write Address Register (RBWAR).....	777
20.2.16	Receive Descriptor Fetch Address Register (RDFAR).....	778
20.2.17	Transmit Buffer Read Address Register (TBRAR) .....	779
20.2.18	Transmit Descriptor Fetch Address Register (TDFAR) .....	780
20.2.19	Flow Control Start FIFO Threshold Setting Register (FCFTR) .....	781
20.2.20	Receive Data Padding Insert Register (RPADIR) .....	783
20.2.21	Transmit Interrupt Setting Register (TRIMD) .....	784
20.2.22	Independent Output Signal Setting Register (IOSR) .....	785
20.3	Operation .....	786
20.3.1	Descriptor Lists and Data Buffers.....	786
20.3.2	Transmission.....	795
20.3.3	Reception .....	797
20.3.4	Transmit/Receive Processing of Multi-Buffer Frame (Single-Frame/ Multi-Descriptor).....	799
Section 21 USB 2.0 Host/Function Module (USB).....		801
21.1	Features.....	801
21.2	Input / Output Pins .....	804
21.3	Register Description .....	805
21.3.1	System Configuration Control Register (SYSCFG) .....	811
21.3.2	CPU Bus Wait Setting Register (BUSWAIT) .....	815
21.3.3	System Configuration Status Register (SYSSTS).....	816
21.3.4	Device State Control Register (DVSTCTR).....	818
21.3.5	Test Mode Register (TESTMODE) .....	823
21.3.6	DMA-FIFO Bus Configuration Registers (D0FBCFG, D1FBCFG) .....	826
21.3.7	FIFO Port Registers (CFIFO, D0FIFO, D1FIFO) .....	827
21.3.8	FIFO Port Select Registers (CFIFOSEL, D0FIFOSEL, D1FIFOSEL).....	830
21.3.9	FIFO Port Control Registers (CFIFOCTR, D0FIFOCTR, D1FIFOCTR) .....	837
21.3.10	Interrupts Enable Register 0 (INTENB0) .....	841
21.3.11	Interrupt Enable Register 1 (INTENB1).....	843
21.3.12	BRDY Interrupt Enable Register (BRDYENB) .....	845
21.3.13	NRDY Interrupt Enable Register (NRDYENB).....	847
21.3.14	BEMP Interrupt Enable Register (BEMPENB).....	849
21.3.15	SOF Control Register (SOFCFG).....	851
21.3.16	Interrupt Status Register 0 (INTSTS0) .....	852
21.3.17	Interrupt Status Register 1 (INTSTS1) .....	858
21.3.18	BRDY Interrupt Status Register (BRDYSTS).....	864
21.3.19	NRDY Interrupt Status Register (NRDYSTS) .....	866
21.3.20	BEMP Interrupt Status Register (BEMPSTS) .....	868
21.3.21	Frame Number Register (FRMNUM) .....	869

21.3.22	μFrame Number Register (UFRMNUM)	872
21.3.23	USB Address Register (USBADDR)	873
21.3.24	USB Request Type Register (USBREQ)	874
21.3.25	USB Request Value Register (USBVAL)	876
21.3.26	USB Request Index Register (USBINDX)	877
21.3.27	USB Request Length Register (USBLENG)	878
21.3.28	DCP Configuration Register (DCPCFG)	879
21.3.29	DCP Maximum Packet Size Register (DCPMAXP)	880
21.3.30	DCP Control Register (DCPCTR)	881
21.3.31	Pipe Window Select Register (PIPESEL)	891
21.3.32	Pipe Configuration Register (PIPECFG)	892
21.3.33	Pipe Buffer Setting Register (PIPEBUF)	899
21.3.34	Pipe Maximum Packet Size Register (PIPEMAXP)	902
21.3.35	Pipe Timing Control Register (PIPEPERI)	904
21.3.36	PIPE <sub>n</sub> Control Registers (PIPE <sub>n</sub> CTR) (n = 1 to 9)	906
21.3.37	PIPE <sub>n</sub> Transaction Counter Enable Registers (PIPE <sub>n</sub> TRE) (n = 1 to 5)	926
21.3.38	PIPE <sub>n</sub> Transaction Counter Registers (PIPE <sub>n</sub> TRN) (n = 1 to 5)	928
21.3.39	Device Address n Configuration Registers (DEVADD <sub>n</sub> ) (n = 0 to A)	930
21.4	Operation	933
21.4.1	System Control and Oscillation Control	933
21.4.2	Interrupt Functions	935
21.4.3	Pipe Control	959
21.4.4	FIFO Buffer Memory	969
21.4.5	Control Transfers (DCP)	979
21.4.6	Bulk Transfers (PIPE1 to PIPE5)	983
21.4.7	Interrupt Transfers (PIPE6 to PIPE9)	985
21.4.8	Isochronous Transfers (PIPE1 and PIPE2)	986
21.4.9	SOF Interpolation Function	997
21.4.10	Pipe Schedule	998
21.5	Usage Notes	1000
21.5.1	USB Startup and Stop Procedures	1000
Section 22	LCD Controller (LCDC)	1001
22.1	Features	1001
22.2	Input/Output Pins	1003
22.3	Register Configuration	1004
22.3.1	LCDC Input Clock Register (LDICKR)	1007
22.3.2	LCDC Module Type Register (LDMTR)	1009
22.3.3	LCDC Data Format Register (LDDFR)	1012
22.3.4	LCDC Start Address Register for Upper Display Data Fetch (LDSARU)	1014

22.3.5	LCDC Start Address Register for Lower Display Data Fetch (LDSARL) .....	1015
22.3.6	LCDC Line Address Offset Register for Display Data Fetch (LDLAOR) .....	1016
22.3.7	LCDC Palette Control Register (LDPALCR).....	1017
22.3.8	Palette Data Registers 00 to FF (LDPR00 to LDPRFF) .....	1018
22.3.9	LCDC Horizontal Character Number Register (LDHCNR) .....	1019
22.3.10	LCDC Horizontal Sync Signal Register (LDHSYNR).....	1020
22.3.11	LCDC Vertical Display Line Number Register (LDVDLNR) .....	1021
22.3.12	LCDC Vertical Total Line Number Register (LDVTLNR).....	1022
22.3.13	LCDC Vertical Sync Signal Register (LDVSYNR) .....	1023
22.3.14	LCDC AC Modulation Signal Toggle Line Number Register (LDACLNR) .....	1024
22.3.15	LCDC Interrupt Control Register (LDINTR) .....	1025
22.3.16	LCDC Power Management Mode Register (LDPMMR) .....	1028
22.3.17	LCDC Power-Supply Sequence Period Register (LDPSPR).....	1030
22.3.18	LCDC Control Register (LDCNTR).....	1032
22.3.19	LCDC User Specified Interrupt Control Register (LDUINTR).....	1033
22.3.20	LCDC User Specified Interrupt Line Number Register (LDUINTLNR) .....	1035
22.3.21	LCDC Memory Access Interval Number Register (LDLIRNR) .....	1036
22.4	Operation .....	1037
22.4.1	LCD Module Sizes which can be Displayed in this LCDC .....	1037
22.4.2	Color Palette Specification .....	1039
22.4.3	Data Format .....	1040
22.4.4	Setting the Display Resolution .....	1044
22.4.5	Power-Supply Control Sequence .....	1044
22.5	Clock and LCD Data Signal Examples .....	1050
22.6	Usage Notes .....	1063
22.6.1	Procedure for Halting Access to Display Data Storage VRAM (SDRAM in Area 1 or 2) .....	1063
22.6.2	Notes on Holding the Access Request by MCU .....	1063
Section 23 G2D .....		1065
23.1	Basic Functions.....	1065
23.1.1	List of Commands and Rendering Attributes .....	1065
23.1.2	Basic Functions.....	1068
23.1.3	Coordinate Systems .....	1078
23.1.4	Data Formats.....	1083
23.1.5	Rendering Attributes.....	1084
23.2	Display List.....	1097
23.2.1	4-Vertex Screen Drawing Commands .....	1097
23.2.2	Line Drawing Commands.....	1108
23.2.3	Work Screen Drawing Commands .....	1139

23.2.4	Work Line Drawing Commands .....	1151
23.2.5	Rectangle Drawing Commands .....	1155
23.2.6	Control Commands .....	1166
23.3	Register Specifications.....	1184
23.3.1	System Control Registers.....	1190
23.3.2	Memory Control Registers.....	1196
23.3.3	Color Control Registers .....	1201
23.3.4	Rendering Control Registers.....	1204
23.3.5	Coordinate Transformation Control Registers .....	1212
Section 24 Video Display Controller (VDC2).....		1221
24.1	Overview .....	1221
24.2	Features.....	1221
24.3	Input/Output Pins .....	1223
24.4	VDC2 Configuration.....	1224
24.5	Functional Descriptions .....	1226
24.5.1	Graphics (Layers 1 to 4) .....	1226
24.5.2	Sync Signal Generation.....	1227
24.5.3	External Sync Mode.....	1230
24.5.4	Digital Video Output .....	1230
24.5.5	Conversion from RGB565 to YC444.....	1231
24.5.6	Conversion from YC444 to YC422 .....	1231
24.5.7	Data Enable Signal (Composite).....	1232
24.6	Register Descriptions .....	1233
24.6.1	Graphics Block Control Registers (GRCMEN1 to GRCMEN4).....	1239
24.6.2	Bus Control Registers (GRCBUSCNT1 to GRCBUSCNT4).....	1241
24.6.3	Graphic Image Base Address Registers (GROPSADR1 to GROPSADR4) .....	1242
24.6.4	Graphic Image Area Registers (GROPSWH1 to GROPSWH4).....	1243
24.6.5	Graphic Image Line Offset Registers (GROPST1 to GROPST4).....	1245
24.6.6	Graphic Image Start Position Registers (GROPDPHV1 to GROPDPHV4).....	1246
24.6.7	$\alpha$ Control Area Registers (GROPEWH2 to GROPEWH4) .....	1246
24.6.8	$\alpha$ Control Area Start Position Registers (GROPEDPHV2 to GROPEDPHV4).....	1247
24.6.9	$\alpha$ Control Registers (GROPEDPA2 to GROPEDPA4) .....	1249
24.6.10	Chroma-Key Control Registers (GROPCKY0_2 to GROPCKY0_4).....	1251
24.6.11	Chroma-Key Color Registers (GROPCKY1_2 to GROPCKY1_4).....	1252
24.6.12	Color Registers for Outside of Graphic Image Area (GROPBASERGB1 to GROPBASERGB4).....	1253
24.6.13	SG Mode Register (SGMODE) .....	1255
24.6.14	Interrupt Output Control Register (SGINTCNT).....	1257

24.6.15	Sync Signal Control Register (SYNCNT) .....	1258
24.6.16	External Sync Signal Input Timing Control Register (EXTSYNCNT) .....	1261
24.6.17	Sync Signal Size Register (SYNSIZE) .....	1262
24.6.18	Vertical Sync Signal Timing Control Register (VSYNCTIM).....	1263
24.6.19	Horizontal Sync Signal Timing Control Register (HSYNCTIM).....	1264
24.6.20	Gate Clock Signal Timing Control Register (CLSTIM).....	1265
24.6.21	Sampling Start Signal Timing Control Register (SPLTIM).....	1266
24.6.22	Gate Control Signal Timing Control Register (COMTIM) .....	1267
24.6.23	SGDE Area Start Position Register (SGDESTART).....	1268
24.6.24	SGDE Area Size Register (SGDESIZE).....	1270
24.6.25	CDE Chroma-Key Color Register (CDECRKY).....	1271
24.6.26	T1004 Control Register (T1004CNT) .....	1272
24.6.27	T1004 Video Start Position Register (T1004OFFSET) .....	1273
24.7	Operating Procedures.....	1275
24.7.1	Display Control Block .....	1275
24.7.2	Graphics Blocks.....	1275
Section 25	NAND Flash Memory Controller (FLCTL).....	1277
25.1	Features.....	1277
25.2	Input/Output Pins .....	1281
25.3	Register Descriptions.....	1282
25.3.1	Common Control Register (FLCMNCR) .....	1284
25.3.2	Command Control Register (FLCMDCR).....	1287
25.3.3	Command Code Register (FLCMCDR) .....	1290
25.3.4	Address Register (FLADR) .....	1291
25.3.5	Address Register 2 (FLADR2) .....	1293
25.3.6	Data Counter Register (FLDTCNTR) .....	1294
25.3.7	Data Register (FLDATAR) .....	1295
25.3.8	Interrupt DMA Control Register (FLINTDMACR) .....	1296
25.3.9	Ready Busy Timeout Setting Register (FLBSYTMR) .....	1301
25.3.10	Ready Busy Timeout Counter (FLBSYCNT).....	1302
25.3.11	Data FIFO Register (FLDTFIFO).....	1303
25.3.12	Control Code FIFO Register (FLECFIFO).....	1304
25.3.13	Transfer Control Register (FLTRCR).....	1305
25.4	Operation .....	1306
25.4.1	Operating Modes .....	1306
25.4.2	Register Setting Procedure.....	1306
25.4.3	Command Access Mode .....	1308
25.4.4	Sector Access Mode .....	1311
25.4.5	ECC Error Correction .....	1315

25.4.6	Status Read .....	1315
25.5	Interrupt Sources .....	1316
25.6	DMA Transfer Specifications .....	1316
Section 26 Sampling Rate Converter (SRC).....		1317
26.1	Features .....	1317
26.2	Register Descriptions .....	1319
26.2.1	SRC Input Data Register (SRCID) .....	1320
26.2.2	SRC Output Data Register (SRCOD) .....	1321
26.2.3	SRC Input Data Control Register (SRCIDCTRL) .....	1322
26.2.4	SRC Output Data Control Register (SRCODCTRL) .....	1323
26.2.5	SRC Control Register (SRCCTRL) .....	1325
26.2.6	SRC Status Register (SRCSTAT).....	1328
26.3	Operation .....	1331
26.3.1	Initial Setting .....	1331
26.3.2	Data Input .....	1332
26.3.3	Data Output.....	1333
26.4	Interrupts .....	1335
26.5	Usage Note.....	1336
26.5.1	Note on Access Register .....	1336
26.5.2	Note on Flash Processing.....	1336
Section 27 General Purpose I/O (GPIO).....		1337
27.1	Features .....	1337
27.2	Register Descriptions .....	1346
27.2.1	Port A Control Register (PTIO_A) .....	1349
27.2.2	Port B Control Register (PTIO_B).....	1351
27.2.3	Port C Control Register (PTIO_C).....	1353
27.2.4	Port D Control Register (PTIO_D) .....	1355
27.2.5	Port E Control Register (PTIO_E) .....	1357
27.2.6	Port F Control Register (PTIO_F) .....	1359
27.2.7	Port G Control Register (PTIO_G) .....	1361
27.2.8	Port H Control Register (PTIO_H) .....	1363
27.2.9	Port I Control Register (PTIO_I) .....	1365
27.2.10	Port J Control Register (PTIO_J).....	1366
27.2.11	Port A Data Register (PTDAT_A).....	1368
27.2.12	Port B Data Register (PTDAT_B) .....	1369
27.2.13	Port C Data Register (PTDAT_C) .....	1370
27.2.14	Port D Data Register (PTDAT_D).....	1371
27.2.15	Port E Data Register (PTDAT_E).....	1372

27.2.16	Port F Data Register (PTDAT_F) .....	1373
27.2.17	Port G Data Register (PTDAT_G).....	1374
27.2.18	Port H Data Register (PTDAT_H).....	1375
27.2.19	Port I Data Register (PTDAT_I).....	1376
27.2.20	Port J Data Register (PTDAT_J) .....	1377
27.2.21	Input-Pin Pull-Up Control Register (PTPUL_SPCL).....	1378
27.2.22	Pin Select Register 0 (PTSEL_A).....	1379
27.2.23	Pin Select Register 1 (PTSEL_B) .....	1381
27.2.24	Pin Select Register 2 (PTSEL_C) .....	1383
27.2.25	Pin Select Register 3 (PTSEL_D).....	1385
27.2.26	Pin Select Register 4 (PTSEL_E) .....	1387
27.2.27	Pin Select Register 5 (PTSEL_F) .....	1389
27.2.28	Pin Select Register 6 (PTSEL_G).....	1391
27.2.29	Pin Select Register 7 (PTSEL_H).....	1393
27.2.30	Pin Select Register 8 (PTSEL_I) .....	1395
27.2.31	Pin Select Register 9 (PTSEL_J) .....	1397
27.2.32	Pin Select Register 10 (PTSEL_K).....	1399
27.2.33	Pin Select Register 11 (PTSEL_P) .....	1401
27.2.34	Pin Select Register 12 (PTSEL_R) .....	1402
27.2.35	Pin Select Register 13 (PTSEL_S) .....	1404
27.2.36	HI-Z Register A (PTHIZ_A) .....	1406
27.2.37	HI-Z Register B (PTHIZ_B).....	1407
27.2.38	Special Select Register (PTSEL_SPCL).....	1409
27.3	Usage Examples.....	1410
27.3.1	Port Function Select.....	1410
27.3.2	Port Output Function .....	1410
27.3.3	Port Input Function .....	1410
27.3.4	On-Chip Module Function.....	1410
Section 28	Power-Down Mode .....	1411
28.1	Features.....	1411
28.1.1	Types of Power-Down Modes .....	1411
28.2	Input/Output Pins .....	1413
28.3	Register Descriptions.....	1413
28.3.1	Standby Control Register (STBCR).....	1414
28.3.2	Module Stop Register 0 (MSTPCR0).....	1415
28.3.3	Module Stop Register 1 (MSTPCR1).....	1419
28.4	Sleep Mode .....	1420
28.4.1	Transition to Sleep Mode.....	1420
28.4.2	Canceling Sleep Mode.....	1420



28.5	Refresh Standby Mode.....	1421
28.5.1	Transition to Refresh Standby Mode .....	1421
28.5.2	Canceling Refresh Standby Mode.....	1421
28.6	Module Standby Mode.....	1422
28.6.1	Transition to Module Standby Mode .....	1422
28.6.2	Canceling Module Standby Mode.....	1422
28.7	STATUS Pin Signal Change Timing .....	1423
28.7.1	Timing at Reset.....	1423
28.7.2	Timing at Sleep Mode Cancellation.....	1423
Section 29 Watchdog Timer and Reset.....		1425
29.1	Features.....	1425
29.2	Input/Output Pins .....	1427
29.3	Register Descriptions .....	1428
29.3.1	Watchdog Timer Stop Time Register (WDTST).....	1429
29.3.2	Watchdog Timer Control/Status Register (WDTC SR).....	1430
29.3.3	Watchdog timer Base Stop Time Register (WDTBST).....	1431
29.3.4	Watchdog Timer Counter (WDCNT).....	1432
29.3.5	Watchdog Timer Base Counter (WDTBCNT) .....	1432
29.4	Operation .....	1433
29.4.1	Reset request.....	1433
29.4.2	Using watchdog timer mode .....	1434
29.4.3	Using Interval timer mode .....	1434
29.4.4	Time for WDT Overflow .....	1434
29.4.5	Clearing WDT Counter.....	1435
29.5	Status Pin Change Timing during Reset .....	1436
29.5.1	Power-On Reset by PRESET.....	1436
29.5.2	Power-On Reset by Watchdog Timer Overflow .....	1439
Section 30 User Break Controller (UBC) .....		1441
30.1	Features.....	1441
30.2	Register Descriptions .....	1443
30.2.1	Match Condition Setting Registers 0 and 1 (CBR0 and CBR1) .....	1445
30.2.2	Match Operation Setting Registers 0 and 1 (CRR0 and CRR1) .....	1451
30.2.3	Match Address Setting Registers 0 and 1 (CAR0 and CAR1).....	1453
30.2.4	Match Address Mask Setting Registers 0 and 1 (CAMR0 and CAMR1).....	1454
30.2.5	Match Data Setting Register 1 (CDR1) .....	1456
30.2.6	Match Data Mask Setting Register 1 (CDMR1) .....	1457
30.2.7	Execution Count Break Register 1 (CETR1).....	1458
30.2.8	Channel Match Flag Register (CCMFR) .....	1459

30.2.9	Break Control Register (CBCR) .....	1460
30.3	Operation Description .....	1461
30.3.1	Definition of Words Related to Accesses .....	1461
30.3.2	User Break Operation Sequence .....	1461
30.3.3	Instruction Fetch Cycle Break .....	1462
30.3.4	Operand Access Cycle Break .....	1464
30.3.5	Sequential Break .....	1465
30.3.6	Program Counter Value to be Saved .....	1467
30.4	User Break Debugging Support Function .....	1468
30.5	User Break Examples .....	1470
30.6	Usage Notes .....	1474
Section 31 User Debugging Interface (H-UDI) .....		1477
31.1	Features .....	1477
31.2	Input/Output Pins .....	1479
31.3	Boundary Scan TAP Controllers (IDCODE, EXTEST, SAMPLE/PRELOAD, BYPASS, CLAMP and HIGHZ) .....	1480
31.4	Register Descriptions .....	1482
31.4.1	Instruction Register (SDIR) .....	1483
31.4.2	Interrupt Source Register (SDINT) .....	1484
31.4.3	Bypass Register (SDBPR) .....	1485
31.4.4	Boundary Scan Register (SDBSR) .....	1485
31.5	Operation .....	1506
31.5.1	TAP Control .....	1506
31.5.2	H-UDI Reset .....	1507
31.5.3	H-UDI Interrupt .....	1507
31.6	Usage Notes .....	1507
Section 32 List of Registers .....		1509
32.1	Register Addresses .....	1509
32.2	Register States in Each Operation Mode .....	1547
Section 33 Electrical Characteristics .....		1573
33.1	Absolute Maximum Ratings .....	1573
33.2	Power-on/Power-off Sequence .....	1574
33.3	DC Characteristics .....	1575
33.4	AC Characteristics .....	1581
33.4.1	Clock and Control Signal Timing .....	1581
33.4.2	Control Signal Timing .....	1584
33.4.3	Bus Timing .....	1586

33.4.4	INTC Module Signal Timing .....	1603
33.4.5	DMAC Module Signal Timing .....	1605
33.4.6	TMU Module Signal Timing .....	1605
33.4.7	IIC Module Signal Timing .....	1606
33.4.8	SCIF Module SignalTiming .....	1608
33.4.9	SSI Module Signal Timing .....	1609
33.4.10	ATAPI Interface Module Signal timing .....	1612
33.4.11	USB Module Signal Timing .....	1640
33.4.12	GPIO Signal Timing .....	1641
33.4.13	H-UDI Module Signal Timing .....	1642
33.4.14	EtherC Module Signal Timing .....	1644
33.4.15	FLCTL Module Signal Timing .....	1648
33.4.16	LCDC Module Signal Timing .....	1652
33.4.17	VDC2 Module Signal Timing .....	1654
33.5	AC Characteristics Measurement Conditions .....	1657
Appendix .....		1659
A.	CPU Operation Mode Register (CPUOPM) .....	1659
B.	Instruction Prefetching and Its Side Effects .....	1661
C.	Speculative Execution for Subroutine Return .....	1662
D.	Version Registers (PVR, PRR) .....	1663
E.	Pin State .....	1665
F.	Pin Treatment When Not in Use .....	1675
G.	Type Name .....	1683
H.	Package Dimensions .....	1684
Index .....		1685



# Figures

## Section 1 Overview

Figure 1.1	Block Diagram .....	13
Figure 1.2	Pin Arrangement .....	14
Figure 1.3	Physical Address Space (1) .....	28
Figure 1.3	Physical Address Space (2) .....	29

## Section 2 Programming Model

Figure 2.1	Data Formats .....	31
Figure 2.2	CPU Register Configuration in Each Processing Mode .....	35
Figure 2.3	General Registers .....	36
Figure 2.4	Floating-Point Registers .....	38
Figure 2.5	Relationship between SZ bit and Endian .....	44
Figure 2.6	Formats of Byte Data and Word Data in Register .....	46
Figure 2.7	Data Formats in Memory .....	47
Figure 2.8	Processing State Transitions .....	48

## Section 4 Pipelining

Figure 4.1	Basic Pipelines .....	71
Figure 4.2	Instruction Execution Patterns (1) .....	73
Figure 4.2	Instruction Execution Patterns (2) .....	74
Figure 4.2	Instruction Execution Patterns (3) .....	75
Figure 4.2	Instruction Execution Patterns (4) .....	76
Figure 4.2	Instruction Execution Patterns (5) .....	77
Figure 4.2	Instruction Execution Patterns (6) .....	78
Figure 4.2	Instruction Execution Patterns (7) .....	79
Figure 4.2	Instruction Execution Patterns (8) .....	80
Figure 4.2	Instruction Execution Patterns (9) .....	81

## Section 5 Exception Handling

Figure 5.1	Instruction Execution and Exception Handling .....	105
Figure 5.2	Example of General Exception Acceptance Order .....	106

## Section 6 Floating-Point Unit (FPU)

Figure 6.1	Format of Single-Precision Floating-Point Number .....	132
Figure 6.2	Format of Double-Precision Floating-Point Number .....	132
Figure 6.3	Single-Precision NaN Bit Pattern .....	135
Figure 6.4	Floating-Point Registers .....	138
Figure 6.5	Relation between SZ Bit and Endian .....	141

## Section 7 Memory Management Unit (MMU)

Figure 7.1	Role of MMU .....	151
Figure 7.2	Virtual Address Space (AT in MMUCR= 0) .....	152
Figure 7.3	Virtual Address Space (AT in MMUCR= 1) .....	153
Figure 7.4	P4 Area .....	154
Figure 7.5	Physical Address Space .....	156
Figure 7.6	UTLB Configuration (TLB Compatible Mode) .....	170
Figure 7.7	Relationship between Page Size and Address Format (TLB Compatible Mode) .....	172
Figure 7.8	ITLB Configuration (TLB Compatible Mode) .....	173
Figure 7.9	Flowchart of Memory Access Using UTLB (TLB Compatible Mode) .....	174
Figure 7.10	Flowchart of Memory Access Using ITLB (TLB Compatible Mode) .....	175
Figure 7.11	UTLB Configuration (TLB Extended Mode) .....	176
Figure 7.12	Relationship between Page Size and Address Format (TLB Extended Mode) .....	179
Figure 7.13	ITLB Configuration (TLB Extended Mode) .....	179
Figure 7.14	Flowchart of Memory Access Using UTLB (TLB Extended Mode) .....	181
Figure 7.15	Flowchart of Memory Access Using ITLB (TLB Extended Mode) .....	182
Figure 7.16	Operation of LDTLB Instruction (TLB Compatible Mode) .....	185
Figure 7.17	Operation of LDTLB Instruction (TLB Extended Mode) .....	186
Figure 7.18	Memory-Mapped ITLB Address Array .....	198
Figure 7.19	Memory-Mapped ITLB Data Array (TLB Compatible Mode) .....	199
Figure 7.20	Memory-Mapped ITLB Data Array 1 (TLB Extended Mode) .....	200
Figure 7.21	Memory-Mapped ITLB Data Array 2 (TLB Extended Mode) .....	201
Figure 7.22	Memory-Mapped UTLB Address Array .....	203
Figure 7.23	Memory-Mapped UTLB Data Array (TLB Compatible Mode) .....	204
Figure 7.24	Memory-Mapped UTLB Data Array 1 (TLB Extended Mode) .....	204
Figure 7.25	Memory-Mapped UTLB Data Array 2 (TLB Extended Mode) .....	205

## Section 8 Caches

Figure 8.1	Configuration of Operand Cache (Cache size = 32 Kbytes) .....	208
Figure 8.2	Configuration of Instruction Cache (Cache size = 32 Kbytes) .....	209
Figure 8.3	Configuration of Write-Back Buffer .....	221
Figure 8.4	Configuration of Write-Through Buffer .....	221
Figure 8.5	Memory-Mapped IC Address Array (Cache size = 32 Kbytes) .....	229
Figure 8.6	Memory-Mapped IC Data Array (Cache size = 32 Kbytes) .....	230
Figure 8.7	Memory-Mapped OC Address Array (Cache size = 32 Kbytes) .....	232
Figure 8.8	Memory-Mapped OC Data Array (Cache size = 32 Kbytes) .....	233
Figure 8.9	Store Queue Configuration .....	234

## Section 10 Clock Pulse Generator (CPG)

Figure 10.1	Block Diagram of CPG .....	246
-------------	----------------------------	-----

## Section 11 Memory Controller Unit (MCU)

Figure 11.1	Block Diagram of MCU	256
Figure 11.2	Correspondence between Virtual Address Space and External Memory Space	260
Figure 11.3	Basic Timing of SRAM Interface	317
Figure 11.4	Example of 32-Bit Data-Width SRAM Connection	318
Figure 11.5	Example of 16-Bit Data-Width SRAM Connection	319
Figure 11.6	Example of 8-Bit Data-Width SRAM Connection	319
Figure 11.7	SRAM Interface Wait Timing (Software Wait Only)	320
Figure 11.8	SRAM Interface Wait Cycle Timing (Wait Cycle Insertion by $\overline{\text{RDY}}$ Signal)	321
Figure 11.9	SRAM Interface Wait State Timing (Read-Strobe Negate Timing Setting)	322
Figure 11.10	Connection Example of Synchronous DRAMs for 64-Bit Data Bus (Area 1)	325
Figure 11.11	Connection Example of Synchronous DRAMs for 32-Bit Data Bus (Area 1)	326
Figure 11.12	Basic SDRAM Interface Timing (1) Burst Read	328
Figure 11.13	Basic SDRAM Interface Timing (2) Burst Write	329
Figure 11.14	Basic SDRAM Interface Timing (3) Single Read	330
Figure 11.15	Basic SDRAM Interface Timing (4) Single Write	331
Figure 11.16	Basic SDRAM Interface Timing (5) Burst Read Timing (Bank Open Mode; Same Row Address Accessed)	333
Figure 11.17	Basic SDRAM Interface Timing (6) Burst Read Timing (Bank Open Mode; Different Row Addresses Accessed)	334
Figure 11.18	Basic SDRAM Interface Timing (7) Burst Write Timing (Bank Open Mode; Same Row Address Accessed)	335
Figure 11.19	Basic SDRAM Interface Timing (8) Burst Write Timing (Bank Open Mode; Different Row Addresses Accessed)	336
Figure 11.20	Arbitration of Access Requests (1)	341
Figure 11.21	Arbitration of Access Requests (2)	341
Figure 11.22	Block Diagram of MCU (with Symbols Shown for Multi-Step Arbitration Circuit)	345
Figure 11.23	Arbitration Sequence	348
Figure 11.24	Reflection of Data Written by SuperHyway Bus Device	351
Figure 11.25	Data Arrangement in Tiled Memory Areas	353
Figure 11.26	Schematic of Linear-to-Tiled Memory Address Translation	354
Figure 11.27	Operation of Linear-to-Tiled Memory Address Translation	356

## Section 12 Direct Memory Access Controller (DMAC)

Figure 12.1	Block Diagram of DMAC	360
Figure 12.2	Round-Robin Mode	389
Figure 12.3	Changes in Channel Priority in Round-Robin Mode	390
Figure 12.4	Data Flow of Dual Address Mode	391
Figure 12.5	Example of DMA Transfer Timing in Dual Address Mode (Source: Ordinary Memory, Destination: Ordinary Memory)	392

Figure 12.6	DMA Transfer Timing Example in Cycle-Steal Normal Mode 1 (DREQ Low Level Detection) .....	393
Figure 12.7	DMA Transfer Timing Example in Cycle-Steal Normal Mode 2 (DREQ Low Level Detection) .....	394
Figure 12.8	Example of DMA Transfer Timing in Cycle Steal Intermittent Mode (DREQ Low Level Detection) .....	394
Figure 12.9	DMA Transfer Timing Example in Burst Mode (DREQ Low Level Detection) .....	395
Figure 12.10	Bus State when Multiple Channels are Operating .....	398
Figure 12.11	DMA Transfer Flowchart .....	399
Figure 12.12	Reload Mode Transfer .....	401
Figure 12.13	Example of DREQ Input Detection in Cycle Steal Mode Edge Detection .....	402
Figure 12.14	Example of DREQ Input Detection in Cycle Steal Mode Level Detection .....	402
Figure 12.15	Example of DREQ Input Detection in Burst Mode Edge Detection .....	403
Figure 12.16	Example of DREQ Input Detection in Burst Mode Level Detection .....	403
Figure 12.17	DMA Transfer End Signal (Cycle Steal Mode Level Detection) .....	404
Figure 12.18	Example of BSC Ordinary Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device) .....	405
 <b>Section 13 Interrupt Controller (INTC)</b>		
Figure 13.1	Block Diagram of INTC .....	410
Figure 13.2	On-Chip Module Interrupt Priority .....	457
Figure 13.3	Interrupt Operation Flowchart .....	463
Figure 13.4	Example of Interrupt Handling Routine .....	466
Figure 13.5	The time requested to detect interrupts from IRQ1 and IRQ0 .....	466
 <b>Section 14 Timer Unit (TMU)</b>		
Figure 14.1	Block Diagram of TMU .....	470
Figure 14.2	Example of Count Operation Setting Procedure .....	482
Figure 14.3	TCNT Auto-Reload Operation .....	483
Figure 14.4	Count Timing when Operating on Internal Clock .....	483
Figure 14.5	Count Timing when Operating on External Clock .....	484
Figure 14.6	Operation Timing when Using Input Capture Function .....	485
 <b>Section 15 Serial Communication Interface with FIFO (SCIF)</b>		
Figure 15.1	Block Diagram of SCIF .....	491
Figure 15.2	Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits) .....	530
Figure 15.3	Sample Flowchart for SCIF Initialization .....	533
Figure 15.4	Sample Flowchart for Transmitting Serial Data .....	534
Figure 15.5	Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit) .....	536
Figure 15.6	Example of Operation Using Modem Control (CTS) .....	536
Figure 15.7	Sample Flowchart for Receiving Serial Data .....	537



Figure 15.8	Sample Flowchart for Receiving Serial Data (cont).....	538
Figure 15.9	Example of SCIF Receive Operation (8-Bit Data, Parity, 1 Stop Bit) .....	540
Figure 15.10	Example of Operation Using Modem Control ( $\overline{\text{RTS}}$ ) .....	540
Figure 15.11	Data Format in Clock Synchronous Communication .....	541
Figure 15.12	Sample Flowchart for SCIF Initialization .....	543
Figure 15.13	Sample Flowchart for Transmitting Serial Data.....	544
Figure 15.14	Example of SCIF Transmit Operation.....	545
Figure 15.15	Sample Flowchart for Receiving Serial Data (1).....	546
Figure 15.16	Sample Flowchart for Receiving Serial Data (2).....	546
Figure 15.17	Example of SCIF Receive Operation .....	547
Figure 15.18	Sample Flowchart for Transmitting/Receiving Serial Data.....	548
Figure 15.19	Receive Data Sampling Timing in Asynchronous Mode (Operation on a Base Clock with a Frequency 16 Times the Bit Rate) .....	552

## **Section 16 I<sup>2</sup>C Bus Interface (IIC)**

Figure 16.1	Block Diagram for I <sup>2</sup> C Bus Interface .....	555
Figure 16.2	I <sup>2</sup> C Bus Timing .....	574
Figure 16.3	Master Data Transmit Format .....	575
Figure 16.4	Master Data Receive Format.....	575
Figure 16.5	Combination Transfer Format of Master Transfer .....	576
Figure 16.6	10-Bit Address Data Transmit Format .....	576
Figure 16.7	10-Bit Address Data Receive Format.....	577
Figure 16.8	10-Bit Address Transmit/Receive Combined Format .....	577
Figure 16.9	Data Transmit Mode Operation Timing.....	579
Figure 16.10	Data Receive Mode Operation Timing .....	581

## **Section 17 ATAPI**

Figure 17.1	ATAPI Block Diagram.....	587
Figure 17.2	PIO Timing Register .....	598
Figure 17.3	Multiword DMA Timing Register .....	599
Figure 17.4	Ultra DMA Timing Register .....	600
Figure 17.5	ATAPI Data Bus Alignment .....	610
Figure 17.6	Procedure in PIO Transfer Mode .....	612
Figure 17.7	Transfer to and from Memory via Pixel Bus by Polling .....	614
Figure 17.8	Transfer to and from Memory via Pixel Bus by Interrupt.....	615
Figure 17.9	Transfer to and from Memory via Pixel Bus by Polling .....	616
Figure 17.10	Transfer to and from Memory via Pixel Bus by Interrupt.....	617
Figure 17.11	Procedure in Hardware Reset for ATAPI Device .....	617

## **Section 18 Serial Sound Interface (SSI)**

Figure 18.1	Block Diagram of SSI .....	620
Figure 18.2	Philips Format (with no Padding) .....	681

Figure 18.3	Philips Format (with Padding) .....	682
Figure 18.4	Sony Format (with Serial Data First, Followed by Padding Bits) .....	682
Figure 18.5	Matsushita Format (with Padding Bits First, Followed by Serial Data) .....	683
Figure 18.6	Multichannel Format (2 Channels, No Padding) .....	685
Figure 18.7	Multichannel Format (3 Channels with High Padding) .....	685
Figure 18.8	Multichannel Format (4 Channels, with Padding Bits First, Followed by Serial Data, with Padding) .....	686
Figure 18.9	Basic Sample Format (Transmit Mode with Example System/ Data Word Length) .....	687
Figure 18.10	Inverted Clock .....	688
Figure 18.11	Inverted Word Select .....	688
Figure 18.12	Inverted Padding Polarity .....	688
Figure 18.13	Padding Bits First, Followed by Serial Data, with Delay .....	689
Figure 18.14	Padding Bits First, Followed by Serial Data, without Delay .....	689
Figure 18.15	Serial Data First, Followed by Padding Bits, without Delay .....	689
Figure 18.16	Parallel Right-Aligned with Delay .....	690
Figure 18.17	Mute Enabled .....	690
Figure 18.18	Transition Diagram between Operation Modes .....	690
Figure 18.19	Transmission Using SSI_DMAC0 and SSI_DMAC1 .....	692
Figure 18.20	Transmission Using Interrupt Data Flow Control .....	693
Figure 18.21	Reception Using SSI_DMAC0 and SSI_DMAC1 .....	695
Figure 18.22	Reception Using Interrupt Data Flow Control .....	696

## Section 19 Ethernet Controller (EtherC)

Figure 19.1	Configuration of EtherC .....	701
Figure 19.2	EtherC Transmitter State Transitions .....	736
Figure 19.3	EtherC Receiver State Transmissions .....	738
Figure 19.4 (1)	MII Frame Transmit Timing (Normal Transmission) .....	739
Figure 19.4 (2)	MII Frame Transmit Timing (Collision) .....	739
Figure 19.4 (3)	MII Frame Transmit Timing (Transmit Error) .....	740
Figure 19.4 (4)	MII Frame Receive Timing (Normal Reception) .....	740
Figure 19.4 (5)	MII Frame Receive Timing (Reception Error (1): Receive Error Notification) ..	740
Figure 19.4 (6)	MII Fame Receive Timing (Reception Error (2): Carrier Error Notification) .....	741
Figure 19.5	MII Management Frame Format .....	741
Figure 19.6 (1)	1-Bit Data Write Flowchart .....	742
Figure 19.6 (2)	Bus Release Flowchart (TA in Read in Figure 19.5) .....	743
Figure 19.6 (3)	1-Bit Data Read Flowchart .....	743
Figure 19.6 (4)	Independent Bus Release Flowchart (IDLE in Write in Figure 19.5) .....	744
Figure 19.7	Changing IPG and Transmission Efficiency .....	745
Figure 19.8	Example of Connection to DP83846AVHG .....	747

<b>Section 20 Ethernet Controller Direct Memory Access Controller (E-DMAC)</b>	
Figure 20.1 Configuration of E-DMAC, and Descriptors and Buffers.....	750
Figure 20.2 Relationship between Transmit Descriptor and Transmit Buffer.....	787
Figure 20.3 Relationship between Receive Descriptor and Receive Buffer.....	791
Figure 20.4 Sample Transmission Flowchart (Single-Frame/Two-Descriptor) .....	796
Figure 20.5 Sample Reception Flowchart (Single-Frame/Two-Descriptor).....	798
Figure 20.6 E-DMAC Operation after Transmit Error .....	799
Figure 20.7 E-DMAC Operation after Receive Error.....	800
<b>Section 21 USB 2.0 Host/Function Module (USB)</b>	
Figure 21.1 UBS Connector Connection .....	935
Figure 21.2 Items Relating to Interrupts.....	940
Figure 21.3 Timing at which a BRDY Interrupt is Generated.....	945
Figure 21.4 Timing at which NRDY Interrupt is Generated when Function Controller Function is Selected .....	951
Figure 21.5 Timing at which BEMP Interrupt is Generated when Function Controller Function is Selected .....	953
Figure 21.6 Device State Transitions.....	954
Figure 21.7 Control Transfer Stage Transitions .....	956
Figure 21.8 Example of SOFR Interrupt Output Timing.....	957
Figure 21.9 Example of a Buffer Memory Map .....	970
Figure 21.10 Example of Buffer Memory Settings .....	974
Figure 21.11 Example of Buffer Memory Operation .....	976
Figure 21.12 Token Issuance when IITV = 0 .....	990
Figure 21.13 Token Issuance when IITV = 1 .....	991
Figure 21.14 Relationship between ( $\mu$ ) Frames and Expected Token Reception when IITV = 0 .....	992
Figure 21.15 Relationship between ( $\mu$ ) Frames and Expected Token Reception when IITV $\neq$ 0 .....	993
Figure 21.16 Example of Data Setup Function Operation.....	995
Figure 21.17 Example of Buffer Flush Function Operation .....	996
Figure 21.18 Example of an Interval Error Being Generated when IITV = 1 .....	997
<b>Section 22 LCD Controller (LCDC)</b>	
Figure 22.1 LCDC Block Diagram.....	1002
Figure 22.2 Valid Display and the Retrace Period .....	1038
Figure 22.3 Color-Palette Data Format .....	1039
Figure 22.4 Power-Supply Control Sequence and States of the LCD Module .....	1045
Figure 22.5 Power-Supply Control Sequence and States of the LCD Module .....	1045
Figure 22.6 Power-Supply Control Sequence and States of the LCD Module .....	1046
Figure 22.7 Power-Supply Control Sequence and States of the LCD Module .....	1046

Figure 22.8	Clock and LCD Data Signal Example	1050
Figure 22.9	Clock and LCD Data Signal Example (STN Monochrome 8-Bit Data Bus Module)	1050
Figure 22.10	Clock and LCD Data Signal Example (STN Color 4-Bit Data Bus Module)	1051
Figure 22.11	Clock and LCD Data Signal Example (STN Color 8-Bit Data Bus Module)	1051
Figure 22.12	Clock and LCD Data Signal Example (STN Color 12-Bit Data Bus Module)	1052
Figure 22.13	Clock and LCD Data Signal Example (STN Color 16-Bit Data Bus Module)	1053
Figure 22.14	Clock and LCD Data Signal Example (DSTN Monochrome 8-Bit Data Bus Module)	1054
Figure 22.15	Clock and LCD Data Signal Example (DSTN Monochrome 16-Bit Data Bus Module)	1055
Figure 22.16	Clock and LCD Data Signal Example (DSTN Color 8-Bit Data Bus Module)	1056
Figure 22.17	Clock and LCD Data Signal Example (DSTN Color 12-Bit Data Bus Module)	1057
Figure 22.18	Clock and LCD Data Signal Example (DSTN Color 16-Bit Data Bus Module)	1058
Figure 22.19	Clock and LCD Data Signal Example (TFT Color 12-Bit Data Bus Module)	1059
Figure 22.20	Clock and LCD Data Signal Example (TFT Color 16-Bit Data Bus Module)	1060
Figure 22.21	Clock and LCD Data Signal Example (8-Bit Interface Color 640 × 480)	1061
Figure 22.22	Clock and LCD Data Signal Example (16-Bit Interface Color 640 × 480)	1062

## Section 23 G2D

Figure 23.1	Coordinate Transformation Flow Image	1074
Figure 23.2	Example of Antialias Specification	1078
Figure 23.3	Screen Coordinates	1079
Figure 23.4	Rendering Coordinates	1080
Figure 23.5	Work Coordinates	1081
Figure 23.6	2-Dimensional Source Coordinates (SS = 1)	1082
Figure 23.7	1-Dimensional Source Coordinates (SS = 0)	1082
Figure 23.8	Example of Source Style Specification	1085
Figure 23.9	Example of Clipping Specification	1086
Figure 23.10	Example of Relative User Clipping Specification	1087
Figure 23.11	Example of Source Address Specification	1090
Figure 23.12	Example of Source Direction Specification	1092
Figure 23.13	Example of Destination Direction Specification	1092
Figure 23.14	Example of Block Enable Specification	1094

## Section 24 Video Display Controller (VDC2)

Figure 24.1	Block Diagram of VDC2	1225
Figure 24.2	Examples of Overlaid Display	1226
Figure 24.3	Format 1 (Vsync, Hsync, DEV, DEH, DEC, and CDE Output)	1228
Figure 24.4	Format 2 (SPS, SPL, CLS, and COM Output)	1229
Figure 24.5	Timing of Conversion from YC444 to YC422	1231

Figure 24.6	Data Enable Signals .....	1232
Figure 24.7	Pixel Bus Endian (ENDIAN = 0) .....	1241
Figure 24.8	Pixel Bus Endian (ENDIAN = 1) .....	1242
Figure 24.9	Graphic Image Area Settings (Reading from Memory) .....	1244
Figure 24.10	Graphic Image Memory Area Settings .....	1245
Figure 24.11	$\alpha$ Control Area Settings .....	1248
Figure 24.12	Screen Format .....	1254
Figure 24.13	COM Signal Timing .....	1268
Figure 24.14	Settings of DE Area Generated in SG Block .....	1269
Figure 24.15	T-1004 Video Output Position .....	1274

## **Section 25 NAND Flash Memory Controller (FLCTL)**

Figure 25.1	FLCTL Block Diagram .....	1280
Figure 25.2	Register Setting Flow .....	1307
Figure 25.3	Read Operation Timing for NAND-Type Flash Memory (1) .....	1308
Figure 25.4	Programming Operation Timing for NAND-Type Flash Memory (1) .....	1309
Figure 25.5	Programming Operation Timing for NAND-Type Flash Memory (2) .....	1309
Figure 25.6	Read Operation Timing for NAND-Type Flash Memory .....	1310
Figure 25.7	Programming Operation Timing for NAND-Type Flash Memory (1) .....	1310
Figure 25.8	Programming Operation Timing for NAND-Type Flash Memory (2) .....	1311
Figure 25.9	Relationship between DMA Transfer and Sector (Data and Control Code), and Memory and DMA Transfer .....	1312
Figure 25.10	Relationship between Sector Number and Address Expansion of NAND-Type Flash Memory .....	1313
Figure 25.11	Sector Access when Unusable Sector Exists in Continuous Sectors .....	1314

## **Section 26 Sampling Rate Converter (SRC)**

Figure 26.1	Block Diagram of SRC .....	1318
Figure 26.2	Sample Flowchart for Initial Setting .....	1331
Figure 26.3	Sample Flowchart for Data Input .....	1332
Figure 26.4	Sample Flowchart for Data Output .....	1333

## **Section 28 Power-Down Mode**

Figure 28.1	STATUS Output when an Interrupt Occurs in Sleep Mode .....	1423
-------------	--	------

## **Section 29 Watchdog Timer and Reset**

Figure 29.1	Block Diagram .....	1426
Figure 29.2	WDT Counting Operations (Example in Interval Timer Mode) .....	1435
Figure 29.3	STATUS Output during Power-on .....	1437
Figure 29.4	STATUS Output by Reset input during Normal Operation .....	1438
Figure 29.5	STATUS Output by Reset input during Sleep Mode .....	1438

Figure 29.6	STATUS Output by Watchdog timer overflow Power-On Reset during Normal Operation .....	1439
Figure 29.7	STATUS Output by Watchdog timer overflow Power-On Reset during Sleep Mode .....	1440
<b>Section 30 User Break Controller (UBC)</b>		
Figure 30.1	Block Diagram of UBC .....	1442
Figure 30.2	Flowchart of User Break Debugging Support Function .....	1469
<b>Section 31 User Debugging Interface (H-UDI)</b>		
Figure 31.1	H-UDI Block Diagram .....	1478
Figure 31.2	Sequence for switching from Boundary-Scan TAP Controller to H-UDI .....	1481
Figure 31.3	TAP Controller State Transitions .....	1506
Figure 31.4	H-UDI Reset .....	1507
<b>Section 33 Electrical Characteristics</b>		
Figure 33.1	Power-on/Power-off Sequence .....	1574
Figure 33.2	EXTAL Clock Input Timing .....	1582
Figure 33.3	CLKOUT Clock Output Timing (1) .....	1583
Figure 33.4	Power-On Oscillation Settling Time .....	1583
Figure 33.5	MODE Pin Setup / Hold Timing .....	1584
Figure 33.6	Pin Drive Timing in Standby .....	1584
Figure 33.7	Control Signal Timing .....	1585
Figure 33.8	Basic Bus Cycle in SRAM Bus Cycle (No Wait Cycle) .....	1587
Figure 33.9	Basic Bus Cycle in SRAM Bus Cycle (One Internal Wait Cycle) .....	1588
Figure 33.10	Basic Bus Cycle in SRAM Bus Cycle (Internal Wait Cycle + One External Wait Cycle) .....	1589
Figure 33.11	Basic Bus Cycle in SRAM Bus Cycle (No Wait Cycle, Address Setup / Hold Time Insert, AnS= 1, AnH= 1) .....	1590
Figure 33.12	SRAM Bus Cycle in Bank Open Mode Read Bus Cycle (ACT-READ) (BOMODE[1:0]= 00, SCL[2:0]= 000, SRCD= 0, CAS Latency= 2cyc, IRCD= 2cyc) .....	1591
Figure 33.13	SRAM Bus Cycle in Bank Open Mode Pre-charge Read Bus Cycle (PRE-ACT-READ) (BOMODE[1:0]= 00, SRP[1:0]= 00, SCL[2:0]= 000, SRCD=0, IRP= 2cyc, CAS Latency= 2cyc, IRCD= 2cyc) .....	1592
Figure 33.14	SRAM Bus Cycle in Bank Open Mode Read Bus Cycle (Read) (BOMODE[1:0]= 00, SCL[2:0]= 000, SRCD=0, CAS Latency= 2cyc, IRCD= 2cyc) .....	1593
Figure 33.15	SRAM Bus Cycle in Bank Open Mode Write Bus Cycle (ACT-WRITE) (BOMODE[1:0]= 00, SRCD= 0, IRCD= 2cyc) .....	1594

Figure 33.16	SRAM Bus Cycle in Bank Open Mode Pre-charge Write Bus Cycle (PRE-ACT-SRITE) (BOMODE[1:0]= 00, SRP[1:0]= 00, SRCD=0, IRP= 2cyc, IRCD= 2cyc).....	1595
Figure 33.17	SRAM Bus Cycle in Bank Open Mode Write Bus Cycle (WRITE) (BOMODE[1:0]= 00, SRCD= 0, IRCD= 2cyc).....	1596
Figure 33.18	SRAM Bus Cycle in Bank Close Mode Read Bus Cycle (ACT-READA) (BOMODE[1:0]= 1, SCL[2:0]= 000, SRCD= 0, IRP= 2cyc, CAS Latency= 2cyc).....	1597
Figure 33.19	SRAM Bus Cycle in Bank Close Mode Write Bus Cycle (ACT-WRITEA) (BOMODE[1:0]= 00, SWR[1:0]= 00, SRP[1:0]= 00, SRCD= 0, IDAL= 4cyc, IRCD= 2cyc) .....	1598
Figure 33.20	SRAM Bus Cycle in Pre-charge Cycle (PALL) (SRP[1:0]= 00, IRP= 2cyc) .....	1599
Figure 33.21	SRAM Bus Cycle in Mode Register Setting Cycle (MRS).....	1600
Figure 33.22	SRAM Bus Cycle in Auto Refresh Cycle (REF) (SRFC[2:0]= 000, IRC= 8cyc).....	1601
Figure 33.23	SRAM Bus Cycle in Refresh Cycle (SREF) .....	1602
Figure 33.24	NMI Input Timing.....	1603
Figure 33.25	IRQ, $\overline{\text{PINT}}$ Input, IRQOUT Output Timing .....	1604
Figure 33.26	$\overline{\text{DREQ}}/\overline{\text{DTEND}}/\overline{\text{DACK}}$ Timing .....	1605
Figure 33.27	TCLK Input Timing .....	1606
Figure 33.28	I <sup>2</sup> C Timing.....	1607
Figure 33.29	AC Characteristics Load Condition .....	1607
Figure 33.30	SCK Input Clock Timing .....	1608
Figure 33.31	SCIF Input/Output Timing in Clocked Synchronous Mode.....	1608
Figure 33.32	Clock Input/Output Timing.....	1609
Figure 33.33	SSI Transmit Timing (1) .....	1610
Figure 33.34	SSI Transmit Timing (2) .....	1610
Figure 33.35	SSI Receive Timing (1).....	1610
Figure 33.36	SSI Receive Timing (2).....	1611
Figure 33.37	AUDIO_CLK Timing .....	1611
Figure 33.38	PIO Data Transmission In-between Devices.....	1619
Figure 33.39	Multiword DMA Data Transmission Start.....	1620
Figure 33.40	Multiword Data Transmission.....	1621
Figure 33.41	End of Multiword Data Transmission from Device .....	1622
Figure 33.42	End of Multiword Data Transmission from Host.....	1623
Figure 33.43	Ultra-DMA Data In-burst Start .....	1624
Figure 33.44	Ultra-DMA Data In-burst.....	1625
Figure 33.45	Ultra-DMA Data In-burst from Host Pause .....	1625
Figure 33.46	End of Ultra-DMA Data In-burst from Device .....	1626
Figure 33.47	End of Ultra-DMA Data In-burst from Host.....	1627

Figure 33.48	Ultra-DMA Data Out-burst Start.....	1628
Figure 33.49	Ultra-DMA Data Out-burst .....	1629
Figure 33.50	Ultra-DMA Data Out-burst from Device Pause .....	1629
Figure 33.51	End of Ultra-DMA Data Out-burst from Host .....	1630
Figure 33.52	End of Ultra-DMA Data Out-burst from Device.....	1631
Figure 33.53	PIO Data Transmission (DIRECTIO) to Device .....	1632
Figure 33.54	PIO Data Transmission (DIRECTIO) from Device .....	1632
Figure 33.55	Multiword DMA Transmission (DIRECTION) .....	1633
Figure 33.56	Ultra-DMA Transmission Data In-burst Start(DIRECTION) .....	1634
Figure 33.57	End of Ultra-DMA Transmission Data In-burst from Device (DIRECTION).....	1635
Figure 33.58	End of Ultra-DMA Transmission Data In-burst from Host (DIRECTION).....	1636
Figure 33.59	Ultra-DMA Transmission Data Out-burst Start (DIRECTION) .....	1637
Figure 33.60	End of Ultra-DMA Transmission Data Out-burst from Host (DIRECTION).....	1638
Figure 33.61	End of Ultra-DMA Transmission Data Out-burst from Device (DIRECTION) ....	1639
Figure 33.62	USB Clock Timing.....	1640
Figure 33.63	GPIO Timing.....	1641
Figure 33.64	TCK Input Timing.....	1642
Figure 33.65	$\overline{\text{PRESET}}$ Hold Timing.....	1643
Figure 33.66	H-UDI Data Transmission Timing.....	1643
Figure 33.67	$\overline{\text{ASEBRKAK}}/\text{BRKACK}$ Pin Break Timing .....	1643
Figure 33.68	MII Transmission Timing (during Normal Operation) .....	1645
Figure 33.69	MII Transmission Timing (in the Event of a Collision).....	1645
Figure 33.70	MII Receive Timing (during Normal Operation) .....	1646
Figure 33.71	MII Receive Timing (in the Event of a Collision).....	1646
Figure 33.72	MDIO Input Timing .....	1646
Figure 33.73	MDIO Output Timing .....	1646
Figure 33.74	WOL Output Timing.....	1647
Figure 33.75	EXOUT Output Timing.....	1647
Figure 33.76	NAND Flush Memory Command Issue Timing .....	1649
Figure 33.77	NAND Flush Memory Address Issue Timing .....	1649
Figure 33.78	NAND Flush Memory Data Read Timing .....	1650
Figure 33.79	NAND Flush Memory Data Write Timing.....	1650
Figure 33.80	NAND Flush Memory Status Read Timing .....	1651
Figure 33.81	LCDC Module Signal Timing.....	1653
Figure 33.82	Clock Input/Output Timing .....	1654
Figure 33.83	VDC2 Transmission Timing (1).....	1655
Figure 33.84	VDC2 Transmission Timing (2).....	1655
Figure 33.85	VDC2 Reception Timing (1).....	1655
Figure 33.86	VDC2 Reception Timing (2).....	1656
Figure 33.87	Output Load Circuit .....	1657



## Appendix

Figure B.1	Instruction Prefetch .....	1661
Figure H.1	Package Dimensions .....	1684



# Tables

## Section 1 Overview

Table 1.1	SH7764 Features .....	2
Table 1.2	Pin Functions .....	15

## Section 2 Programming Model

Table 2.1	Initial Register Values.....	34
Table 2.2	Bit Allocation for FPU Exception Handling .....	44

## Section 3 Instruction Set

Table 3.1	Execution Order of Delayed Branch Instructions .....	52
Table 3.2	Addressing Modes and Effective Addresses .....	53
Table 3.3	Notation Used in Instruction List.....	58
Table 3.4	Fixed-Point Transfer Instructions .....	59
Table 3.5	Arithmetic Operation Instructions .....	61
Table 3.6	Logic Operation Instructions .....	63
Table 3.7	Shift Instructions .....	64
Table 3.8	Branch Instructions .....	65
Table 3.9	System Control Instructions.....	65
Table 3.10	Floating-Point Single-Precision Instructions .....	68
Table 3.11	Floating-Point Double-Precision Instructions.....	69
Table 3.12	Floating-Point Control Instructions .....	70
Table 3.13	Floating-Point Graphics Acceleration Instructions .....	70

## Section 4 Pipelining

Table 4.1	Representations of Instruction Execution Patterns.....	72
Table 4.2	Instruction Groups .....	82
Table 4.3	Combination of Preceding and Following Instructions.....	84
Table 4.4	Issue Rates and Execution Cycles.....	86

## Section 5 Exception Handling

Table 5.1	Register Configuration.....	95
Table 5.2	States of Register in Each Operating Mode .....	96
Table 5.3	Exceptions.....	102
Table 5.4	UTLB Protection Information (TLB Compatible Mode).....	113
Table 5.5	UTLB Protection Information (TLB Extended Mode) .....	113
Table 5.6	ITLB Protection Information (TLB Compatible Mode) .....	115
Table 5.7	ITLB Protection Information (TLB Extended Mode).....	115

## **Section 6 Floating-Point Unit (FPU)**

Table 6.1	Floating-Point Number Formats and Parameters .....	133
Table 6.2	Floating-Point Ranges .....	134
Table 6.3	Bit Allocation for FPU Exception Handling .....	142

## **Section 7 Memory Management Unit (MMU)**

Table 7.1	Register Configuration .....	158
Table 7.2	Register States in Each Processing State .....	158
Table 7.3	Cache Size and Countermeasure for Avoiding Synonym Problems .....	188

## **Section 8 Caches**

Table 8.1	Cache Features .....	207
Table 8.2	Store Queue Features .....	207
Table 8.3	Register Configuration .....	211
Table 8.4	Register States in Each Processing State .....	211

## **Section 9 On-Chip Memory**

Table 9.1	IL Memory Addresses .....	237
Table 9.2	Register Configuration .....	238
Table 9.3	Register States in Each Processing Mode .....	238
Table 9.4	Protective Function Exceptions to Access On-Chip Memory .....	242

## **Section 10 Clock Pulse Generator (CPG)**

Table 10.1	Pin Configuration and Functions of CPG .....	248
Table 10.2	Clock Operating Modes .....	249
Table 10.3	Register Configuration .....	250
Table 10.4	Register States in Each Operating Mode .....	250

## **Section 11 Memory Controller Unit (MCU)**

Table 11.1	MCU Pin Configuration .....	258
Table 11.2	External Memory Space Map .....	261
Table 11.3	MODE Pin Settings for Memory Bus Width of Area 0 .....	261
Table 11.4	Endian Setting by Pin .....	262
Table 11.5	Register Configuration .....	263
Table 11.6	Address Multiplexing .....	277
Table 11.7	32-Bit External Device/Big-Endian Access and Data Alignment (Areas 0 and 3) .....	308
Table 11.8	16-Bit External Device/Big-Endian Access and Data Alignment (Areas 0 and 3) .....	308
Table 11.9	8-Bit External Device/Big-Endian Access and Data Alignment (Areas 0 and 3) .....	309
Table 11.10	32-Bit External Device/Little-Endian Access and Data Alignment (Areas 0 and 3) .....	310

Table 11.11	16-Bit External Device/Little-Endian Access and Data Alignment (Areas 0 and 3).....	310
Table 11.12	8-Bit External Device/Little-Endian Access and Data Alignment (Areas 0 and 3).....	311
Table 11.13	32-Bit External Device/Big-Endian Access and Data Alignment (Areas 1 and 2).....	312
Table 11.14	32-Bit External Device/Little-Endian Access and Data Alignment (Areas 1 and 2).....	313
Table 11.15	64-Bit External Device/Big-Endian Access and Data Alignment (Areas 1 and 2).....	314
Table 11.16	64-Bit External Device/Little-Endian Access and Data Alignment (Areas 1 and 2).....	315
Table 11.17	Supported Commands for SDRAM .....	324
Table 11.18	SDRAM Bus Widths and Address Multiplexing (External Bus Width is 32 Bits) .....	327
Table 11.19	Correspondence between Linear Addresses and Tiled Memory Addresses.....	355

## **Section 12 Direct Memory Access Controller (DMAC)**

Table 12.1	Pin Configuration.....	361
Table 12.2	Register Configuration of DMAC.....	362
Table 12.3	State of Registers in Each Operating Mode .....	364
Table 12.4	Transfer Request Sources .....	383
Table 12.5	Setting External Request Mode with RS Bits .....	384
Table 12.6	Selecting External Request Detection with DL, DS Bits .....	385
Table 12.7	Selecting External Request Detection with DO Bit .....	385
Table 12.8	Selecting On-Chip Peripheral Module Request Modes with Bits RS[3:0] .....	386
Table 12.9	DMA Transfer Matrix in Auto-Request Mode .....	396
Table 12.10	DMA Transfer Matrix in External Request Mode (Only Channels 0 and 1) ....	396
Table 12.11	DMA Transfer Matrix in On-Chip Peripheral module Request Mode.....	397

## **Section 13 Interrupt Controller (INTC)**

Table 13.1	Interrupt Types.....	412
Table 13.2	INTC Pin Configuration .....	414
Table 13.3	INTC Register Configuration .....	415
Table 13.4	Register States in Each Operating Mode .....	417
Table 13.5	Interrupt Request Sources and INT2PRI0 to INT2PRI12.....	431
Table 13.6	Interrupt Exception Handling and Priority.....	458
Table 13.7	Interrupt Response Time.....	465
Table 13.8	Switching Sequence of IRQ1 and IRQ0 Pin Function.....	467

## Section 14 Timer Unit (TMU)

Table 14.1	Pin Configuration.....	471
Table 14.2	Register Configuration.....	472
Table 14.3	Register States in Each Processing Mode .....	473
Table 14.4	TMU Interrupt Sources.....	486

## Section 15 Serial Communication Interface with FIFO (SCIF)

Table 15.1	Pin Configuration.....	492
Table 15.2	Register Configuration.....	493
Table 15.3	Register State in Each Operation Mode.....	494
Table 15.4	SCSMR Settings .....	514
Table 15.5	Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0).....	515
Table 15.6	Bit Rates and SCBRR Settings (Clock Synchronous Mode).....	516
Table 15.7	Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode) .....	517
Table 15.8	Maximum Bit Rates with External Clock Input (Asynchronous Mode).....	517
Table 15.9	Maximum Bit Rates with External Clock Input (Clock Synchronous Mode, $t_{\text{Seyc}} = 12t_{\text{pccyc}}$ ).....	518
Table 15.10	SCSMR Settings and SCIF Communication Formats .....	528
Table 15.11	SCSMR and SCSCR Settings and SCIF Clock Source Selection.....	529
Table 15.12	Serial Communication Formats (Asynchronous Mode).....	531
Table 15.13	SCIF Interrupt Sources .....	549

## Section 16 I<sup>2</sup>C Bus Interface (IIC)

Table 16.1	Pin Configuration.....	556
Table 16.2	Register Configuration.....	556
Table 16.3	Register State in Each Operating Mode.....	557
Table 16.4	Suggested Settings for CDF and SCGD .....	571
Table 16.5	Description on Symbols of I <sup>2</sup> C Bus Data Format .....	574

## Section 17 ATAPI

Table 17.1	Pin Configuration.....	588
Table 17.2	ATA Task File Register Map.....	589
Table 17.3	ATAPI Packet Command Task File Register Map.....	590
Table 17.4	ATAPI Interface Control Register Map.....	591
Table 17.5	Data Transfer Modes .....	611

## Section 18 Serial Sound Interface (SSI)

Table 18.1	Pin Configuration.....	621
Table 18.2	SSI_DMACH0 Register Configuration .....	621
Table 18.3	SSI_DMACH0 Register State in Each Operating Mode .....	624

Table 18.4	SSI_DMACH1 Register Configuration .....	626
Table 18.5	SSI_DMACH1 Register State in Each Operating Mode .....	628
Table 18.6	Register Configuration of SSI_CH0 to SSI_CH5 .....	631
Table 18.7	Register State in Each Operating Mode for SSI_CH0 to SSI_CH5 .....	632
Table 18.8	Bus Formats of SSI Module .....	680
Table 18.9	Number of Padding Bits for Each Valid Configuration .....	684
<b>Section 19 Ethernet Controller (EtherC)</b>		
Table 19.1	Pin Configuration .....	702
Table 19.2	Register Configuration .....	703
Table 19.3	Register States in Each Operation Mode .....	704
<b>Section 20 Ethernet Controller Direct Memory Access Controller (E-DMAC)</b>		
Table 20.1	Register Configuration .....	751
Table 20.2	Register States in Each Operating Mode .....	752
<b>Section 21 USB 2.0 Host/Function Module (USB)</b>		
Table 21.1	USB Pin Configuration .....	804
Table 21.2	Register Configuration .....	805
Table 21.3	Register State in Each Processing Mode .....	808
Table 21.4	Register Bits Initialized by Writing USBE = 0 (when Function Controller Function is Selected) .....	814
Table 21.5	Register Bits Initialized by Writing USBE = 0 (when Host Controller Function is Selected) .....	814
Table 21.6	USB Data Bus Line Status .....	817
Table 21.7	Test Mode Operation .....	825
Table 21.8	Endian Operation in 32-Bit Access (when MBW = 10) .....	828
Table 21.9	Endian Operation in 16-Bit Access (when MBW = 01) .....	829
Table 21.10	Endian Operation in 8-Bit Access (when MBW = 00) .....	829
Table 21.11	Meaning of BSTS Bit .....	915
Table 21.12	Information Cleared by this Module by Setting ACLRM = 1 .....	916
Table 21.13	Operation of This Module depending on PID Setting (when Host Controller Function is Selected) .....	916
Table 21.14	Operation of This Module depending on PID Setting (when Function Controller Function is Selected) .....	917
Table 21.15	Information Cleared by this Module by Setting ACLRM = 1 .....	926
Table 21.16	Types of Reset .....	933
Table 21.17	Interrupt Generation Conditions .....	936
Table 21.18	Pipe Setting Items .....	960
Table 21.19	Buffer Status Indicated by the BSTS Bit .....	971
Table 21.20	Buffer Status Indicated by the INBUFM Bit .....	971
Table 21.21	List of Buffer Clearing Methods .....	972

Table 21.22	Buffer Memory Map .....	973
Table 21.23	Relationship between Transfer Mode Settings by CNTMD Bit and Timings at which Reading Data or Transmitting Data from FIFO Buffer is Enabled .....	975
Table 21.24	FIFO Port Function Settings .....	977
Table 21.25	FIFO Port Access Categorized by Pipe .....	977
Table 21.26	Packet Reception and Buffer Memory Clearing Processing .....	979
Table 21.27	NYET Handshake Responses .....	984
Table 21.28	Error Detection when a Token is Received .....	987
Table 21.29	Error Detection when a Data Packet is Received .....	988
Table 21.30	Functions of the Interval Counter when the Function Controller Function is Selected .....	989
Table 21.31	Conditions for Generating a Transaction .....	998

## **Section 22 LCD Controller (LCDC)**

Table 22.1	Pin Configuration .....	1003
Table 22.2	Register Configuration .....	1004
Table 22.3	Register State in Each Operating Mode .....	1005
Table 22.4	I/O Clock Frequency and Clock Division Ratio .....	1008
Table 22.5	Available Power-Supply Control-Sequence Periods at Typical Frame Rates .....	1047
Table 22.6	LCDC Operating Modes .....	1048
Table 22.7	LCD Module Power-Supply States .....	1048

## **Section 23 G2D**

Table 23.1	Commands and Rendering Attributes .....	1065
Table 23.2	Commands and Rendering Attributes .....	1067
Table 23.3	Setting Ranges of Parameters Set by Registers and Saturation Processing .....	1075
Table 23.4	Vertex Ranges after Operation and Saturation Processing .....	1075
Table 23.5	Register Configuration .....	1184
Table 23.6	Register Bit Configuration .....	1187
Table 23.7	Initial Register Values at Hardware Reset and Software Reset .....	1189

## **Section 24 Video Display Controller (VDC2)**

Table 24.1	Pin Configuration .....	1223
Table 24.2	Functional Blocks in VDC2 .....	1224
Table 24.3	Register Configuration in Graphics Block 1 .....	1234
Table 24.4	Register Configuration in Graphics Block 2 .....	1235
Table 24.5	Register Configuration in Graphics Block 3 .....	1236
Table 24.6	Register Configuration in Graphics Block 4 .....	1237
Table 24.7	Register Configuration in Display Control Block .....	1238
Table 24.8	Functions of Display Enable Bits .....	1240
Table 24.9	$\alpha$ Value and Blending Ratio .....	1250



Table 24.10	Example of Register Settings for T-1004 Output.....	1273
<b>Section 25</b>	<b>NAND Flash Memory Controller (FLCTL)</b>	
Table 25.1	Pin Configuration.....	1281
Table 25.2	Register Configuration of FLCTL .....	1282
Table 25.3	Register State of FLCTL in Each Processing Mode .....	1283
Table 25.4	Status Read of NAND-Type Flash Memory .....	1315
Table 25.5	FLCTL Interrupt Requests .....	1316
Table 25.6	DMA Transfer Specifications .....	1316
<b>Section 26</b>	<b>Sampling Rate Converter (SRC)</b>	
Table 26.1	Register Configuration.....	1319
Table 26.2	State of Registers in Each Operating Mode .....	1319
Table 26.3	Alignment of Data before Sampling Rate Conversion.....	1320
Table 26.4	Alignment of Data in SRCOD .....	1321
Table 26.5	Relationship between Sampling Rate Setting and Number of Output Data.....	1328
Table 26.6	Interrupt Requests and Generation Conditions .....	1335
<b>Section 27</b>	<b>General Purpose I/O (GPIO)</b>	
Table 27.1	Multiplexed Pins Controlled by Port Control Registers.....	1338
Table 27.2	Multiplexed Pins Controlled by Pin Select Registers .....	1343
Table 27.3	Register Configuration.....	1346
Table 27.4	Register States in Each Operating Mode .....	1348
<b>Section 28</b>	<b>Power-Down Mode</b>	
Table 28.1	States in Power-Down Modes.....	1412
Table 28.2	Pin Configuration.....	1413
Table 28.3	Register Configuration.....	1413
Table 28.4	Register States in Each Operating Mode .....	1413
<b>Section 29</b>	<b>Watchdog Timer and Reset</b>	
Table 29.1	Pin Configuration.....	1427
Table 29.2	Register Configuration.....	1428
Table 29.3	Register States in Each Processing Mode .....	1428
<b>Section 30</b>	<b>User Break Controller (UBC)</b>	
Table 30.1	Register Configuration.....	1443
Table 30.2	Register Status in Each Processing State .....	1444
Table 30.3	Settings for Match Data Setting Register.....	1456
Table 30.4	Relation between Operand Sizes and Address Bits to be Compared .....	1464
<b>Section 31</b>	<b>User Debugging Interface (H-UDI)</b>	
Table 31.1	Pin Configuration.....	1479
Table 31.2	Commands Supported by Boundary-Scan TAP Controller .....	1481

Table 31.3	Register Configuration (1) .....	1482
Table 31.4	Register Configuration (2) .....	1482
Table 31.5	Register Status in Each Processing State .....	1482
Table 31.6	SDBSR Configuration .....	1486

## **Section 32 List of Registers**

Table 32.1	Register Configuration .....	1510
Table 32.2	Register States in Each Operation Mode (1) .....	1547

## **Section 33 Electrical Characteristics**

Table 33.1	Absolute Maximum Ratings .....	1573
Table 33.2	Recommended Time for Power-on/Power-off Sequence .....	1574
Table 33.3	DC Characteristics [Common Items] .....	1575
Table 33.4	DC Characteristics [Excluding the Pins Related to USB Transceiver and I <sup>2</sup> C] .....	1576
Table 33.5	DC Characteristics [Pins Related to I <sup>2</sup> C] .....	1578
Table 33.6	DC Characteristics [Pins Related to USB (1)] .....	1578
Table 33.7	DC Characteristics [Pins Related to USB (2) (for Full-Speed/High-Speed Common Items)] .....	1579
Table 33.8	DC Characteristics [Pins Related to USB (3) (for Full Speed)] .....	1579
Table 33.9	DC Characteristics [Pins Related to USB (4) (for High Speed)] .....	1580
Table 33.10	DC Characteristics [Pins Related to USB (5) (for Low Speed)] .....	1580
Table 33.11	Output Permissible Current Value .....	1581
Table 33.12	Clock and Control Signal Timing .....	1581
Table 33.13	Control Signal Timing .....	1584
Table 33.14	Bus Timing .....	1586
Table 33.15	INTC Module Signal Timing .....	1603
Table 33.16	DMAC Module Signal Timing .....	1605
Table 33.17	TMU Module Signal Timing .....	1605
Table 33.18	I <sup>2</sup> C Module Signal Timing .....	1606
Table 33.19	SCIF Module Signal Timing .....	1608
Table 33.20	SSI Module Signal Timing .....	1609
Table 33.21	ATAPI Interface Resister Access Timing in PIO Transmission .....	1612
Table 33.22	ATAPI Interface Data Transmission Timing in PIO Transmission .....	1613
Table 33.23	ATAPI Interface Multiword Transmission Timing .....	1614
Table 33.24	ATAPI Interface Ultra-DMA Transmission Timing .....	1615
Table 33.25	Symbol for ATAPI Interface Ultra-DMA Transmission Timing .....	1617
Table 33.26	ATAPI Interface DIRECTION Timing .....	1618
Table 33.27	USB Module Clock Timing .....	1640
Table 33.28	USB Electrical Characteristics (for Full Speed) .....	1640
Table 33.29	USB Electrical Characteristics (for Low Speed) .....	1641

Table 33.30	GPIO Signal Timing .....	1641
Table 33.31	H-UDI Module Signal Timing.....	1642
Table 33.32	Ether Net Controller Timing.....	1644
Table 33.33	NAND Flush Memory Interface Timing.....	1648
Table 33.34	LCDC Module Signal timing.....	1652
Table 33.35	VDC2 Module Signal Timing.....	1654
Appendix		
Table D.1	Register Configuration.....	1663
Table E.1	Pin States in Reset, Power-Down State, and Bus-Released State .....	1665
Table F.1	Treatment of Unused Pins.....	1675
Table G.1	Type name of the products.....	1683



# Section 1 Overview

## 1.1 SH7764 Features

This is a system LSI that integrates a Renesas Technology original RISC CPU core with peripheral functions required for system configuration.

The CPU in this LSI has a RISC-type (Reduced Instruction Set Computer) instruction set and uses a superscalar architecture, which greatly improves instruction execution speed. This LSI features the SH-4A CPU, and it has become possible to assemble low-cost, high-performance, and high-functioning systems, even for applications that were previously impossible with microprocessors, such as realtime control, which demands high speeds.

This LSI has a 32-Kbyte instruction cache and a 32-Kbyte operand cache that can be switched between copy-back and write-through modes. It also has a memory management unit (MMU), which enables access to a 4-Gbyte virtual address space, a 4-entry fully-associative TLB for instructions, and a 64-entry fully-associative TLB for both instructions and operands. It includes 16-Kbyte on-chip SRAM, which can be accessed at a high speed and used as the system stack area or the resident area for the core of the functions requiring high performance.

This LSI also has a 2D graphic engine (G2D) for fast display processing. The pictures drawn by the G2D can be displayed through the LCD controller (LCDC).

In addition, this LSI provides on-chip peripheral functions necessary for system configuration, such as an Ethernet controller (EtherC), a USB host interface (supporting V2.0 high speed and full speed), an ATAPI controller (supporting Ultra DMA), a serial communication interface with FIFO (SCIF), an I<sup>2</sup>C bus interface (IIC), a serial sound interface with dedicated DMAC (SSI), a 32-bit timer (TMU), a watchdog timer (WDT), an interrupt controller (INTC), and I/O ports.

This LSI also provides an external memory access support function to enable direct connection to various memory devices such as SDRAM or peripheral LSIs.

These on-chip functions significantly reduce costs of designing and manufacturing application systems.

The features of this LSI are listed in table 1.1.

**Table 1.1 SH7764 Features**

<b>Items</b>	<b>Specification</b>
CPU	<ul style="list-style-type: none"><li>• Renesas Technology original SuperH architecture (SH-4A)</li><li>• Compatible with SH-1, SH-2, SH-3, and SH-4 at object code level</li><li>• 32-bit internal data bus</li><li>• General register file:<ul style="list-style-type: none"><li>— Sixteen 32-bit general registers (and eight 32-bit shadow registers)</li><li>— Seven 32-bit control registers</li><li>— Four 32-bit system registers</li><li>— Register bank for high-speed response to interrupts</li></ul></li><li>• RISC-type instruction set (upward compatible with SH series):<ul style="list-style-type: none"><li>— Instruction length: 16-bit fixed-length basic instructions for improved code efficiency</li><li>— Load/store architecture</li><li>— Delayed branch instructions</li><li>— Conditional execution</li><li>— Instruction set based on C language</li></ul></li><li>• Superscalar architecture (providing simultaneous execution of two instructions) including FPU</li><li>• Instruction execution time: Up to two instructions/cycle</li><li>• Address space: 4 Gbytes</li><li>• Space identifier ASIDs: 8 bits, 256 virtual address spaces</li><li>• Internal multiplier</li><li>• Eight-stage pipeline</li><li>• Harvard architecture</li></ul>

---

Items	Specification
FPU	<ul style="list-style-type: none"> <li>• On-chip floating-point coprocessor</li> <li>• Supports single precision (32 bits) and double precision (64 bits)</li> <li>• Supports IEEE754-compliant data types and exceptions</li> <li>• Two rounding modes: Round to Nearest and Round to Zero</li> <li>• Handling of denormalized numbers: Truncation to zero or interrupt generation for compliance with IEEE754</li> <li>• Floating-point registers: 32 bits x 16 words x 2 banks (single-precision x 16 words or double-precision x 8 words) x 2 banks</li> <li>• 32-bit CPU-FPU floating-point communication register (FPUL)</li> <li>• Supports FMAC (multiply-and-accumulate) instruction</li> <li>• Supports FDIV (divide) and FSQRT (square root) instructions</li> <li>• Supports FLDI0/FLDI1 (load constant 0/1) instructions</li> <li>• Instruction execution time: <ul style="list-style-type: none"> <li>— Latency (FADD/FSUB): 3 cycles (single-precision) or 5 cycles (double-precision)</li> <li>— Latency (FMAC/FMUL): 5 cycles (single-precision) or 7 cycles (double-precision)</li> <li>— Pitch (FADD/FSUB): 1 cycle (single-precision) or 1 cycle (double-precision)</li> <li>— Pitch (FMAC/FMUL): 1 cycle (single-precision) or 3 cycles (double-precision)</li> </ul> <p>Note: FMAC is supported for single-precision only.</p> </li> <li>• 3-D graphics instructions (single-precision only) <ul style="list-style-type: none"> <li>— 4-dimensional vector conversion and matrix operations (FTRV): 4 cycles (pitch), 8 cycles (latency)</li> <li>— 4-dimensional vector inner product (FIPR): 1 cycle (pitch), 5 cycles (latency)</li> </ul> </li> <li>• 11-stage pipeline</li> </ul>

Items	Specification
Memory management unit (MMU)	<ul style="list-style-type: none"><li>• 4-Gbyte address space, 256 address space identifiers (8-bit ASIDs)</li><li>• Single virtual memory mode and multiple virtual memory mode</li><li>• Supports multiple page sizes: 1 Kbyte, 4 Kbytes, 8 Kbytes, 64 Kbytes, 256 Kbytes, 1 Mbyte, 4 Mbytes, and 64 Mbytes</li><li>• 4-entry fully-associative TLB for instructions</li><li>• 64-entry fully-associative TLB for instructions and operands</li><li>• Supports software-controlled replacement and random-counter replacement algorithm</li><li>• TLB contents can be accessed directly by address mapping</li><li>• Access right check</li></ul>
Cache memory	<ul style="list-style-type: none"><li>• Instruction cache (IC)<ul style="list-style-type: none"><li>— 32-Kbyte, 4-way set associative</li><li>— 256 entries/way, 32-byte block length</li><li>— Power-down function (way prediction)</li></ul></li><li>• Operand cache (OC)<ul style="list-style-type: none"><li>— 32-Kbyte, 4-way set associative</li><li>— 256 entries/way, 32-byte block length</li></ul></li><li>• Single-stage copy-back buffer and single-stage write-through buffer</li><li>• Store queue (32 bytes <math>\times</math> 2 entries)</li></ul>
On-chip memory (IL memory)	<ul style="list-style-type: none"><li>• 16-Kbyte fast RAM</li><li>• Consists of one page</li><li>• Accessible from the following three read/write ports<ul style="list-style-type: none"><li>— SuperHyway bus</li><li>— Cache/RAM internal bus</li><li>— Instruction bus</li></ul></li><li>• Supports 8-, 16-, 32-, or 64-bit operand access from the CPU</li><li>• Supports 8-, 16-, 32-, or 64-bit access and 16- or 32-byte access through external requests</li></ul>



Items	Specification
User break controller (UBC)	<ul style="list-style-type: none"> <li>• Supports debugging by means of user break interrupts</li> <li>• Two break channels</li> <li>• Address, data value, access type, and data size can all be set as break conditions</li> <li>• Supports sequential break function</li> </ul>
Clock pulse generator (CPG)	<ul style="list-style-type: none"> <li>• Choice of main clock: 10 to 12 times EXTAL input clock</li> <li>• Clock modes: <ul style="list-style-type: none"> <li>— CPU clock: 324 MHz max.</li> <li>— Local bus clock: 108 MHz max.</li> <li>— SDRAM clock: 108 MHz max.</li> <li>— USB clock: 48 MHz</li> <li>— VDC2 clock: Appropriate frequency input depending on the display panel size</li> </ul> </li> </ul>
Watchdog timer (WDT)	<ul style="list-style-type: none"> <li>• Supports watchdog timer mode, in which a counter overflow resets the internal circuits, and interval mode, in which a counter overflow generates an interrupt</li> <li>• Outputs an overflow signal externally and can assert a reset signal (power-on reset) for the circuits in the LSI in watchdog timer mode</li> <li>• One channel</li> </ul>
Interrupt controllers (INTC)	<ul style="list-style-type: none"> <li>• Direct jump mode (compatible with SH-4)</li> <li>• Three external interrupt pins: NMI, IRQ1, and IRQ0</li> <li>• On-chip peripheral module interrupts: Priority level can be set for each module</li> </ul>
Direct memory access controller (DMAC)	<ul style="list-style-type: none"> <li>• Six channels; external requests available for two of them</li> <li>• Transfer data size: Byte, word (2 bytes), longword (4 bytes), 16 bytes, or 32 bytes</li> <li>• Maximum transfer byte count: 16,777,216</li> <li>• Address mode: Dual address mode</li> <li>• Bus mode: Cycle-steal or burst mode</li> <li>• Transfer requests: External requests (channels 0 and 1), on-chip peripheral module requests, or auto-requests selectable</li> <li>• Priority: Fixed channel priority mode or round robin mode selectable</li> </ul>

Items	Specification
Memory control unit (MCU)	<ul style="list-style-type: none"><li>• Supports external memory access<ul style="list-style-type: none"><li>— Outputs four external memory select signals</li><li>— Supports four external memory areas (FLASH, SDRAM), each of which has 64 Mbytes max.</li></ul></li><li>• SRAM: 32-, 16-, or 8-bit data bus width selectable</li><li>• SDRAM: 64- or 32-bit data bus width selectable</li><li>• Big endian or little endian mode can be set</li></ul> <p>[SRAM interface]</p> <ul style="list-style-type: none"><li>• NOR-type flash memory can be connected</li><li>• Cycle wait function: Wait control by hardware through signals</li><li>• Wait control for preventing collisions on the data bus (idle cycle insertion):<ul style="list-style-type: none"><li>— Wait setting between read cycles</li><li>— Wait setting between a read cycle and a write cycle</li></ul></li></ul> <p>[SDRAM interface]</p> <ul style="list-style-type: none"><li>• Refresh function:<ul style="list-style-type: none"><li>— Auto-refresh (programmable refresh counter provided)</li><li>— Self-refresh</li></ul></li><li>• Timing control:<p>Row-column latency, column latency, row active period, write recovery period, row precharge period, auto-refresh request interval, initial precharge cycle count, and initial auto-refresh request interval</p></li><li>• Burst access mode: Random column (SDRAM burst length: eight for 32-bit bus or four for 64-bit bus)</li><li>• Initialization sequencer function: Issues precharge and auto-refresh commands</li></ul>

---

Items	Specification
Timer (TMU)	<ul style="list-style-type: none"> <li>• 6-channel auto-reload 32-bit timer</li> <li>• Input-capture function (only channel 2)</li> <li>• Choice of six counter input clocks for each channel               <ul style="list-style-type: none"> <li>— External clock and five peripheral clocks (Pck/4, Pck/16, Pck/64, Pck/256, and Pck/1024)</li> <li>(Pck is the peripheral module clock)</li> </ul> </li> </ul>
Serial communication interface with FIFO (SCIF)	<ul style="list-style-type: none"> <li>• Three channels</li> <li>• Separate 16-byte FIFOs for transmission and reception</li> <li>• Asynchronous mode or clock synchronous mode selectable</li> <li>• Full-duplex communications</li> <li>• Transmit/receive clock source: Internal clock from the baudrate generator or external clock from the SCK pin selectable</li> <li>• Modem control function (in asynchronous mode)</li> </ul>
I <sup>2</sup> C bus interface (IIC)	<ul style="list-style-type: none"> <li>• Supports the Philips I<sup>2</sup>C bus (Inter IC Bus) interface</li> <li>• Master and slave functions</li> <li>• Multi-master function</li> <li>• Supports transfer speed up to 400 kbps</li> <li>• Programmably generates a clock from the system clock</li> </ul>
ATAPI interface (ATAPI)	<ul style="list-style-type: none"> <li>• Supports primary channel</li> <li>• Master and slave functions</li> <li>• Transfer mode: PIO modes 0 to 4, multiword DMA modes 0 to 2, and Ultra DMA modes 0 to 4 (66 Mbps max.)</li> <li>• High-speed transfer using 32-byte double buffer</li> <li>• Supports descriptor mode</li> <li>• On-chip dedicated DMAC (one channel)</li> <li>• I/O: Supports 3.3 V</li> </ul> <p>Note: The ATAPI controller has two groups of I/O pins: primary I/O group and secondary I/O group (mirror pins). Both groups always work in the same way but pins in different groups should not be used together.</p>

Items	Specification
Serial sound interface (SSI)	<ul style="list-style-type: none"><li>• Six-channel bidirectional serial transfers</li><li>• Supports various real audio formats</li><li>• Master and slave functions</li><li>• Programmably generates a word clock or bit clock</li><li>• Supports multichannel format</li><li>• Supports 8-, 16-, 18-, 20-, 22-, 24-, or 32-bit data format</li><li>• SSI network function</li></ul> <p>Any audio clock channels can be connected. See the following examples.</p> <ul style="list-style-type: none"><li>— Example 1: All audio clocks are connected.</li><li>— Example 2: Audio clocks for channels 0 to 2 are connected.</li><li>— Example 3: Audio clocks are connected for channels 0 and 1, channels 2 and 3, and channels 4 and 5, respectively.</li><li>— Example 4: Audio clocks for all channels are used independently.</li></ul> <p>In the same way, the SSISCK and SSIWS pins for any channels can be connected as a set. This setting can be made independently from the audio clock connection settings.</p> <ul style="list-style-type: none"><li>• SSI-DMAC</li></ul> <p>A dedicated DMAC for the SSI is provided to transfer data between the SSI and external or on-chip memory.</p> <ul style="list-style-type: none"><li>— Number of channels: Six for transmission and six for reception</li><li>— Transfer data size: 8, 16, or 32 bytes</li><li>— Maximum transfer byte count: 4,294,967,296</li><li>— Bus mode: Cycle-steal mode</li><li>— Priority: Fixed channel priority mode or round robin mode selectable</li></ul>

---

Items	Specification
Ethernet controller (EtherC)	<ul style="list-style-type: none"> <li>• Ethernet MAC (Media Access Control) function</li> <li>• Data frame assembly/disassembly (frame format conforming to IEEE802.3)</li> <li>• CSMA/CD link management (data collision prevention and collision processing)</li> <li>• CRC processing</li> <li>• Separate 2-Kbyte FIFOs for transmission and reception</li> <li>• Full-duplex or half-duplex transmission and reception</li> <li>• Detects short packets and long packets</li> <li>• Conforms to MII (Media Independent Interface) standard</li> <li>• Station management (STA function)</li> <li>• 10 or 100-Mbps transfer rate</li> <li>• Magic Packet detection (WOL (Wake-On-LAN) signal output)</li> </ul>
DMAC for Ethernet controller (E-DMAC)	<ul style="list-style-type: none"> <li>• Reduces the load on the CPU by means of a descriptor management system</li> <li>• One channel for data transfer from the Ether receive FIFO (2 Kbytes) to receive buffer</li> <li>• One channel for data transfer from the transmit buffer to EtherC transmit FIFO (2 Kbytes)</li> <li>• Achieves efficient bus utilization through 32-byte burst transfer</li> <li>• Supports single-frame multi-buffer transfer</li> </ul>
USB host/function interface (USB)	<ul style="list-style-type: none"> <li>• Conforms to USB version 2.0</li> <li>• Supports 480-Mbps and 12-Mbps transfer speeds</li> <li>• Can be switched between the USB host and function by software</li> <li>• PHY is provided</li> <li>• Connectable with multiple peripheral devices through a hub</li> <li>• 5-Kbyte RAM provided as a communication buffer</li> </ul>
LCD controller (LCDC)	<ul style="list-style-type: none"> <li>• Supports 16 x 1 to 1024 x 1024-dot display size</li> <li>• Supports 4, 8, 15, and 16 bpp color modes</li> <li>• Supports 1, 2, 4, and 6 bpp grayscale modes</li> <li>• Supports TFT, DSTN, and STN display</li> <li>• Selectable signal polarities</li> <li>• 24-bit color palette memory (16 of 24 bits are valid: R: 5; G: 6; B: 5)</li> <li>• Unified graphics memory architecture</li> </ul>

Items	Specification
2D graphic engine (2D Engine: G2D)	<p>[Drawing functions]</p> <ul style="list-style-type: none"><li>• Four-vertex drawing</li><li>• Polygon drawing</li><li>• Line drawing</li><li>• High-functional bold line drawing</li><li>• Antialiasing</li><li>• Raster operation/BitBLT with alpha blending</li></ul> <p>[Coordinate transformation functions]</p> <ul style="list-style-type: none"><li>• 4 × 4 matrix operation + W division of perspective</li></ul> <p>[Color representation]</p> <ul style="list-style-type: none"><li>• Source: 1, 8, or 16 bits/pixel Drawing: 8 or 16 bits/pixel</li><li>• Work: Binary</li></ul> <p>[Screen coordinates]</p> <ul style="list-style-type: none"><li>• X-direction: 0 to 4095 Y-direction: 0 to 4095</li></ul> <p>[Register setting]</p> <ul style="list-style-type: none"><li>• Current pointer setting</li><li>• Local offset setting</li><li>• Specific address mapped register setting</li></ul> <p>[Sequence control]</p> <ul style="list-style-type: none"><li>• Vsync wait</li><li>• Jump</li><li>• Subroutine (nesting level: 1)</li></ul>
Video display controller 2 (VDC2)	<p>[Graphic processing functions]</p> <ul style="list-style-type: none"><li>• Plane configuration: Graphic display composed of four planes</li><li>• Alpha blending and chroma-key functions for graphics (supports RGB16-format input data)</li></ul> <p>[Output functions]</p> <ul style="list-style-type: none"><li>• Digital RGB output (6 bits for each color)</li><li>• Panel output conforming to the VESA standard (RGB6:6:6, HD, VD, DE)</li><li>• BTA T-1004 digital (8:4:4 parallel) interface output</li><li>• Supports external synchronous mode</li></ul>

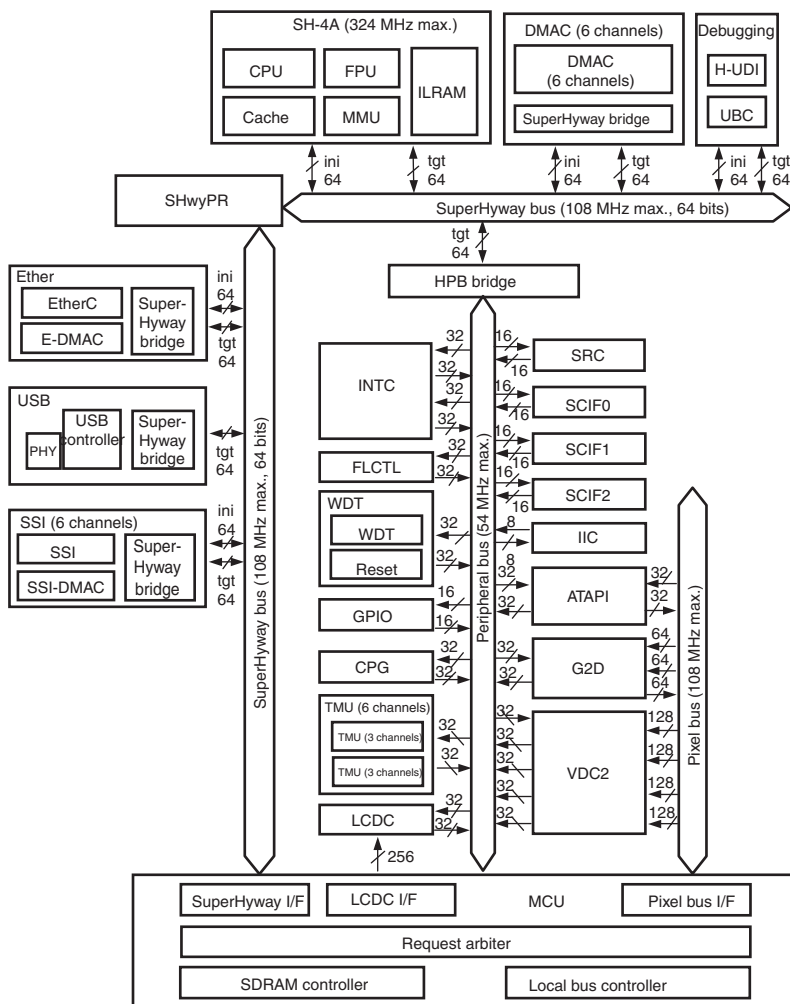
---

Items	Specification
NAND flash memory controller (FLCTL)	<ul style="list-style-type: none"> <li>• Directly connectable to a NAND-type flash memory</li> <li>• Read or write in sector units (512 + 16 bytes) and ECC processing</li> <li>• Read or write in byte units</li> <li>• 256-byte FIFO provided</li> <li>• Does not support multil-level (MLC) flash memory</li> </ul>
Sampling rate converter (SRC)	<ul style="list-style-type: none"> <li>• Data format: 32-bit stereo (16 bits each for L and R) or 16-bit monaural data</li> <li>• Input sampling rate: 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, or 48 kHz</li> <li>• Output sampling rate: 44.1 or 48 kHz</li> </ul>
I/O ports (GPIO)	<ul style="list-style-type: none"> <li>• 77 general I/O ports</li> <li>• Input or output can be selected for each bit</li> <li>• Multiplexed with interrupt pins or on-chip peripheral module pins</li> </ul>
Power-down modes	<ul style="list-style-type: none"> <li>• Three power-down modes provided to reduce the power consumption in this LSI <ul style="list-style-type: none"> <li>— Sleep mode: Stops clock supply to the CPU.</li> <li>— Refresh standby mode: The CPU and on-chip peripheral modules stop operation. The CPG continues operation and the SDRAM can continue refresh operation.-</li> <li>— Module standby mode: Stops clock supply to the on-chip peripheral modules.</li> </ul> </li> </ul>
Debugging interfaces	<ul style="list-style-type: none"> <li>• H-UDI (User Debugging Interface)</li> <li>• AUD (Advanced User Debugger)</li> </ul>
Power supply voltage	<ul style="list-style-type: none"> <li>• VDD and VDD-PLL: 1.2 ± 0.1 V</li> <li>• VDDQ: 3.3 ± 0.3 V</li> </ul>
Package	BGA-404 pin (19 mm x 19 mm)
Process	90-μm CMOS process

## 1.2 Block Diagram

Figure 1.1 shows a block diagram of the SH7764.





## [Legend]

ATAPI: ATAPI controller  
 CPG: Clock pulse generator  
 CPU: Central processing unit  
 DMAC: Direct memory access controller  
 E-DMAC: Direct memory access controller for Ethernet controller  
 EtherC: Ethernet controller  
 FLCTL: NAND flash memory controller  
 FPU: Floating-point unit  
 GPIO: General I/O  
 G2D: 2D graphics engine  
 HPB: Peripheral bus bridge  
 H-UDI: User debugging interface  
 IIC: I<sup>2</sup>C bus interface  
 INTC: Interrupt controller  
 LCDC: LCD controller

ILRAM: IL memory  
 MCU: Memory controller unit  
 MMU: Memory management unit  
 SCIF: Serial communication interface with FIFO  
 SHwPR: SuperHyway bus packet router  
 SRC: Sampling rate converter  
 SSI: Serial sound interface  
 SSI-DMAC: DMAC for serial sound interface  
 TMU: Timer unit  
 UBC: User break controller  
 USB: USB host/function interface  
 VDC2: Video display controller 2  
 WDT: Watchdog timer  
 ini: Port for outputting requests to the bus  
 tgt: Port for receiving requests from the bus

Figure 1.1 Block Diagram

## 1.3 Pin Arrangement

Figure 1.2 shows the pin arrangement.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22		
A	VSS	VSS	VSS	VDDA_USB	VDD_USB	DP	DM	VDDQ_USB	VDDQ	D02 IDE09	D01 IDE11	D04 IDE13	WE1 DQMALL	WE5 DQMALL	D08 IDE15	D06 IDEA2	D04 PFS	D02 PF7	A2	A3	CE1	VSS	A	
B	XIN	VSS	VSS	VSSA_USB	VSS_USB	VBUS	VSS	VSSQ_USB	VSS	D03 IDE10	D01 IDE10	D09 IDE12	WE3 DQMALL	WE2 DQMALL	D09 IDE14	D07 IDEA1	D05 IDEA0	D03 PFS	A1	CE2	VSS	CLKOUT	B	
C	SQOUT	VSS	VDDQ	VSS	VSS	VSS	VSS	VSS	VSS	D04 IDE11	D06 IDE16	D08 IDE14	D01 IDE10	D03 IDE10	D05 IDE12	D07 IDE14	D09 IDE16	D01 IDE10	A0	VDDQ	RAS	CNE	C	
D	EXOUT PFS IDECST_M	UNKSTA PFS IDECSC_M	WOL PFS IDEAL_M	VDDQ	VSS	UV12	VSSQA_USB	VSS	VSS	D05 IDECST	D07 IDE17	D09 IDE15	D01 IDE10	D03 IDE11	D05 IDE12	D07 IDE14	D09 IDE16	D01 IDE10	D03 IDE11	VDDQ	CAS	RW	A4	D
E	COL PFS IDEA2_M	CPS PFS IDEA1_M	SSISC02 PC3	VSS	VDDQ	UV12	VDDQA_USB	REFRIN	VDDQ	VDDQ	VSS	VDD	VSS	VSS	VSS	VSS	VDDQ	VDDQ	A7	A6	A5	A10	E	
F	MR_TXD0 AUDIO_CLKS IDEINT_M P01	MR_TXD0 SSIGATA PC3	AUDIO_CLKS VDDQ	SSIGATA PC3	VDDQ													VDDQ	A9	A8	A14	A13	F	
G	MR_TXD0 SSIGATA PC3	MR_TXD1 TX_ER PFS IDEIORDY_M P02	SSIW02 PC4	VSS														VDDQ	A12	A11	A18	A15	G	
H	TX_CLK PFS IDEI15_M	TX_ER PFS IDEI15_M	RX_ER PFS IDEI15_M	MPMD	VDDQ													VSS	D05	D04	D03	D02	H	
J	RX_CLK PFS IDEI15_M	RX_ER PFS IDEI15_M	SSIW03 PC4	VDDQ														VSS	D07	D06	D01	D00	J	
K	MR_RXD0 SSIGATA PC3	MR_RXD0 SSIGATA PC3	RIO OTTEND1	VSS														VSS	D09	D08	D18	D18	K	
L	MR_RXD0 SSIGATA PC3	MR_RXD0 SSIGATA PC3	AUDIO_CLKS P02	VSS														VDDQ	D01	D00	D17	D16	L	
M	MDIO PFS IDEI11_M	MDIO PFS IDEI11_M	SSIW02 PC4	VDDQ														VDDQ	D01	D00	D17	D16	M	
N	AUDIO_CLKS PC7	SSIW03 PC4	SSIW03 PC4	VDDQ														VDDQ	D01	D00	D17	D16	N	
P	AUDIO_CLKS PC7	SSIW03 PC4	SSIW03 PC4	VDDQ														VSS	D11	D10	D5	D4	P	
R	PJ2 IDEI10_M	PJ2 IDEI10_M	SSIW03 PC4	VDDQ														VSS	D13	D12	D3	D2	R	
T	PJ3 IDEI10_M	PJ3 IDEI10_M	SSIW03 PC4	VDDQ														VSS	D15	D14	D1	D0	T	
U	PJ4 IDEI10_M	PJ4 IDEI10_M	SSIW03 PC4	VDDQ														VDDQ	A17	A16	A19	A18	U	
V	PJ5 IDEI10_M	PJ5 IDEI10_M	SSIW03 PC4	VDDQ														VSS	D17	D16	D1	D0	V	
W	PJ6 IDEI10_M	PJ6 IDEI10_M	SSIW03 PC4	VDDQ														VSS	D19	D18	D1	D0	W	
Y	PJ7 IDEI10_M	PJ7 IDEI10_M	SSIW03 PC4	VDDQ														VSS	D21	D20	D1	D0	Y	
AA	PJ8 IDEI10_M	PJ8 IDEI10_M	SSIW03 PC4	VDDQ														VSS	D23	D22	D1	D0	AA	
AB	PJ9 IDEI10_M	PJ9 IDEI10_M	SSIW03 PC4	VDDQ														VSS	D25	D24	D1	D0	AB	

Figure 1.2 Pin Arrangement

## 1.4 Pin Functions

Table 1.2 lists the pin functions.

**Table 1.2 Pin Functions**

Classification	Symbol	I/O	Name	Function
Power supply	VDD	I	Power supply for internal power	Power supply pins for the internal core power. All the VDD pins must be connected to the system power supply. This LSI does not operate if there is a pin left open.
	VSS	I	Ground for internal power and I/O circuits	Ground pins for the internal core power and I/O circuits. All the VSS pins must be connected to the system power supply (0 V). This LSI does not operate if there is a pin left open
	VDDQ	I	Power supply for I/O circuits	Power supply pins for the I/O pins. All the VDDQ pins must be connected to the system power supply. This LSI does not operate if there is a pin left open.
	VDD_PLL1	I	Power supply for PLL1	Power supply pins for the on-chip PLL1 oscillator. This LSI does not operate if there is a pin left open.
	VSS_PLL1	I	Ground for PLL1	Ground pins for the on-chip PLL1 oscillator. This LSI does not operate if there is a pin left open.
	VDD_PLL2	I	Power supply for PLL2	Power supply pins for the on-chip PLL2 oscillator. This LSI does not operate if there is a pin left open.
	VSS_PLL2	I	Ground for PLL2	Ground pins for the on-chip PLL2 oscillator. This LSI does not operate if there is a pin left open.
Clock	EXTAL	I	Crystal resonator/external clock	Connected to a crystal resonator. An external clock signal can also be input to the EXTAL pin.
	XTAL	O	Crystal resonator	Connected to a crystal resonator.
	CLKOUT	O	System clock output	Supplies the system clock to external devices.

Classification	Symbol	I/O	Name	Function
Operating mode control	MODE2 MODE1 MODE0	I	Clock mode set	These pins set the clock operating mode. Do not change the signal levels on these pins during operation.
	MODE4 MODE3	I	Bus mode set	These pins set the bus operating mode. Do not change the signal levels on these pins during operation.
	MODE5	I	Endian set	Selects the endian for the CPU. Do not change the signal level on this pin during operation.
	MODE7	I	XIN/XOUT pin function set	Enables the external clock or crystal resonator for the USB.
	MODE8	I	EXTAL/XTAL pin function set	Enables the external clock or crystal resonator.
System control	$\overline{\text{PRESET}}$	I	Power-on reset	This LSI enters the power-on reset state when this signal goes low.
	$\overline{\text{WDTOVF}}$	O	Watchdog timer overflow	Outputs an overflow signal from the WDT.
	$\overline{\text{BREQ}}$	I	Bus-mastership request	A low level should be input to this pin when an external device requests the release of the bus mastership.
	$\overline{\text{BACK}}$	O	Bus-mastership request acknowledge	Indicates that the bus mastership has been released to an external device. Reception of the $\overline{\text{BACK}}$ signal informs the device which has output the $\overline{\text{BREQ}}$ signal that it has acquired the bus.

Classification	Symbol	I/O	Name	Function
Interrupts	NMI	I	Non-maskable interrupt	Non-maskable interrupt request pin. Fix it high when not in use.
	IRQ, IRQ0	I	Interrupt requests 1 and 0	Maskable interrupt request pins. Level-input or edge-input detection can be selected. When the edge-input detection is selected, the rising edge, falling edge, or both edges can also be selected.
	PINT15 to PINT0	I	Port interrupts	Pins for interrupt requests from ports.  PA7 to PA0 and PB7 to PB0 are used to generate interrupts. A low level should be input to generate an interrupt.
	IRQOUT	O	Interrupt detection	Status signal that indicates that an interrupt request has been detected and accepted.
Address bus	A25 to A0	O	Address bus	Address output.
Data bus	D63 to D0	I/O	Data bus	Bidirectional data bus.
Operation status	STATUS1, STATUS0	O	Internal operation status indication	These pins indicate the following status.  00: Normal state 01: Standby state 10: Sleep state 11: Reset state

Classification	Symbol	I/O	Name	Function
Bus control	$\overline{\text{CS3}}$ to $\overline{\text{CS0}}$	O	Chip select 3 to 0	Chip-select signals for external memory or devices.
	$\overline{\text{BS}}$	O	Bus cycle start	Bus-cycle start signal. It is asserted for the first of the multiple bus cycles of a bus transaction.
	$\overline{\text{RD}}$	O	Read	Indicates that data is read from an external device.
	$\text{R}/\overline{\text{W}}$	O	Read/write	Indicates the read/write state for an external device. It outputs a high level for a read access or a low level for a write access.
	$\overline{\text{RDY}}$	I	Wait	Input signal for inserting a wait cycle into the bus cycles during access to the external space.
	$\overline{\text{WE0}}$	O	Byte select	Indicates a write access to bits 7 to 0 of data of an external memory or device (for 8-, 16-, or 32-bit access). This pin is multiplexed with DQM64LL.
	$\overline{\text{WE1}}$	O	Byte select	Indicates a write access to bits 15 to 8 of data of an external memory or device (for 16-, or 32-bit access). This pin is multiplexed with DQM64LU.
	$\overline{\text{WE2}}$	O	Byte select	Indicates a write access to bits 23 to 16 of data of an external memory or device (for 16-, or 32-bit access). This pin is multiplexed with DQM64UL.
	$\overline{\text{WE2}}$	O	Byte select	Indicates a write access to bits 31 to 24 of data of an external memory or device (for 8-, 16-, or 32-bit access). This pin is multiplexed with DQM64UU.

Classification	Symbol	I/O	Name	Function
Bus control DRAM interface	$\overline{\text{RAS}}$	O	RAS	Connected to the $\overline{\text{RAS}}$ pin when SDRAM is connected.
	$\overline{\text{CAS}}$	O	CAS	Connected to the $\overline{\text{CAS}}$ pin when SDRAM is connected.
	CKE	O	CK enable	Connected to the CKE pin when SDRAM is connected.
	DQM0U	O	Byte select 0	Selects bits 31 to 24 when SDRAM is connected.
	DQM1U	O	Byte select 1	Selects bits 23 to 16 when SDRAM is connected.
	DQM2U	O	Byte select 2	Selects bits 15 to 8 when SDRAM is connected.
	DQM3U	O	Byte select 3	Selects bits 7 to 0 when SDRAM is connected.
	DQM64U0	O	Byte select 0 for 64 bits	Selects bits 63 to 56 when SDRAM is connected. This pin is multiplexed with $\overline{\text{WE3}}$ .
	DQM64U1	O	Byte select 1 for 64 bits	Selects bits 55 to 48 when SDRAM is connected. This pin is multiplexed with $\overline{\text{WE2}}$ .
	DQM64U2	O	Byte select 2 for 64 bits	Selects bits 47 to 40 when SDRAM is connected. This pin is multiplexed with $\overline{\text{WE1}}$ .
	DQM64U3	O	Byte select 3 for 64 bits	Selects bits 39 to 32 when SDRAM is connected. This pin is multiplexed with $\overline{\text{WE0}}$ .

Classification	Symbol	I/O	Name	Function
Video display controller 2 (VDC2)	DR5 to DR0	O	Digital red data output	Video data output.
	DG5 to DG0	O	Digital green data output	Video data output.
	DB5 to DB0	O	Digital blue data output	Video data output.
	VSYNC/SPS	O	Vertical sync/gate start signal	Vertical sync signal/gate start signal.
	HSYNC/SPL	O	Horizontal sync/sampling start signal	Horizontal sync/ sampling start signal.
	DE_V/CLS	O	Vertical data enable/gate clock signal	Vertical data enable/gate clock signal.
	DE_H/DE_C	O	Horizontal data enable/display enable signal	Horizontal data enable/display enable signal.
	COM/CDE	O	Gate control/chroma data enable signal	Gate control/display enable signal (asserted when the data matches the chroma-key target color specified in the register).
	BT_DATA7 to BT_DATA0	I/O	BTA-T1004 display data	BTA-T1004 display data output.
	BT_HSYNC	O	BTA-T1004 horizontal sync	BTA-T1004 horizontal sync signal.
	BT_VSYNC	O	BTA-T1004 vertical sync	BTA-T1004 vertical sync signal.
	BT_DE_C	O	BTA-T1004 display enable	BTA-T1004 display enable signal.
	EX_HSYNC	I	HSYNC input	HSYNC input in external synchronous mode.
	EX_VSYNC	I	VSYNC input	VSYNC input in external synchronous mode.
	DCLKIN	I	Panel source clock input	Display source clock input. Input an appropriate frequency depending on the display panel size.
	DCLKOUT	O	Panel clock output	Panel clock output.



Classification	Symbol	I/O	Name	Function
Direct memory access controller (DMAC)	<u>DREQ0</u> , <u>DREQ1</u>	I	DMA-transfer request	Input pins to receive external requests for DMA transfer.
	<u>DACK0</u> , <u>DACK1</u>	O	DMA-transfer request acknowledge	Output pins for signals indicating acknowledge of external requests from external devices.
	<u>DTEND0</u> , <u>DTEND1</u>	O	DMA-transfer end output	Output pins for DMA transfer end.
Ethernet controller (EtherC)	CRS	I	Carrier sense	Carrier sense signal input.
	COL	I	Collision	Signal collision detection signal input.
	MII_TXD3 to MII_TXD0	O	Transmit data	4-bit transmit data. Connect them to the data transmit pins of the PHY.
	TX_EN	O	Transmit enable	Indicates that transmit data is ready on MII_TXD pins.
	TX_CLK	I	Transmit clock	Clock signal for TX_EN, TX_ER, and MII_TXD.
	TX_ER	O	Transmit error	Notifies PHY_LSI of error during transmission.
	MII_RXD3 to MII_RXD0	I	Receive data	4-bit receive data. Connect them to the data receive pins of the PHY.
	RX_DV	I	Receive data valid	Indicates that receive data is ready on MII_RXD pins.
	RX_CLK	I	Receive clock	Clock signal for RX_DV, RX_ER, and MII_RXD.
	RX_ER	I	Receive error	Indicates the error during reception.
	MDC	O	Management clock	Clock signal for information transfer via MDIO.
	MDIO	I/O	Management data	Bidirectional data for exchange of management information.
	WOL	O	MAGIC packet receive	Receives Magic packets.
	LNKSTA	I	Link status	Inputs link status from the PHY-LSI.
	EXOUT	O	General output	External output.

Classification	Symbol	I/O	Name	Function
ATAPI interface (ATAPI)	IDED15 to IDED0, IDED15_M to IDED0_M	I/O	IDE data bus	Bidirectional data bus. IDED15_M to IDED0_M are mirror pins.
	IDEA2 to IDEA0, IDEA2_M to IDEA0_M	O	IDE address bus	IDE address output. IDEA2_M to IDEA0_M are mirror pins.
	$\overline{\text{IODACKI}}$ , $\overline{\text{ODACK\_M}}$	O	IDEDMA acknowledge	Primary channel DMA acknowledge signal (active low). $\overline{\text{IODACK\_M}}$ is a mirror pin.
	IODREQ, IODREQ_M	I	IDEDMA request	Primary channel DMA request signal (active high). IODREQ_M is a mirror pin.
	$\overline{\text{IDECS1}}$ , $\overline{\text{IDECS0}}$ , $\overline{\text{IDECS1\_M}}$ , $\overline{\text{IDECS0\_M}}$	O	IDE chip select	Primary channel chip select signal (active low). $\overline{\text{IDECS1\_M}}$ and $\overline{\text{IDECS0\_M}}$ are mirror pins.
	$\overline{\text{IDEIOWR}}$ , $\overline{\text{IDEIOWR\_M}}$	O	IDE write	Primary channel write signal (active low). $\overline{\text{IDEIOWR\_M}}$ is a mirror pin.
	$\overline{\text{IDEIORD}}$ , $\overline{\text{IDEIORD\_M}}$	O	IDE read	Primary channel read signal (active low). $\overline{\text{IDEIORD\_M}}$ is a mirror pin.
	IDEIORDY, IDEIORDY_M	I	IDE ready	Primary channel ready signal (active high). IDEIORDY_M is a mirror pin.
	IDEINT, IDEINT_M	I	IDE interrupt	Primary channel interrupt request signal (active high). IDEINT_M is a mirror pin.
	$\overline{\text{IDERST}}$ , $\overline{\text{IDERST\_M}}$	O	IDE reset	Primary channel ATAPI device reset signal (active low). $\overline{\text{IDERST\_M}}$ is a mirror pin.
	DIRECTION, DIRECTION_M	O	Direction	External level shifter direction signal (0 when writing to the device). DIRECTION_M is a mirror pin.

Classification	Symbol	I/O	Name	Function
Serial communication interface with FIFO (SCIF)	SCK0, SCK1, SCK2	I/O	Serial clock	Serial clock I/O.
	TXD0, TXD1, TXD2	O	Transmit data	Serial data output.
	RXD0, RXD1, RXD2	I	Receive data	Serial data input.
	CTS0, CTS1, CTS2	I/O	Modem control transmit enable	Modem control signals to stop or restart data transmission.
	RTS0, RTS1, RTS2	I/O	Modem control transmit request	Modem control signals to stop or restart data reception.
I <sup>2</sup> C bus interface (IIC)	SCL	I/O	Serial clock	Serial clock I/O.
	SDA	I/O	Serial data	Serial data I/O.
USB host/function controller (USB)	XIN	I	Crystal resonator/external clock for USB	Connected to a crystal resonator or the external clock for USB operation.
	XOUT	O	Crystal resonator for USB	Connected to a crystal resonator for USB operation.
	DP	I/O	D+	USB D+ signal
	DM	I/O	D-	USB D- signal
	VBUS	I	Vbus	USB Vbus signal
	REFRIN	I	Reference input	Connected to the analog ground through a resistance of 5.6 kΩ (±1%)
	VDD_USB	Digital power supply	Power supply for USB PHY digital section	Power supply for the digital section of the USB PHY. Input 1.2 V.
	VSS_USB	Digital ground	Ground for USB PHY digital section	Ground for the digital section of the USB PHY. Input 0 V.
	VDDQ_USB	Digital power supply	Power supply for USB PHY digital section	Power supply for the digital section of the USB PHY. Input 3.3 V.
	VSSQ_USB	Digital ground	Ground for USB PHY digital section	Ground for the digital section of the USB PHY. Input 0 V.

Classification	Symbol	I/O	Name	Function
USB host/function controller (USB)	VDDA_USB	Analog power supply	Power supply for USB PHY analog section	Power supply for the analog section of the USB PHY. Input 1.2 V.
	VSSA_USB	Analog ground	Ground for USB PHY analog section	Ground for the analog section of the USB PHY. Input 0 V.
	VDDQA_USB	Analog power supply	Power supply for USB PHY analog section	Power supply for the analog section of the USB PHY. Input 3.3 V.
	VSSQA_USB	Analog ground	Ground for USB PHY analog section	Ground for the analog section of the USB PHY. Input 0 V.
	UV12	Analog power supply	USB480 MHz power supply	Power supply for 480 MHz operation block. Input 1.2 V.
	UG12	Analog ground	USB480 MHz ground	Ground for 480 MHz operation block. Input 0 V.
32-bit timer (TMU)	TCLK	I	Timer clock	External clock input for the timer. It can also be used for the input capture signal in channel 2.

Classification	Symbol	I/O	Name	Function
Serial sound interface (SSI)	SSIDATA0, SSIDATA1, SSIDATA2, SSIDATA3, SSIDATA4, SSIDATA5	I/O	SSI data I/O	Serial data I/O.
	SSISCK0, SSISCK1, SSISCK2, SSISCK3, SSISCK4, SSISCK5,	I/O	SSI clock I/O	Serial clock I/O.
	SSIWS0, SSIWS1, SSIWS2, SSIWS3, SSIWS4, SSIWS5	I/O	SSI clock L R I/O	Word select I/O.
	AUDIO_CLK0, AUDIO_CLK1, AUDIO_CLK2, AUDIO_CLK3, AUDIO_CLK4, AUDIO_CLK5	I	SSI audio external clock	External clock input for audio. This clock is input to the frequency divider.
LCD controller (LCDC)	LCD_DATA15 to LCD_DATA0	O	LCD data	LCD panel data output.
	LCD_DON	O	Display start	Display start (DON) signal.
	LCD_CL1	O	Shift clock 1	LCD shift clock 1/horizontal sync signal.
	LCD_CL2	O	Shift clock 2	LCD shift clock 2/dot clock.
	LCD_CLK	I	Clock source	LCD clock source input.
	LCD_FLM	O	Line marker	First line marker/vertical sync signal.
	LCD_VCPWC	O	Power control (VCC)	LCD module power control (VCC).
	LCD_VEPWC	O	Power control (VEE)	LCD module power control (VEE).
	LCD_M_DISP	O	LCD current-alternating	LCD current-alternating/DISP signal.

Classification	Symbol	I/O	Name	Function
NAND flash memory controller (FLCTL)	$\overline{\text{FCE}}$	O	Chip enable	Chip enable pin.
	FD7 to FD0	I/O	Data I/O	Command, address, and data I/O.
	FCLE	O	Command latch enable	Command latch enable (CLE). Asserted when a command is output.
	FALE	O	Address latch enable	Address latch enable (ALE). Asserted when an address is output and negated when data is input or output.
	$\overline{\text{FRE}}$	O	Read enable	Read enable (RE). Reads data at the falling edge of RE.
	$\overline{\text{FWE}}$	O	Write enable	Write enable. Flash memory latches a command, address, and data at the rising edge of WE.
	FR/ $\overline{\text{B}}$	I	Ready/busy	Ready/busy. Indicates ready state at a high level or busy state at a low level.
I/O ports (GPIO)	PA7 to PA0	I/O	General port	8-bit general I/O port.
	PB7 to PB0	I/O	General port	8-bit general I/O port.
	PC7 to PC0	I/O	General port	8-bit general I/O port.
	PD7 to PD0	I/O	General port	8-bit general I/O port.
	PE7 to PE0	I/O	General port	8-bit general I/O port.
	PF7 to PF0	I/O	General port	8-bit general I/O port.
	PG7 to PG0	I/O	General port	8-bit general I/O port.
	PH7 to PH0	I/O	General port	8-bit general I/O port.
	PI4 to PI0	I/O	General port	5-bit general I/O port.
	PJ7 to PJ0	I/O	General port	8-bit general I/O port.

Classification	Symbol	I/O	Name	Function
User debugging interface (H-UDI)	TCK	I	Test clock	Test clock input.
	TMS	I	Test mode select	Test mode select signal input.
	TRST	I	Test reset	Initialization signal input.
	TDI	I	Test data input	Serial input for instructions and data.
	TDO	O	Test data output	Serial output for instructions and data.
Advanced user debugger (AUD)	AUDATA3 to AUDATA0, AUDCK, AUDSYNC	O	Emulator pins	Dedicated emulator pins.
	ASEBRKAK/ BRKACK	I/O	Emulator pins	Dedicated emulator pins.
	MPMD	I	Chip mode pin	Selects emulation support mode (MPMD = low) or LSI operation mode (MPMD = high).

## 1.5 Address Map

Figure 1.3 shows the address map of this LSI.

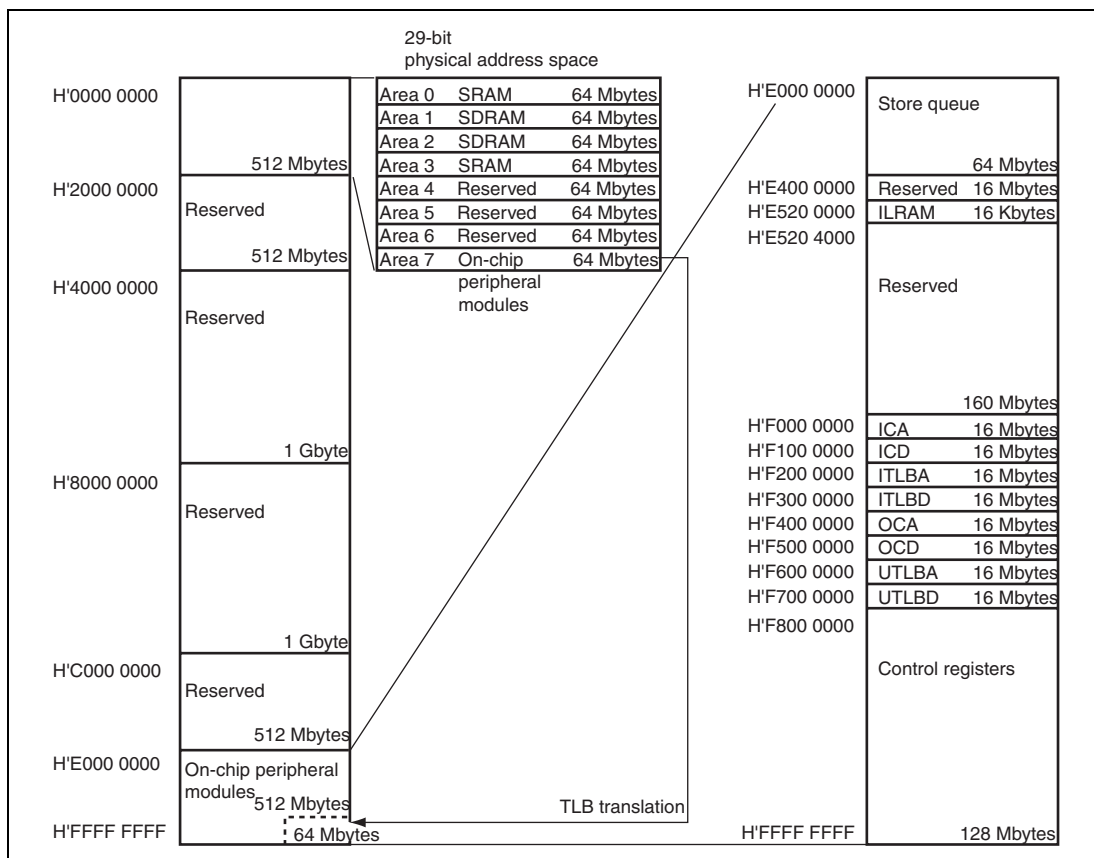


Figure 1.3 Physical Address Space (1)



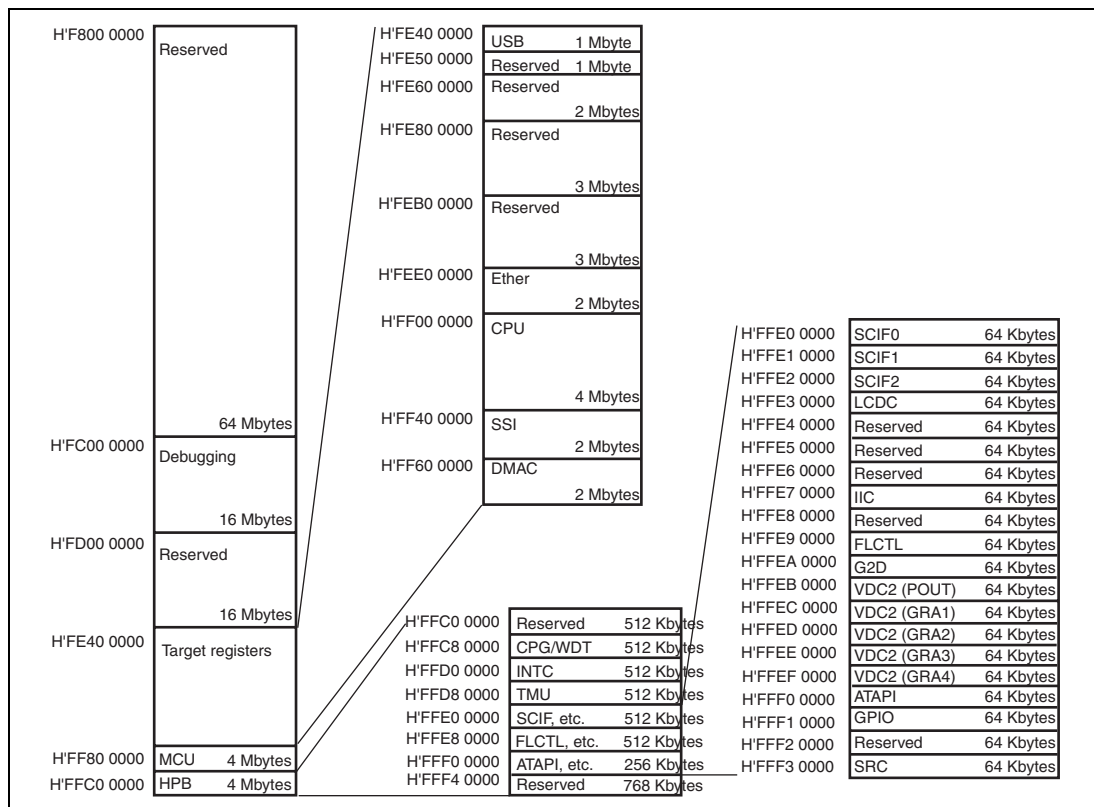


Figure 1.3 Physical Address Space (2)

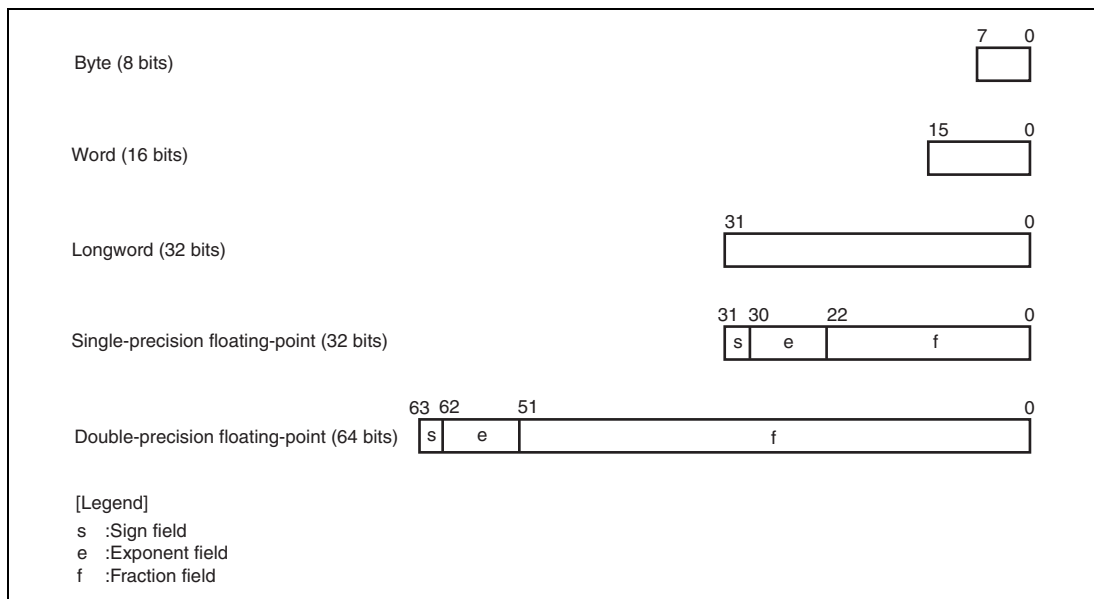


## Section 2 Programming Model

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

### 2.1 Data Formats

The data formats supported in the SH-4A 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.

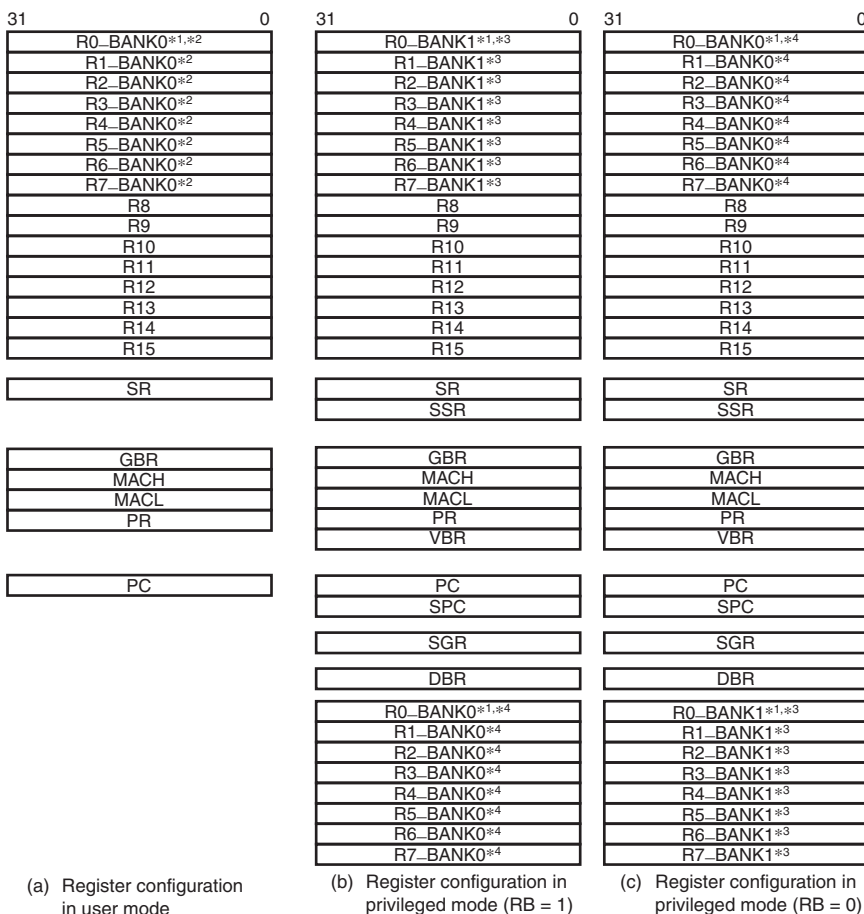
**Table 2.1 Initial Register Values**

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.

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. The SH-4A 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. The SH-4A 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.

SR.MD = 0 or (SR.MD = 1, SR.RB = 0)		(SR.MD = 1, SR.RB = 1)	
R0	R0_BANK0	R0-BANK0	
R1	R1_BANK0	R1-BANK0	
R2	R2_BANK0	R2-BANK0	
R3	R3_BANK0	R3-BANK0	
R4	R4_BANK0	R4-BANK0	
R5	R5_BANK0	R5-BANK0	
R6	R6_BANK0	R6-BANK0	
R7	R7_BANK0	R7-BANK0	
R0-BANK1	R0_BANK1	R0	
R1-BANK1	R1_BANK1	R1	
R2-BANK1	R2_BANK1	R2	
R3-BANK1	R3_BANK1	R3	
R4-BANK1	R4_BANK1	R4	
R5-BANK1	R5_BANK1	R5	
R6-BANK1	R6_BANK1	R6	
R7-BANK1	R7_BANK1	R7	
R8	R8	R8	
R9	R9	R9	
R10	R10	R10	
R11	R11	R11	
R12	R12	R12	
R13	R13	R13	
R14	R14	R14	
R15	R15	R15	

**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, XD<sub>i</sub> (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.

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

FPSR.FR = 0			FPSR.FR = 1			
FV0	DR0	FR0	FPR0_BANK0	XF0	XD0	XMTRX
		FR1	FPR1_BANK0	XF1		
	DR2	FR2	FPR2_BANK0	XF2	XD2	
		FR3	FPR3_BANK0	XF3		
FV4	DR4	FR4	FPR4_BANK0	XF4	XD4	
		FR5	FPR5_BANK0	XF5		
	DR6	FR6	FPR6_BANK0	XF6	XD6	
		FR7	FPR7_BANK0	XF7		
FV8	DR8	FR8	FPR8_BANK0	XF8	XD8	
		FR9	FPR9_BANK0	XF9		
	DR10	FR10	FPR10_BANK0	XF10	XD10	
		FR11	FPR11_BANK0	XF11		
FV12	DR12	FR12	FPR12_BANK0	XF12	XD12	
		FR13	FPR13_BANK0	XF13		
	DR14	FR14	FPR14_BANK0	XF14	XD14	
		FR15	FPR15_BANK0	XF15		
XMTRX	XD0	XF0	FPR0_BANK1	FR0	DR0	FV0
		XF1	FPR1_BANK1	FR1		
	XD2	XF2	FPR2_BANK1	FR2	DR2	
		XF3	FPR3_BANK1	FR3		
	XD4	XF4	FPR4_BANK1	FR4	DR4	FV4
		XF5	FPR5_BANK1	FR5		
	XD6	XF6	FPR6_BANK1	FR6	DR6	
		XF7	FPR7_BANK1	FR7		
	XD8	XF8	FPR8_BANK1	FR8	DR8	FV8
		XF9	FPR9_BANK1	FR9		
	XD10	XF10	FPR10_BANK1	FR10	DR10	
		XF11	FPR11_BANK1	FR11		
	XD12	XF12	FPR12_BANK1	FR12	DR12	FV12
		XF13	FPR13_BANK1	FR13		
	XD14	XF14	FPR14_BANK1	FR14	DR14	
		XF15	FPR15_BANK1	FR15		

**Figure 2.4 Floating-Point Registers**

## 2.2.4 Control Registers

### (1) Status Register (SR)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 2.2.5 System Registers

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

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

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

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

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

PC indicates the address of the instruction currently being executed.

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

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

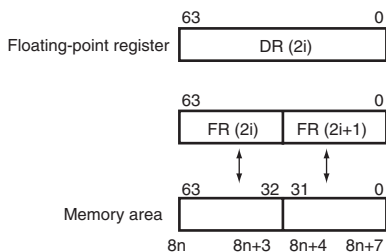
  

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause				Enable (EN)						Flag				RM	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

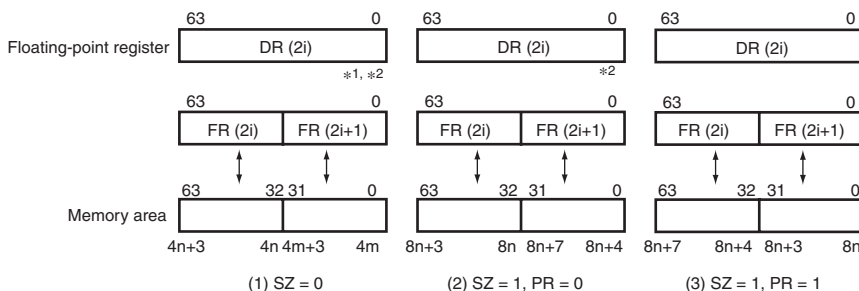
Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
21	FR	0	R/W	Floating-Point Register Bank 0: FPR0_BANK0 to FPR15_BANK0 are assigned to FR0 to FR15 and FPR0_BANK1 to FPR15_BANK1 are assigned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

Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	000000	R/W	FPU Exception Cause Field
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

&lt;Big endian&gt;



&lt;Little endian&gt;



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

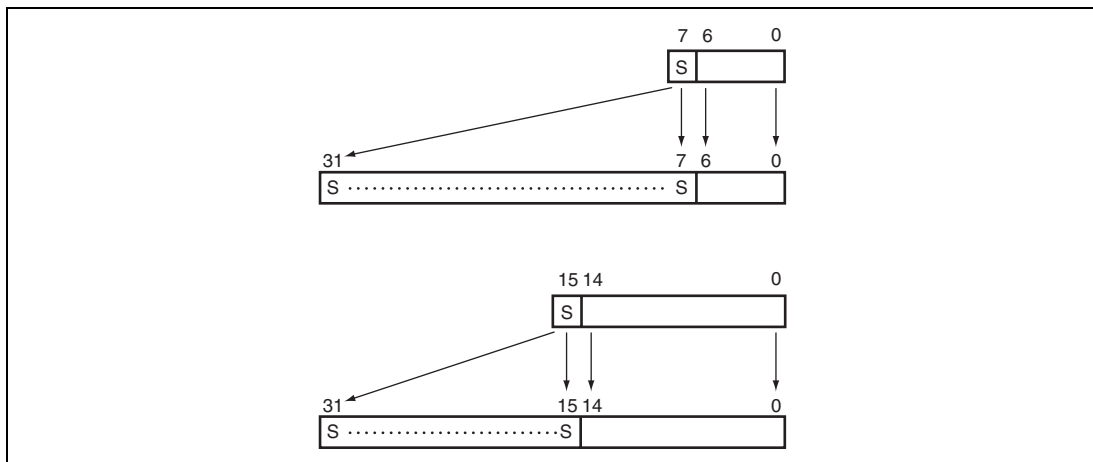
Access to area H'FC00 0000 to H'FFFF FFFF in user mode will cause an address error.

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**

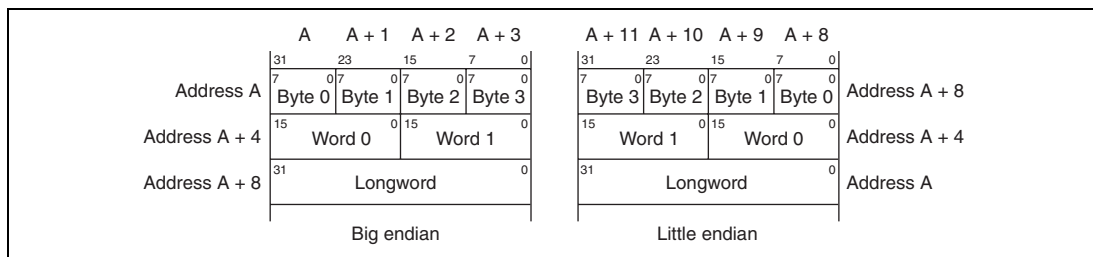
## 2.5 Data Formats in Memory

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.

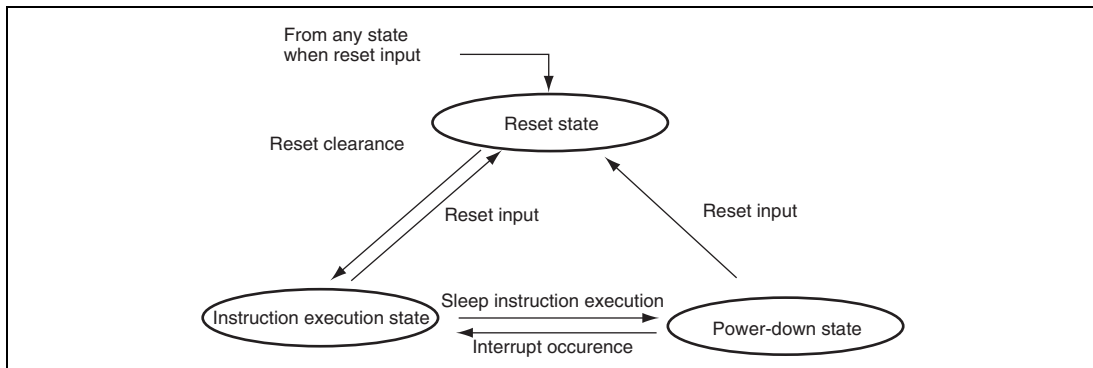
In the power-on reset state, the internal state of the CPU and the on-chip peripheral module registers are initialized. For details, see register descriptions for each section.

### (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 power-down state: sleep mode and standby mode. For details, see section 28, 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 the SH-4A has been significantly enhanced from that of the SH-4. Therefore, in the case when a code in memory is rewritten and attempted to be executed immediately, there is increased possibility that the code before being modified, which has already been prefetched, is executed.

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

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

```
SYNCO
ICBI @Rn
```

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

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

```
SYNCO
ICBI @Rn
```

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

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

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

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

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



## Section 3 Instruction Set

The SH-4A's instruction set is implemented with 16-bit fixed-length instructions. The SH-4A 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 the SH-4A moves byte-size or word-size data from memory to a register, the data is sign-extended.

### 3.1 Execution Environment

#### (1) PC

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

#### (2) Load-Store Architecture

The SH-4A 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, the SH-4A'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
```

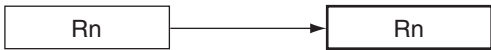
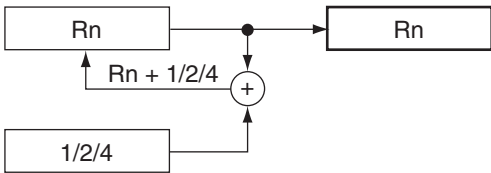
There are no PC-relative load instructions for floating-point operations. However, it is possible to set 0.0 or 1.0 by using the FLDI0 or FLDI1 instruction on a single-precision floating-point register.

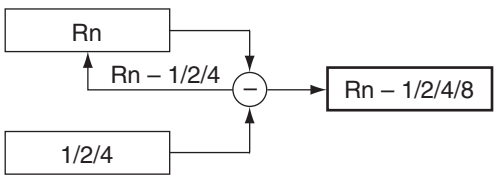
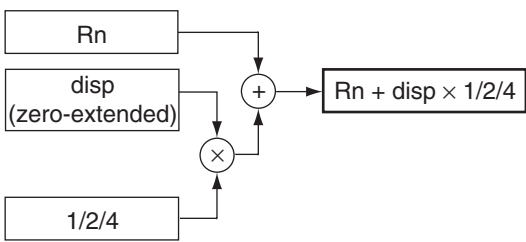
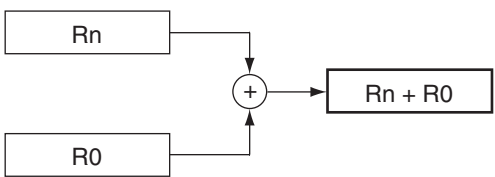


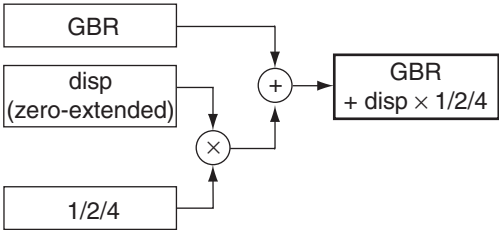
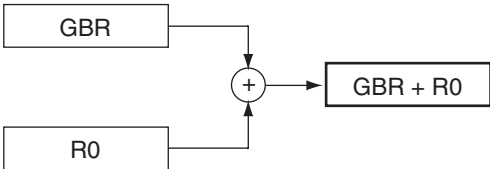
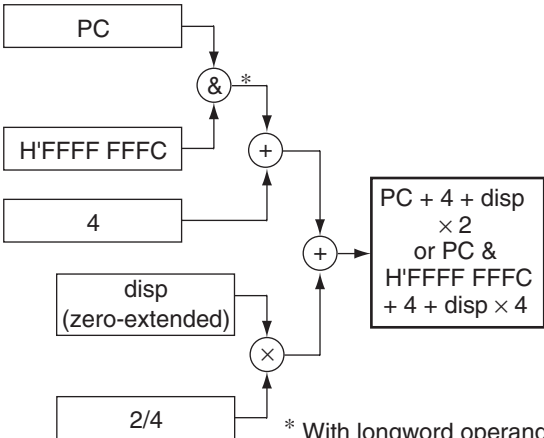
## 3.2 Addressing Modes

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	<p>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.</p>  <pre> graph LR     Rn[Rn] --&gt; Minus((−))     Const[1/2/4] --&gt; Minus     Minus --&gt; EA[Rn - 1/2/4/8]           </pre>	<p>Byte: <math>Rn - 1 \rightarrow Rn</math></p> <p>Word: <math>Rn - 2 \rightarrow Rn</math></p> <p>Longword: <math>Rn - 4 \rightarrow Rn</math></p> <p>Quadword: <math>Rn - 8 \rightarrow Rn</math></p> <p><math>Rn \rightarrow EA</math> (Instruction executed with Rn after calculation)</p>
Register indirect with displacement	@(disp:4, Rn)	<p>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.</p>  <pre> graph LR     Rn[Rn] --&gt; Plus1((+))     Disp[disp (zero-extended)] --&gt; Mult((×))     Const[1/2/4] --&gt; Mult     Mult --&gt; Plus1     Plus1 --&gt; EA[Rn + disp × 1/2/4]           </pre>	<p>Byte: <math>Rn + disp \rightarrow EA</math></p> <p>Word: <math>Rn + disp \times 2 \rightarrow EA</math></p> <p>Longword: <math>Rn + disp \times 4 \rightarrow EA</math></p>
Indexed register indirect	@(R0, Rn)	<p>Effective address is sum of register Rn and R0 contents.</p>  <pre> graph LR     Rn[Rn] --&gt; Plus2((+))     R0[R0] --&gt; Plus2     Plus2 --&gt; EA[Rn + R0]           </pre>	$Rn + R0 \rightarrow EA$

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: $\text{GBR} + \text{disp} \rightarrow \text{EA}$ Word: $\text{GBR} + \text{disp} \times 2 \rightarrow \text{EA}$ Longword: $\text{GBR} + \text{disp} \times 4 \rightarrow \text{EA}$
			
Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents.	$\text{GBR} + \text{R0} \rightarrow \text{EA}$
			
PC-relative with displacement	@(disp:8, PC)	Effective address is $\text{PC} + 4$ 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: $\text{PC} + 4 + \text{disp} \times 2 \rightarrow \text{EA}$ Longword: $\text{PC} \& \text{H'FFFF FFFC} + 4 + \text{disp} \times 4 \rightarrow \text{EA}$
			

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.	$PC + 4 + \text{disp} \times 2 \rightarrow \text{Branch-Target}$
<pre> graph TD     PC[PC] --&gt; A((+))     4[4] --&gt; A     A --&gt; B((+))     disp[disp sign-extended] --&gt; C((x))     2[2] --&gt; C     C --&gt; B     B --&gt; Result[PC + 4 + disp x 2] </pre>			
PC-relative	disp:12	Effective address is PC + 4 with 12-bit displacement disp added after being sign-extended and multiplied by 2.	$PC + 4 + \text{disp} \times 2 \rightarrow \text{Branch-Target}$
<pre> graph TD     PC[PC] --&gt; A((+))     4[4] --&gt; A     A --&gt; B((+))     disp[disp sign-extended] --&gt; C((x))     2[2] --&gt; C     C --&gt; B     B --&gt; Result[PC + 4 + disp x 2] </pre>			
Rn		Effective address is sum of PC + 4 and Rn.	$PC + 4 + Rn \rightarrow \text{Branch-Target}$
<pre> graph TD     PC[PC] --&gt; A((+))     4[4] --&gt; A     A --&gt; B((+))     Rn[Rn] --&gt; B     B --&gt; Result[PC + 4 + Rn] </pre>			

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 ( $\times 1$ ,  $\times 2$ , or  $\times 4$ ) 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 iii: 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 ( $\times 1$ ,  $\times 2$ ,  $\times 4$ , or  $\times 8$ ) 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 $\rightarrow$ sign extension $\rightarrow$ Rn	1110nnnniiiiiii	—	—	—
MOV.W	@(disp*,PC), Rn	(disp $\times 2$ + PC + 4) $\rightarrow$ sign extension $\rightarrow$ Rn	1001nnnnddddddd	—	—	—
MOV.L	@(disp*,PC), Rn	(disp $\times 4$ + PC & H'FFFF FFFC + 4) $\rightarrow$ Rn	1101nnnnddddddd	—	—	—
MOV	Rm,Rn	Rm $\rightarrow$ Rn	0110nnnnnnmm0011	—	—	—
MOV.B	Rm,@Rn	Rm $\rightarrow$ (Rn)	0010nnnnnnmm0000	—	—	—
MOV.W	Rm,@Rn	Rm $\rightarrow$ (Rn)	0010nnnnnnmm0001	—	—	—
MOV.L	Rm,@Rn	Rm $\rightarrow$ (Rn)	0010nnnnnnmm0010	—	—	—
MOV.B	@Rm,Rn	(Rm) $\rightarrow$ sign extension $\rightarrow$ Rn	0110nnnnnnmm0000	—	—	—
MOV.W	@Rm,Rn	(Rm) $\rightarrow$ sign extension $\rightarrow$ Rn	0110nnnnnnmm0001	—	—	—
MOV.L	@Rm,Rn	(Rm) $\rightarrow$ Rn	0110nnnnnnmm0010	—	—	—
MOV.B	Rm,@-Rn	Rn-1 $\rightarrow$ Rn, Rm $\rightarrow$ (Rn)	0010nnnnnnmm0100	—	—	—
MOV.W	Rm,@-Rn	Rn-2 $\rightarrow$ Rn, Rm $\rightarrow$ (Rn)	0010nnnnnnmm0101	—	—	—
MOV.L	Rm,@-Rn	Rn-4 $\rightarrow$ Rn, Rm $\rightarrow$ (Rn)	0010nnnnnnmm0110	—	—	—
MOV.B	@Rm+,Rn	(Rm) $\rightarrow$ sign extension $\rightarrow$ Rn, Rm + 1 $\rightarrow$ Rm	0110nnnnnnmm0100	—	—	—
MOV.W	@Rm+,Rn	(Rm) $\rightarrow$ sign extension $\rightarrow$ Rn, Rm + 2 $\rightarrow$ Rm	0110nnnnnnmm0101	—	—	—
MOV.L	@Rm+,Rn	(Rm) $\rightarrow$ Rn, Rm + 4 $\rightarrow$ Rm	0110nnnnnnmm0110	—	—	—
MOV.B	R0,@(disp*,Rn)	R0 $\rightarrow$ (disp + Rn)	10000000nnnndddd	—	—	—
MOV.W	R0,@(disp*,Rn)	R0 $\rightarrow$ (disp $\times 2$ + Rn)	10000001nnnndddd	—	—	—
MOV.L	Rm,@(disp*,Rn)	Rm $\rightarrow$ (disp $\times 4$ + Rn)	0001nnnnnnmmdddd	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
MOV.B	@(disp*,Rm),R0	(disp + Rm) → sign extension → R0	10000100mmmmddddd	—	—	—
MOV.W	@(disp*,Rm),R0	(disp × 2 + Rm) → sign extension → R0	10000101mmmmddddd	—	—	—
MOV.L	@(disp*,Rm),Rn	(disp × 4 + Rm) → Rn	0101nnnnmmmmddddd	—	—	—
MOV.B	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0100	—	—	—
MOV.W	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0101	—	—	—
MOV.L	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	—	—	—
MOV.B	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1100	—	—	—
MOV.W	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1101	—	—	—
MOV.L	@(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	—	—
MOV.B	R0,@(disp*,GBR)	R0 → (disp + GBR)	11000000ddddddddd	—	—	—
MOV.W	R0,@(disp*,GBR)	R0 → (disp × 2 + GBR)	11000001ddddddddd	—	—	—
MOV.L	R0,@(disp*,GBR)	R0 → (disp × 4 + GBR)	11000010ddddddddd	—	—	—
MOV.B	@(disp*,GBR),R0	(disp + GBR) → sign extension → R0	11000100ddddddddd	—	—	—
MOV.W	@(disp*,GBR),R0	(disp × 2 + GBR) → sign extension → R0	11000101ddddddddd	—	—	—
MOV.L	@(disp*,GBR),R0	(disp × 4 + GBR) → R0	11000110ddddddddd	—	—	—
MOVA	@(disp*,PC),R0	disp × 4 + PC & H'FFFF FFFC + 4 → R0	11000111ddddddddd	—	—	—
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



Instruction		Operation	Instruction Code	Privileged	T Bit	New
MOVT	Rn	$T \rightarrow Rn$	0000nnnn00101001	—	—	—
SWAP.B	Rm,Rn	$Rm \rightarrow \text{swap lower 2 bytes} \rightarrow Rn$	0110nnnnnnnnnn1000	—	—	—
SWAP.W	Rm,Rn	$Rm \rightarrow \text{swap upper/lower words} \rightarrow Rn$	0110nnnnnnnnnn1001	—	—	—
XTRCT	Rm,Rn	$Rm:Rn \text{ middle 32 bits} \rightarrow Rn$	0010nnnnnnnnnn1101	—	—	—
Note: * The assembler of Renesas uses the value after scaling ( $\times 1$ , $\times 2$ , or $\times 4$ ) as the displacement (disp).						

**Table 3.5 Arithmetic Operation Instructions**

Instruction		Operation	Instruction Code	Privileged	T Bit	New
ADD	Rm,Rn	$Rn + Rm \rightarrow Rn$	0011nnnnnnnnnn1100	—	—	—
ADD	#imm,Rn	$Rn + \text{imm} \rightarrow Rn$	0111nnnniiiiiii	—	—	—
ADDC	Rm,Rn	$Rn + Rm + T \rightarrow Rn$ , carry $\rightarrow T$	0011nnnnnnnnnn1110	—	Carry	—
ADDV	Rm,Rn	$Rn + Rm \rightarrow Rn$ , overflow $\rightarrow T$	0011nnnnnnnnnn1111	—	Overflow	—
CMP/EQ	#imm,R0	When $R0 = \text{imm}$ , $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	10001000iiiiiii	—	Comparison result	—
CMP/EQ	Rm,Rn	When $Rn = Rm$ , $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	0011nnnnnnnnnn0000	—	Comparison result	—
CMP/HS	Rm,Rn	When $Rn \geq Rm$ (unsigned), $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	0011nnnnnnnnnn0010	—	Comparison result	—
CMP/GE	Rm,Rn	When $Rn \geq Rm$ (signed), $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	0011nnnnnnnnnn0011	—	Comparison result	—
CMP/HI	Rm,Rn	When $Rn > Rm$ (unsigned), $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	0011nnnnnnnnnn0110	—	Comparison result	—
CMP/GT	Rm,Rn	When $Rn > Rm$ (signed), $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	0011nnnnnnnnnn0111	—	Comparison result	—
CMP/PZ	Rn	When $Rn \geq 0$ , $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	0100nnnn00010001	—	Comparison result	—
CMP/PL	Rn	When $Rn > 0$ , $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	0100nnnn00010101	—	Comparison result	—

Instruction	Operation	Instruction Code	Privileged	T Bit	New
CMP/STR Rm,Rn	When any bytes are equal, $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	0010nnnnnnmmmm1100	—	Comparison result	—
DIV1 Rm,Rn	1-step division ( $Rn \div Rm$ )	0011nnnnnnmmmm0100	—	Calculation result	—
DIV0S Rm,Rn	MSB of $Rn \rightarrow Q$ , MSB of $Rm \rightarrow M$ , $M \div Q \rightarrow T$	0010nnnnnnmmmm0111	—	Calculation result	—
DIV0U	$0 \rightarrow M/Q/T$	0000000000011001	—	0	—
DMULS.L Rm,Rn	Signed, $Rn \times Rm \rightarrow MAC$ , $32 \times 32 \rightarrow 64$ bits	0011nnnnnnmmmm1101	—	—	—
DMULU.L Rm,Rn	Unsigned, $Rn \times Rm \rightarrow MAC$ , $32 \times 32 \rightarrow 64$ bits	0011nnnnnnmmmm0101	—	—	—
DT Rn	$Rn - 1 \rightarrow Rn$ ; when $Rn = 0$ , $1 \rightarrow T$ When $Rn \neq 0$ , $0 \rightarrow T$	0100nnnn00010000	—	Comparison result	—
EXTS.B Rm,Rn	Rm sign-extended from byte $\rightarrow Rn$	0110nnnnnnmmmm1110	—	—	—
EXTS.W Rm,Rn	Rm sign-extended from word $\rightarrow Rn$	0110nnnnnnmmmm1111	—	—	—
EXTU.B Rm,Rn	Rm zero-extended from byte $\rightarrow Rn$	0110nnnnnnmmmm1100	—	—	—
EXTU.W Rm,Rn	Rm zero-extended from word $\rightarrow Rn$	0110nnnnnnmmmm1101	—	—	—
MAC.L @Rm+,@Rn+	Signed, $(Rn) \times (Rm) + MAC \rightarrow MAC$ $Rn + 4 \rightarrow Rn$ , $Rm + 4 \rightarrow Rm$ $32 \times 32 + 64 \rightarrow 64$ bits	0000nnnnnnmmmm1111	—	—	—
MAC.W @Rm+,@Rn+	Signed, $(Rn) \times (Rm) + MAC \rightarrow MAC$ $Rn + 2 \rightarrow Rn$ , $Rm + 2 \rightarrow Rm$ $16 \times 16 + 64 \rightarrow 64$ bits	0100nnnnnnmmmm1111	—	—	—
MUL.L Rm,Rn	$Rn \times Rm \rightarrow MACL$ $32 \times 32 \rightarrow 32$ bits	0000nnnnnnmmmm0111	—	—	—
MULS.W Rm,Rn	Signed, $Rn \times Rm \rightarrow MACL$ $16 \times 16 \rightarrow 32$ bits	0010nnnnnnmmmm1111	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
MULU.W	Rm,Rn	Unsigned, $Rn \times Rm \rightarrow MACL$ $16 \times 16 \rightarrow 32$ bits	0010nnnnnnmm1110	—	—	—
NEG	Rm,Rn	$0 - Rm \rightarrow Rn$	0110nnnnnnmm1011	—	—	—
NEGC	Rm,Rn	$0 - Rm - T \rightarrow Rn$ , borrow $\rightarrow T$	0110nnnnnnmm1010	—	Borrow	—
SUB	Rm,Rn	$Rn - Rm \rightarrow Rn$	0011nnnnnnmm1000	—	—	—
SUBC	Rm,Rn	$Rn - Rm - T \rightarrow Rn$ , borrow $\rightarrow T$	0011nnnnnnmm1010	—	Borrow	—
SUBV	Rm,Rn	$Rn - Rm \rightarrow Rn$ , underflow $\rightarrow T$	0011nnnnnnmm1011	—	Underflow	—

Table 3.6 Logic Operation Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
AND	Rm,Rn	$Rn \& Rm \rightarrow Rn$	0010nnnnnnmm1001	—	—	—
AND	#imm,R0	$R0 \& imm \rightarrow R0$	11001001iiiiiii	—	—	—
AND.B	#imm, @(R0,GBR)	$(R0 + GBR) \& imm \rightarrow (R0 + GBR)$	11001101iiiiiii	—	—	—
NOT	Rm,Rn	$\sim Rm \rightarrow Rn$	0110nnnnnnmm0111	—	—	—
OR	Rm,Rn	$Rn   Rm \rightarrow Rn$	0010nnnnnnmm1011	—	—	—
OR	#imm,R0	$R0   imm \rightarrow R0$	11001011iiiiiii	—	—	—
OR.B	#imm, @(R0,GBR)	$(R0 + GBR)   imm \rightarrow (R0 + GBR)$	11001111iiiiiii	—	—	—
TAS.B	@Rn	When $(Rn) = 0$ , $1 \rightarrow T$ Otherwise, $0 \rightarrow T$ In both cases, $1 \rightarrow$ MSB of $(Rn)$	0100nnnn00011011	—	Test result	—
TST	Rm,Rn	$Rn \& Rm$ ; when result = 0, $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	0010nnnnnnmm1000	—	Test result	—
TST	#imm,R0	$R0 \& imm$ ; when result = 0, $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	11001000iiiiiii	—	Test result	—
TST.B	#imm, @(R0,GBR)	$(R0 + GBR) \& imm$ ; when result = 0, $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	11001100iiiiiii	—	Test result	—
XOR	Rm,Rn	$Rn \wedge Rm \rightarrow Rn$	0010nnnnnnmm1010	—	—	—

Instruction	Operation	Instruction Code	Privileged	T Bit	New
XOR    #imm,R0	$R0 \wedge \text{imm} \rightarrow R0$	11001010iiiiiii	—	—	—
XOR.B   #imm, @(R0,GBR)	$(R0 + \text{GBR}) \wedge \text{imm} \rightarrow (R0 + \text{GBR})$	11001110iiiiiii	—	—	—

**Table 3.7    Shift Instructions**

Instruction	Operation	Instruction Code	Privileged	T Bit	New
ROTL    Rn	$T \leftarrow Rn \leftarrow \text{MSB}$	0100nnnn00000100	—	MSB	—
ROTR    Rn	$\text{LSB} \rightarrow Rn \rightarrow T$	0100nnnn00000101	—	LSB	—
ROTCL   Rn	$T \leftarrow Rn \leftarrow T$	0100nnnn00100100	—	MSB	—
ROTCR   Rn	$T \rightarrow Rn \rightarrow T$	0100nnnn00100101	—	LSB	—
SHAD    Rm,Rn	When $Rm \geq 0$ , $Rn \ll Rm \rightarrow Rn$ When $Rm < 0$ , $Rn \gg Rm \rightarrow [MSB \rightarrow Rn]$	0100nnnnnnnnnn1100	—	—	—
SHAL    Rn	$T \leftarrow Rn \leftarrow 0$	0100nnnn00100000	—	MSB	—
SHAR    Rn	$\text{MSB} \rightarrow Rn \rightarrow T$	0100nnnn00100001	—	LSB	—
SHLD    Rm,Rn	When $Rm \geq 0$ , $Rn \ll Rm \rightarrow Rn$ When $Rm < 0$ , $Rn \gg Rm \rightarrow [0 \rightarrow Rn]$	0100nnnnnnnnnn1101	—	—	—
SHLL    Rn	$T \leftarrow Rn \leftarrow 0$	0100nnnn00000000	—	MSB	—
SHLR    Rn	$0 \rightarrow Rn \rightarrow T$	0100nnnn00000001	—	LSB	—
SHLL2   Rn	$Rn \ll 2 \rightarrow Rn$	0100nnnn00001000	—	—	—
SHLR2   Rn	$Rn \gg 2 \rightarrow Rn$	0100nnnn00001001	—	—	—
SHLL8   Rn	$Rn \ll 8 \rightarrow Rn$	0100nnnn00011000	—	—	—
SHLR8   Rn	$Rn \gg 8 \rightarrow Rn$	0100nnnn00011001	—	—	—
SHLL16   Rn	$Rn \ll 16 \rightarrow Rn$	0100nnnn00101000	—	—	—
SHLR16   Rn	$Rn \gg 16 \rightarrow Rn$	0100nnnn00101001	—	—	—

**Table 3.8 Branch Instructions**

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

**Table 3.9 System Control Instructions**

Instruction		Operation	Instruction Code	Privileged	T Bit	New
CLRMACH		$0 \rightarrow \text{MACH}$ , $\text{MACL}$	0000000000101000	—	—	—
CLRS		$0 \rightarrow \text{S}$	0000000001001000	—	—	—
CLRT		$0 \rightarrow \text{T}$	0000000000001000	—	0	—
ICBI	@Rn	Invalidates instruction cache block	0000nnnn11100011	—	—	New
LDC	Rm,SR	$\text{Rm} \rightarrow \text{SR}$	0100mmmm00001110	Privileged	LSB	—
LDC	Rm,GBR	$\text{Rm} \rightarrow \text{GBR}$	0100mmmm00011110	—	—	—
LDC	Rm,VBR	$\text{Rm} \rightarrow \text{VBR}$	0100mmmm00101110	Privileged	—	—
LDC	Rm,SGR	$\text{Rm} \rightarrow \text{SGR}$	0100mmmm00111010	Privileged	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	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	—	—
LDC.L	@Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	Privileged	—	—
LDC.L	@Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	Privileged	—	—
LDC.L	@Rm+,DBR	(Rm) → DBR, Rm + 4 → Rm	0100mmmm11110110	Privileged	—	—
LDC.L	@Rm+,Rn_BANK	(Rm) → Rn_BANK, Rm + 4 → Rm	0100mmmm1nnn0111	Privileged	—	—
LDS	Rm,MACH	Rm → MACH	0100mmmm00001010	—	—	—
LDS	Rm,MACL	Rm → MACL	0100mmmm00011010	—	—	—
LDS	Rm,PR	Rm → PR	0100mmmm00101010	—	—	—
LDS.L	@Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	—	—	—
LDS.L	@Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	—	—	—
LDS.L	@Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	—	—	—
LDTLB		PTEH/PTEL (/PTEA) → TLB	0000000000111000	Privileged	—	—
MOVCA.L	R0,@Rn	R0 → (Rn) (without fetching cache block)	0000nnnn11000011	—	—	—
NOP		No operation	0000000000001001	—	—	—
OCBI	@Rn	Invalidates operand cache block	0000nnnn10010011	—	—	—
OCBP	@Rn	Writes back and invalidates operand cache block	0000nnnn10100011	—	—	—
OCBWB	@Rn	Writes back operand cache block	0000nnnn10110011	—	—	—
PREF	@Rn	(Rn) → operand cache	0000nnnn10000011	—	—	—
PREFI	@Rn	Reads 32-byte instruction block into instruction cache	0000nnnn11010011	—	—	New
RTE		Delayed branch, SSR/SPC → SR/PC	0000000000101011	Privileged	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
SETS		$1 \rightarrow S$	0000000001011000	—	—	—
SETT		$1 \rightarrow T$	0000000000011000	—	1	—
SLEEP		Sleep or standby	0000000000011011	Privileged	—	—
STC	SR,Rn	$SR \rightarrow Rn$	0000nnnn00000010	Privileged	—	—
STC	GBR,Rn	$GBR \rightarrow Rn$	0000nnnn00010010	—	—	—
STC	VBR,Rn	$VBR \rightarrow Rn$	0000nnnn00100010	Privileged	—	—
STC	SSR,Rn	$SSR \rightarrow Rn$	0000nnnn00110010	Privileged	—	—
STC	SPC,Rn	$SPC \rightarrow Rn$	0000nnnn01000010	Privileged	—	—
STC	SGR,Rn	$SGR \rightarrow Rn$	0000nnnn00111010	Privileged	—	—
STC	DBR,Rn	$DBR \rightarrow Rn$	0000nnnn11111010	Privileged	—	—
STC	Rm_BANK,Rn	$Rm\_BANK \rightarrow Rn$ (m = 0 to 7)	0000nnnn1mmmm0010	Privileged	—	—
STC.L	SR,@-Rn	$Rn - 4 \rightarrow Rn, SR \rightarrow (Rn)$	0100nnnn00000011	Privileged	—	—
STC.L	GBR,@-Rn	$Rn - 4 \rightarrow Rn, GBR \rightarrow (Rn)$	0100nnnn00010011	—	—	—
STC.L	VBR,@-Rn	$Rn - 4 \rightarrow Rn, VBR \rightarrow (Rn)$	0100nnnn00100011	Privileged	—	—
STC.L	SSR,@-Rn	$Rn - 4 \rightarrow Rn, SSR \rightarrow (Rn)$	0100nnnn00110011	Privileged	—	—
STC.L	SPC,@-Rn	$Rn - 4 \rightarrow Rn, SPC \rightarrow (Rn)$	0100nnnn01000011	Privileged	—	—
STC.L	SGR,@-Rn	$Rn - 4 \rightarrow Rn, SGR \rightarrow (Rn)$	0100nnnn00110010	Privileged	—	—
STC.L	DBR,@-Rn	$Rn - 4 \rightarrow Rn, DBR \rightarrow (Rn)$	0100nnnn11110010	Privileged	—	—
STC.L	Rm_BANK,@-Rn	$Rn - 4 \rightarrow Rn, Rm\_BANK \rightarrow (Rn)$ (m = 0 to 7)	0100nnnn1mmmm0011	Privileged	—	—
STS	MACH,Rn	$MACH \rightarrow Rn$	0000nnnn00001010	—	—	—
STS	MACL,Rn	$MACL \rightarrow Rn$	0000nnnn00011010	—	—	—
STS	PR,Rn	$PR \rightarrow Rn$	0000nnnn00101010	—	—	—
STS.L	MACH,@-Rn	$Rn - 4 \rightarrow Rn, MACH \rightarrow (Rn)$	0100nnnn00000010	—	—	—
STS.L	MACL,@-Rn	$Rn - 4 \rightarrow Rn, MACL \rightarrow (Rn)$	0100nnnn00010010	—	—	—
STS.L	PR,@-Rn	$Rn - 4 \rightarrow Rn, PR \rightarrow (Rn)$	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	11000011iiiiiii	—	—	—

**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	1111nnnnnnnnm1100	—	—	—
FMOV.S	@Rm,FRn	(Rm) → FRn	1111nnnnnnnnm1000	—	—	—
FMOV.S	@(R0,Rm),FRn	(R0 + Rm) → FRn	1111nnnnnnnnm0110	—	—	—
FMOV.S	@Rm+,FRn	(Rm) → FRn, Rm + 4 → Rm	1111nnnnnnnnm1001	—	—	—
FMOV.S	FRm,@Rn	FRm → (Rn)	1111nnnnnnnnm1010	—	—	—
FMOV.S	FRm,@-Rn	Rn-4 → Rn, FRm → (Rn)	1111nnnnnnnnm1011	—	—	—
FMOV.S	FRm,@(R0,Rn)	FRm → (R0 + Rn)	1111nnnnnnnnm0111	—	—	—
FMOV	DRm,DRn	DRm → DRn	1111nnn0nnnn01100	—	—	—
FMOV	@Rm,DRn	(Rm) → DRn	1111nnn0nnnn1000	—	—	—
FMOV	@(R0,Rm),DRn	(R0 + Rm) → DRn	1111nnn0nnnn0110	—	—	—
FMOV	@Rm+,DRn	(Rm) → DRn, Rm + 8 → Rm	1111nnn0nnnn1001	—	—	—
FMOV	DRm,@Rn	DRm → (Rn)	1111nnnnnnnm01010	—	—	—
FMOV	DRm,@-Rn	Rn-8 → Rn, DRm → (Rn)	1111nnnnnnnm01011	—	—	—
FMOV	DRm,@(R0,Rn)	DRm → (R0 + Rn)	1111nnnnnnnm00111	—	—	—
FLDS	FRm,FPUL	FRm → FPUL	1111nnnn00011101	—	—	—
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	$FRn + FRm \rightarrow FRn$	1111nnnnmmmm0000	—	—	—
FCMP/EQ	FRm,FRn	When $FRn = FRm$ , $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	1111nnnnmmmm0100	—	Comparison on result	—
FCMP/GT	FRm,FRn	When $FRn > FRm$ , $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	1111nnnnmmmm0101	—	Comparison on result	—
FDIV	FRm,FRn	$FRn/FRm \rightarrow FRn$	1111nnnnmmmm0011	—	—	—
FLOAT	FPUL,FRn	(float) FPUL $\rightarrow$ FRn	1111nnnn00101101	—	—	—
FMAC	FR0,FRm,FRn	$FR0 * FRm + FRn \rightarrow FRn$	1111nnnnmmmm1110	—	—	—
FMUL	FRm,FRn	$FRn * FRm \rightarrow FRn$	1111nnnnmmmm0010	—	—	—
FNEG	FRn	$FRn \wedge H'8000\ 0000 \rightarrow FRn$	1111nnnn01001101	—	—	—
FSQRT	FRn	$\sqrt{FRn} \rightarrow FRn$	1111nnnn01101101	—	—	—
FSUB	FRm,FRn	$FRn - FRm \rightarrow FRn$	1111nnnnmmmm0001	—	—	—
FTRC	FRm,FPUL	(long) FRm $\rightarrow$ FPUL	1111mmmm00111101	—	—	—

**Table 3.11 Floating-Point Double-Precision Instructions**

Instruction		Operation	Instruction Code	Privileged	T Bit	New
FABS	DRn	$DRn \wedge H'7FFF\ FFFF\ FFFF\ FFFF \rightarrow DRn$	1111nnnn001011101	—	—	—
FADD	DRm,DRn	$DRn + DRm \rightarrow DRn$	1111nnnn0mmmm00000	—	—	—
FCMP/EQ	DRm,DRn	When $DRn = DRm$ , $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	1111nnnn0mmmm00100	—	Comparison result	—
FCMP/GT	DRm,DRn	When $DRn > DRm$ , $1 \rightarrow T$ Otherwise, $0 \rightarrow T$	1111nnnn0mmmm00101	—	Comparison result	—
FDIV	DRm,DRn	$DRn/DRm \rightarrow DRn$	1111nnnn0mmmm00011	—	—	—
FCNVDS	DRm,FPUL	double_to_float(DRm) $\rightarrow$ FPUL	1111mmmm010111101	—	—	—
FCNVSD	FPUL,DRn	float_to_double(FPUL) $\rightarrow$ DRn	1111nnnn010101101	—	—	—
FLOAT	FPUL,DRn	(float)FPUL $\rightarrow$ DRn	1111nnnn000101101	—	—	—
FMUL	DRm,DRn	$DRn * DRm \rightarrow DRn$	1111nnnn0mmmm00010	—	—	—
FNEG	DRn	$DRn \wedge H'8000\ 0000\ 0000\ 0000 \rightarrow DRn$	1111nnnn001001101	—	—	—
FSQRT	DRn	$\sqrt{DRn} \rightarrow DRn$	1111nnnn001101101	—	—	—
FSUB	DRm,DRn	$DRn - DRm \rightarrow DRn$	1111nnnn0mmmm00001	—	—	—
FTRC	DRm,FPUL	(long) DRm $\rightarrow$ FPUL	1111mmmm000111101	—	—	—

**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	1111nnn1mmmm01100	—	—	—
FMOV XDm,DRn	XDm → DRn	1111nnn0mmmm11100	—	—	—
FMOV XDm,XDn	XDm → XDn	1111nnn1mmmm11100	—	—	—
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)	1111nnnnmmmm11010	—	—	—
FMOV XDm,@-Rn	Rn – 8 → Rn, XDm → (Rn)	1111nnnnmmmm11011	—	—	—
FMOV XDm,@(R0,Rn)	XDm → (R0 + Rn)	1111nnnnmmmm10111	—	—	—
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

The SH-4A 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.

#### 1. General Pipeline

I1	I2	I3	ID	E1	E2	E3	WB
-Instruction fetch		-Predecode	-Instruction decode -Issue -Register read	-Forwarding	-Operation		-Write-back

#### 2. General Load/Store Pipeline

I1	I2	I3	ID	E1	E2	E3	WB
-Instruction fetch		-Predecode	-Instruction decode -Issue -Register read	-Address calculation	-Memory data access		-Write-back

#### 3. Special Pipeline

I1	I2	I3	ID	E1	E2	E3	WB
-Instruction fetch		-Predecode	-Instruction decode -Issue -Register read	-Forwarding	-Operation		-Write-back

#### 4. Special Load/Store Pipeline

I1	I2	I3	ID	E1	E2	E3	WB
-Instruction fetch		-Predecode	-Instruction decode -Issue -Register read				

#### 5. Floating-Point Pipeline

I1	I2	I3	ID	FS1	FS2	FS3	FS4	FS
-Instruction fetch		-Predecode	-Instruction decode -Issue	-Register read -Forwarding	-Operation	-Operation	-Operation	-Operation -Write-back

#### 6. Floating-Point Extended Pipeline

I1	I2	I3	ID	FE1	FE2	FE3	FE4	FE5	FE6	FS
-Instruction fetch		-Predecode	-Instruction decode -Issue	-Register read -Forwarding	-Operation	-Operation	-Operation	-Operation	-Operation	-Operation -Write-back

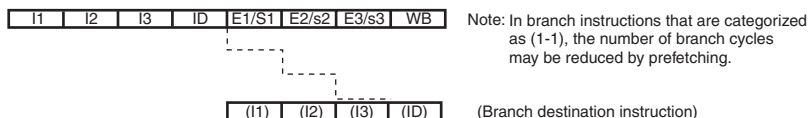
**Figure 4.1 Basic Pipelines**

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

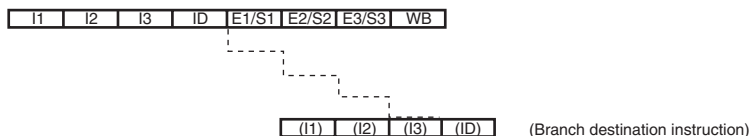
**Table 4.1 Representations of Instruction Execution Patterns**

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

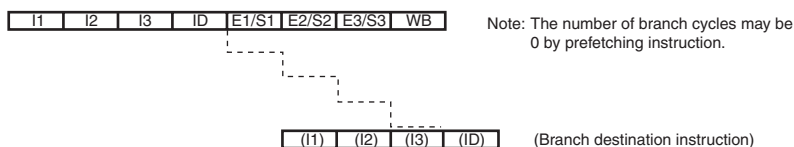
(1-1) BF, BF/S, BT, BT/S, BRA, BSR: 1 issue cycle + 0 to 3 branch cycles



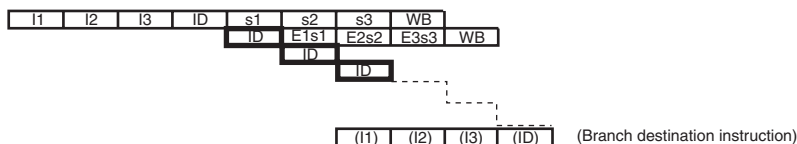
(1-2) JSR, JMP, BRAF, BSRF: 1 issue cycle + 4 branch cycles



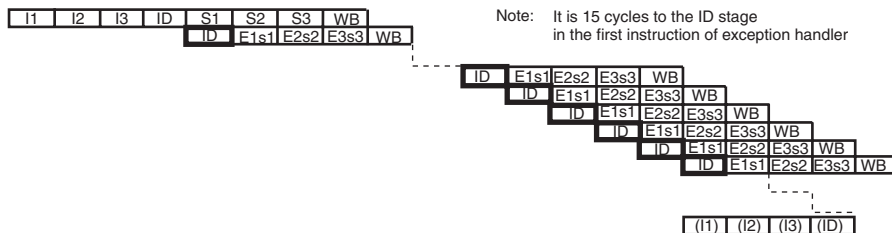
(1-3) RTS: 1 issue cycle + 0 to 4 branch cycles



(1-4) RTE: 4 issue cycles + 2 branch cycles



(1-5) TRAPA: 8 issue cycles + 5 cycles + 2 branch cycle



(1-6) SLEEP: 2 issue cycles

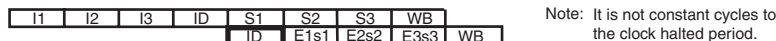


Figure 4.2 Instruction Execution Patterns (1)

(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
----	----	----	----	-------	-------	-------	----

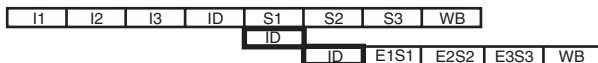
**Figure 4.2 Instruction Execution Patterns (2)**

(3-1) Load/store: 1 issue cycle

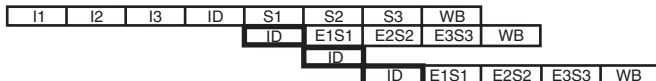
MOV.[BWL], MOV.[BWL] @(d,GBR)



(3-2) AND.B, OR.B, XOR.B, TST.B: 3 issue cycles



(3-3) TAS.B: 4 issue cycles



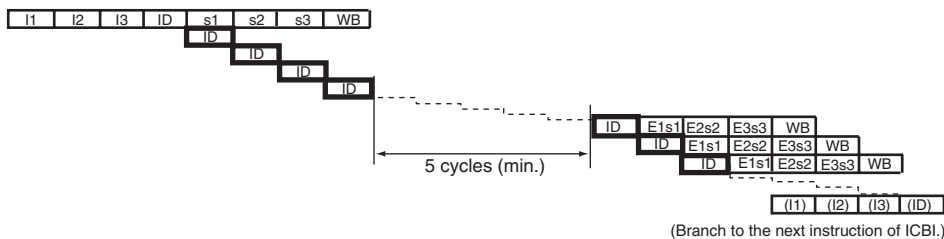
(3-4) PREF, OCBI, OCBP, OCBWB, MOVCA.L, SYNCO: 1 issue cycle



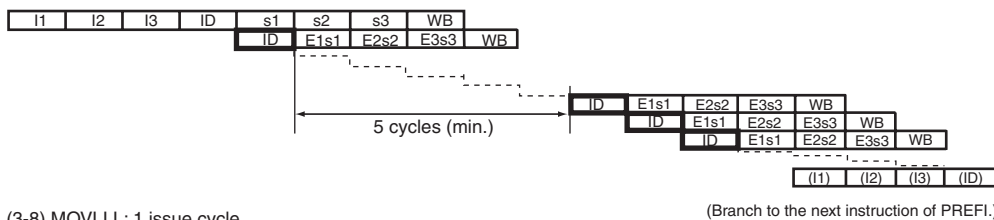
(3-5) LDTLB: 1 issue cycle



(3-6) ICBI: 8 issue cycles + 5 cycles + 4 branch cycle



(3-7) PREFI: 5 issue cycles + 5 cycles + 4 branch cycle



(3-8) MOVL.I.L: 1 issue cycle



(3-9) MOVCO.L: 1 issue cycle



(3-10) MOVUA.L: 2 issue cycles

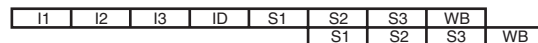


Figure 4.2 Instruction Execution Patterns (3)

(4-1) LDC to Rp\_BANK/SSR/SPC/VBR: 1 issue cycle



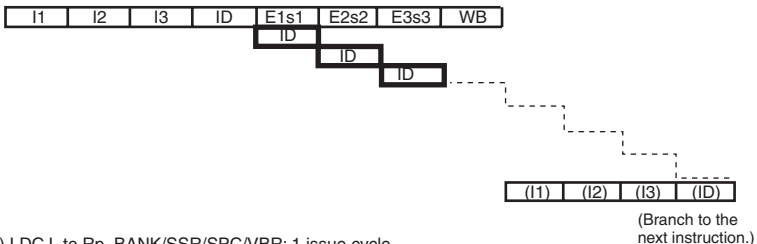
(4-2) LDC to DBR/SGR: 4 issue cycles



(4-3) LDC to GBR: 1 issue cycle



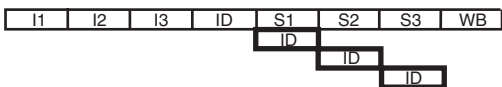
(4-4) LDC to SR: 4 issue cycles + 4 branch cycles



(4-5) LDC.L to Rp\_BANK/SSR/SPC/VBR: 1 issue cycle



(4-6) LDC.L to DBR/SGR: 4 issue cycles



(4-7) LDC.L to GBR: 1 issue cycle



(4-8) LDC.L to SR: 6 issue cycles + 4 branch cycles



**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)**

(5-1) LDS to MACH/L: 1 issue cycle

I1	I2	I3	ID	s1	s2	s3	WB
							MS

(5-2) LDS.L to MACH/L: 1 issue cycle

I1	I2	I3	ID	S1	S2	S3	WB
							MS

(5-3) STS from MACH/L: 1 issue cycle

I1	I2	I3	ID	s1	s2	s3	WB
							MS

(5-4) STS.L from MACH/L: 1 issue cycle

I1	I2	I3	ID	S1	S2	S3	WB
							MS

(5-5) MULS.W, MULU.W: 1 issue cycle

I1	I2	I3	ID	E1	M2	M3	MS
----	----	----	----	----	----	----	----

(5-6) DMULS.L, DMULU.L, MUL.L: 1 issue cycle

I1	I2	I3	ID	E1	M2	M3	
					M2	M3	MS

(5-7) CLRMAC: 1 issue cycle

I1	I2	I3	ID	E1	M2	M3	MS
----	----	----	----	----	----	----	----

(5-8) MAC.W: 2 issue cycle

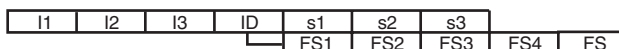
I1	I2	I3	ID	S1	S2	S3	WB			
				ID	S1	S2	S3	WB		
								M2	M3	MS

(5-9) MAC.L: 2 issue cycle

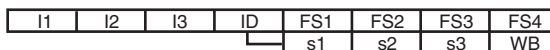
I1	I2	I3	ID	S1	S2	S3	WB			
				ID	S1	S2	S3	WB		
								M2	M3	
									M2	M3
										MS

**Figure 4.2 Instruction Execution Patterns (6)**

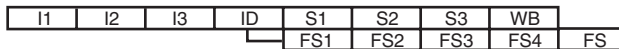
(6-1) LDS to FPUL: 1 issue cycle



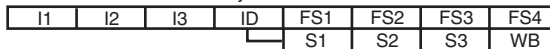
(6-2) STS from FPUL: 1 issue cycle



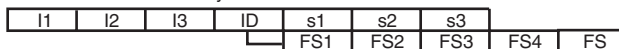
(6-3) LDS.L to FPUL: 1 issue cycle



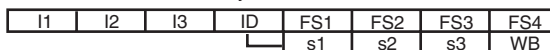
(6-4) STS.L from FPUL: 1 issue cycle



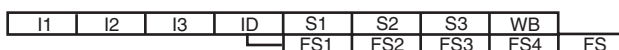
(6-5) LDS to FPSCR: 1 issue cycle



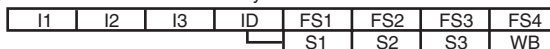
(6-6) STS from FPSCR: 1 issue cycle



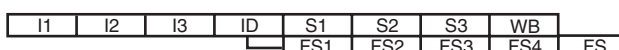
(6-7) LDS.L to FPSCR: 1 issue cycle



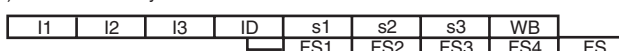
(6-8) STS.L from FPSCR: 1 issue cycle



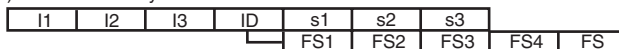
(6-9) FPU load/store instruction FMOV: 1 issue cycle



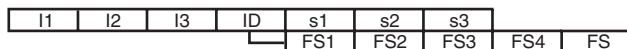
(6-10) FLDS: 1 issue cycle



(6-11) FSTS: 1 issue cycle

**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

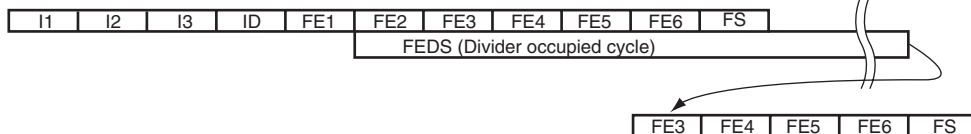


(6-14) Single-precision floating-point computation: 1 issue cycle

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

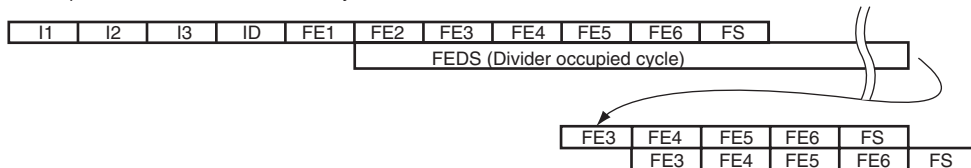


(6-17) Double-precision floating-point computation: 1 issue cycle

FMUL



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



**Figure 4.2 Instruction Execution Patterns (8)**

(6-19) FIPR: 1 issue cycle

I1	I2	I3	ID	FE1	FE2	FE3	FE4	FE5	FE6	FS
----	----	----	----	-----	-----	-----	-----	-----	-----	----

(6-20) FTRV: 1 issue cycle

I1	I2	I3	ID	FE1	FE2	FE3	FE4	FE5	FE6	FS									
					FE1	FE2	FE3	FE4	FE5	FE6	FS								
						FE1	FE2	FE3	FE4	FE5	FE6	FS							
							FE1	FE2	FE3	FE4	FE5	FE6	FS						

(6-21) FSRRA: 1 issue cycle

I1	I2	I3	ID	FE1	FE2	FE3	FE4	FE5	FE6	FS
					FEPL					

Function computing unit occupied cycle

(6-22) FSCA: 1 issue cycle

I1	I2	I3	ID	FE1	FE2	FE3	FE4	FE5	FE6	FS		
					FE1	FE2	FE3	FE4	FE5	FE6	FS	
						FE1	FE2	FE3	FE4	FE5	FE6	FS
					FEPL							

Function computing unit occupied cycle

**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)					CO
		EX	MT	BR	LS	FE	
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



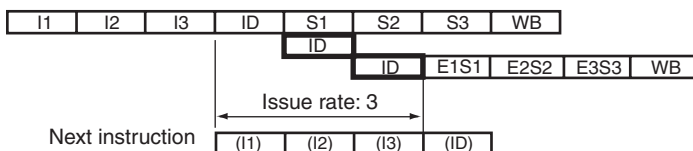
### 4.3 Issue Rates and Execution Cycles

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.

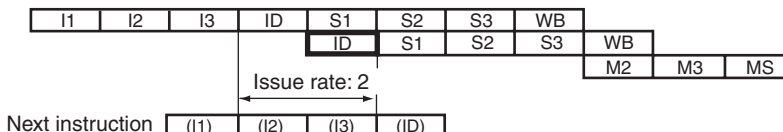
#### 1. Issue Rate

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

E.g. AND.B instruction



E.g. MAC.W instruction

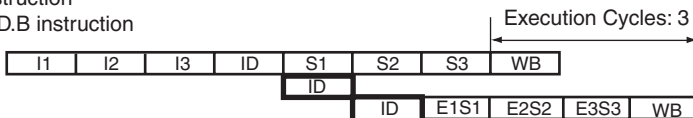


#### 2. Execution Cycles

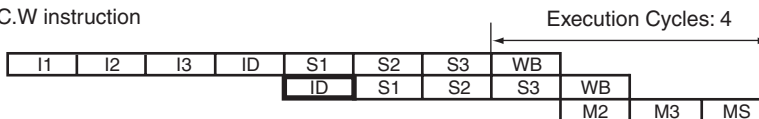
Execution cycles indicates the cycle counts an instruction occupied the pipeline based on the next rules.

CPU instruction

E.g. AND.B instruction

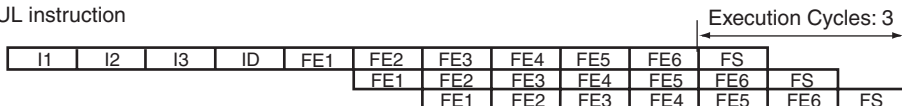


E.g. MAC.W instruction

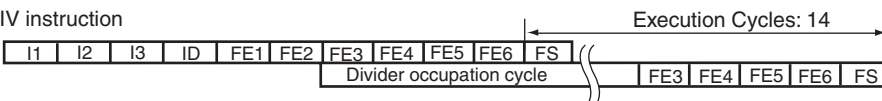


FPU instruction

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	MOV.T 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	6-13
	181	FLDI1	FRn	LS	1	6-13
	182	FMOV	FRm,FRn	LS	1	6-9
	183	FMOV.S	@Rm,FRn	LS	1	6-9
	184	FMOV.S	@Rm+,FRn	LS	1	6-9
	185	FMOV.S	@(R0,Rm),FRn	LS	1	6-9
	186	FMOV.S	FRm,@Rn	LS	1	6-9
	187	FMOV.S	FRm,@-Rn	LS	1	6-9
	188	FMOV.S	FRm,@(R0,Rn)	LS	1	6-9
	189	FLDS	FRm,FPUL	LS	1	6-10
	190	FSTS	FPUL,FRn	LS	1	6-11
	191	FABS	FRn	LS	1	6-12
	192	FADD	FRm,FRn	FE	1	6-14
	193	FCMP/EQ	FRm,FRn	FE	1	6-14
	194	FCMP/GT	FRm,FRn	FE	1	6-14
	195	FDIV	FRm,FRn	FE	14	6-15
	196	FLOAT	FPUL,FRn	FE	1	6-14
	197	FMAC	FR0,FRm,FRn	FE	1	6-14
	198	FMUL	FRm,FRn	FE	1	6-14
	199	FNEG	FRn	LS	1	6-12
	200	FSQRT	FRn	FE	30	6-15
	201	FSUB	FRm,FRn	FE	1	6-14
	202	FTRC	FRm,FPUL	FE	1	6-14
	203	FMOV	DRm,DRn	LS	1	6-9
	204	FMOV	@Rm,DRn	LS	1	6-9
	205	FMOV	@Rm+,DRn	LS	1	6-9
	206	FMOV	@(R0,Rm),DRn	LS	1	6-9
	207	FMOV	DRm,@Rn	LS	1	6-9
	208	FMOV	DRm,@-Rn	LS	1	6-9
	209	FMOV	DRm,@(R0,Rn)	LS	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	14	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 the SH-4A is of three kinds: resets, general exceptions, and interrupts.

### 5.2 Register Descriptions

Table 5.1 lists the configuration of registers related exception handling.

**Table 5.1 Register Configuration**

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

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

**Table 5.2 States of Register in Each Operating Mode**

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

### 5.2.1 TRAPA Exception Register (TRA)

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

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

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

## 5.2.2 Exception Event Register (EXPEVT)

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

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

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

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

### 5.2.3 Interrupt Event Register (INTEVT)

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

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

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

## 5.2.4 Non-Support Detection Exception Register (EXPMASK)

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

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

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

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

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

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved For details on reading/writing these bits, see General Precautions on Handling of Product.
4	MMCAW	0	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 For details on reading/writing these bits, see General Precautions on Handling of Product.
1	BRDSSLP	0	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	0	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* <sup>2</sup>	Priority Order* <sup>2</sup>	Exception Transition Direction* <sup>3</sup>		Exception Code* <sup>4</sup>
					Vector Address	Offset	
Reset	Abort type	Power-on reset	1	1	H'A000 0000	—	H'000
		H-UDI reset	1	1	H'A000 0000	—	H'000
		Instruction TLB multiple-hit exception	1	2	H'A000 0000	—	H'140
		Data TLB multiple-hit exception	1	3	H'A000 0000	—	H'140
General exception	Re-execution type	User break before instruction execution*	2	0	(VBR/DBR)	H'100/—	H'1E0
		Instruction address error	2	1	(VBR)	H'100	H'0E0
		Instruction TLB miss exception	2	2	(VBR)	H'400	H'040
		Instruction TLB protection violation exception	2	3	(VBR)	H'100	H'0A0
		General illegal instruction exception	2	4	(VBR)	H'100	H'180
		Slot illegal instruction exception	2	4	(VBR)	H'100	H'1A0
		General FPU disable exception	2	4	(VBR)	H'100	H'800
		Slot FPU disable exception	2	4	(VBR)	H'100	H'820
		Data address error (read)	2	5	(VBR)	H'100	H'0E0
		Data address error (write)	2	5	(VBR)	H'100	H'100
		Data TLB miss exception (read)	2	6	(VBR)	H'400	H'040
		Data TLB miss exception (write)	2	6	(VBR)	H'400	H'060
		Data TLB protection violation exception (read)	2	7	(VBR)	H'100	H'0A0
		Data TLB protection violation exception (write)	2	7	(VBR)	H'100	H'0C0
		FPU exception	2	8	(VBR)	H'100	H'120
		Initial page write exception	2	9	(VBR)	H'100	H'080

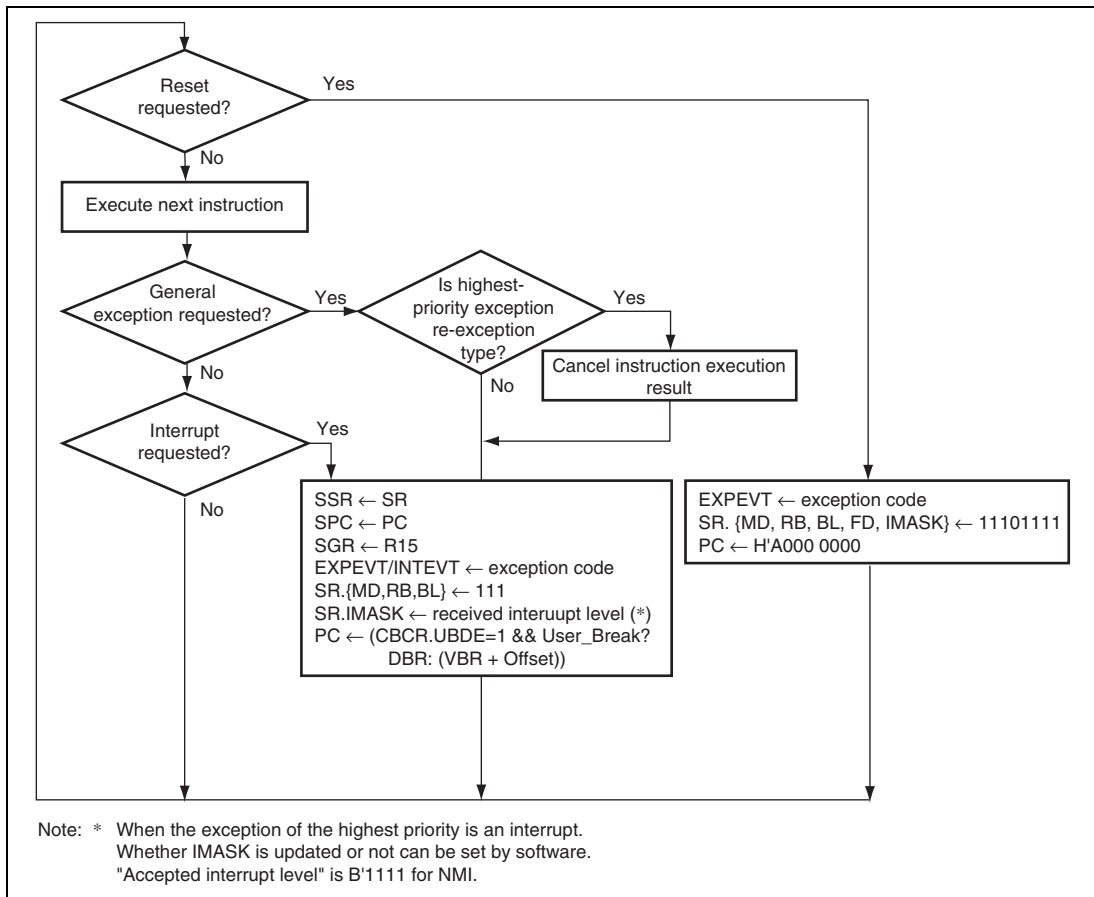
Exception Category	Execution Mode	Exception	Priority Level* <sup>2</sup>	Priority Order* <sup>2</sup>	Exception Transition Direction* <sup>3</sup>		Exception Code* <sup>4</sup>
					Vector Address	Offset	
General exception	Completion type	Unconditional trap (TRAPA)	2	4	(VBR)	H'100	H'160
		User break 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	—

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

## 5.5 Exception Flow

### 5.5.1 Exception Flow

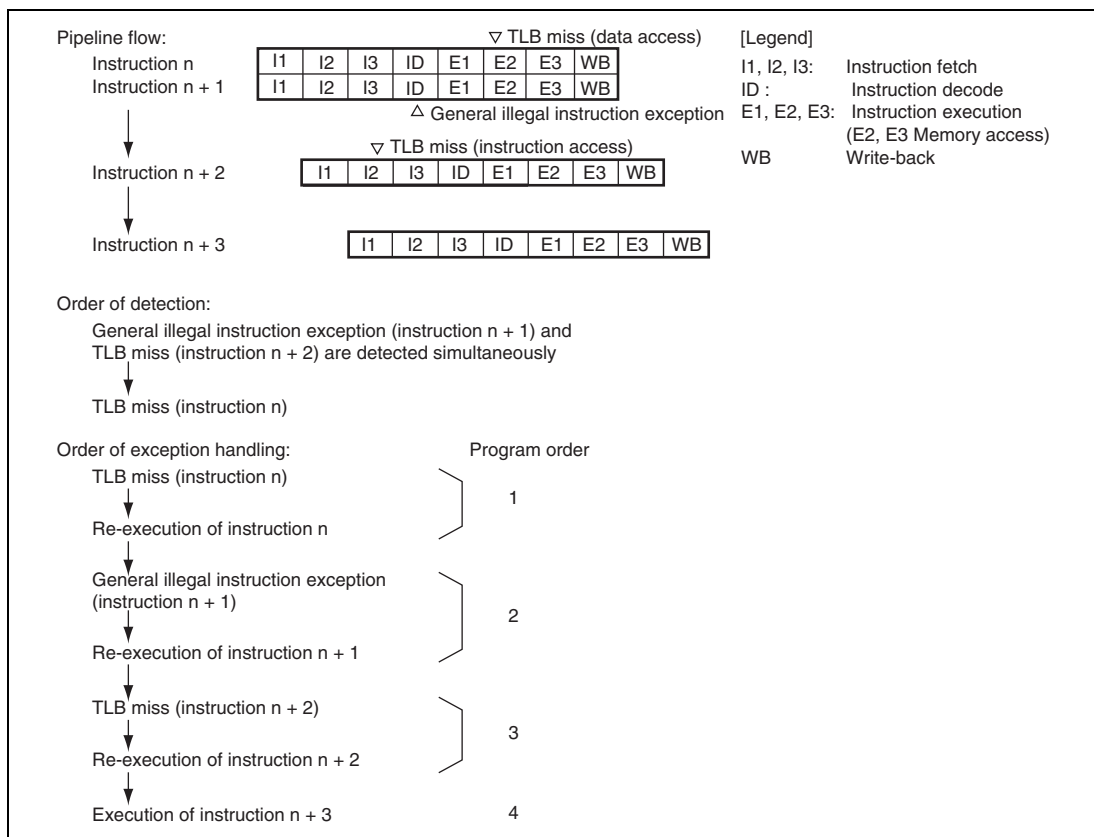
Figure 5.1 shows an outline flowchart of the basic operations in instruction execution and exception handling. For the sake of clarity, the following description assumes that instructions are executed sequentially, one by one. Figure 5.1 shows the relative priority order of the different kinds of exceptions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



**Figure 5.1 Instruction Execution and Exception Handling**

## 5.5.2 Exception Source Acceptance

A priority ranking is provided for 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 power-on reset, and the CPU branches to the same address as in a reset (H'A0000000). For the operation in the event of a user break, see section 30, User Break Controller (UBC). If an ordinary interrupt occurs, the interrupt request is held pending and is accepted after the BL bit has been cleared to 0 by software. If a nonmaskable interrupt (NMI) occurs, it can be held pending or accepted according to the setting made by software.

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

### 5.5.4 Return from Exception Handling

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

## 5.6 Description of Exceptions

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

### 5.6.1 Resets

#### (1) Power-On Reset

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

#### (2) H-UDI Reset

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



### (3) Instruction TLB Multiple Hit Exception

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

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

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

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

### (4) Data TLB Multiple-Hit Exception

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

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

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

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

## 5.6.2 General Exceptions

### (1) Data TLB Miss Exception

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

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

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

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

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

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

## (2) Instruction TLB Miss Exception

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

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

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

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

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

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

### (3) Initial Page Write Exception

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

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

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

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

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

#### (4) Data TLB Protection Violation Exception

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

**Table 5.4 UTLB Protection Information (TLB Compatible Mode)**

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

**Table 5.5 UTLB Protection Information (TLB Extended Mode)**

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

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

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

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

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

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

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

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

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

### (5) Instruction TLB Protection Violation Exception

- Source: The access does not accord with the ITLB protection information (PR bits or EPR bits) shown in table 5.6 and table 5.7.

**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, 01	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'FFFFFFF in user mode  
Area H'E5000000 to H'E5FFFFFFF can be accessed in user mode. For details, see section 9, On-Chip Memory.
- Transition address:  $VBR + H'00000100$
- Transition operations:

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

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

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

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

## (8) Unconditional Trap

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

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

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

## (9) General Illegal Instruction Exception

- Sources:

- Decoding of an undefined instruction not in a delay slot

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

Undefined instruction: H'FFFD

- Decoding in user mode of a privileged instruction not in a delay slot

Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC instructions that access GBR

- Transition address: VBR + H'00000100

- Transition operations:

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

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

```
General_illegal_instruction_exception()
```

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

## (10) Slot Illegal Instruction Exception

- Sources:
  - Decoding of an undefined instruction in a delay slot  
 Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S  
 Undefined instruction: H'FFFD
  - Decoding of an instruction that modifies PC in a delay slot  
 Instructions that modify PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm,SR, LDC.L @Rm+,SR, ICBI, PREFI
  - Decoding in user mode of a privileged instruction in a delay slot  
 Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC instructions that access GBR
  - Decoding of a PC-relative MOV instruction or MOVA instruction in a delay slot
  - The BRDSSLP bit in EXPMASK is 0, and the SLEEP instruction in the delay slot is executed.
  - The RTEDS bit in EXPMASK is 0, and an instruction other than the NOP instruction in the delay slot is executed.
- Transition address: VBR + H'000 0100
- Transition operations:
 

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

Exception code H'1A0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined code other than H'FFFD is decoded.

```
Slot_illegal_instruction_exception()
```

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

### (11) General FPU Disable Exception

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

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

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

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

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

## (12) Slot FPU Disable Exception

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

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

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

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

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

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

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

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

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

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

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



## (14) FPU Exception

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

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

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

### 5.6.3 Interrupts

#### (1) NMI (Nonmaskable Interrupt)

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

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

Exception code H'1C0 is set in INTEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0600. When the BL bit in SR is 0, this interrupt is not masked by the interrupt mask bits in SR, and is accepted at the highest priority level. When the BL bit in SR is 1, a software setting can specify whether this interrupt is to be masked or accepted. When the INTMU bit in CPUOPM is 1 and the NMI interrupt is accessed, B'1111 is set to IMASK bit in SR. For details, see section 13, 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 13, 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, BSRRF, 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 power-on reset is executed. The value in EXPEVT at this time is H'00000020; the SSR contents are undefined.

#### B. Interrupt

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

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

### (3) SPC when an Exception Occurs

#### A. Re-execution type general exception

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

#### B. Completion type general exception or interrupt

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

### (4) RTE Instruction Delay Slot

- A. The instruction in the delay slot of the RTE instruction is executed only after the value saved in SSR has been restored to SR. The acceptance of the exception related to the instruction access is determined depending on SR before restoring, while the acceptance of

other exceptions is determined depending on the processing mode by SR after restoring or the BL bit. The completion type exception is accepted before branching to the destination of RTE instruction. However, if the re-execution type exception is occurred, the operation cannot be guaranteed.

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

#### **(5) Changing the SR Register Value and Accepting Exception**

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

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

## Section 6 Floating-Point Unit (FPU)

### 6.1 Features

The FPU has the following features.

- Conforms to IEEE754 standard
- 32 single-precision floating-point registers (can also be referenced as 16 double-precision registers)
- Two rounding modes: Round to Nearest and Round to Zero
- Two denormalization modes: Flush to Zero and Treat Denormalized Number
- Six exception sources: FPU Error, Invalid Operation, Divide By Zero, Overflow, Underflow, and Inexact
- Comprehensive instructions: Single-precision, double-precision, graphics support, and system control
- In 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).

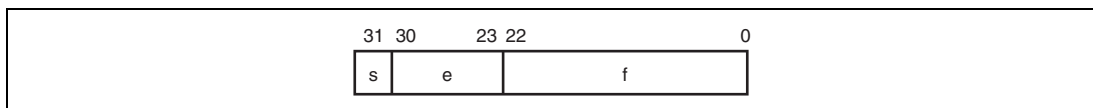
## 6.2 Data Formats

### 6.2.1 Floating-Point Format

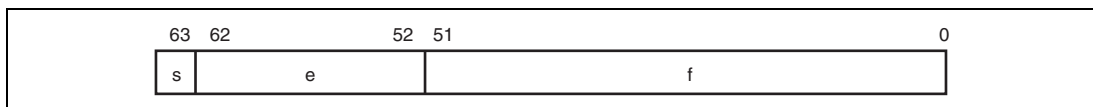
A floating-point number consists of the following three fields:

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

The SH-4A 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:

$$e = E + \text{bias}$$

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  $f \neq 0$ ,  $v = (-1)^s 2^{E_{\min}} (0.f)$  [denormalized number]

If  $E = E_{\min} - 1$  and  $f = 0$ ,  $v = (-1)^s 0$  [positive or negative zero]

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

**Table 6.2 Floating-Point Ranges**

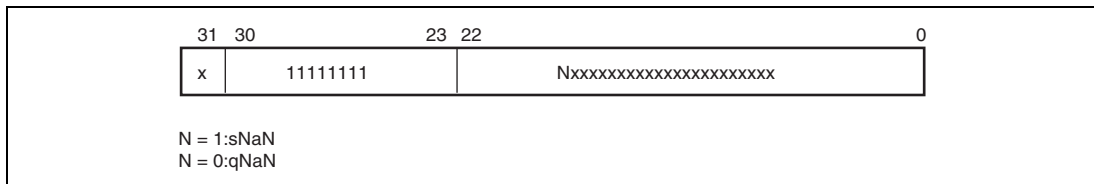
Type	Single-Precision	Double-Precision
Signaling non-number	H'7FFF FFFF to H'7FC0 0000	H'7FFF FFFF FFFF FFFF to H'7FF8 0000 0000 0000
Quiet non-number	H'7FBF FFFF to H'7F80 0001	H'7FF7 FFFF FFFF FFFF to H'7FF0 0000 0000 0001
Positive infinity	H'7F80 0000	H'7FF0 0000 0000 0000
Positive normalized number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

## 6.2.2 Non-Numbers (NaN)

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

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

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



**Figure 6.3 Single-Precision NaN Bit Pattern**

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

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

Following three instructions are used as transfer instructions between registers.

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

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

The qNaN values as operation results are as follows:

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

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

### **6.2.3 Denormalized Numbers**

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

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

When the DN bit in FPSCR is 0, a denormalized number (source operand or operation result) is processed as it is. See section 11, Instruction Descriptions of 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, XD<sub>i</sub> (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.

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

FPSCR.FR = 0				FPSCR.FR = 1			
FV0	DR0	FR0	FPR0 BANK0	XF0	XD0	XMTRX	
		FR1	FPR1 BANK0	XF1			
	DR2	FR2	FPR2 BANK0	XF2	XD2		
		FR3	FPR3 BANK0	XF3			
FV4	DR4	FR4	FPR4 BANK0	XF4	XD4		
		FR5	FPR5 BANK0	XF5			
	DR6	FR6	FPR6 BANK0	XF6	XD6		
		FR7	FPR7 BANK0	XF7			
FV8	DR8	FR8	FPR8 BANK0	XF8	XD8		
		FR9	FPR9 BANK0	XF9			
	DR10	FR10	FPR10 BANK0	XF10	XD10		
		FR11	FPR11 BANK0	XF11			
FV12	DR12	FR12	FPR12 BANK0	XF12	XD12		
		FR13	FPR13 BANK0	XF13			
	DR14	FR14	FPR14 BANK0	XF14	XD14		
		FR15	FPR15 BANK0	XF15			
XMTRX	XD0	XF0	FPR0 BANK1	FR0	DR0	FV0	
		XF1	FPR1 BANK1	FR1			
	XD2	XF2	FPR2 BANK1	FR2	DR2		
		XF3	FPR3 BANK1	FR3			
	XD4	XF4	FPR4 BANK1	FR4	DR4	FV4	
		XF5	FPR5 BANK1	FR5			
	XD6	XF6	FPR6 BANK1	FR6	DR6		
		XF7	FPR7 BANK1	FR7			
	XD8	XF8	FPR8 BANK1	FR8	DR8	FV8	
		XF9	FPR9 BANK1	FR9			
	XD10	XF10	FPR10 BANK1	FR10	DR10		
		XF11	FPR11 BANK1	FR11			
	XD12	XF12	FPR12 BANK1	FR12	DR12	FV12	
		XF13	FPR13 BANK1	FR13			
	XD14	XF14	FPR14 BANK1	FR14	DR14		
		XF15	FPR15 BANK1	FR15			

**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

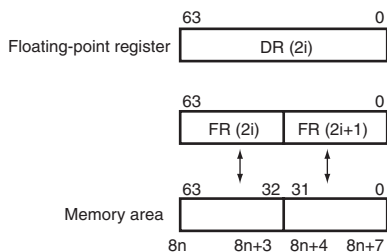
bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause				Enable (EN)						Flag				RM	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

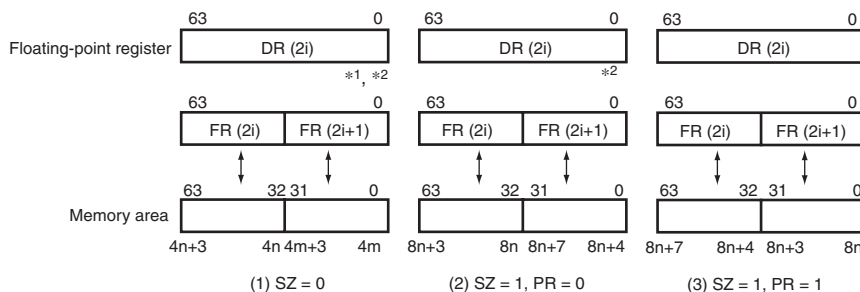
Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	All 0	R/W	FPU Exception Cause Field
11 to 7	Enable	All 0	R/W	FPU Exception Enable Field
6 to 2	Flag	All 0	R/W	FPU Exception Flag Field Each time an FPU operation instruction is executed, the FPU exception cause field is cleared to 0. When an FPU 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



&lt;Big endian&gt;



&lt;Little endian&gt;

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

### (2) Round to Zero

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

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

## 6.5 Floating-Point Exceptions

### 6.5.1 General FPU Disable Exceptions and Slot FPU Disable Exceptions

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

### 6.5.2 FPU Exception Sources

The exception sources are as follows:

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

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

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

### 6.5.3 FPU Exception Handling

FPU exception handling is initiated in the following cases:

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

See section 11, Instruction Descriptions of SH-4A Extended Functions Software Manual for details of FPU exception case.

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

The SH-4A 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, the SH-4A 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:

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

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 FV<sub>m</sub>, FV<sub>n</sub> (m, n: 0, 4, 8, 12)

This instruction is basically used for the following purposes:

- Inner product (m ≠ 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  $(4 \times 4) \cdot$  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  $4 \times 4$  matrix, the SH-4A supports 4-dimensional operations.

- Matrix  $(4 \times 4) \times$  matrix  $(4 \times 4)$ :

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, the SH-4A also supports high-speed data transfer instructions.

When the SZ bit is 1, the SH-4A 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 ( $2 \times 32$ -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)

The SH-4A supports an 8-bit address space identifier, a 32-bit virtual address space, and a 29-bit physical address space. Address translation from virtual addresses to physical addresses is enabled by the memory management unit (MMU) in the SH-4A. The MMU performs high-speed address translation by caching user-created address translation table information in an address translation buffer (translation lookaside buffer: TLB).

The SH-4A 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.

In view of flag functions of the MMU, TLB compatible mode (four paging sizes with four protection bits) and TLB extended mode (eight paging sizes with six protection bits) are provided.

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

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

## 7.1 Overview of MMU

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

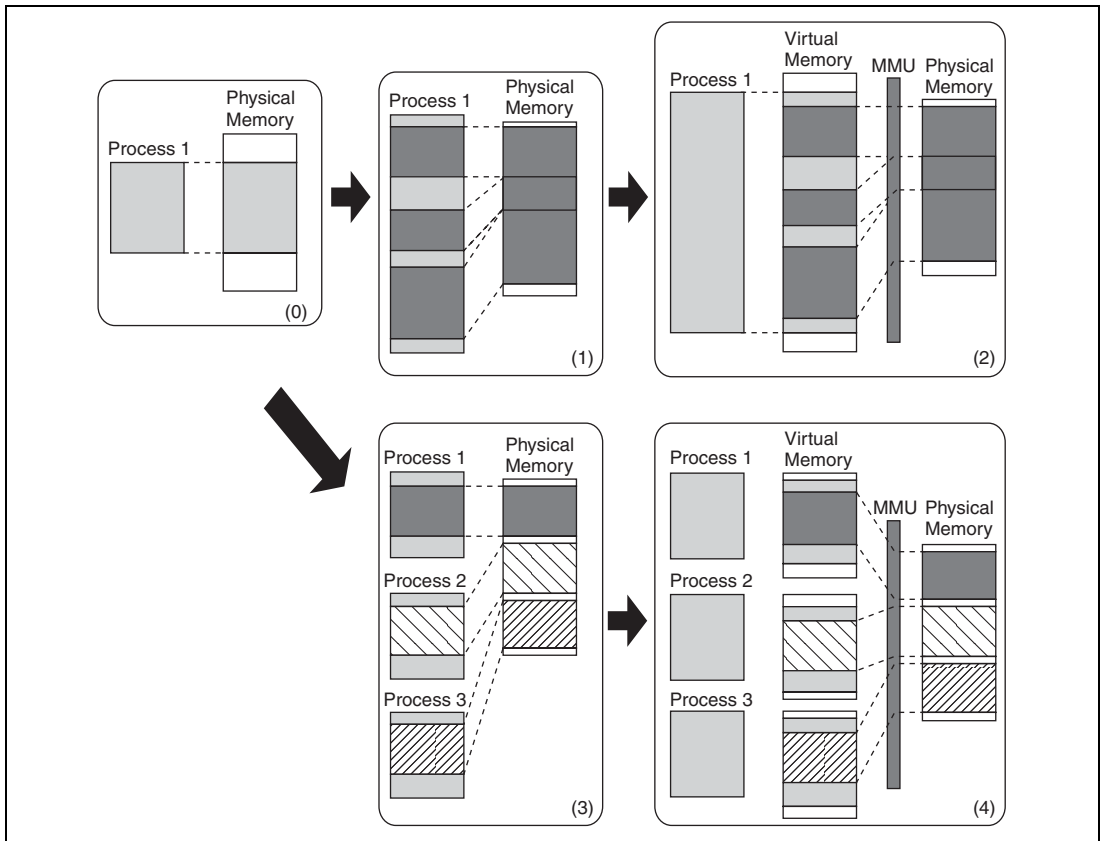
The virtual memory system that came into being in this way works to best effect in a time sharing system (TSS) that allows a number of processes to run simultaneously ((3) in figure 7.1). Running a number of processes in a TSS did not increase efficiency since each process had to take account of physical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

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

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

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

In the following descriptions, the address space in virtual memory in the SH-4A is referred to as virtual address space, and the address space in physical memory as physical address space.



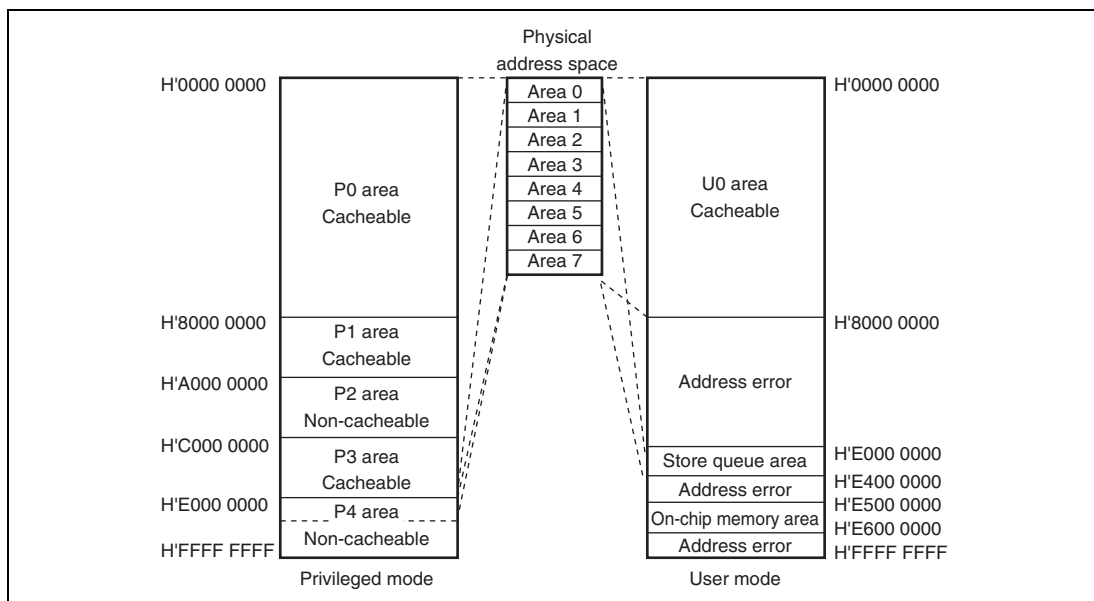
**Figure 7.1 Role of MMU**

### 7.1.1 Address Spaces

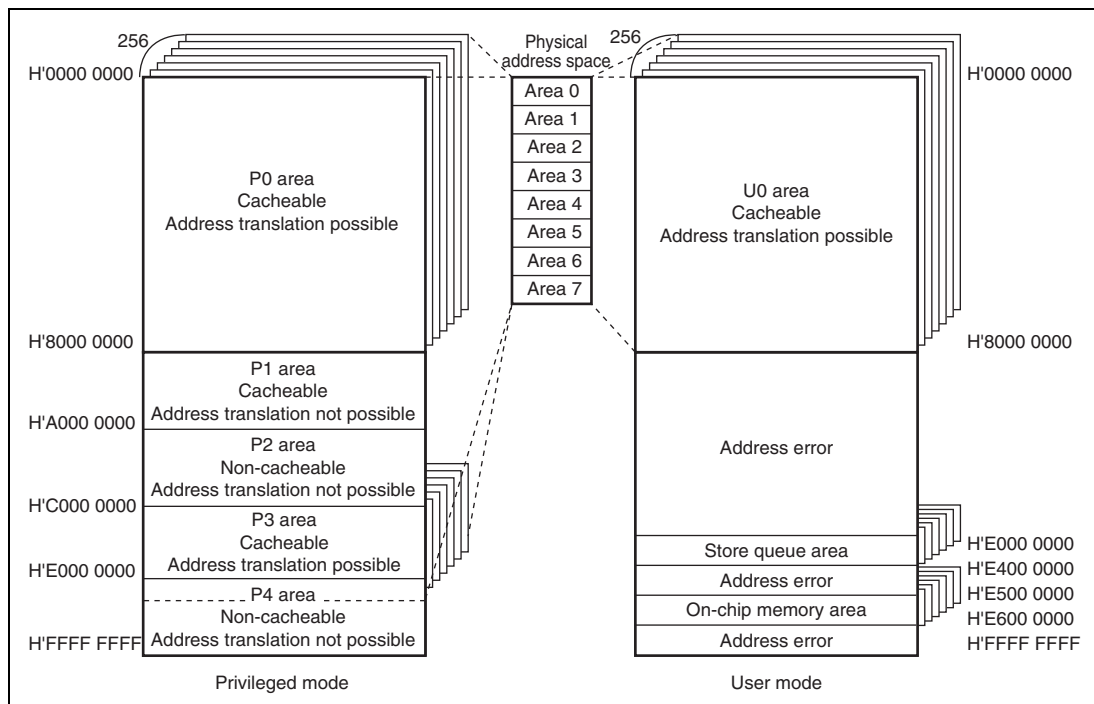
#### (1) Virtual Address Space

The SH-4A 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 the SH-4A. 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.

H'E000 0000	Store queue
H'E400 0000	Reserved area
H'E500 0000	On-chip memory area
H'E600 0000	Reserved area
H'F000 0000	Instruction cache address array
H'F100 0000	Instruction cache data array
H'F200 0000	Instruction TLB address array
H'F300 0000	Instruction TLB data array
H'F400 0000	Operand cache address array
H'F500 0000	Operand cache data array
H'F600 0000	Unified TLB address array
H'F700 0000	Unified TLB data array
H'F800 0000	Reserved area
H'FC00 0000	Control register area
H'FFFF FFFF	

**Figure 7.4 P4 Area**

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

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

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

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

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

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

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

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

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

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

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

## **(2) Physical Address Space**

The SH-4A 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, Memory Controller Unit (MCU).

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.

H'0000 0000	Area 0
H'0400 0000	Area 1
H'0800 0000	Area 2
H'0C00 0000	Area 3
H'1000 0000	Area 4
H'1400 0000	Area 5
H'1800 0000	Area 6
H'1C00 0000	Area 7 (reserved area)
H'1FFF FFFF	

**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 the SH-4A, 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	PTL	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	Sleep	Standby
Page table entry high register	PTEH	Undefined	Retained	Retained
Page table entry low register	PTL	Undefined	Retained	Retained
Translation table base register	TTB	Undefined	Retained	Retained
TLB exception address register	TEA	Undefined	Retained	Retained
MMU control register	MMUCR	H'0000 0000	Retained	Retained
Page table entry assistance register	PTEA	H'0000 xxx0	Retained	Retained
Physical address space control register	PASCR	H'0000 0000	Retained	Retained

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

### 7.2.1 Page Table Entry High Register (PTEH)

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

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

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

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

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

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

## 7.2.2 Page Table Entry Low Register (PTEL)

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	PPN												
Initial value:	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPN						—	V	SZ1	PR1	PR0	SZ0	C	D	SH	WT
Initial value:	—	—	—	—	—	—	0	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

### 7.2.3 Translation Table Base Register (TTB)

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

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

### 7.2.4 TLB Exception Address Register (TEA)

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEA 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 IRMCRR 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	<p>Least Recently Used ITLB</p> <p>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.</p> <p>LRUI is updated by means of the algorithm shown below. x means that updating is not performed.</p> <p>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</p> <p>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 reset, the LRUI bits are initialized to 0, and therefore a prohibited setting is never made by a hardware update.</p> <p>x means "don't care".</p> <p>111xxx: ITLB entry 0 is updated  0xx11x: ITLB entry 1 is updated  x0x0x1: ITLB entry 2 is updated  xx0x00: ITLB entry 3 is updated</p> <p>Other than above: Setting prohibited</p>

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



Bit	Bit Name	Initial Value	R/W	Description
7	ME	0	R/W	<p>TLB Extended Mode Switching</p> <p>0: TLB compatible mode</p> <p>1: TLB extended mode</p> <p>For modifying the ME bit value, always set the TI bit to 1 to invalidate the contents of ITLB and UTLB.</p>
6 to 3	—	All 0	R	<p>Reserved</p> <p>For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.</p>
2	TI	0	R/W	<p>TLB Invalidate Bit</p> <p>Writing 1 to this bit invalidates (clears to 0) all valid UTLB/ITLB bits. This bit is always read as 0.</p>
1	—	0	R	<p>Reserved</p> <p>For details on reading from or writing to this bit, see description in General Precautions on Handling of Product.</p>
0	AT	0	R/W	<p>Address Translation Enable Bit</p> <p>These bits enable or disable the MMU.</p> <p>0: MMU disabled</p> <p>1: MMU enabled</p> <p>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.</p>

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

## 7.2.7 Physical Address Space Control Register (PASCR)

PASCR controls the operation in the physical address space.

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

## 7.2.8 Instruction Re-Fetch Inhibit Control Register (IRMCR)

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

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

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	R2	R1	LT	MT	MC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
3	R1	0	R/W	<p>Re-Fetch Inhibit 1 after Register Change</p> <p>When a register allocated in addresses H'FF200000 to H'FF2FFFFFF is changed, this bit controls whether re-fetch is performed for the next instruction.</p> <p>0: Re-fetch is performed</p> <p>1: Re-fetch is not performed</p>
2	LT	0	R/W	<p>Re-Fetch Inhibit after LDTLB Execution</p> <p>This bit controls whether re-fetch is performed for the next instruction after the LDTLB instruction has been executed.</p> <p>0: Re-fetch is performed</p> <p>1: Re-fetch is not performed</p>
1	MT	0	R/W	<p>Re-Fetch Inhibit after Writing Memory-Mapped TLB</p> <p>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.</p> <p>0: Re-fetch is performed</p> <p>1: Re-fetch is not performed</p>
0	MC	0	R/W	<p>Re-Fetch Inhibit after Writing Memory-Mapped IC</p> <p>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.</p> <p>0: Re-fetch is performed</p> <p>1: Re-fetch is not performed</p>

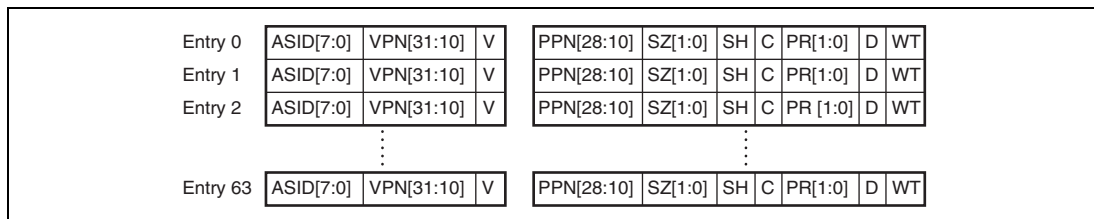
## 7.3 TLB Functions (TLB Compatible Mode; MMUCR.ME = 0)

### 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.



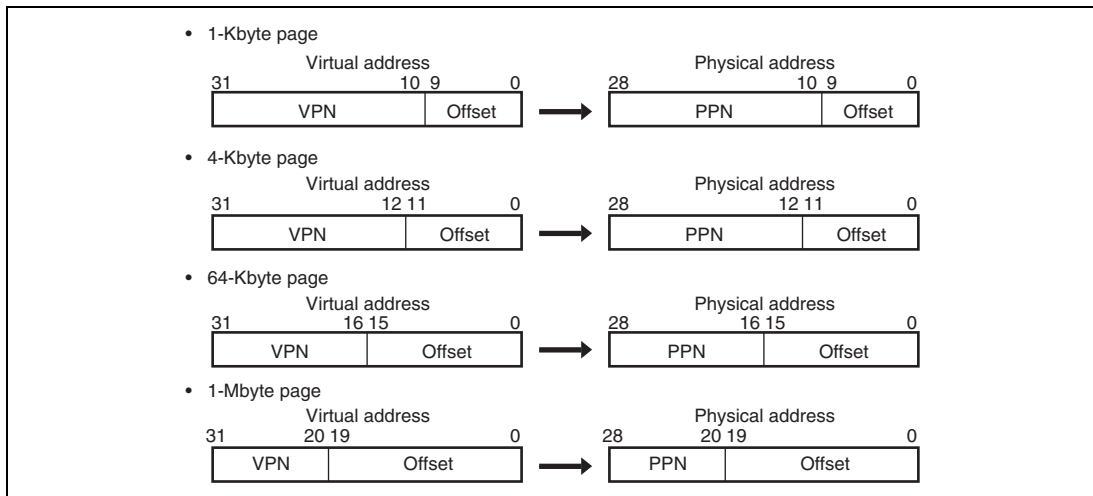
**Figure 7.6 UTLB Configuration (TLB Compatible Mode)**

[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.
- **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.

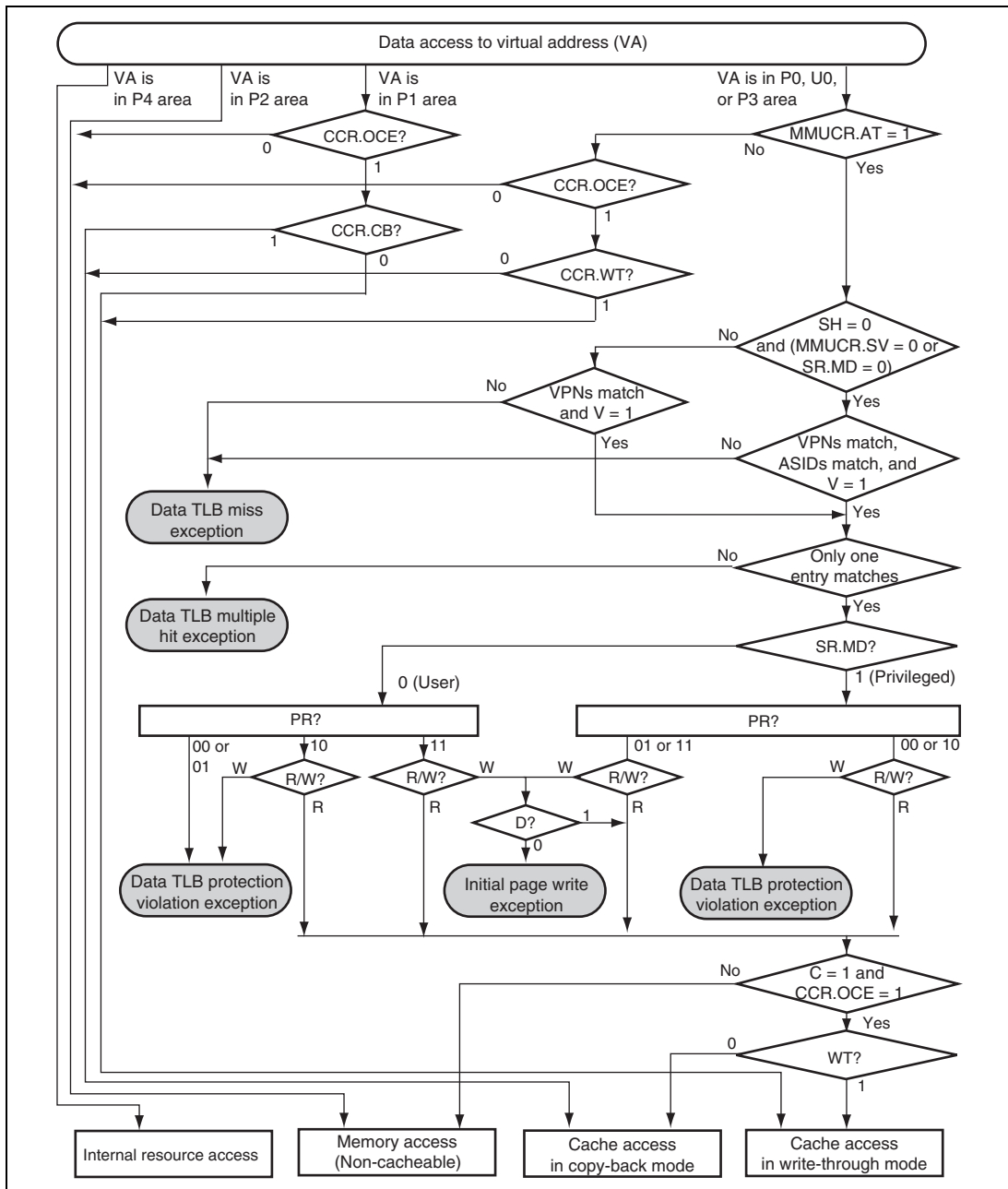
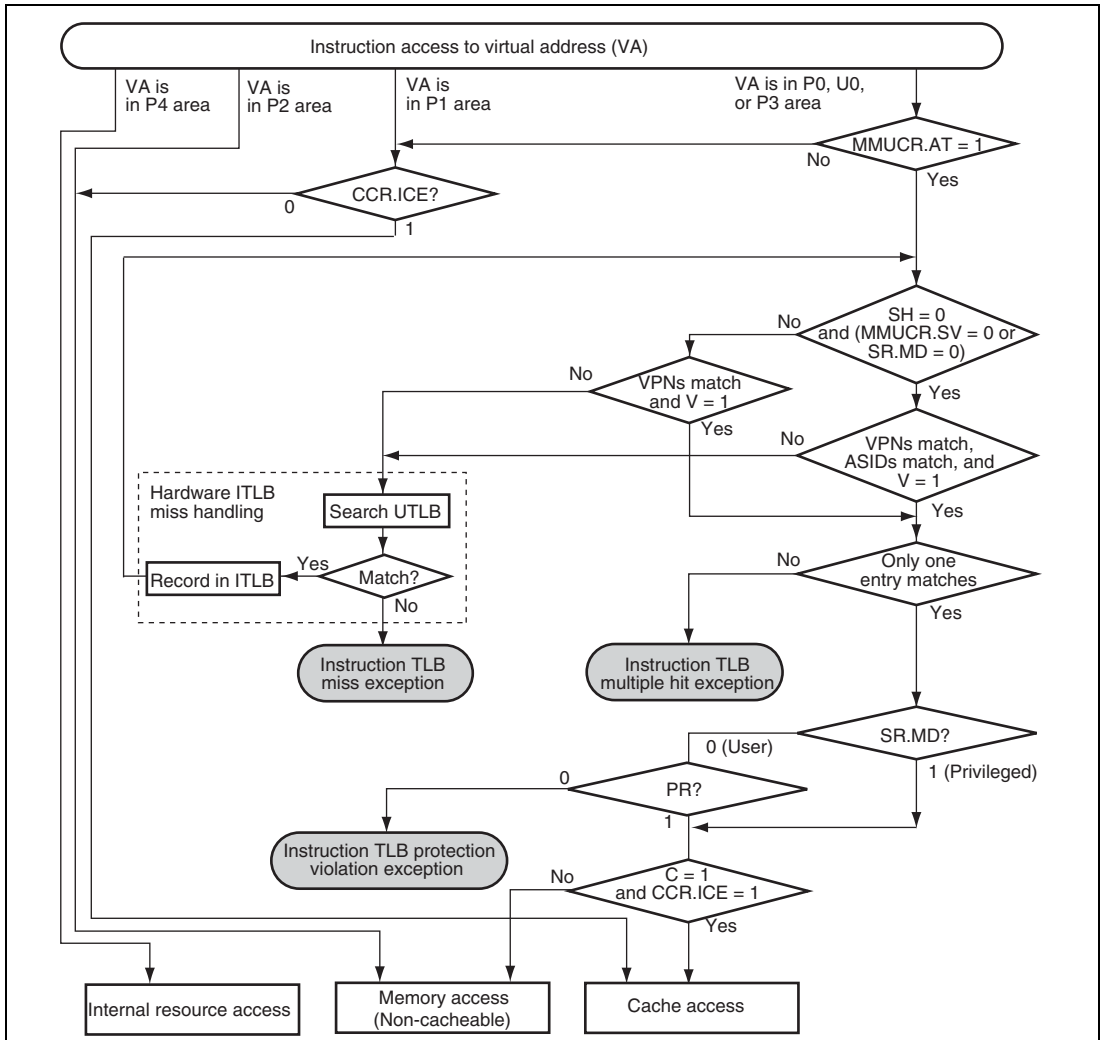


Figure 7.9 Flowchart of Memory Access Using UTLB (TLB Compatible Mode)

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

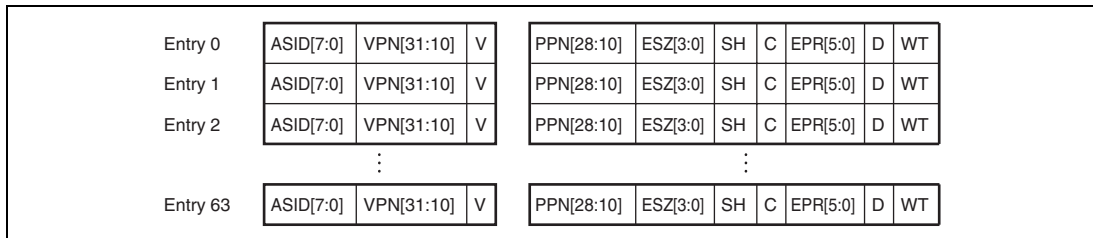


**Figure 7.10 Flowchart of Memory Access Using ITLB (TLB Compatible Mode)**

## 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.

- 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.

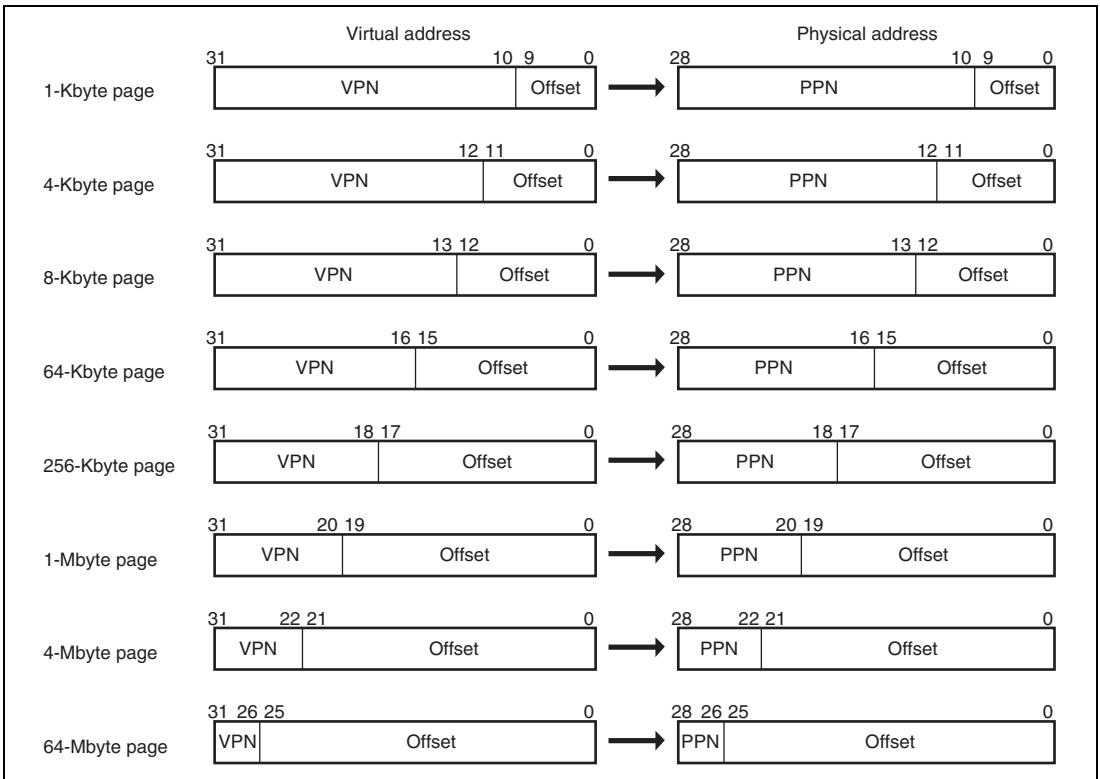
1: Write has been performed.

- 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.



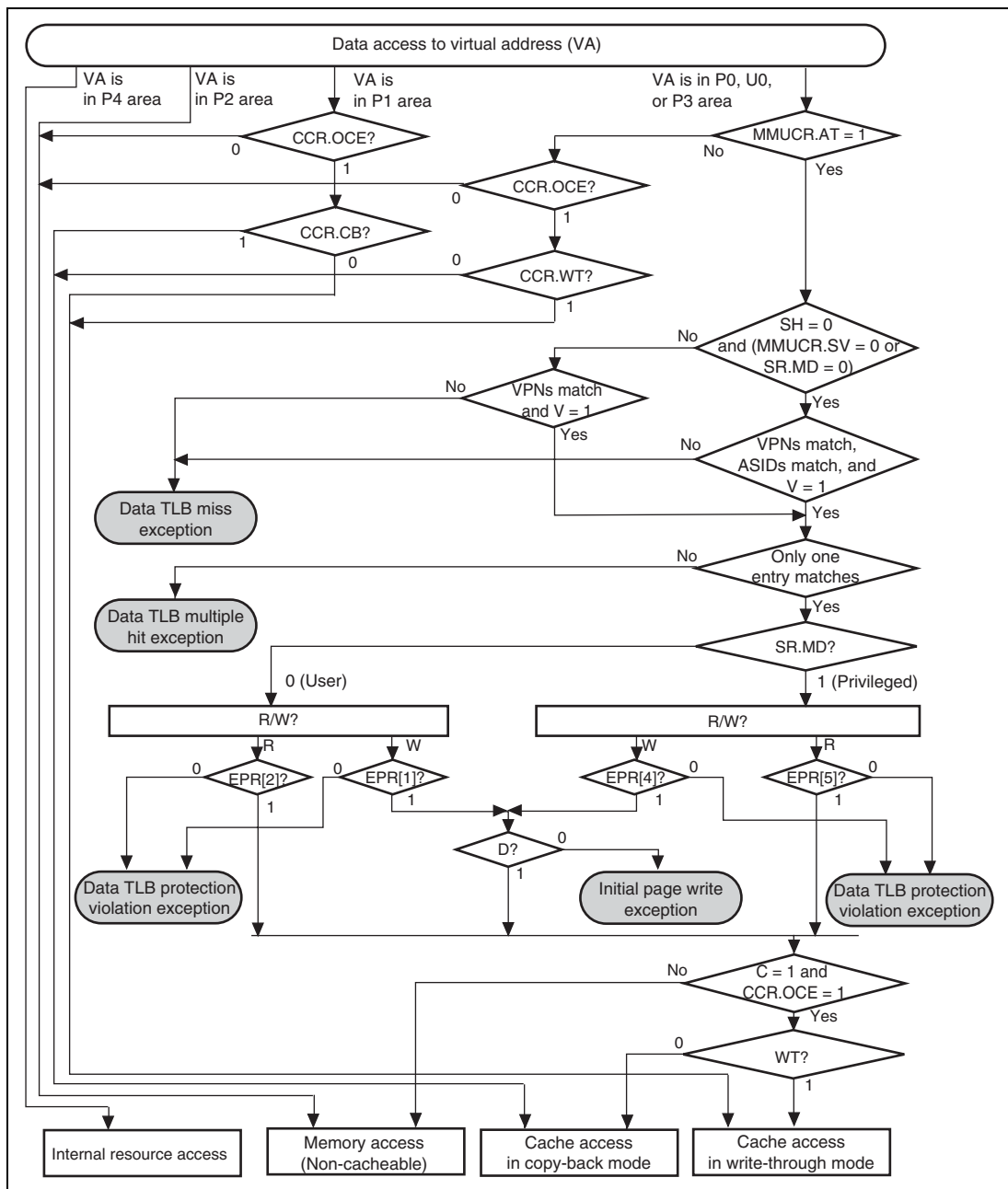
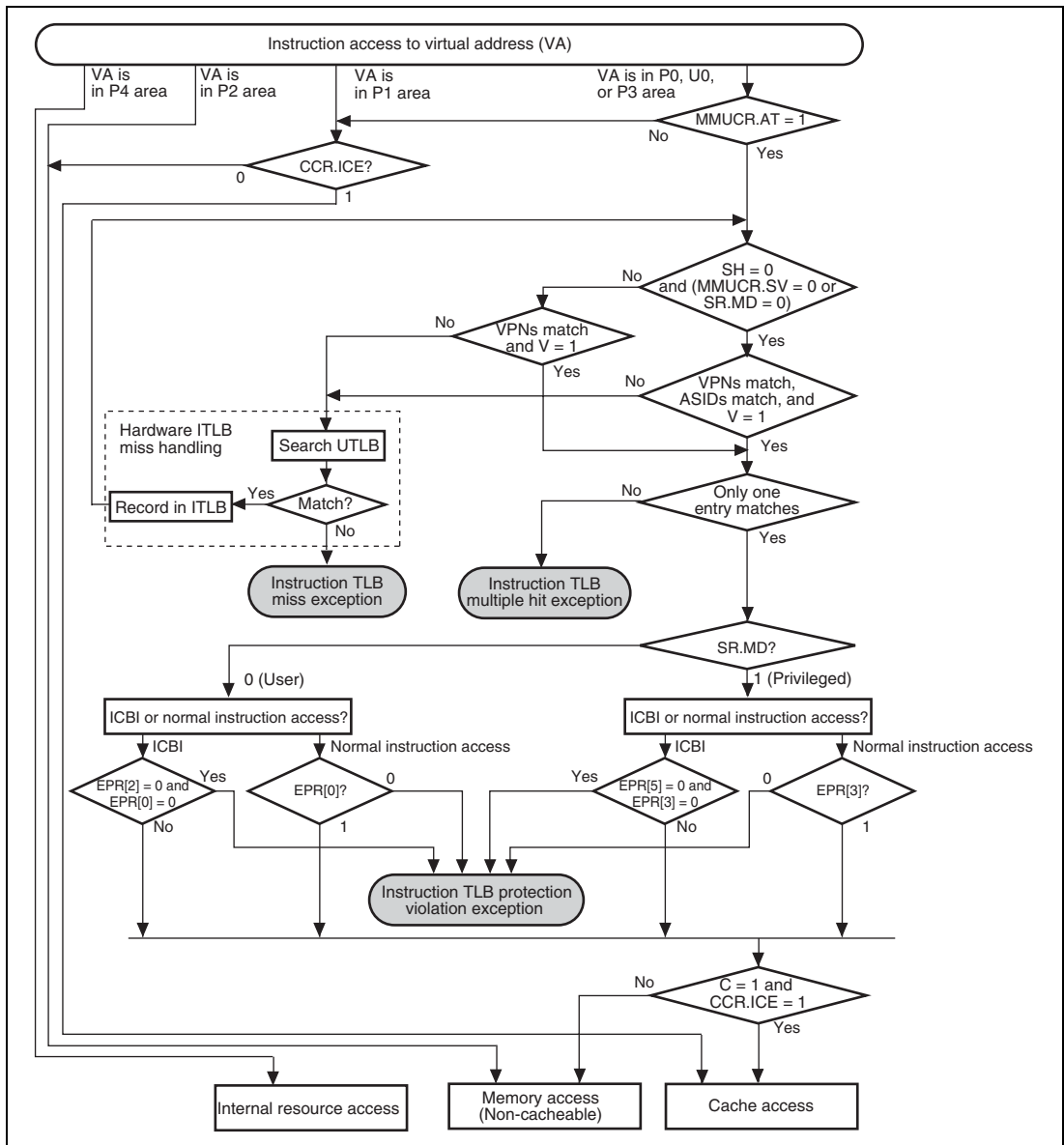


Figure 7.14 Flowchart of Memory Access Using UTLB (TLB Extended Mode)

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

The SH-4A 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)

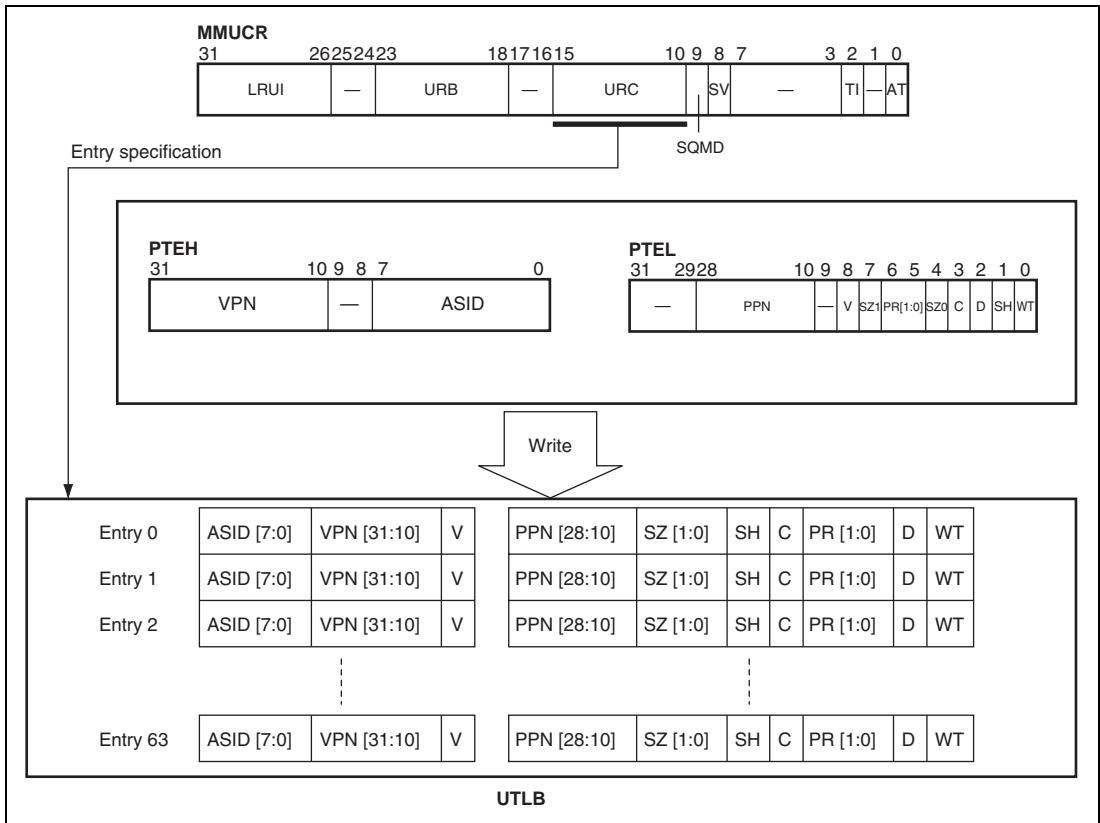
A TLB load instruction (LDTLB) is provided for recording UTLB entries. When an LDTLB instruction is issued, the SH-4A copies the contents of PTEH and PTEL (also the contents of PTEA in TLB extended mode) to the UTLB entry indicated by the URC bit in MMUCR. ITLB entries are not updated by the LDTLB instruction, and therefore address translation information purged from the UTLB entry may still remain in the ITLB entry. As the LDTLB instruction changes address translation information, ensure that it is issued by a program in the P1 or P2 area.

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

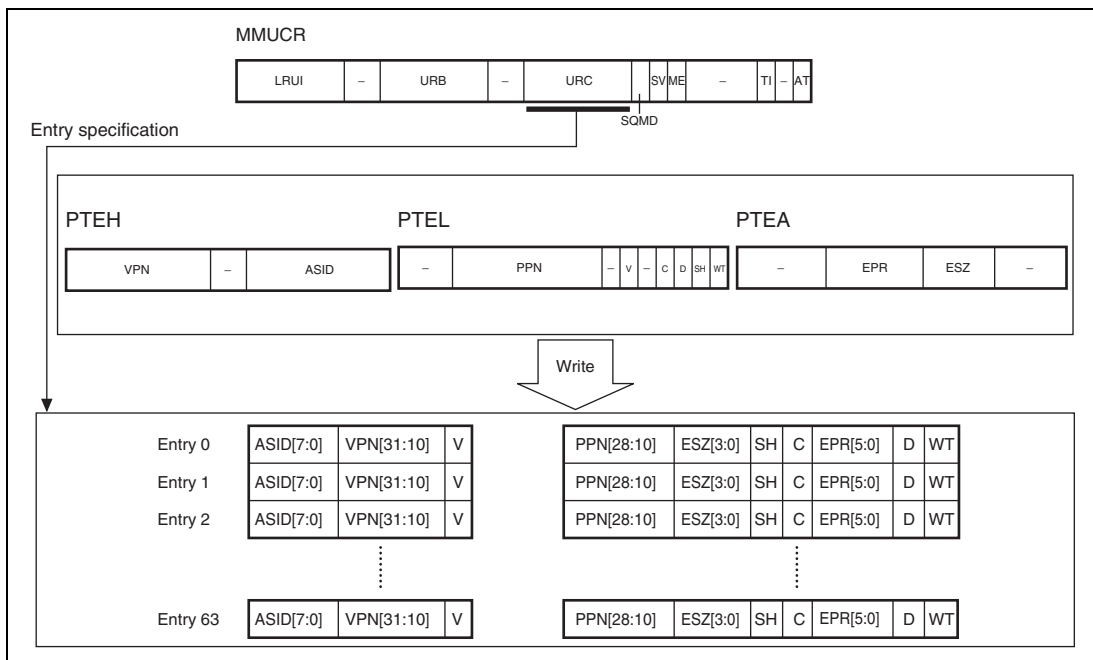
1. Execute a branch using the RTE instruction. In this case, the branch destination may be the area where TLB is used to translate the address.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the LT bit in IRMCR is 0 (initial value) before executing the LDTLB instruction,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

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

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



**Figure 7.16 Operation of LDTLB Instruction (TLB Compatible Mode)**



**Figure 7.17 Operation of LDTLB Instruction (TLB Extended Mode)**

#### 7.5.4 Hardware ITLB Miss Handling

In an instruction access, the SH-4A searches the ITLB. If it cannot find the necessary address translation information (ITLB miss occurred), the UTLB is searched by hardware, and if the necessary address translation information is present, it is recorded in the ITLB. This procedure is known as hardware ITLB miss handling. If the necessary address translation information is not found in the UTLB search, an instruction TLB miss exception is generated and processing passes to software.

### 7.5.5 Avoiding Synonym Problems

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

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

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

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

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

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

**Table 7.3 Cache Size and Countermeasure for Avoiding Synonym Problems**

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

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



## 7.6 MMU Exceptions

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

### 7.6.1 Instruction TLB Multiple Hit Exception

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

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

#### (1) Hardware Processing

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

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

#### (2) Software Processing (Reset Routine)

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

## 7.6.2 Instruction TLB Miss Exception

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

### (1) Hardware Processing

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

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

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

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

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

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

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

### 7.6.3 Instruction TLB Protection Violation Exception

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

#### (1) Hardware Processing

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

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

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

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

### **7.6.4 Data TLB Multiple Hit Exception**

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

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

#### **(1) Hardware Processing**

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

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

#### **(2) Software Processing (Reset Routine)**

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

### **7.6.5 Data TLB Miss Exception**

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

#### **(1) Hardware Processing**

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

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

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

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

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

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

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

### **7.6.6 Data TLB Protection Violation Exception**

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

#### **(1) Hardware Processing**

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

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

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

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

### 7.6.7 Initial Page Write Exception

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

#### (1) Hardware Processing

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

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

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

Software is responsible for the following processing:

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

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



## 7.7 Memory-Mapped TLB Configuration

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

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

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

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

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

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

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

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

7.7.1 ITLB Address Array

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

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

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

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

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

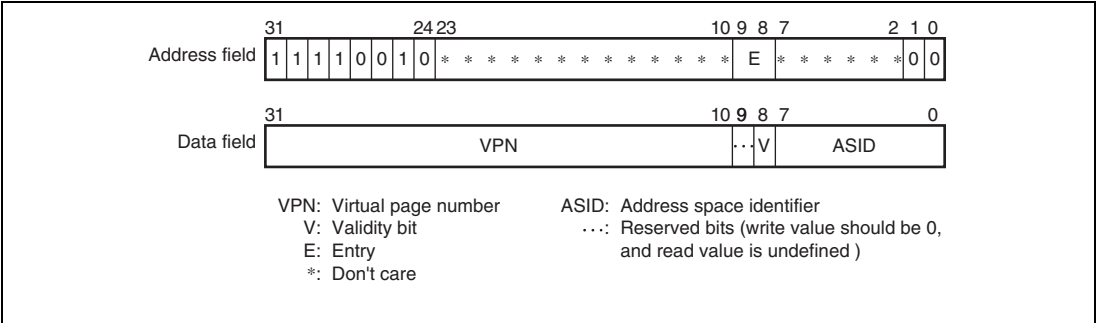


Figure 7.18 Memory-Mapped ITLB Address Array

### 7.7.2 ITLB Data Array (TLB Compatible Mode)

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

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

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

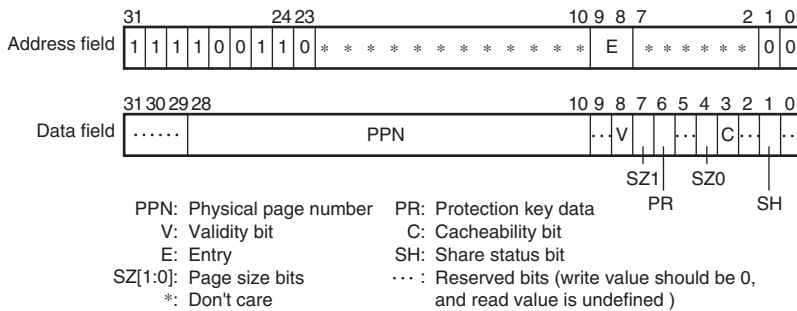
The following two kinds of operation can be used on ITLB data array:

1. ITLB data array read

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

2. ITLB data array write

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



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

7.7.3 ITLB Data Array (TLB Extended Mode)

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

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

(1) ITLB Data Array 1

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

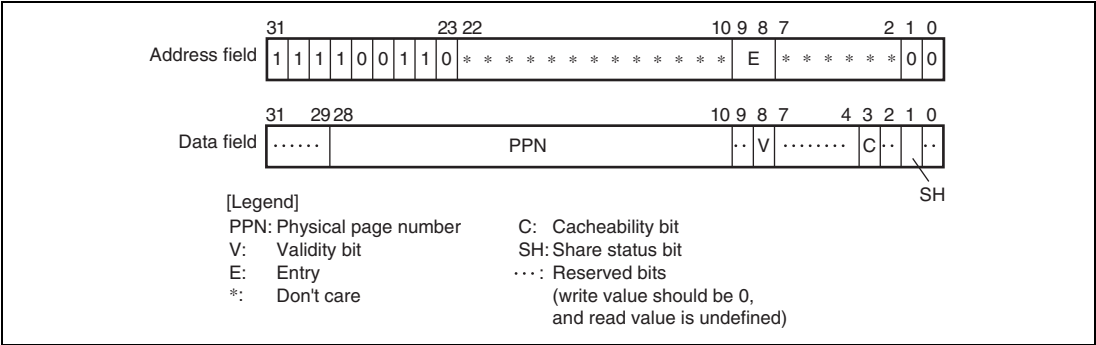


Figure 7.20 Memory-Mapped ITLB Data Array 1 (TLB Extended Mode)

## (2) ITLB Data Array 2

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

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

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

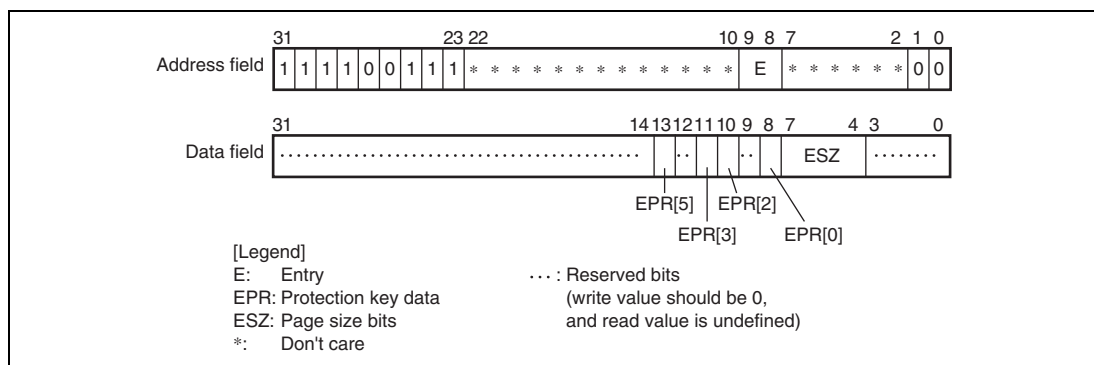
The following two kinds of operation can be applied to ITLB data array 2:

### 1. ITLB data array 2 read

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

### 2. ITLB data array 2 write

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



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

#### 7.7.4 UTLB Address Array

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

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

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

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

1. UTLB address array read

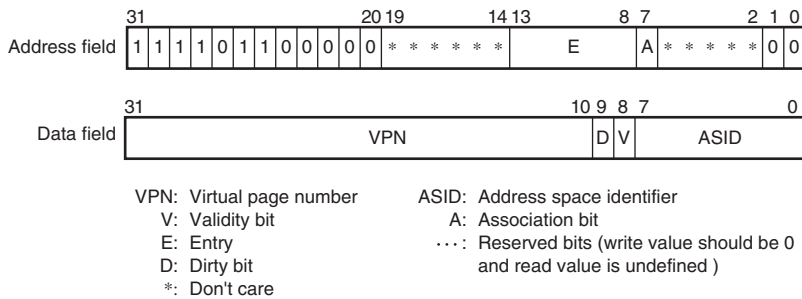
VPN, D, V, and ASID are read into the data field from the UTLB entry corresponding to the entry set in the address field. In a read, associative operation is not performed regardless of whether the association bit specified in the address field is 1 or 0.

2. UTLB address array write (non-associative)

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

3. UTLB address array write (associative)

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



**Figure 7.22 Memory-Mapped UTLB Address Array**

### 7.7.5 UTLB Data Array (TLB Compatible Mode)

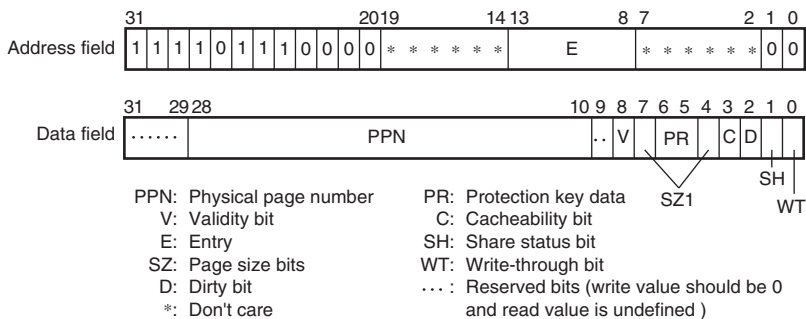
The UTLB data array is allocated to addresses H'F700 0000 to H'F70F FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in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

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

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

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

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



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

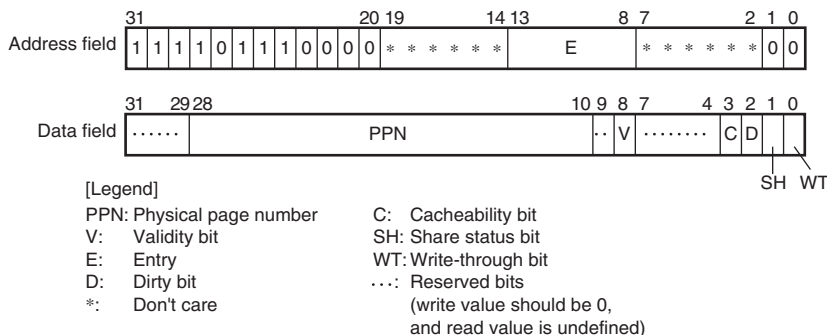
### 7.7.6 UTLB Data Array (TLB Extended Mode)

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

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

#### (1) UTLB Data Array 1

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



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



## (2) UTLB Data Array 2

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

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

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

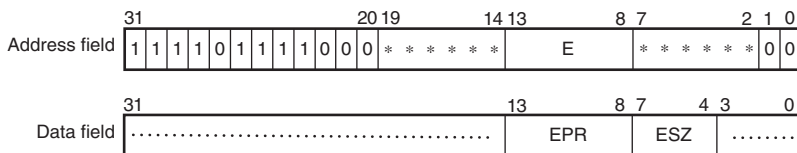
The following two kinds of operation can be applied to UTLB data array 2:

### 1. UTLB data array 2 read

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

### 2. UTLB data array 2 write

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



[Legend]

E: Entry

EPR: Protection key data

ESZ: Page size bits

\*: Don't care

...: Reserved bits

(write value should be 0,  
and read value is undefined)

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

## 7.8 Usage Notes

### 7.8.1 Note on Using LDTLB Instruction

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

1. Place the TLB miss exception handling routine\*<sup>1</sup> only in the P1, P2 area, or the on-chip memory so that all the instruction accesses\*<sup>2</sup> in the TLB miss exception handling routine should occur solely in the P1, P2 area, or the on-chip memory. Clear the RP bit in the RAMCR register to 0 (initial value), when the TLB miss exception handling routine is placed in the on-chip memory.  
Do not make an attempt to execute the FDIV or FSQRT instruction in the TLB miss exception handling routine.
2. If a TLB miss exception occurs, add 1 to MMUCR.URC before executing an LDTLB instruction.

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

2. Instruction accesses include the PREFI and ICBI instructions.

## Section 8 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.

### 8.1 Features

The features of the cache are given in table 8.1.

The SH-4A 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

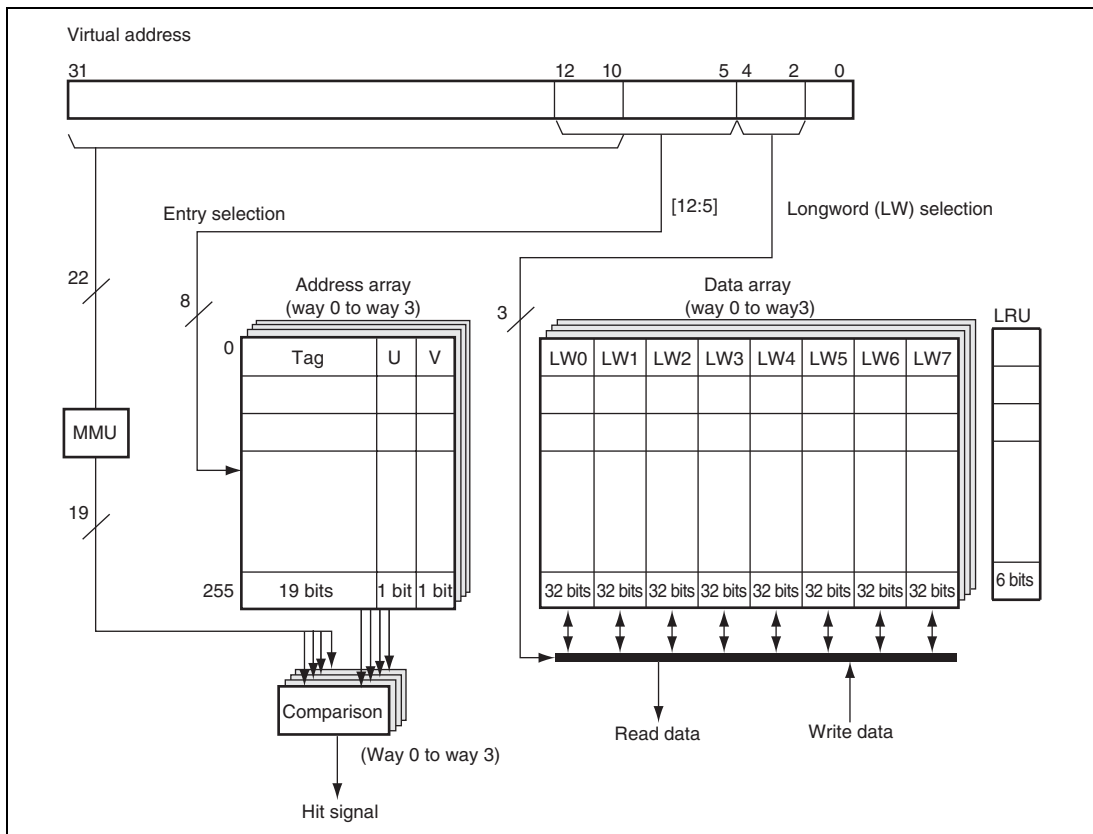
**Table 8.2 Store Queue Features**

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

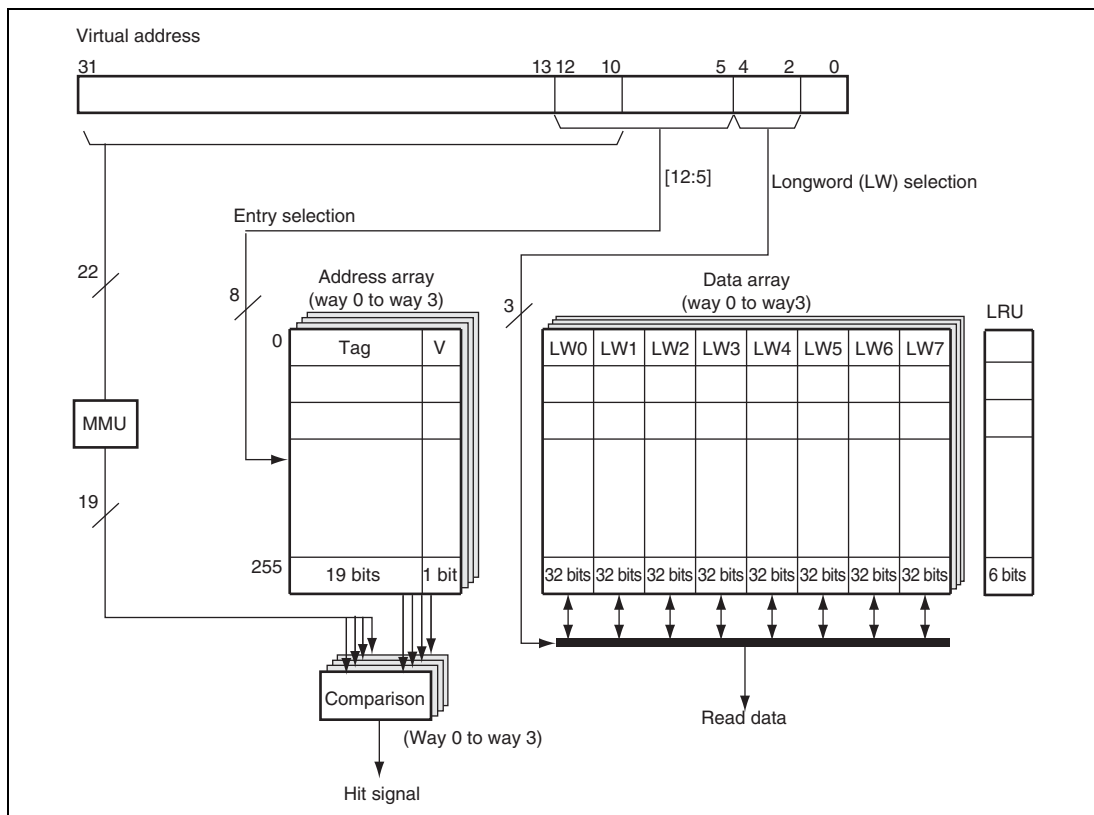
The operand cache of the SH-4A is 4-way set associative, each may comprising 256 cache lines. Figure 8.1 shows the configuration of the operand cache.

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

The SH-4A 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.
- **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.
- **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.

- 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.

- 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. 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	Sleep	Standby
Cache control register	CCR	H'0000 0000	Retained	Retained
Queue address control register 0	QACR0	Undefined	Retained	Retained
Queue address control register 1	QACR1	Undefined	Retained	Retained
On-chip memory control register	RAMCR	H'0000 0000	Retained	Retained

### 8.2.1 Cache Control Register (CCR)

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

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

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

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

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

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



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

## 8.2.2 Queue Address Control Register 0 (QACR0)

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	AREA0			—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	—	—	—	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

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

### 8.2.3 Queue Address Control Register 1 (QACR1)

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	AREA1			—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	—	—	—	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

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

## 8.2.4 On-Chip Memory Control Register (RAMCR)

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

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

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

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RMD	RP	IC2W	OC2W	ICWPW	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

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

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

## 8.3 Operand Cache Operation

### 8.3.1 Read Operation

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

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

3. Cache hit

The data indexed by virtual address bits [4:0] is read from the data field of the cache line on the 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 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 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 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.
  - 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)

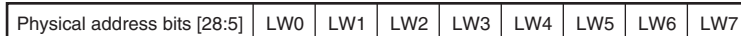
The tag and data field of the cache line on the way which is selected to replace are saved in the write-back buffer. Then a data write in accordance with the access size is performed for the data field on the hit way which is indexed by virtual address bits [4:0]. Then, the data, excluding the cache-missed 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, the SH-4A 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

The SH-4A 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

The SH-4A 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 the SH-4A, the following six instructions are supported for cache operations. Details of these instructions are given in section 11, Instruction Descriptions of the SH-4A Extended Functions Software Manual.

- Operand cache invalidate instruction: OCBI @Rn  
Operand cache invalidation (no write-back)
- Operand cache purge instruction: OCBP @Rn  
Operand cache invalidation (with write-back)
- Operand cache write-back instruction: OCBWB @Rn  
Operand cache write-back
- Operand cache allocate instruction: MOVCA.L R0,@Rn  
Operand cache allocation
- Instruction cache invalidate instruction: ICBI @Rn  
Instruction cache invalidation
- Operand access synchronization instruction: SYNCO  
Wait for data transfer completion

#### (2) Coherency Control

The operand cache can receive "PURGE" and "FLUSH" transaction from SuperHyway bus to control the cache coherency. Since the address used by the PURGE and FLUSH transaction is a physical address, do not use the 1 Kbyte page size to avoid cache synonym problem in MMU enable mode.

- PURGE transaction  
When the operand cache is enabled, the PURGE transaction checks the operand cache and invalidates the hit entry. If the invalidated entry is dirty, the data is written back to the external memory. If the transaction is not hit to the cache, it is no-operation.

- **FLUSH transaction**

When the operand cache is enabled, the FLUSH transaction checks the operand cache and if the hit line is dirty, then the data is written back to the external memory. If the transaction is not hit to the cache or the hit entry is not dirty, it is no-operation.

### **(3) Changes in Instruction Specifications Regarding Coherency Control**

Of the operand cache operating instructions, the coherency control-related specifications of OCBI, OCBP, and OCBWB have been changed from those of the SH-4A with H'20-valued VER bits in the processor version register (PVR).

- **Changes in the invalidate instruction OCBI@Rn**

When Rn is designating an address in a non-cacheable area, this instruction is executed as NOP in the SH-4A with H'20-valued VER bits in the processor version register (PVR). In the SH-4A with extended functions, this instruction invalidates the operand cache line designated by way = Rn[14:13] and entry = Rn[12:5] provided that Rn[31:24] = H'F4 (OC address array area). In this process, writing back of the line does not take place even if the line to be invalidated is dirty. This operation is only executable in privileged mode, and an address error exception occurs in user mode. TLB-related exceptions do not occur.

Do not execute this instruction to invalidate the memory-mapped array areas and control register areas for which Rn[31:24] is not H'F4, and their reserved areas (H'F0 to H'F3, H'F5 to H'FF).

- **Changes in the purge instruction OCBP@Rn**

When Rn is designating an address in a non-cacheable area, this instruction is executed as NOP in the SH-4A with H'20-valued VER bits in the processor version register (PVR). In the SH-4A with extended functions, this instruction invalidates the operand cache line designated by way = Rn[14:13] and entry = Rn[12:5] provided that Rn[31:24] = H'F4 (OC address array area). In this process, writing back of the line takes place when the line to be invalidated is dirty. This operation is only executable in privileged mode, and an address error exception occurs in user mode. TLB-related exceptions do not occur.

Do not execute this instruction to invalidate the memory-mapped array areas and control register areas for which Rn[31:24] is not H'F4, and their reserved areas (H'F0 to H'F3, H'F5 to H'FF).

- **Changes in the write-back instruction OCBWB@Rn**

When Rn is designating an address in a non-cacheable area, this instruction is executed as NOP in the SH-4A with H'20-valued VER bits in the processor version register (PVR). In the SH-4A with extended functions, provided that Rn[31:24] = H'F4 (OC address array area), this instruction writes back the operand cache line designated by way = Rn[14:13] and entry = Rn[12:5] if it is dirty and clears the dirty bit to 0. This operation is only executable in

privileged mode, and an address error exception occurs in user mode. TLB-related exceptions do not occur.

Do not execute this instruction to invalidate the memory-mapped array areas and control register areas for which  $Rn[31:24]$  is not H'F4, and their reserved areas (H'F0 to H'F3, H'F5 to H'FF).

### 8.5.2 Prefetch Operation

The SH-4A 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.



V : Validity bit  
A : Association bit  
... : Reserved bits (write value should be 0 and read value is undefined )  
\* : Don't care

**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.

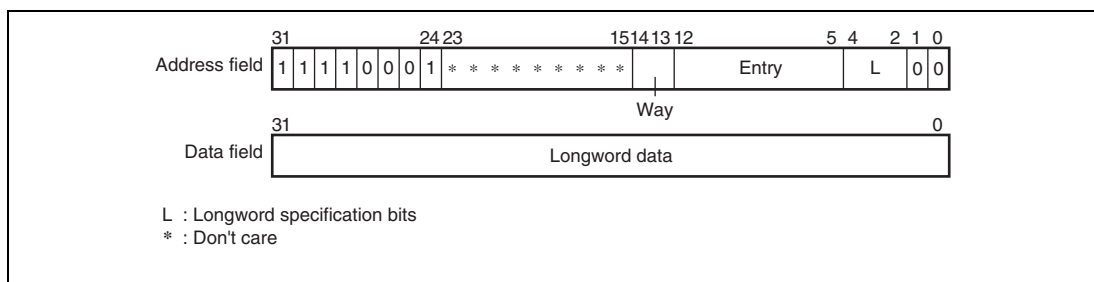
The following two kinds of operation can be used on the IC data array:

#### 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.

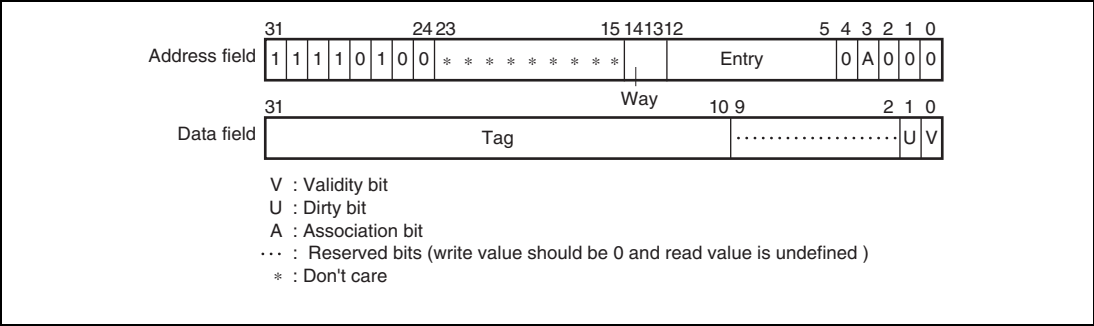
2. OC address array write (non-associative)

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**

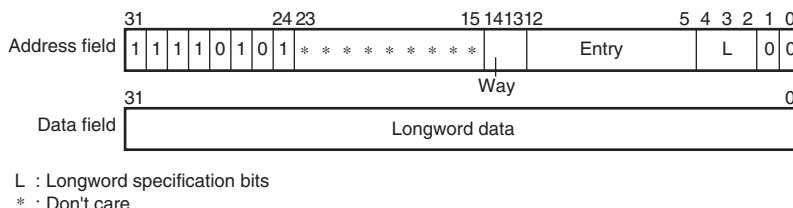
The OC data array is allocated to addresses H'F500 0000 to H'F5FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword 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

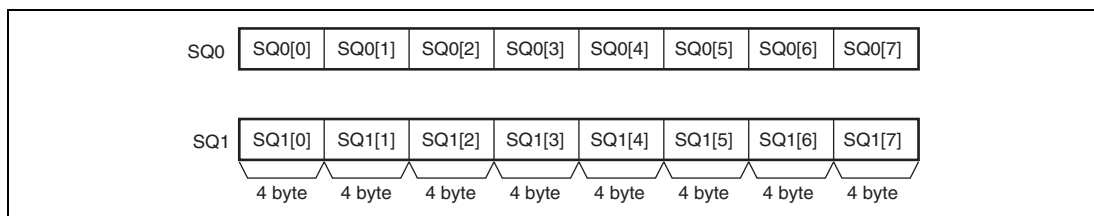
Associative writing to the IC and OC address arrays may not be supported in future SuperH-family products. The use of instructions ICBI, OCBI, OCBP, and OCBWB is recommended. These instructions handle ITLB misses, and notify instruction TLB miss exceptions and data TLB miss exceptions, thus providing a sure way of controlling the IC and OC. As a transitional measure, the SH-4A 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

The SH-4A 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 the SH-4A, reading the contents of the SQs may be performed by means of a load instruction for addresses H'FF00 1000 to H'FF00 103C in the P4 area. Only longword access is possible.

[31:6]	: H'FF00 1000	Store queue specification
[5]	: 0/1	0: SQ0 specification 1: SQ1 specification
[4:2]	: LW specification	Specifies longword position in SQ0/SQ1
[1:0]	: 00	Fixed at 0



## Section 9 On-Chip Memory

This LSI includes the IL memory which is suitable for instruction storage.

### 9.1 Features

#### (1) IL Memory

- Capacity  
The IL memory in this LSI is 16 Kbytes.
- Page  
The IL memory is divided into four pages (pages 0, 1, 2, and 3).
- Memory map  
The IL memory is allocated to the addresses shown in table 9.1 in both the virtual address space and the physical address space.

**Table 9.1 IL Memory Addresses**

	Memory Size
Page	16 Kbytes
Page 0	H'E520 0000 to H'E520 0FFF
Page 1	H'E520 1000 to H'E520 1FFF
Page 2	H'E520 2000 to H'E520 2FFF
Page 3	H'E520 3000 to H'E520 3FFF

- Ports  
The page has three independent read/write ports and is connected to the SuperHyway bus, the cache/RAM internal bus, and the instruction bus. The instruction bus is used when the IL memory is accessed through instruction fetch. The cache/RAM internal bus is used when the IL memory is accessed through operand access. The SuperHyway bus is used for IL memory access from the SuperHyway bus master module.
- Priority  
In the event of simultaneous accesses to the same page from different buses, the access requests are processed according to priority. The priority order is: SuperHyway bus > cache/RAM internal bus > instruction bus.

## 9.2 Register Descriptions

The following register is related to the on-chip memory.

**Table 9.2 Register Configuration**

Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
On-chip memory control register	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32

Note: \* The P4 address is the address used when using P4 area in the virtual address space.  
The area 7 address is the address used when accessing from area 7 in the physical address space using the TLB.

**Table 9.3 Register States in Each Processing Mode**

Name	Abbreviation	Power-On Reset	Sleep	Standby
On-chip memory control register	RAMCR	H'0000 0000	Retained	Retained

## 9.2.1 On-Chip Memory Control Register (RAMCR)

RAMCR controls the protective functions in the on-chip memory.

When updating RAMCR, please follow limitation described at section 7.2.4, On-Chip Memory Control Register (RAMCR).

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

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

Bit	Bit Name	Initial Value	R/W	Description
31to10	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.
9	RMD	0	R/W	On-Chip Memory Access Mode Specifies the right of access to the on-chip memory from the virtual address space. 0: An access in privileged mode is allowed. (An address error exception occurs in user mode.) 1: An access in user/ privileged mode is allowed.
8	RP	0	R/W	On-Chip Memory Protection Enable Selects whether or not to use the protective functions using ITLB and UTLB for accessing the on-chip memory from the virtual address space. 0: Protective functions are not used. 1: Protective functions are used. For further details, refer to section 9.4, On-Chip Memory Protective Functions.
7	IC2W	0	R/W	IC Two-Way Mode For further details, refer to section 8.4.3, IC Two-Way Mode.

Bit	Bit Name	Initial Value	R/W	Description
6	OC2W	0	R/W	OC Two-Way Mode For further details, refer to section 8.3.6, OC Two-Way Mode.
5	ICWPD	0	R/W	IC Way Prediction Disable For further details, refer to section 8.4.4, Instruction Cache Way Prediction Operation.
4 to 0	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.

---

## **9.3 Operation**

### **9.3.1 Instruction Fetch Access from the CPU**

Instruction fetch access from the CPU is performed directly via the instruction bus for a given virtual address. In the case of successive accesses to the same page of IL memory and as long as no page conflict occurs, the access takes one cycle.

### **9.3.2 Operand Access from the CPU and Access from the FPU**

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.

Note: Operand access is applied for PC relative access (@(disp,pc)).

### **9.3.3 Access from the SuperHyway Bus Master Module**

On-chip memory is always accessed by the SuperHyway bus master module, such as DMAC, via the SuperHyway bus which is a physical address bus. The same addresses as for the virtual addresses must be used.

## 9.4 On-Chip Memory Protective Functions

The SH-4A implements the following protective functions to the on-chip memory by using the on-chip memory access mode bit (RMD) and the on-chip memory protection enable bit (RP) in the on-chip memory control register (RAMCR).

- Protective functions for access from the CPU and FPU

When RAMCR.RMD = 0, and the on-chip memory is accessed in user mode, it is determined to be an address error exception.

When MMUCR.AT = 1 and RAMCR.RP = 1, MMU exception and address error exception are checked in the on-chip memory area which is a part of area P4 as with the area P0/P3/U0.

The above descriptions are summarized in table 9.4.

**Table 9.4 Protective Function Exceptions to Access On-Chip Memory**

MMUCR.AT	RAMCR.RP	SR.MD	RAMCR. RMD	Always Occurring Exceptions	Possibly Occurring Exceptions
0	x	0	0	Address error exception	—
			1	—	—
		1	x	—	—
1	0	0	0	Address error exception	—
			1	—	—
		1	x	—	—
	1	0	0	Address error exception	—
			1	—	MMU exception
		1	x	—	MMU exception

[Legend] x: Don't care

## 9.5 Usage Notes

### 9.5.1 Page Conflict

In the event of simultaneous access to the same page from different buses, page conflict occurs. Although each access is completed correctly, this kind of conflict tends to lower on-chip memory accessibility. Therefore it is advisable to provide all possible preventative software measures. For example, conflicts will not occur if each bus accesses different pages.

### 9.5.2 Access Across Different Pages

Access from the instruction bus is performed in one cycle when the access is made successively to the same page but takes multiple cycles (a maximum of two wait cycles may be required) when the access is made across pages or the previous access was made to memory other than IL memory. For this reason, from the viewpoint of performance optimization, it is recommended to design the software such that the target page does not change so often in access from the instruction bus. For example, allocating a separate program for each page will deliver better efficiency.

### 9.5.3 On-Chip Memory Coherency

In order to allocate instructions in the IL memory, write an instruction to the IL memory, execute the following sequence, then branch to the rewritten instruction.

- SYNCO
- ICBI @Rn

In this case, the target for the ICBI instruction can be any address (IL memory address may be possible) within the range where no address error exception occurs, and cache hit/miss is possible.

### 9.5.4 Sleep Mode

The SuperHyway bus master module, such as DMAC, cannot access IL memory in sleep mode.





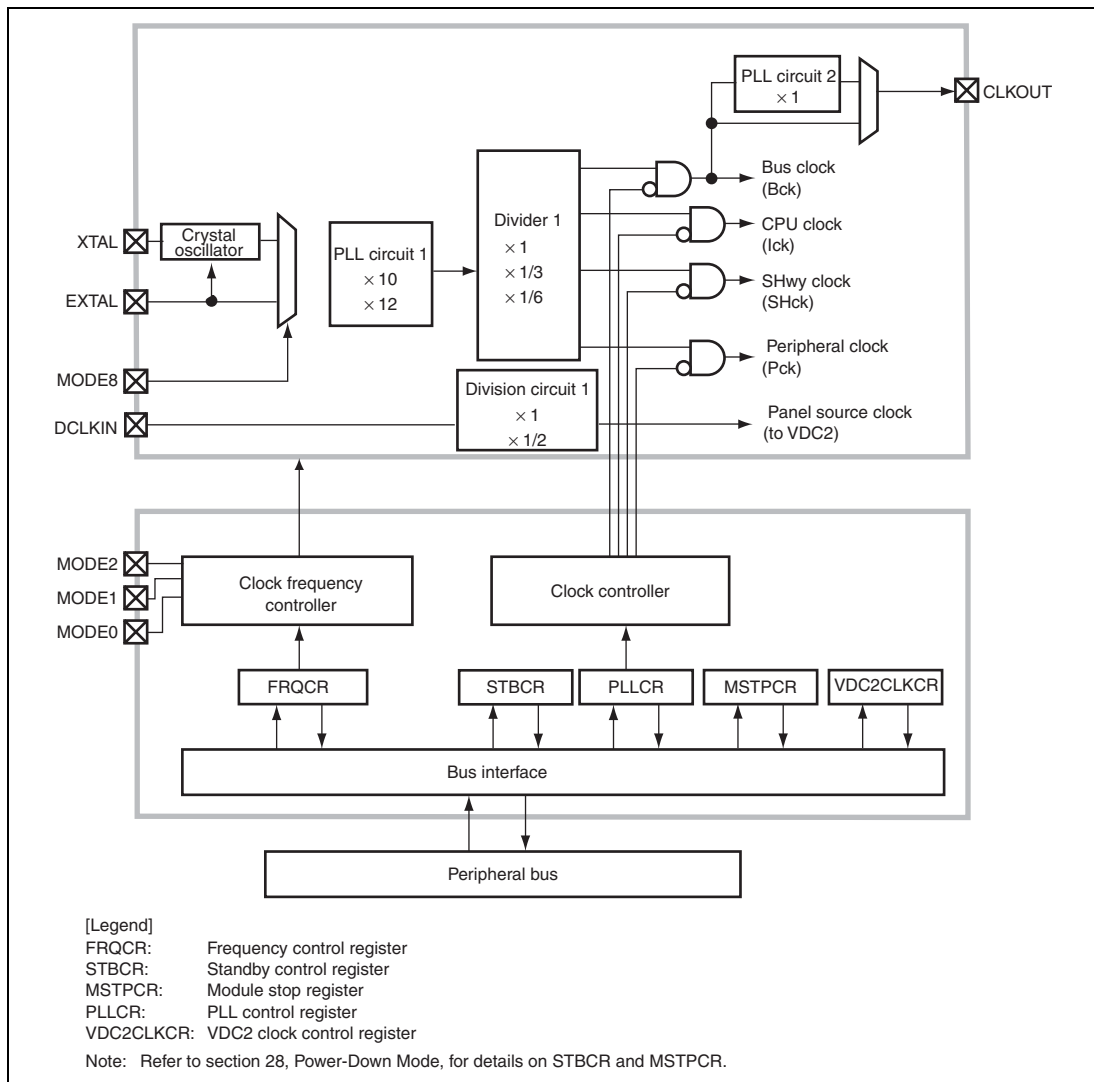
## Section 10 Clock Pulse Generator (CPG)

The CPG generates clocks provided to the on-chip peripheral modules and external bus interface of this LSI, and controls the power-down mode function. The CPG consists of an oscillator, PLL circuits, frequency dividers, and control circuits.

### 10.1 Features

- Clocks used for LSI internal operation  
Generates the CPU clock (Ick) used by the CPU, FPU, cache, and TLB, SHwy clock (SHck) used by the SuperHyway, and peripheral clock (Pck) supplied to the peripheral modules.
- Clocks supplied to outside modules  
Generates the bus clock (Bck) used by the external bus interface.
- Clock modes  
Either a crystal resonator or an externally input clock can be selected as the CPG clock input. The combination of the division ratios for the CPU clock, SHwy clock, bus clock, and peripheral clock after a power-on reset can be selected from two clock operating modes.
- Power-down mode control  
The clock can be stopped for sleep mode and refresh standby mode, and specific modules can be stopped in module standby mode.

A block diagram of the CPG is shown in figure 10.1.



**Figure 10.1 Block Diagram of CPG**

The functions of the blocks in the CPG are as follows.

### **(1) PLL Circuit 1**

PLL circuit 1 multiplies the frequency of the crystal oscillator or the clock input from the EXTAL pin by the ratio of  $\times 10$  or  $\times 12$ . The multiplication ratio is selected by the combination of mode control pins MODE0, MODE1, and MODE2.

### **(2) PLL Circuit 2**

PLL circuit 2 aligns the phases of the bus clock (Bck) and the clock signal output from the CLKOUT pin that is used by the external peripheral interface.

### **(3) Crystal Oscillator**

The crystal oscillator is a clock pulse generator used when a crystal resonator is connected to the XTAL or EXTAL pin. The crystal oscillator can be enabled by the MODE8 pin setting.

### **(4) Divider 1**

Divider 1 generates the CPU clock (Ick), SHwy clock (SHck), peripheral module clocks (Pck), and bus clock (Bck).

### **(5) Frequency Control Register (FRQCR)**

The frequency control register is a read-only register that shows the frequency division ratios for the Ick, SHck, Pck, and Bck.

### **(6) PLL Control Register (PLLCR)**

The PLL control register has control bits assigned for enabling or disabling the CLKOUT pin output.

### **(7) Module Stop Registers (MSTPCR)**

The module stop register has control bits for running/stopping the individual peripheral modules. For the details of MSTPCR, see section 28, Power-Down Mode.

### **(8) Standby Control Register (STBCR)**

The standby control register has bits for controlling the stand by modes. For the details of STBCR, see section 28, Power-Down Mode.

## 10.2 Input/Output Pins

Table 10.1 lists the CPG pin configuration.

**Table 10.1 Pin Configuration and Functions of CPG**

Pin Name	Function	I/O	Description
MODE0	Mode control pins 0, 1, 2 (Clock operating mode)	Input	Sets the clock operating mode after a power-on reset.
MODE1		Input	
MODE2		Input	
MODE8	Mode control pin 8 (Clock input mode)	Input	<p>Selects the use of the crystal resonator.</p> <p>MODE8 = low: External clock is input from the EXTAL pin.</p> <p>MODE8 = high: Crystal resonator is connected to the EXTAL and XTAL pins.</p>
XTAL	Clock pins	Output	A crystal resonator is connected.
EXTAL		Input	A crystal resonator is connected, or an external clock is input.
CLKOUT		Output	Used as an external bus clock output pin.

**Note:** For the guaranteed AC timing of the CLKOUT pin, refer to the section on electrical characteristics. Pay attention to the relationship between the input frequency of the crystal oscillator and the multiplication ratio.

### 10.3 Clock Operating Mode

Table 10.2 shows the relationship between the mode control pin (MODE0, MODE1, and MODE2) combinations and the clock operating mode after a power-on reset.

**Table 10.2 Clock Operating Modes**

Clock operating mode	External pin combination*1					EXTAL frequency (MHz)		Clock generated by CPG				Initial value of FRQCR
	MODE2	MODE1	MODE0	PLL1	PLL2			lck	SHck	Bck	Pck	
2	0	1	0	ON	ON	25 to 33.1 15 to 32.4	Frequency ratio*2	10	10/3	10/3	10/6	H'30320044
							Max. frequency	324	108	108	54	
3	0	1	1	ON	ON	25 only 12.5 to 27	Frequency ratio*2	12	4	4	2	H'40320044
							Max. frequency	324	108	108	54	

Notes: 1. Mode pin (MODE0, MODE1, and MODE2) combinations other than above are prohibited.

2. The ratio of the frequency of each clock to that of the crystal oscillator or the clock input from the EXTAL pin.

## 10.4 Register Descriptions

Table 10.3 shows the CPG register configuration. Table 10.4 shows the register states in each operating mode.

**Table 10.3 Register Configuration**

Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
Frequency control register	FRQCR	R	H'FFC8 0000	H'1FC8 0000	32
PLL control register	PLLCR	R/W	H'FFC8 0024	H'1FC8 0024	32
VDC2 clock control register	VDC2CLKCR	R/W	H'FFC8 0004	H'1FC8 0004	32

**Table 10.4 Register States in Each Operating Mode**

Register Name	Abbreviation	Power-On Reset	Standby	Sleep
Frequency control register	FRQCR	H'x032 0044*	Retained	Retained
PLL control register	PLLCR	H'0000 E001	Retained	Retained
VDC2 clock control register	VDC2CLKCR	H'0000 0080	Retained	Retained

Note: \* The initial value after a power-on reset is determined by the combination of the external pins, MODE0, MODE1, and MODE2.

### 10.4.1 Frequency Control Register (FRQCR)

FRQCR is a 32-bit read-only register used to confirm the division ratios for the CPU clock (Ick), SHwy clock (SHck), peripheral clock (Pck), and the bus clock (Bck) after a power-on reset. For the frequency ratios, refer to table 10.2, Clock Operating Mode. This register can be accessed only in longwords. Operation cannot be guaranteed if this register is written to.

FRQCR is only initialized by a power-on reset caused by the  $\overline{\text{PRESET}}$  pin or watchdog timer overflow.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CFC[2:0]			—	BFC[2:0]		
Initial value:	—	—	—	—	0	0	0	0	0	0	1	1	0	0	1	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
	—	—	—	—	—	—	—	—	—	PFC[2:0]			—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	1	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 28	—	Undefined	R	Reserved These bits are read as B'0011 when the clock operating mode is mode 2, and read as B'0100 when mode 3.
27 to 23	—	All 0	R	Reserved These bits are always read as all 0.
22 to 20	CFC[2:0]	011	R	CPU Clock (Ick) and SHwy Clock (SHck) Frequency Division Ratios CFC[2:0]    Ick    SHck 011:        ×1    ×1/3
19	—	0	R	Reserved This bit is always read as 0.
18 to 16	BFC[2:0]	010	R	Bus Clock (Bck) Frequency Division Ratio 010: ×1/3
15 to 7	—	All 0	R	Reserved These bits are always read as all 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PFC[2:0]	100	R	Peripheral Clock (Pck) Frequency Division Ratio 100: $\times 1/6$
3	—	0	R	Reserved This bit is always read as 0.
2	—	1	R	Reserved This bit is always read as 1.
1, 0	—	All 0	R	Reserved These bits are always read as 0.



## 10.4.2 PLL Control Register (PLLCR)

PLLCR is a 32-bit readable/writable register that enables or disables clock output from the CLKOUT pin. PLLCR can be accessed only in longwords.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved The write value should be the same as the initial values.
15 to 13	—	All 1	R	Reserved The write value should be the same as the initial values.
12 to 2	—	All 0	R	Reserved The write value should be the same as the initial values.
1	CKOFF	0	R/W	CLKOUT Output Stop 0: Clock is output from the CLKOUT pin 1: Clock is not output from the CLKOUT pin (The pin is placed in a Hi-Z state)
0	CKONE	1	R/W	Clock Output Enable Selects whether to output clock from the CLKOUT pin or tie the CLKOUT pin to a low level during software standby mode. 0: Tied to a low level 1: Clock is output

### 10.4.3 VDC2 Clock Control Register (VDC2CLKCR)

VDC2CLKCR is a 32-bit readable/writable register that selects the VDC2 clock.

VDC2CLKCR can be accessed only in longwords.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CKSEL	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R	R	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	CKSEL	1	R/W	Panel Source Clock Division Ratio select This bit specifies the frequency division ratio of the panel source clock that is input from the DCLKIN pin and supplied to the VDC2 module. 0: DCLKIN input x 1/1 times 1: DCLKIN input x 1/2 times
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

## Section 11 Memory Controller Unit (MCU)

The memory controller unit (MCU) arbitrates accesses from the CPU and various modules and outputs the appropriate control signals for SRAM and SDRAM interfaces. This module allows direct connection with the SRAM, ROM, and SDRAM.

This module is provided with the SuperHyway bus interface (SHIF), pixel bus interface (PXIF), LCD controller bus interface (LCDIF), SRAM controller (LBSC), SDRAM controller (SBSC), and arbiter (ARBT) that arbitrates accesses from the interface modules to the controllers.

### 11.1 Features

- Supports external memory access
  - Outputs four external memory select signals
  - Supports four external memory areas (FLASH, SDRAM), each of which has 64 Mbytes max.
- SRAM: 32-, 16-, or 8-bit data bus width selectable
- SDRAM: 64- or 32-bit data bus width selectable
- Big endian or little endian mode can be set

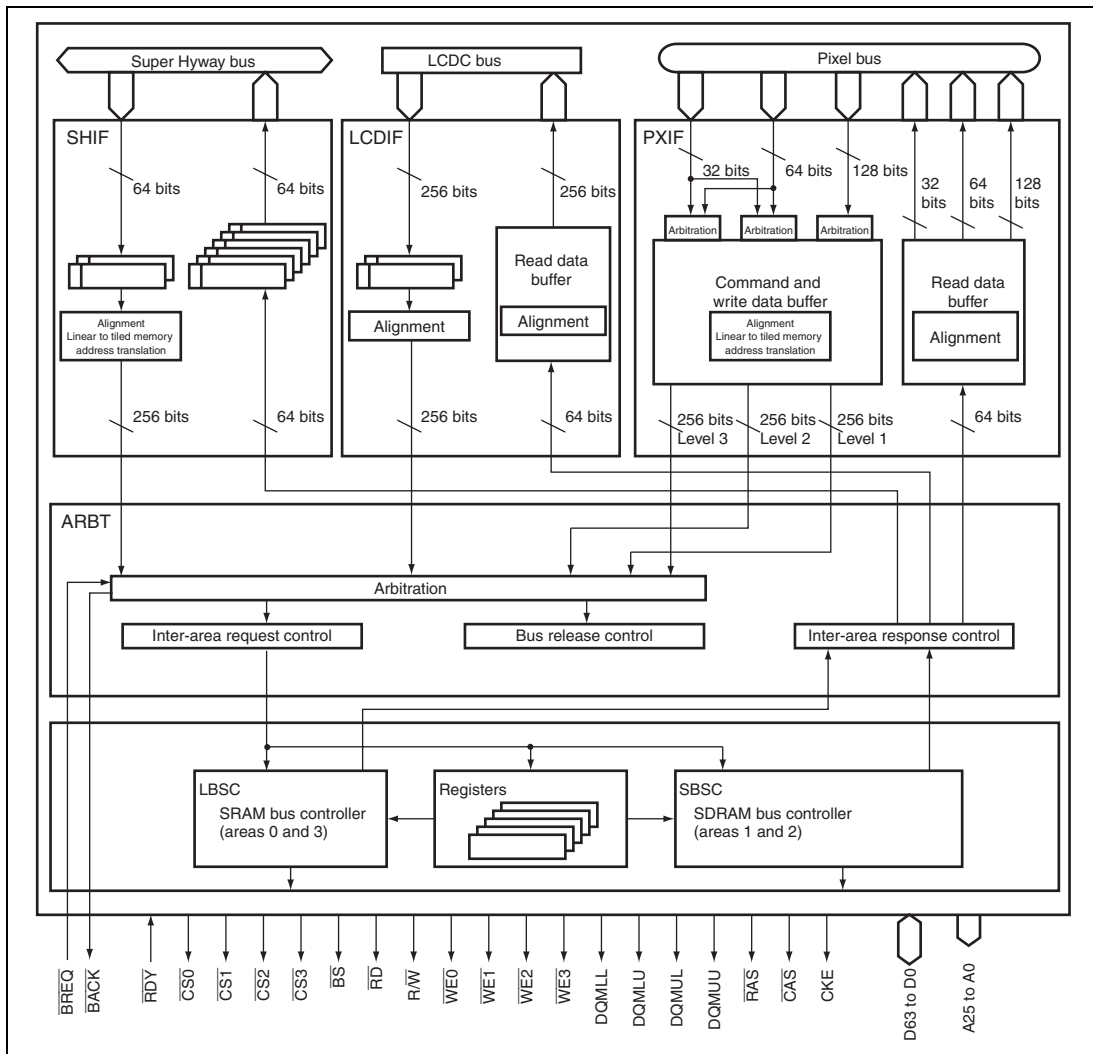
#### [SRAM interface]

- NOR-type flash memory can be connected
- Cycle wait function: Wait control by hardware through signals
- Wait control for preventing collisions on the data bus (idle cycle insertion):
  - Wait setting between read cycles
  - Wait setting between a read cycle and a write cycle

#### [SDRAM interface]

- Refresh function:
  - Auto-refresh (programmable refresh counter provided)
  - Self-refresh
- Timing control: Row-column latency, column latency, row active period, write recovery period, row precharge period, auto-refresh request interval, initial precharge cycle count, and initial auto-refresh request interval
- Burst access mode: Random column (SDRAM burst length: eight for 32-bit bus or four for 64-bit bus)
- Initialization sequencer function: Issues precharge and auto-refresh commands

Figure 11.1 shows the block diagram of the MMU.



**Figure 11.1 Block Diagram of MCU**

1. SHIF (SuperHyway bus interface): An interface between the CPU and SRAM or SDRAM. The SuperHyway protocol is used for interfacing. The bus is 64 bits wide. The SuperHyway bus is a split-transaction bus allowing efficient data transfer. In the split transaction, communication is split into request packets and response packets. Using this system prevents the bus from being occupied throughout one communication session.
2. PXIF (pixel bus interface): An interface block to which peripheral modules are connected that access unified memory (SDRAM). Separate ports are provided for each module; signals are connected from pier to pier using the pixel bus protocol. The 32-, 64-, and 128-bit-wide buses are supported.
3. LCDIF (LCD controller bus interface): An interface to which an LCD controller (LCDC) is directly connected. Only an LCDC is to be connected, and the exclusive bus protocol is used. The bus is 256 bits wide. Only the SDRAM space can be accessed via this bus.
4. ARBT (arbiter): Arbitrates accesses between the SHIF, PXIF, and LCDIF based on the priority determined separately.
5. LBSC (SRAM local bus controller): Controls read and write accesses to the SRAM. The LBSC has the following features.
  - Areas 0 and 3 can be controlled in the external memory space.
  - Memory size of each area is 64 Mbytes maximum.
  - Bus width of 8, 16, or 32 bits can be set (by the external pins for area 0 and by the register for area 3).
  - Wait states can be inserted using the  $\overline{\text{RDY}}$  pin.
  - Wait-state insertion can be controlled through programming.
  - Wait cycles can be inserted automatically between consecutive memory accesses to prevent data bus conflict.
  - Setup time and hold time can be inserted for the write strobe on a write cycle to enable connection to low-speed memory.
6. SBSC (SDRAM bus controller): Controls read and write accesses to the SDRAM. Commands are issued and read/write data is transmitted and received according to the SDRAM timing specifications. The 32- and 64-bit wide buses are supported; the burst length is eight for the 32-bit bus and four for the 64-bit bus; and the data transfer unit is basically 32 bytes. Auto-refresh and self-refresh modes are supported. When a row address hit occurs in bank open mode, data is consecutively transferred in burst mode.

## 11.2 Input/Output Pins

Table 11.1 shows the MCU pin configuration.

**Table 11.1 MCU Pin Configuration**

Pin Name	Function	I/O	Description
A25 to A0	Address bus	Output	Address output
D63 to 32	Data bus	I/O	Data input/output (multiplexed with the other pins)
D31 to D0	Data bus	I/O	Data input/output
$\overline{\text{BS}}$	Bus cycle start	Output	Signal that indicates the start of a bus cycle.
$\overline{\text{CS3}}$ to $\overline{\text{CS0}}$	Chip select	Output	Chip select signal that indicates the area being accessed.
$\overline{\text{RD}}$	Read	Output	Read signal from the external device
$\overline{\text{R/W}}$	Read/write	Output	Data bus input/output direction designation signal. Also used as WE signal during SDRAM access.
$\overline{\text{RAS}}$	Row address strobe	Output	SDRAM RAS signal
$\overline{\text{CAS}}$	Column address strobe	Output	SDRAM CAS signal
$\overline{\text{CKE}}$	Clock enable	Output	SDRAM clock enable signal
$\overline{\text{DQMLL}}$	Data mask	Output	SDRAM data mask signal for D7 to D0
$\overline{\text{DQMLU}}$	Data mask	Output	SDRAM data mask signal for D15 to D8
$\overline{\text{DQMUL}}$	Data mask	Output	SDRAM data mask signal for D23 to D16
$\overline{\text{DQMUU}}$	Data mask	Output	SDRAM data mask signal for D31 to D24
$\overline{\text{WE0}}$	Data enable 0	Output	During SRAM access: write strobe signal for D7 to D0.  When setting SDRAM interface: data mask signal for D39 to D32 (high active)

Pin Name	Function	I/O	Description
$\overline{WE1}$	Data enable 1	Output	During SRAM access: write strobe signal for D15 to D8  When setting SDRAM interface: data mask signal for D47 to D40 (high active)
$\overline{WE2}$	Data enable 2	Output	During SRAM access: write strobe signal for D23 to D16  When setting SDRAM interface: data mask signal for D55 to D48 (high active)
$\overline{WE3}$	Data enable 3	Output	During SRAM access: write strobe signal for D31 to D24  When setting SDRAM interface: data mask signal for D63 to D56 (high active)
$\overline{RDY}$	Ready	Input	Wait cycle request signal
$\overline{BREQ}$	Bus release request	Input	Bus release request signal
$\overline{BACK}$	Bus request acknowledge/ bus return request	Output	Bus release acknowledge signal/bus return request signal
MODE3, MODE4	Area 0 bus width	Input	Signal setting area 0 bus width at power-on reset
MODE5	Endian switchover	Input	Endian setting at power-on reset
$\overline{DACK0}^*$	DMAC0 acknowledge signal	Output	Data acknowledge of DMAC channel 0
$\overline{DACK1}^*$	DMAC1 acknowledge signal	Output	Data acknowledge of DMAC channel 1
$\overline{DTEND0}^*$	DMAC0 transfer end signal	Output	Transfer end of DMAC channel 0
$\overline{DTEND1}^*$	DMAC1 transfer end signal	Output	Transfer end of DMAC channel 1

Note: \* The polarity (initial state is low active) of the  $\overline{DACK0}$ ,  $\overline{DACK1}$ ,  $\overline{DTEND0}$ , and  $\overline{DTEND1}$  pins can be selected by the AL bit in CHCR0 and CHCR1 of the DMAC.

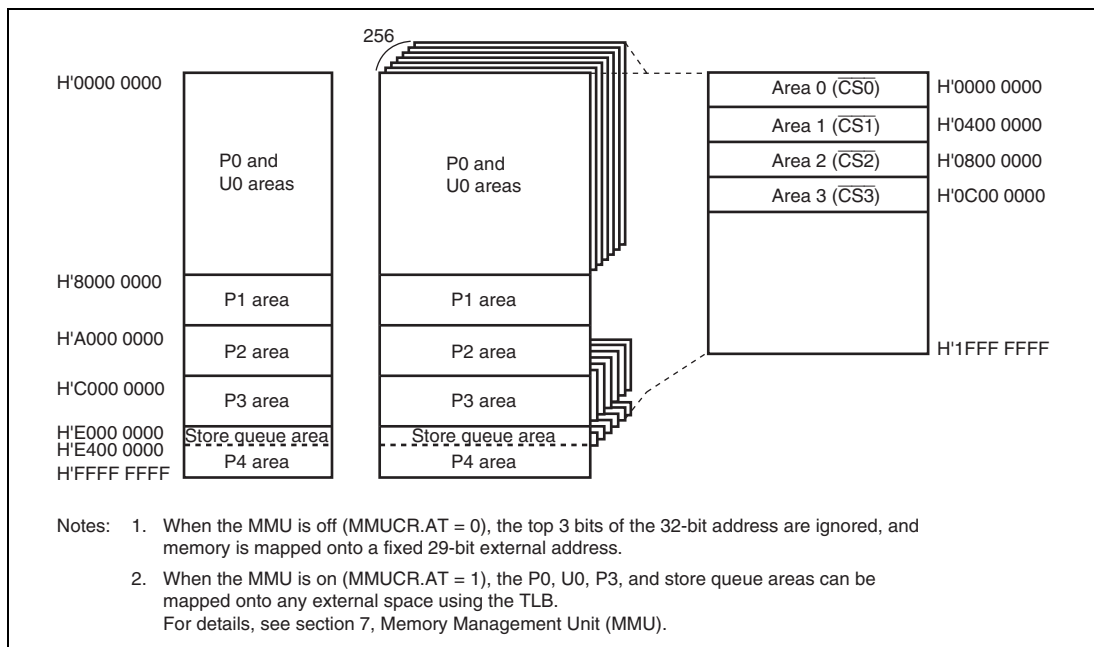
## 11.3 Area Overview

### 11.3.1 Space Divisions

The architecture of this LSI provides a 32-bit virtual address space. The virtual address space is divided into five areas according to the upper address value. The external memory space indicated by the remaining 29 address bits is divided into four areas.

The virtual addresses can be allocated to any external address using the address translation function of the MMU. For details, see section 7, Memory Management Unit (MMU). This section describes the area division of the external address space.

With this LSI, SRAM or SDRAM can be connected to each of the four areas in the external address space as shown in table 11.2.



**Figure 11.2 Correspondence between Virtual Address Space and External Memory Space**



**Table 11.2 External Memory Space Map**

Area	External Address	Size	Connectable Memory	Specifiable Bus Width (Bit)	Access Size
0	H'0000 0000 to H'03FF FFFF	64 Mbytes	SRAM	8, 16, 32* <sup>1</sup>	8, 16, 32
1	H'0400 0000 to H'07FF FFFF	64 Mbytes	SDRAM	32, 64* <sup>2</sup>	32, 64
2	H'0800 0000 to H'0BFF FFFF	64 Mbytes	SDRAM	32, 64* <sup>2</sup>	32, 64
3	H'0C00 0000 to H'0FFF FFFF	64 Mbytes	SRAM	8, 16, 32* <sup>2</sup>	8, 16, 32

Notes: 1. The memory bus width is specified by external pins.

2. The memory bus width is specified by the register.

### 11.3.2 Memory Bus Width

The memory bus width is set differently for each area. For area 0, a bus width of 8, 16, or 32 bits is set according to the external pin settings at a power-on reset by the PRESET pin. The correspondence between the external pins (MODE4 and MODE3) and the bus width is shown in table 11.3.

**Table 11.3 MODE Pin Settings for Memory Bus Width of Area 0**

MODE4	MODE3	Bus Width
0	0	Reserved
0	1	8 bits
1	0	16 bits
1	1	32 bits

For areas 1 and 2, a bus width of 32 or 64 bits can be selected through the memory interface mode register (MIM) (the bus width is same for areas 0 and 1). For details, see section 11.4.2, Memory Interface Mode Register (MIM).

For area 3, a bus width of 8, 16, or 32 bits can be selected through the CS3 bus control register (CS3BCR). For details, see section 11.4.15, CS3 Bus Control Register (CS3BCR).

### 11.3.3 Endian Setting

The endian mode is set by the state of the external pin (MODE5). The endian setting by the pin applies to areas 0, 1, 2, and 3. The correspondence between the MODE5 pin and the endian is shown in table 11.4.

**Table 11.4 Endian Setting by Pin**

<b>MODE5</b>	<b>Endian</b>
0	Big endian
1	Little endian

## 11.4 Register Description

Table 11.5 shows the MCU registers. The registers are 64 bits wide, but they should be accessed in longword (32-bit) units. When data is written, it is reflected in the state in longword units. When data is read, a longword value set at the point of access is referenced. When accessing data in bits 63 to 32 and bits 31 to 0, specify the addresses  $8n + 0$  and  $8n + 4$ , respectively. These registers control the interface with various types of memories, and the number of wait states.

**Table 11.5 Register Configuration**

Address	Register Name	Abbr.	Initial Value	Access Size
H'FF800000	Version control register	VCR	H'0B04000000000000	32
H'FF800008	Memory interface mode register	MIM	H'00000000061A0x40	32
H'FF800010	SDRAM control register	SCR	H'0000000000000000	32
H'FF800018	SDRAM timing register	STR	H'0000000000FFFE7	32
H'FF800030	SDRAM row attribute register	SDRA	H'0000000000000200	32
H'FFAxxxxx	SDRAM mode register	SDMR	—	32
H'FF800200	Arbitration mode register	AMR	H'0000000004000000	32
H'FF800100	Linear-to-tiled memory address translation control register 0	LTC0	H'0000000000000000	32
H'FF800108	Linear-to-tiled memory address translation area start address register 0	LTAD0	H'0000000000000000	32
H'FF800110	Linear-to-tiled memory address translation area start address mask register 0	LTAM0	H'0000000000000000	32
H'FF800118	Linear-to-tiled memory address translation control register 1	LTC1	H'0000000000000000	32
H'FF800120	Linear-to-tiled memory address translation area start address register 1	LTAD1	H'0000000000000000	32
H'FF800128	Linear-to-tiled memory address translation area start address mask register 1	LTAM1	H'0000000000000000	32
H'FF800130	Linear-to-tiled memory address translation control register 2	LTC2	H'0000000000000000	32

Address	Register Name	Abbr.	Initial Value	Access Size
H'FF800138	Linear-to-tiled memory address translation area start address register 2	LTAD2	H'0000000000000000	32
H'FF800140	Linear-to-tiled memory address translation area start address mask register 2	LTAM2	H'0000000000000000	32
H'FF800148	Linear-to-tiled memory address translation control register 3	LTC3	H'0000000000000000	32
H'FF800150	Linear-to-tiled memory address translation area start address register 3	LTAD3	H'0000000000000000	32
H'FF800158	Linear-to-tiled memory address translation area start address mask register 3	LTAM3	H'0000000000000000	32
H'FF800160	Linear-to-tiled memory address translation control register 4	LTC4	H'0000000000000000	32
H'FF800168	Linear-to-tiled memory address translation area start address register 4	LTAD4	H'0000000000000000	32
H'FF800170	Linear-to-tiled memory address translation area start address mask register 4	LTAM4	H'0000000000000000	32
H'FF800178	Linear-to-tiled memory address translation control register 5	LTC5	H'0000000000000000	32
H'FF800180	Linear-to-tiled memory address translation area start address register 5	LTAD5	H'0000000000000000	32
H'FF800188	Linear-to-tiled memory address translation area start address mask register 5	LTAM5	H'0000000000000000	32
H'FF800190	Linear-to-tiled memory address translation control register 6	LTC6	H'0000000000000000	32
H'FF800198	Linear-to-tiled memory address translation area start address register 6	LTAD6	H'0000000000000000	32
H'FF8001A0	Linear-to-tiled memory address translation area start address mask register 6	LTAM6	H'0000000000000000	32

Address	Register Name	Abbr.	Initial Value	Access Size
H'FF8001A8	Linear-to-tiled memory address translation control register 7	LTC7	H'0000000000000000	32
H'FF8001B0	Linear-to-tiled memory address translation area start address register 7	LTAD7	H'0000000000000000	32
H'FF8001B8	Linear-to-tiled memory address translation area start address mask register 7	LTAM7	H'0000000000000000	32
H'FF800218	Request mask setting register	RQM	H'0000000000000000	32
H'FF801000	Bus control register	BCR	H'0000000038000000	32
H'FF802000	CS0 bus control register	CS0BCR	H'0000000077777x80	32
H'FF802008	CS0 wait control register	CS0WCR	H'000000007777770F	32
H'FF802030	CS3 bus control register	CS3BCR	H'0000000077777380	32
H'FF802038	CS3 wait control register	CS3 WCR	H'000000007777770F	32

Note: Registers should not be accessed in the size other than the specified access size.

### 11.4.1 Version Control Register (VCR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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
	—	—	—	—	—	—	DRAM_SELFREF	DRAM_INACTIVE	—	—	BAD_OPF	—	—	—	ERR_SNT	—
Initial value:	0	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/W	R	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
63 to 10	—	0/1	R	Reserved  These bits always return the initial values shown in the bit map above. The write value should always be 0.
9	DRAM_SELFREF	0	R/W	This bit is set to 1 when a data block area is accessed while self-refresh mode is enabled by the RMODE, DRE, and DCE bits in MIM. This bit is cleared by writing 0 to it.
8	DRAM_INACTIVE	0	R/W	This bit is set to 1 when a data block area is accessed while the SDRAM controller is disabled by the DCE bit in MIM. This bit is cleared by writing 0 to it.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	BAD_OPC	0	R/W	This bit is set to 1 when a request that is not supported by the MCU is received via the SuperHyway bus. This bit is cleared by writing 0 to it.
4 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	ERR_SNT	0	R/W	This bit is set to 1 when the MCU returns an error response via the SuperHyway bus. This bit is cleared by writing 0 to it.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

## 11.4.2 Memory Interface Mode Register (MIM)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	BOMODE[1:0]	—	PCKE	—	—	—	—	—	—	—	—	—	—	SELF	RMODE	—
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/W	R	R	R	R	R	R	R	R	R	R	R/W	R

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

Bit	Bit Name	Initial Value	R/W	Description
63 to 48	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
47, 46	BOMODE1 and BOMODE0	00	R/W	These bits are readable/writable bits that select the SDRAM access mode. The MCU supports two SDRAM access modes. For details on the operation in each mode, see section 11.7.7, Bank Open Mode. 00: Bank open mode 01: Bank close mode Other than above: Setting prohibited
45	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
44	PCKE	0	R/W	Setting this bit to 1 sets the CKE pin low and places the MCU in power-down mode when the SDRAM is not accessed (in the idle state or bank active state). This function can reduce power consumption of the SDRAM.
43 to 35	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
34	SELFS	0	R	This bit indicates whether the SDRAM is in the self-refresh state. A value of 1 indicates that the SDRAM is in the self-refresh state, and a value of 0 indicates that it is not in the self-refresh state.
33	RMODE	0	R/W	This readable/writable bit specifies whether to perform auto-refreshing or self-refreshing. The bit is valid only when the DRE bit is set to 1. 0: Auto-refreshing 1: Self-refreshing
32 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	R/W	Description
27 to 16	DRI11 to DRI0	H'61A	R/W	<p><b>DRAM Refresh Interval</b></p> <p>When refreshing is valid (DRE = 1), the maximum refresh interval (for auto-refreshing) can be specified by these bits. One count is the same as the cycle of the memory clock.</p> <p>At 100-MHz operation, one count corresponds to 10 ns. The minimum settable value is H'020. If a set value is less than H'020, H'020 is added to the count value.</p> <p>The MCU has a 12-bit internal counter. When the DCE or DRE bit is 0, or the RMODE bit is 1, this counter is cleared to 0. Otherwise, this counter increments on each external clock pulse. The counter value is compared with the DRI bits, and if a match occurs, an auto-refresh request is generated in the MCU and auto-refreshing is performed. Note that the counter is cleared to 0 at a match, and then begins incrementing again. The maximum of one internally generated auto-refresh request is recorded, and if bits DCE, DRE, and RMODE are 110, respectively, an auto-refresh request is never cleared until auto-refreshing is performed. The DRE bit should be cleared to 0 before writing to the DRI bits, and set to 1 after the writing has completed. In this case, the previous written value should be set to the DRI bits.</p> <p><b>Note:</b> While the bus is released, a refresh request is generated when the counter increments up to one-half of the set value.</p>
15 to 10	—	All 0	R	<p><b>Reserved</b></p> <p>These bits are always read as 0. The write value should always be 0.</p>
9	DRE	0	R/W	<p><b>DRAM Refresh Enable</b></p> <p>This bit sets whether refresh mode is valid or invalid.</p> <p>1: Valid</p> <p>0: Invalid</p>

Bit	Bit Name	Initial Value	R/W	Description
8	ENDIAN	*	R	<p>This bit indicates whether the external data bus is operating in big endian mode or little endian mode.</p> <p>1: Big endian mode</p> <p>0: Little endian mode</p> <p>Writing to this bit is invalid.</p>
7, 6	BW1 and BW0	01	R/W	<p>Bus Width</p> <p>These bits specify the SDRAM bus width. The width is either 32 bits or 64 bits according to the setting of these bits.</p> <p>00: Setting prohibited</p> <p>01: 32 bits wide</p> <p>10: 64 bits wide</p> <p>11: Setting prohibited</p>
5 to 1	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
0	DCE	0	R/W	<p>DRAM Controller Enable</p> <p>This bit enables the SDRAM control by the MCU. When this bit is 1, the SDRAM control by the MCU is enabled. When this bit is 0, the MCU returns an error response to the request sent via the SuperHyway bus. Accordingly, the DCE bit should always be set to 1 while the SDRAM is operating.</p>

Note: \* Setting of the registers used for SDRAM control is applied to both area 1 and area 2. Individual setting for each area cannot be made.

### 11.4.3 SDRAM Control Register (SCR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	SMS[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
63 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	SMS2 to SMS0	000	R/W	<p>SDRAM Mode Select</p> <p>These bits initialize the SDRAM at a power-on or after release of a reset. By setting these bits by software, the following commands are issued. For details on the initialization procedure, see section 11.7.9, SDRAM Initialization Sequence.</p> <p>Each time this register is written, the command is issued once.</p> <p>000: Normal operation</p> <p>001: The NOP command is issued (valid only when the DCE bit in MIM is set to 1).</p> <p>010: The PALL command is issued (valid only when the DCE bit in MIM is set to 1).</p> <p>011: The CKE signal is enabled, and at the same time, the DESELECT command is issued (valid only when the DCE bit in MIM is set to 1).</p> <p>100: The CBR (auto) refresh command is issued (valid only when the DCE bit in MIM is set to 1).</p> <p>Settings other than above are prohibited. If a prohibited value is set, correct operation is not guaranteed.</p>

### 11.4.4 SDRAM Timing Register (STR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	—	—	—	—	—	—	—	WR[2:0]			RW[2:0]			SWR[1:0]	
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRFC[2:0]			SRAS[2:0]			SRP[1:0]		SRC[2:0]			SCL[2:0]			SRCD	—
Initial value:	1	1	1	1	1	1	1	1	1	1	1	0	0	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

Bit	Bit Name	Initial Value	R/W	Description
63 to 24	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
23 to 21	WR2 to WR0	111	R/W	These bits specify the minimum number of cycles from the WRITE command issuance to the READ command issuance for the SDRAM.  000: 4 cycles 001: 5 cycles 010: 6 cycles 011: 7 cycles 100: 8 cycles 101: 9 cycles 110: 10 cycles 111: 11 cycles

Bit	Bit Name	Initial Value	R/W	Description
20 to 18	RW2 to RW0	111	R/W	<p>These bits specify the minimum number of cycles from the READ command issuance to the WRITE command issuance for the SDRAM.</p> <p>000: 6 cycles  001: 7 cycles  010: 8 cycles  011: 9 cycles  100: 10 cycles  101: 11 cycles  110: 12 cycles  111: 13 cycles</p>
17, 16	SWR1 and SWR0	11	R/W	<p>These bits specify the number of cycles (Twr) from the last postamble to PRE/PALL command issuance in a write cycle.</p> <p>00: 2 cycles  01: 3 cycles  10: 4 cycles  11: 5 cycles</p>
15 to 13	SRFC2 to SRFC0	111	R/W	<p>These bits specify the number of cycles in the same bank for the following access times (Trfc).</p> <p>(1) From auto-refresh to ACT command issuance  (2) From auto-refresh to auto-refresh</p> <p>000: 8 cycles  001: 9 cycles  010: 10 cycles  011: 11 cycles  100: 12 cycles  101: 13 cycles  110: 14 cycles  111: 15 cycles</p>

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	SRAS2 to SRAS0	111	R/W	<p>These bits specify the minimum number of cycles (Tras) from ACT command issuance to PRE command issuance in the same bank.</p> <p>000: 7 cycles  001: 8 cycles  010: 9 cycles  011: 10 cycles  100: 11 cycles  101: 12 cycles  110: 13 cycles  111: 14 cycles</p>
9, 8	SRP1 and SRP0	11	R/W	<p>These bits specify the number of cycles (Trp) from PRE command issuance to ACT command issuance.</p> <p>00: 2 cycles  01: 3 cycles  10: 4 cycles  11: 5 cycles</p>
7 to 5	SRC2 to SRC0	111	R/W	<p>These bits specify the number of cycles (Trc) in the same bank for the following times.</p> <p>(1) From ACT command issuance to auto refresh  (2) From ACT command issuance to ACT command issuance</p> <p>000: 8 cycles  001: 9 cycles  010: 10 cycles  011: 11 cycles  100: 12 cycles  101: 13 cycles  110: 14 cycles  111: 15 cycles</p>

Bit	Bit Name	Initial Value	R/W	Description
4 to 2	SCL2 to SCL0	001	R/W	These bits specify the CAS latency (CL) in data read. 000: 2 cycles 001: 3 cycles Other than above: Reserved
1	SRCD	1	R/W	This bit specifies the number of cycles (Trcd) from RAS (ACT) command issuance to the CAS (READ/READA or WRITE/WRITEA) command issuance. 0: 2 cycles 1: 3 cycles
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

#### 11.4.5 SDRAM Row Attribute Register (SDRA)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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
	—	—	—	—	SPLIT[3:0]			—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R



Bit	Bit Name	Initial Value	R/W	Description
63 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	SPLIT3 to SPLIT0	0010	R/W	These bits specify the row/column configuration of the connected SDRAM. 0010: $12 \times 9$ (= $8M \times 16$ bits product or $8M \times 32$ bits product) 0100: $13 \times 9$ (= $16M \times 16$ bits product) Other than above: Setting prohibited The relation between the SPLIT bits and row/column is shown in table 11.6.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

**Table 11.6 Address Multiplexing**

- When the external bus is 32 bits wide:

External Bus	SDRAM Address BA1 BA0 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0																
32 bits	SH7764 Address A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0																
128 Mbits	Bank (2)	12	13														
(16 Mbytes)	Row (12)				11	24	23	22	21	20	19	18	17	16	15	14	
8 M × 16	Column (9)							10	9	8	7	6	5	4	3	2	
256 Mbits	Bank (2)	12	13														
(32 Mbytes)	Row (12)				11	24	23	22	21	20	19	18	17	16	15	14	
8 M × 32	Column (9)							10	9	8	7	6	5	4	3	2	
256 Mbits	Bank (2)	12	13														
(32 Mbytes)	Row (13)				11	25	24	23	22	21	20	19	18	17	16	15	14
16 M × 16	Column (9)							10	9	8	7	6	5	4	3	2	

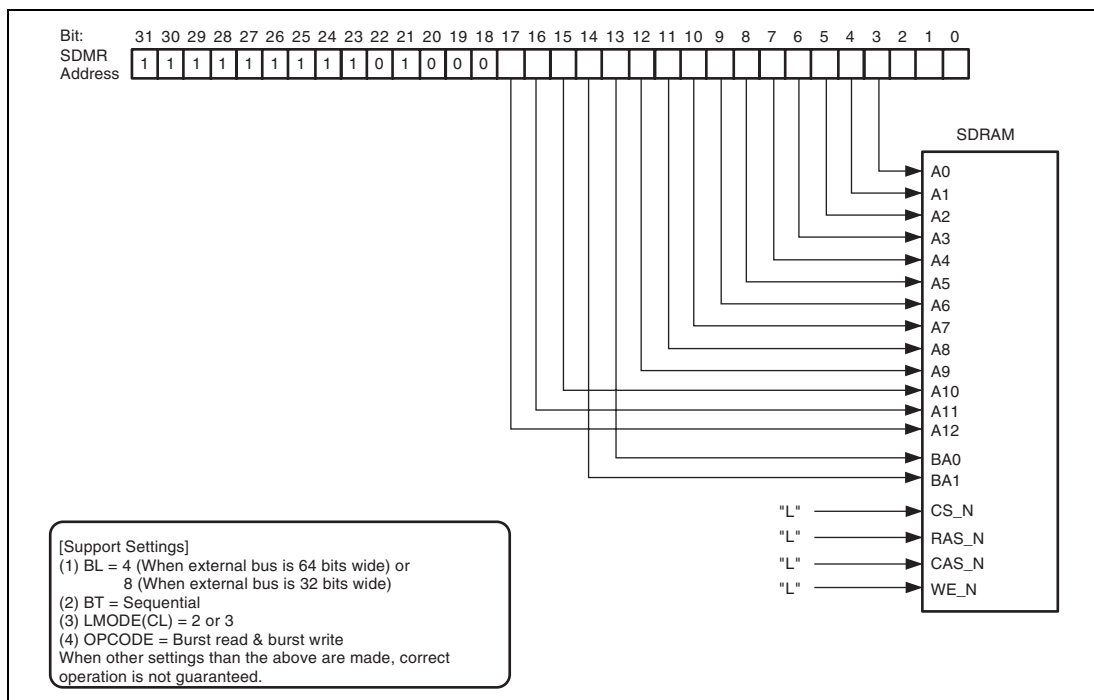
- When the external bus is 64 bits wide:

External Bus	SDRAM Address	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
64 bits	SH7764 Address	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
128 Mbits	Bank (2)	14	13													
(16 Mbytes)	Row (12)				12	25	24	23	22	21	20	19	18	17	16	15
8 M × 16	Column (9)							11	10	9	8	7	6	5	4	3
256 Mbits	Bank (2)	14	13													
(32 Mbytes)	Row (12)				12	25	24	23	22	21	20	19	18	17	16	15
8 M × 32	Column (9)							11	10	9	8	7	6	5	4	3

#### 11.4.6 SDRAM Mode Register (SDMR)

This register is used to set the SDRAM mode register.

Since SDMR is not physically contained in the MCU, reading SDMR is invalid. Only the write addresses have a meaning for the SDRAM, and the write data is ignored.



## 11.4.7 Arbitration Mode Register (AMR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	—	—	—	—	LAM[2:0]			—	—	—	—	—	—	—	SWAM
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
63 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 24	LAM2 to LAM0	100	R/W	LCDC Arbitration Select These bits set the arbitration priority level of the LCDC. 100: The arbitration priority of the LCDC is level 1 (default). 010: The arbitration priority of the LCDC is level 2. 001: The arbitration priority of the LCDC is level 3.
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	SWAM	0	R/W	SuperHyway Module Level 2 Arbitration Enable This bit sets the priority of the SuperHyway module to level 2.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	PAM7 to PAM0	H'00	R/W	Pixel Bus Module Level 2 Arbitration Enable These bits set the priority of the Pixel bus module to level 2. The following is the correspondence between bits and modules. PAM[7]: G2D (command) PAM[6]: G2D (data) PAM[5]: (reserved) PAM[4]: (reserved) PAM[3]: (reserved) PAM[2]: ATAPI PAM[1]: (reserved) PAM[0]: (reserved)

#### 11.4.8 Linear-to-Tiled Memory Address Translation Control Register (LTCn)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	LTE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
	—	—	—	LTMWX[3:0]				—	—	—	—	—	—	—	—	LTGBM
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	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
63 to 32	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
31	LTE	0	R/W	Linear-to-Tiled Memory Address Translation Enable This bit enables linear-to-tiled memory address translation to be performed in the space specified by the LTAD and LTAM registers.
30 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 9	LTMWX3 to LTMWX0	H'0	R/W	Memory Width Setting These bits specify the image area width. 0001: 512 0010: 1024 0100: 2048 1000: 4096 Other than above: Setting prohibited.
8 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	LTGBM	0	R/W	16 Bits Per Pixel Graphics Mode Enable This bit specifies the graphics mode. 0: 8 bits per pixel 1: 16 bits per pixel

Note: This register should be set while the SDRAM is not accessed by any modules; for example, during initial setting after a power on (except for auto-refreshing).

### 11.4.9 Linear-to-Tiled Memory Address Translation Area Start Address Register (LTADn)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	—	—	LTAD[8:0]									—	—	—	—
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/W	R/W	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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
63 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 20	LTAD8 to LTAD0	H'000	R/W	Linear-to-Tiled Memory Address Translation Start Address These bits specify the linear-to-tiled memory address translation start address.
19 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

**Note:** This register should be set while the SDRAM is not accessed by any modules; for example, during initial setting after a power on (except for auto-refreshing).

### 11.4.10 Linear-to-Tiled Memory Address Translation Area Start Address Mask Register (LTAMn)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	—	—	LTAM[8:0]										—	—	—
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/W	R/W	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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
63 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 20	LTAM8 to LTAM0	H'000	R/W	Linear-to-Tiled Memory Address Translation Start Address Mask These bits specify the range for comparison between the LTAD bits and a real address.
19 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

In the LTAM8 to LTAM0 bits, specify a physical address in the unified memory. The data should contain contiguous 1s from the left. So, one of (H'000), (H'100), H'180, H'1C0, H'1E0, H'1F0, H'1F8, H'1FC, H'1FE, and H'1FF should be specified.

Note: The unified memory space of this LSI consists of areas 1 and 2, which are 64 Mbytes each. Accordingly, (H'000) and (H'100) should not be specified.

Example:

(a) LTAD[8:0] == B'01000000

LTAM[8:0] == B'111111000

In the above case, the address space in which bits 28 to 23 of an address are B'010000 (8 Mbytes) is translated to tiled memory space.

(b) LTAD[8:0] == B'001010101

LTAM[8:0] == B'111111100

In the above case, the address space in which bits 28 to 22 of an address are B'0010101 (4 Mbytes) is translated to tiled memory space.

(c) LTAM[8:0] == B'111111111

In the above case, an address space of 1 Mbyte is translated to tiled memory space.

(d) LTAM[8:0] == B'110000000

In the above case, an address space of 128 Mbytes (whole space) is translated to tiled memory space.

Note that, this register should be set while the SDRAM is not accessed by any modules; for example, during initial setting after a power on (except for auto-refreshing).

#### 11.4.11 Request Mask Setting Register (RQM)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	—	—	—	—	—	—	NMIME	—	—	—	—	—	—	—	LCDM
Initial value:	0	0	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	R	R/W

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



Bit	Bit Name	Initial Value	R/W	Description
63 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	NMIME	0	R/W	Request Mask Enable during NMI 0: Main memory access request is not masked when an NMI occurs. 1: Main memory access request is masked when an NMI occurs.
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	LCDM	0	R/W	LCDC Request Mask Enable 0: LCDC request is not masked. 1: LCDC request is masked.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	VDCM	0	R/W	VDC2 Request Mask Enable 0: VDC2 request is not masked. 1: VDC2 request is masked.
9 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	2DDM	0	R/W	2DD (G2D Data) Request Mask Enable 0: 2DDM request is not masked. 1: 2DDM request is masked.
5	2DCM	0	R/W	2DC (G2D Command) Request Mask Enable 0: 2DCM request is not masked. 1: 2DCM request is masked.
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	ATAM	0	R/W	ATAPI Request Mask Enable 0: ATAPI request is not masked. 1: ATAPI request is masked.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Arbitration of memory access can be masked for each module during NMI. The setting of this register is reflected in the arbitrating operation. Accordingly, it is not applied to the memory access in progress.

## 11.4.12 Bus Control Register (BCR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	—	IRSD[2:0]			DPUP	—	OPUP	—	—	—	—	BREQEN	—	—	—
Initial value:	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	IPUP	—	—	—	—	—	—	ASYNC1	ASYNC0
Initial value:	0	0	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	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
63 to 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29 to 27	IRSD2 to IRSD0	111	R/W	Idle Cycles between SRAM Access and SDRAM Access These bits specify the number of idle cycles to be inserted between SRAM access (areas 0 and 3) and SDRAM access (areas 1 and 2). 000: 4 idle cycles 001: 5 idle cycles 010: 6 idle cycles 011: 7 idle cycles 100: 8 idle cycles 101: 9 idle cycles 110: 10 idle cycles 111: 11 idle cycles

Bit	Bit Name	Initial Value	R/W	Description
26	DPUP	0	R/W	<p>Data Pin Pull-Up Resistor Control</p> <p>This bit specifies the state of the pull-up resistors on the data pins (D63 to D0). This bit is initialized by a power-on reset.</p> <p>0: The pull-up resistors on the data pins (D63 to D0) are turned on in some cycles before or after memory access.</p> <p>1: The pull-up resistors on the data pins (D63 to D0) are off.</p> <p>Note: If a data pin needs to be pulled up, use of an external pull-up resistor is recommended.</p>
25	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
24	OPUP	0	R/W	<p>Control Output Pin Pull-Up Resistor Control</p> <p>This bit specifies the state of the pull-up resistors on the pins A25 to A0, <math>\overline{BS}</math>, <math>\overline{CSn}</math>, RD, <math>\overline{WEn/DQMn}</math>, RD/<math>\overline{WR}</math>, <math>\overline{RAS}</math>, and <math>\overline{CAS}</math> when these pins are in Hi-Z state. This bit is initialized by a power-on reset.</p> <p>0: The pull-up resistors on the control output pins are on.</p> <p>1: The pull-up resistors on the control output pins are off.</p>
23 to 20	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
19	BREQEN	0	R/W	<p><math>\overline{BREQ}</math> Enable</p> <p>This bit specifies whether or not an external request can be accepted. This bit is initialized by a power-on reset.</p> <p>0: An external request is not accepted.</p> <p>1: An external request is accepted.</p>
18 to 9	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
8	IPUP	0	R/W	<p>Input Pin Pull-Up Resistor Control</p> <p>This bit specifies the state of the pull-up resistors on the input pins (<math>\overline{\text{RDY}}</math> and <math>\overline{\text{BREQ}}</math>). This bit is initialized by a power-on reset.</p> <p>0: The pull-up resistors on the input pins (<math>\overline{\text{RDY}}</math> and <math>\overline{\text{BREQ}}</math>) are on.</p> <p>1: The pull-up resistors on the input pins (<math>\overline{\text{RDY}}</math> and <math>\overline{\text{BREQ}}</math>) are off.</p>
7 to 2	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
1	ASYNC1	0	R/W	<p>Asynchronous Input 1</p> <p>This register enables asynchronous input to the <math>\overline{\text{BREQ}}</math> pin.</p> <p>0: The <math>\overline{\text{BREQ}}</math> input signal is synchronized with CLKOUT.</p> <p>1: The <math>\overline{\text{BREQ}}</math> input signal is asynchronous to CLKOUT.</p>
0	ASYNC0	0	R/W	<p>Asynchronous Input 0</p> <p>This register enables asynchronous input to the <math>\overline{\text{RDY}}</math> pin.</p> <p>0: The <math>\overline{\text{RDY}}</math> input signal is synchronized with CLKOUT.</p> <p>1: The <math>\overline{\text{RDY}}</math> input signal is asynchronous to CLKOUT.</p>

## 11.4.13 CS0 Bus Control Register (CS0BCR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	IWW[2:0]			—	IWRWD[2:0]			—	IWRWS[2:0]			—	IWRRD[2:0]		
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[2:0]			—	—	SZ[1:0]		RDSPL	—	—	—	—	—	—	—
Initial value:	0	1	1	1	0	0	Undefined	Undefined	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
63 to 31	—	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
30 to 28	IWW2 to IWW0	111	R/W	<p>Idle Cycles between Write and Read/Write and Write Access Cycles</p> <p>These bits specify the number of idle cycles to be inserted after a write access to the memory connected to the SRAM area (areas 0 and 3).</p> <p>The idle cycles specified in these bits are inserted between write and read cycles or between write and write cycles, and at the same time, inserted between accesses to area 0 and area 0 or between accesses to area 0 and area 3.</p> <p>000: No idle cycles 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</p>
27	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
26 to 24	IWRWD2 to IWRWD0	111	R/W	<p>Idle Cycles between Read and Write Access Cycles to Different Areas</p> <p>These bits specify the number of idle cycles to be inserted after a read access to the memory connected to area 0.</p> <p>The idle cycles specified in these bits are inserted between a read access cycle to area 0 and a write access cycle to area 3.</p> <p>000: No idle cycles 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</p>
23	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
22 to 20	IWRWS2 to IWRWS0	111	R/W	<p>Idle Cycles between Read and Write Access Cycles to Same Area (Area 0)</p> <p>These bits specify the number of idle cycles to be inserted after a read access to the memory connected to area 0.</p> <p>The idle cycles specified in these bits are inserted between consecutive read and write access cycles to the same area (area 0).</p> <p>000: No idle cycles 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</p>



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	IWRRD2 to IWRRD0	111	R/W	Idle Cycles between Read and Read Access Cycles to Different Area These bits specify the number of idle cycles to be inserted after a read access to the memory connected to area 0. The idle cycles specified in these bits are inserted between a read access cycle to area 0 and a read access cycle to area 3. 000: No idle cycles 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.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	IWRRS2 to IWRRS0	111	R/W	<p>Idle Cycles between Read and Read Access Cycles to Same Area (Area 0)</p> <p>These bits specify the number of idle cycles to be inserted after a read access to the memory connected to area 0.</p> <p>The idle cycles specified in these bits are inserted between consecutive read and read access cycles to the same area (area 0).</p> <p>000: No idle cycles 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</p>
11, 10	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
9, 8	SZ1 and SZ0	Undefined	R	<p>Bus Width</p> <p>The external pins (MODE4 and MODE3) for specifying the bus width are sampled at a power-on reset.</p> <p>00: Reserved  01: 8 bits  10: 16 bits  11: 32 bits</p>

Bit	Bit Name	Initial Value	R/W	Description
7	RDSPL	1	R/W	<p><math>\overline{RD}</math> Hold Cycle</p> <p>This bit specifies the number of cycles to be inserted into the <math>\overline{RD}</math> assertion period to ensure the data hold time to the read data sample timing. When this bit is set to 1, the number of delay cycles between the <math>\overline{RD}</math> negation and the <math>\overline{CS0}</math> negation should be set to 1 or more by setting the RDH bits in CS0WCR.</p> <p>Note that, by setting this bit to 1, the number of delay cycles between the <math>\overline{RD}</math> negation and the <math>\overline{CS0}</math> negation is reduced by 1.</p> <p>0: No hold cycles inserted 1: 1 hold cycle inserted</p>
6 to 0	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

#### 11.4.14 $\overline{CSn}$ Wait Control Register (CSnWCR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	ADS[2:0]			—	ADH[2:0]			—	RDS[2:0]			—	RDH[2:0]		
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[2:0]			—	WTH[2:0]			—	BSH[2:0]			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	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
63 to 31	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
30 to 28	ADS2 to ADS0	111	R/W	Address Setup Cycles These bits specify the number of cycles to be inserted to ensure the address setup time to the CSn assertion. 000: No cycles 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.
26 to 24	ADH2 to ADH0	111	R/W	Address Hold Cycles These bits specify the number of cycles to be inserted to ensure the address hold time to the CSn negation. 000: No cycles 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.

Bit	Bit Name	Initial Value	R/W	Description
22 to 20	RDS2 to RDS0	111	R/W	<p><math>\overline{\text{CSn}}</math> Assert–<math>\overline{\text{RD}}</math> Assert Delay Cycle</p> <p>These bits specify the number of cycles to be inserted between the <math>\overline{\text{CSn}}</math> assertion and the <math>\overline{\text{RD}}</math> assertion.</p> <p>000: No cycles inserted (delay of 1 cycle)</p> <p>001: 1 cycle inserted (delay of 2 cycles)</p> <p>010: 2 cycles inserted (delay of 3 cycles)</p> <p>011: 3 cycles inserted (delay of 4 cycles)</p> <p>100: 4 cycles inserted (delay of 5 cycles)</p> <p>101: 5 cycles inserted (delay of 6 cycles)</p> <p>110: 6 cycles inserted (delay of 7 cycles)</p> <p>111: 7 cycles inserted (delay of 8 cycles)</p>
19	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
18 to 16	RDH2 to RDH0	111	R/W	<p><math>\overline{\text{RD}}</math> Negate–<math>\overline{\text{CSn}}</math> Negate Delay Cycle</p> <p>These bits specify the number of cycles to be inserted between the <math>\overline{\text{RD}}</math> negation and the <math>\overline{\text{CSn}}</math> negation.</p> <p>000: No cycles inserted (delay of 0 cycles)</p> <p>001: 1 cycle inserted (delay of 1 cycle)</p> <p>010: 2 cycles inserted (delay of 2 cycles)</p> <p>011: 3 cycles inserted (delay of 3 cycles)</p> <p>100: 4 cycles inserted (delay of 4 cycles)</p> <p>101: 5 cycles inserted (delay of 5 cycles)</p> <p>110: 6 cycles inserted (delay of 6 cycles)</p> <p>111: 7 cycles inserted (delay of 7 cycles)</p>
15	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	WTS2 to WTS0	111	R/W	<p><math>\overline{\text{CSn}}</math> Assert–<math>\overline{\text{WEn}}</math> Assert Delay Cycle</p> <p>These bits specify the number of cycles to be inserted between the <math>\overline{\text{CSn}}</math> assertion and the <math>\overline{\text{WEn}}</math> assertion.</p> <p>000: No cycles inserted (delay of 0.5 cycle)</p> <p>001: 1 cycle inserted (delay of 1.5 cycles)</p> <p>010: 2 cycles inserted (delay of 2.5 cycles)</p> <p>011: 3 cycles inserted (delay of 3.5 cycles)</p> <p>100: 4 cycles inserted (delay of 4.5 cycles)</p> <p>101: 5 cycles inserted (delay of 5.5 cycles)</p> <p>110: 6 cycles inserted (delay of 6.5 cycles)</p> <p>111: 7 cycles inserted (delay of 7.5 cycles)</p>
11	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
10 to 8	WTH2 to WTH0	111	R/W	<p><math>\overline{\text{WEn}}</math> Negate–<math>\overline{\text{CSn}}</math> Negate Delay Cycle</p> <p>These bits specify the number of cycles to be inserted between the <math>\overline{\text{WEn}}</math> negation and the <math>\overline{\text{CSn}}</math> negation.</p> <p>000: No cycles inserted (delay of 0.5 cycle)</p> <p>001: 1 cycle inserted (delay of 1.5 cycles)</p> <p>010: 2 cycles inserted (delay of 2.5 cycles)</p> <p>011: 3 cycles inserted (delay of 3.5 cycles)</p> <p>100: 4 cycles inserted (delay of 4.5 cycles)</p> <p>101: 5 cycles inserted (delay of 5.5 cycles)</p> <p>110: 6 cycles inserted (delay of 6.5 cycles)</p> <p>111: 7 cycles inserted (delay of 7.5 cycles)</p>
7	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	BSH2 to BSH0	000	R/W	<p>These bits specify the number of cycles to be inserted to elongate the <math>\overline{BS}</math> assertion time during an access to the <math>\overline{CSn}</math> space.</p> <p>Cycle insertion is enabled when a value other than 000 is set in the RDS and WTS bits in CSnWCR in reading and writing, respectively. The total number of access cycles is not changed by the setting of these bits.</p> <p>000: 1 cycle inserted for the <math>\overline{BS}</math> assertion  001: 2 cycles inserted for the <math>\overline{BS}</math> assertion  010: Setting prohibited  011: Setting prohibited  100: Setting prohibited  101: Setting prohibited  110: Setting prohibited  111: Setting prohibited</p>

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	IW3 to IW0	1111	R/W	<p>These bits specify the number of wait cycles to be inserted during an access to the <math>\overline{\text{CSn}}</math> space.</p> <p>When 0000 is set (wait cycles are not inserted), external wait insertion using the RDY pin is disabled.</p> <p>0000: No wait cycles inserted</p> <p>0001: 1 wait cycle inserted</p> <p>0010: 2 wait cycles inserted</p> <p>0011: 3 wait cycles inserted</p> <p>0100: 4 wait cycles inserted</p> <p>0101: 5 wait cycles inserted</p> <p>0110: 6 wait cycles inserted</p> <p>0111: 7 wait cycles inserted</p> <p>1000: 8 wait cycles inserted</p> <p>1001: 9 wait cycles inserted</p> <p>1010: 11 wait cycles inserted</p> <p>1011: 13 wait cycles inserted</p> <p>1100: 15 wait cycles inserted</p> <p>1101: 17 wait cycles inserted</p> <p>1110: 21 wait cycles inserted</p> <p>1111: 25 wait cycles inserted</p>



### 11.4.15 CS3 Bus Control Register (CS3BCR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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:	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
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
	—	IWW[2:0]			—	IWRWD[2:0]			—	IWRWS[2:0]			—	IWRRD[2:0]		
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[2:0]			—	—	SZ[1:0]		RDSPL	—	—	—	—	—	—	—
Initial value:	0	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
63 to 31	—	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
30 to 28	IWW2 to IWW0	111	R/W	<p>Idle Cycles between Write and Read/Write and Write Access Cycles</p> <p>These bits specify the number of idle cycles to be inserted after a write access to the memory connected to the SRAM area (areas 0 and 3).</p> <p>The idle cycles specified in these bits are inserted between write and read cycles or between write and write cycles, and at the same time, inserted between accesses to area 3 and area 0 or between accesses to area 3 and area 3.</p> <p>000: No idle cycles 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</p>
27	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
26 to 24	IWRWD2 to IWRWD0	111	R/W	<p>Idle Cycles between Read and Write Access Cycles to Different Areas</p> <p>These bits specify the number of idle cycles to be inserted after a read access to the memory connected to area 3.</p> <p>The idle cycles specified in these bits are inserted between a read cycle for area 3 and a write cycle for area 0.</p> <p>000: No idle cycles 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</p>
23	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
22 to 20	IWRWS2 to IWRWS0	111	R/W	<p>Idle Cycles between Read and Write Access Cycles to Same Area (Area 3)</p> <p>These bits specify the number of idle cycles to be inserted after a read access to the memory connected to area 3.</p> <p>The idle cycles specified in these bits are inserted between consecutive read and write access cycles to the same area (area 3).</p> <p>000: No idle cycles 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</p>

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	IWRRD2 to IWRRD0	111	R/W	Idle Cycles between Read and Read Access Cycles to Different Area These bits specify the number of idle cycles to be inserted after a read access to the memory connected to area 3. The idle cycles specified in these bits are inserted between a read access cycle to area 3 and a read access cycle to area 0. 000: No idle cycles 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.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	IWRRS2 to IWRRS0	111	R/W	<p>Idle Cycles between Read and Read Access Cycles to Same Area (Area 3)</p> <p>These bits specify the number of idle cycles to be inserted after a read access to the memory connected to area 3.</p> <p>The idle cycles specified in these bits are inserted between consecutive read and read access cycles to the same area (area 3).</p> <p>000: No idle cycles 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</p>
11, 10	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
9, 8	SZ1 and SZ0	11	R/W	<p>Bus Width</p> <p>These bits specify the bus width of area 3.</p> <p>00: Reserved  01: 8 bits  10: 16 bits  11: 32 bits</p>

Bit	Bit Name	Initial Value	R/W	Description
7	RDSPL	1	R/W	<p><math>\overline{RD}</math> Hold Cycle</p> <p>This bit specifies the number of cycles to be inserted into the <math>\overline{RD}</math> assertion period to ensure the data hold time to the read data sample timing. When this bit is set to 1, the number of delay cycles between the <math>\overline{RD}</math> negation and the <math>\overline{CS3}</math> negation should be set to 1 or more by setting the RDH bits in CS3WCR.</p> <p>Note that, by setting this bit to 1, the number of delay cycles between the <math>\overline{RD}</math> negation and the <math>\overline{CS3}</math> negation is reduced by 1.</p> <p>0: No hold cycles inserted 1: 1 hold cycle inserted</p>
6 to 0	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

## 11.5 Operation

### 11.5.1 Endian/Access Size and Data Alignment

This LSI supports both big-endian mode, in which the upper byte (MSByte) in a string of byte data is at address 0, and little-endian mode, in which the lower byte (LSByte) in a string of byte data is at address 0. The mode is specified by the external pin (MODE5 pin) at a power-on reset through the RESET pin. At a power-on reset by PRESET, big-endian mode is specified when the MD5 pin is low, and little-endian mode is specified when the MD5 pin is high.

A data bus width of 8, 16, or 32 bits can be selected for the normal memory interface (areas 0 and 3), and one of 32 or 64 bits can be selected for the SDRAM interface (areas 1 and 2). Data alignment is carried out according to the data bus width and endian mode of each device. Accordingly, 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 at the area with an 8-bit width in the SRAM interface, each address is incremented one by one, and then access is performed four times. In the 32-byte transfer, a total of 32-byte data is continuously transferred 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 wrap around method according to the set bus width. The bus is not released during these transfers. In this LSI, data alignment and data length conversion between different interfaces is performed automatically. (The accesses from the pixel bus and LCDC are not performed in the wrap around method.)

The relationship between the endian mode, device data length, and access unit are shown in tables 11.7 to 11.16.

**Table 11.7 32-Bit External Device/Big-Endian Access and Data Alignment (Areas 0 and 3)**

Operation			Data Bus				Strobe Signal			
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	—	—	—	Assert			
	4n + 1	1	—	Data 7 to 0	—	—		Assert		
	4n + 2	1	—	—	Data 7 to 0	—			Assert	
	4n + 3	1	—	—	—	Data 7 to 0				Assert
Word	4n	1	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert		
	4n + 2	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

**Table 11.8 16-Bit External Device/Big-Endian Access and Data Alignment (Areas 0 and 3)**

Operation			Data Bus				Strobe Signal			
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	—			Assert	
	2n + 1	1	—	—	—	Data 7 to 0				Assert
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword	4n	1	—	—	Data 31 to 24	Data 23 to 16			Assert	Assert
	4n + 2	2	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert



**Table 11.9 8-Bit External Device/Big-Endian Access and Data Alignment (Areas 0 and 3)**

Operation			Data Bus				Strobe Signal			
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				Assert
Word	2n	1	—	—	—	Data 15 to 8				Assert
	2n + 1	2	—	—	—	Data 7 to 0				Assert
Longword	4n	1	—	—	—	Data 31 to 24				Assert
	4n + 1	2	—	—	—	Data 23 to 16				Assert
	4n + 2	3	—	—	—	Data 15 to 8				Assert
	4n + 3	4	—	—	—	Data 7 to 0				Assert

**Table 11.10 32-Bit External Device/Little-Endian Access and Data Alignment  
(Areas 0 and 3)**

Operation			Data Bus				Strobe Signal			
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				Assert
	4n + 1	1	—	—	Data 7 to 0	—			Assert	
	4n + 2	1	—	Data 7 to 0	—	—		Assert		
	4n + 3	1	Data 7 to 0	—	—	—	Assert			
Word	4n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
	4n + 2	1	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert		
Longword	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

**Table 11.11 16-Bit External Device/Little-Endian Access and Data Alignment  
(Areas 0 and 3)**

Operation			Data Bus				Strobe Signal			
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				Assert
	2n + 1	1	—	—	Data 7 to 0	—			Assert	
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword	4n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
	4n + 2	2	—	—	Data 31 to 24	Data 23 to 16			Assert	Assert

**Table 11.12 8-Bit External Device/Little-Endian Access and Data Alignment (Areas 0 and 3)**

Operation			Data Bus				Strobe Signal			
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				Assert
Word	2n	1	—	—	—	Data 7 to 0				Assert
	2n + 1	2	—	—	—	Data 15 to 8				Assert
Longword	4n	1	—	—	—	Data 7 to 0				Assert
	4n + 1	2	—	—	—	Data 15 to 8				Assert
	4n + 2	3	—	—	—	Data 23 to 16				Assert
	4n + 3	4	—	—	—	Data 31 to 24				Assert

**Table 11.13 32-Bit External Device/Big-Endian Access and Data Alignment (Areas 1 and 2)**

	D63 to D56	D55 to D48	D47 to D40	D39 to D32	D31 to D24	D23 to D16	D15 to D8	D7 to D0
Byte access at address 0					Data 7 to 0			
Byte access at address 1						Data 7 to 0		
Byte access at address 2							Data 7 to 0	
Byte access at address 3								Data 7 to 0
Byte access at address 4					Data 7 to 0			
Byte access at address 5						Data 7 to 0		
Byte access at address 6							Data 7 to 0	
Byte access at address 7								Data 7 to 0
Word access at address 0					Data 15 to 8	Data 7 to 0		
Word access at address 2							Data 15 to 8	Data 7 to 0
Word access at address 4					Data 15 to 8	Data 7 to 0		
Word access at address 6							Data 15 to 8	Data 7 to 0
Longword access at address 0					Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Longword access at address 4					Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Quadword access at address 0 (first time: address 0)					Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32
Quadword access at address 4 (second time: address 4)					Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0

**Table 11.14 32-Bit External Device/Little-Endian Access and Data Alignment  
(Areas 1 and 2)**

	D63 to D56	D55 to D48	D47 to D40	D39 to D32	D31 to D24	D23 to D16	D15 to D8	D7 to D0
Byte access at address 0								Data 7 to 0
Byte access at address 1							Data 7 to 0	
Byte access at address 2						Data 7 to 0		
Byte access at address 3					Data 7 to 0			
Byte access at address 4								Data 7 to 0
Byte access at address 5							Data 7 to 0	
Byte access at address 6						Data 7 to 0		
Byte access at address 7					Data 7 to 0			
Word access at address 0							Data 15 to 8	Data 7 to 0
Word access at address 2					Data 15 to 8	Data 7 to 0		
Word access at address 4							Data 15 to 8	Data 7 to 0
Word access at address 6					Data 15 to 8	Data 7 to 0		
Longword access at address 0					Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Longword access at address 4					Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Quadword access at address 0 (first time: address 0)					Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Quadword access at address 4 (second time: address 4)					Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32

**Table 11.15 64-Bit External Device/Big-Endian Access and Data Alignment (Areas 1 and 2)**

	D63 to D56	D55 to D48	D47 to D40	D39 to D32	D31 to D24	D23 to D16	D15 to D8	D7 to D0
Byte access at address 0	Data 7 to 0							
Byte access at address 1	Data 7 to 0							
Byte access at address 2	Data 7 to 0							
Byte access at address 3	Data 7 to 0							
Byte access at address 4	Data 7 to 0							
Byte access at address 5	Data 7 to 0							
Byte access at address 6	Data 7 to 0							
Byte access at address 7	Data 7 to 0							
Word access at address 0	Data 15 to 8	Data 7 to 0						
Word access at address 2			Data 15 to 8	Data 7 to 0				
Word access at address 4					Data 15 to 8	Data 7 to 0		
Word access at address 6							Data 15 to 8	Data 7 to 0
Longword access at address 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0				
Longword access at address 4					Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Quadword access at address 0	Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0

**Table 11.16 64-Bit External Device/Little-Endian Access and Data Alignment  
(Areas 1 and 2)**

	D63 to D56	D55 to D48	D47 to D40	D39 to D32	D31 to D24	D23 to D16	D15 to D8	D7 to D0
Byte access at address 0								Data 7 to 0
Byte access at address 1							Data 7 to 0	
Byte access at address 2						Data 7 to 0		
Byte access at address 3					Data 7 to 0			
Byte access at address 4				Data 7 to 0				
Byte access at address 5			Data 7 to 0					
Byte access at address 6		Data 7 to 0						
Byte access at address 7	Data 7 to 0							
Word access at address 0							Data 15 to 8	Data 7 to 0
Word access at address 2					Data 15 to 8	Data 7 to 0		
Word access at address 4			Data 15 to 8	Data 7 to 0				
Word access at address 6	Data 15 to 8	Data 7 to 0						
Longword access at address 0					Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Longword access at address 4	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0				
Quadword access at address 0	Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0

### 11.5.2 Data Alignment in Various Modules

The endian mode in the MCU matches that in the CPU, and either big endian or little endian can be used. Data should be aligned in each of the modules connected to the SuperHyway bus, the modules connected to the pixel bus, and the LCDC connected to the LCD bus according to their own bus width.

## 11.6 SRAM Interface

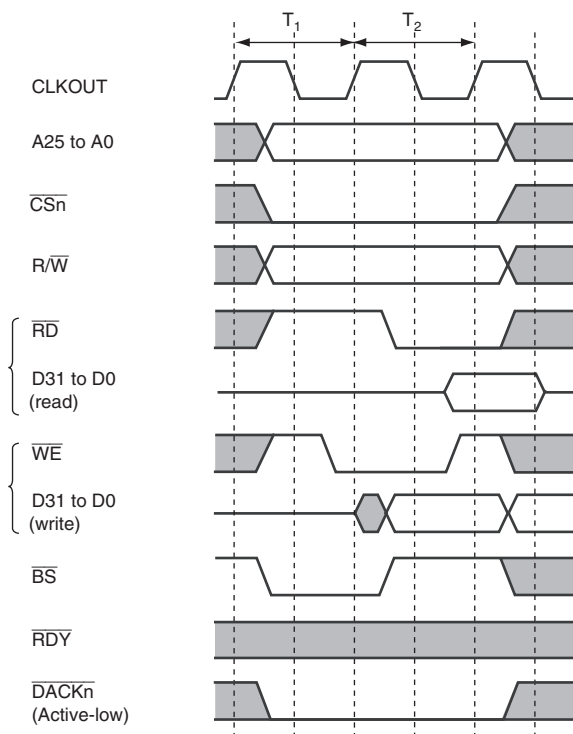
### 11.6.1 Basic Timing

The strobe signals for the SRAM interface of this LSI are output primarily based on the SRAM connection. Figure 11.4 shows the basic timing of the SRAM interface. A no-wait normal access is completed in two cycles. The  $\overline{BS}$  signal is asserted for one cycle to indicate the start of a bus cycle. The  $\overline{CS0}$  and  $\overline{CS3}$  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 the case of access at minimum pitch.

During reading, specification of an access size is not needed. 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.

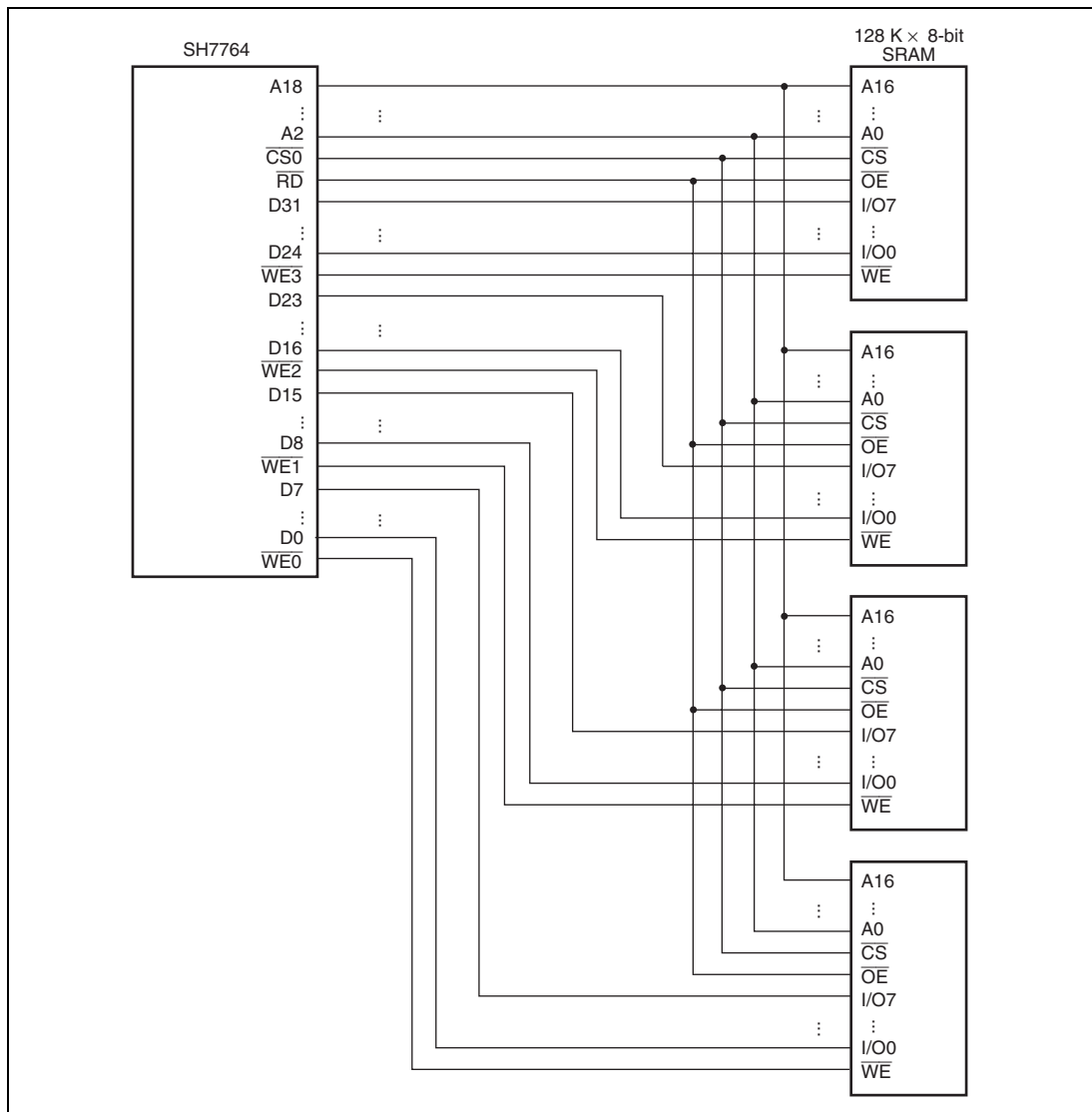
In 32-byte transfer, a total of 32 bytes are transferred continuously according to the bus width set. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wrap around method according to the set bus width. The bus is not released during this transfer.



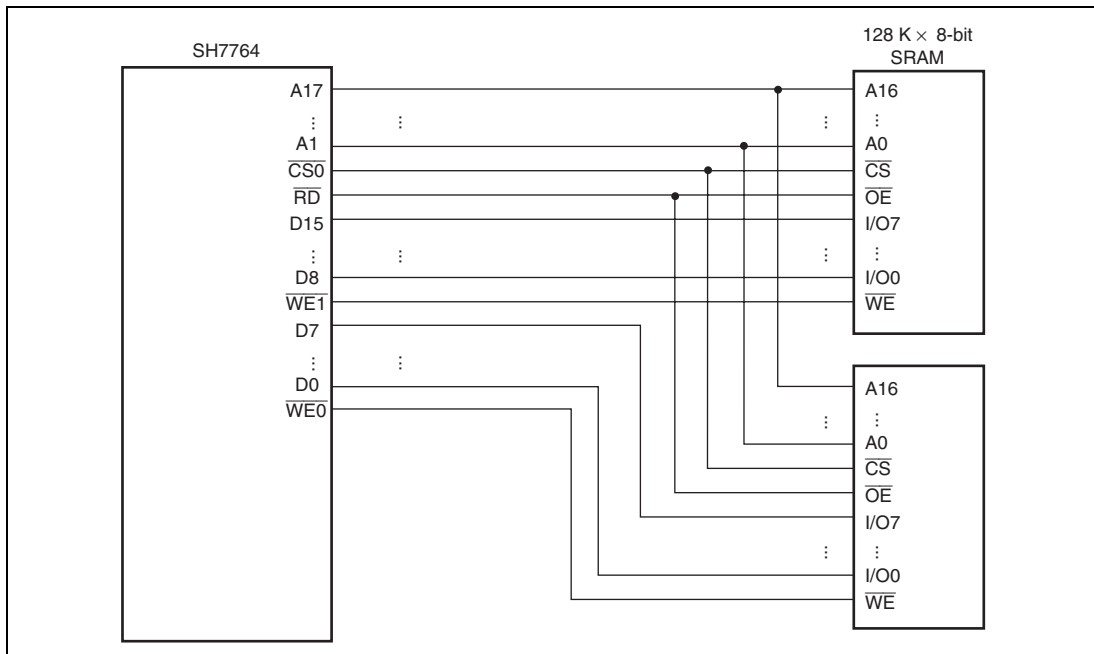


**Figure 11.3 Basic Timing of SRAM Interface**

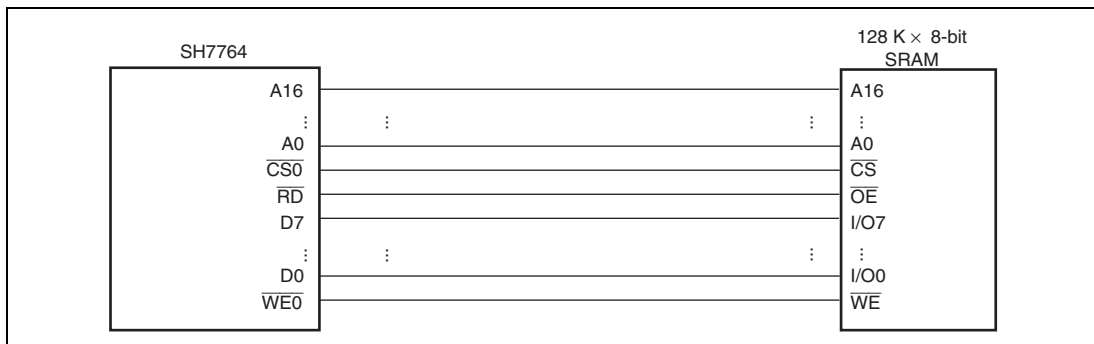
Figures 11.4 to 11.6 show examples of the connection to SRAM with data width of 32, 16, and 8 bits.



**Figure 11.4 Example of 32-Bit Data-Width SRAM Connection**



**Figure 11.5 Example of 16-Bit Data-Width SRAM Connection**

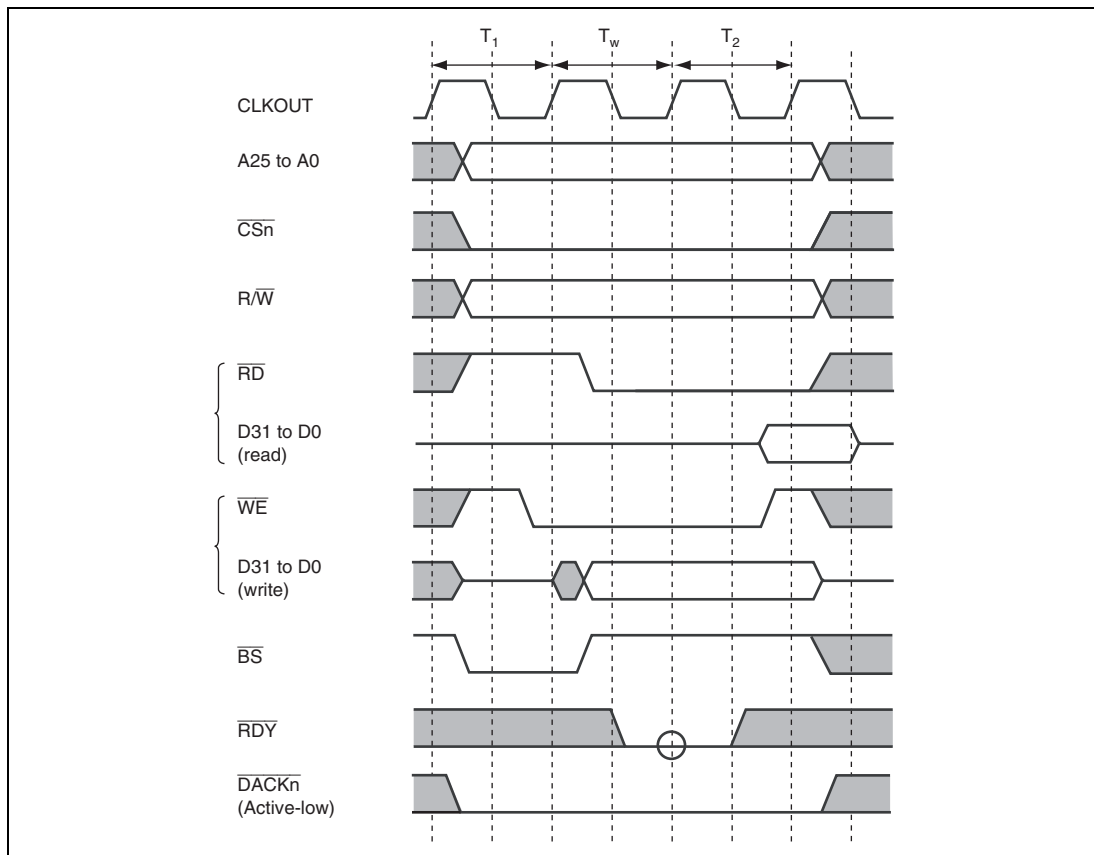


**Figure 11.6 Example of 8-Bit Data-Width SRAM Connection**

## 11.6.2 Wait Cycle Control

Wait cycle insertion for the SRAM interface can be controlled by CSnWCR. If the IW bits for each area in CSnWCR is not 0, a software wait is inserted in accordance with the wait-control bits. For details, see section 11.4.14,  $\overline{\text{CSn}}$  Wait Control Register (CSnWCR).

A specified number of  $T_w$  cycles is inserted as wait cycles in accordance with the CSnWCR setting. The insertion timing of the wait cycle is shown in figure 11.7.



**Figure 11.7 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 11.8, 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.

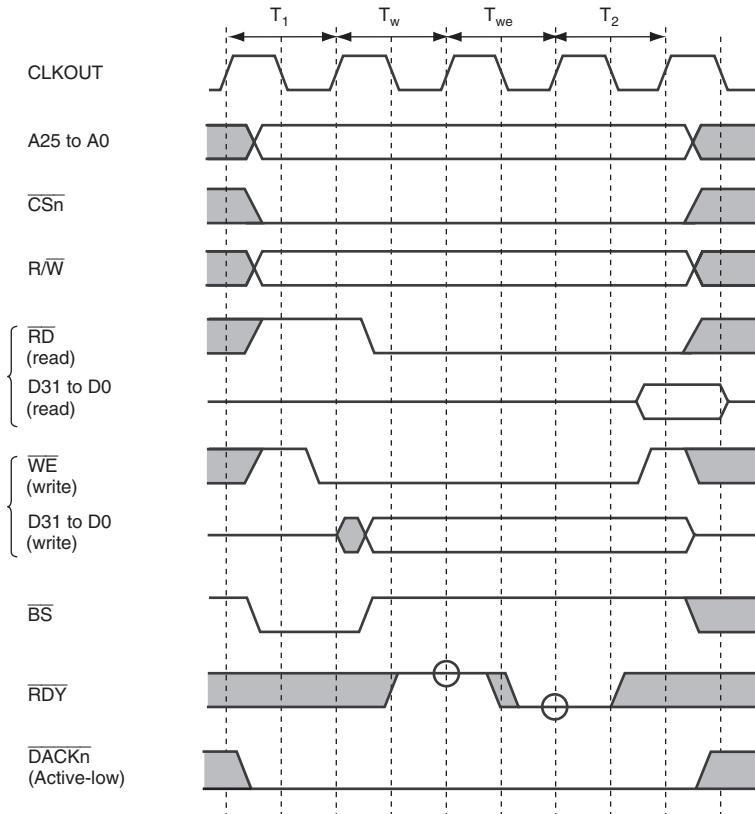
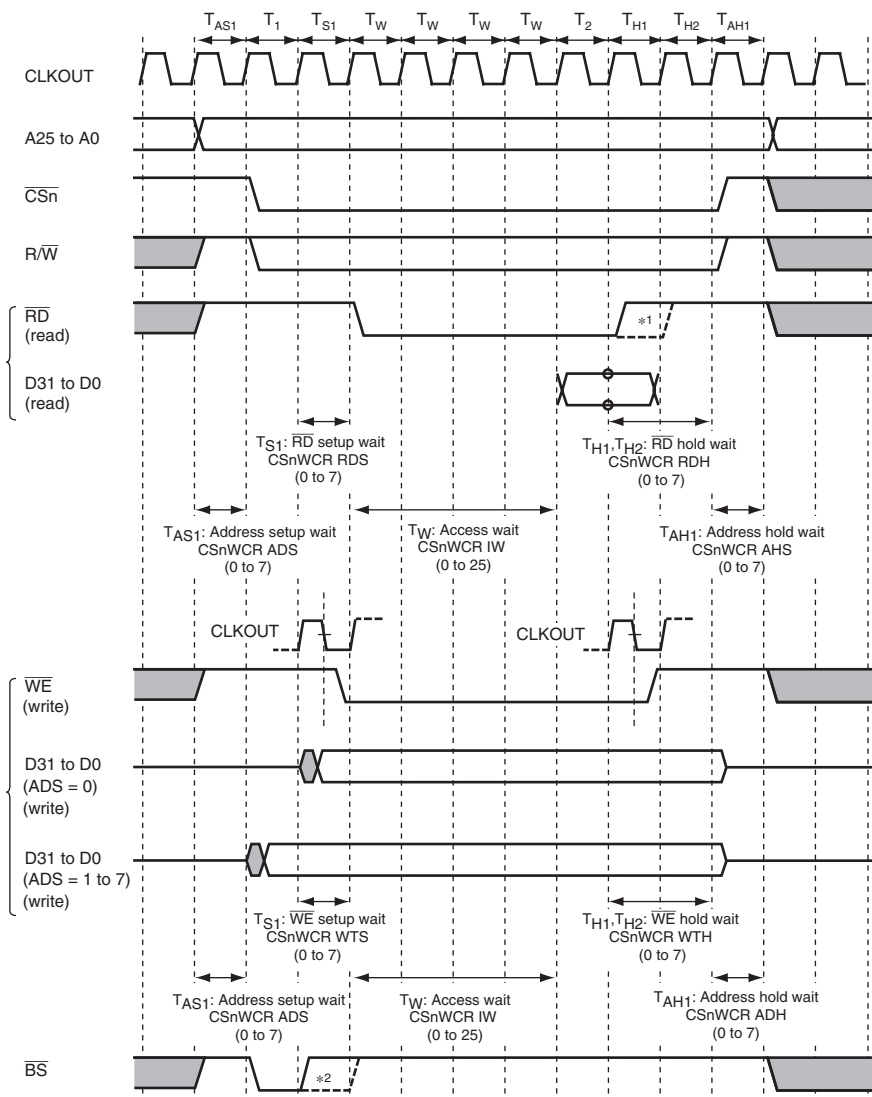


Figure 11.8 SRAM Interface Wait Cycle Timing (Wait Cycle Insertion by  $\overline{\text{RDY}}$  Signal)

### 11.6.3 Read-Strobe Negate Timing

When the SRAM interface is used, the negation timing of the strobe signal during a read operation can be specified through the RDH bit in CSnWCR.



Notes: 1. When CSnBCR RDSPL is set to 1.  
2. When CSnWCR.BSH is set to 1.

**Figure 11.9 SRAM Interface Wait State Timing (Read-Strobe Negate Timing Setting)**

## 11.7 SDRAM Interface

### 11.7.1 SDRAM Direct Connection

Since SDRAMs can be selected by the CS signals, they can be connected to physical areas 0 and 2 while sharing the control signals such as  $\overline{\text{RAS}}$ .

The control signals used for direct connection with SDRAM are:  $\overline{\text{RAS}}$ ,  $\overline{\text{CAS}}$ ,  $\text{R}/\overline{\text{W}}$ ,  $\overline{\text{CS1}}$  or  $\overline{\text{CS2}}$ ,  $\overline{\text{DQMLL}}$ ,  $\overline{\text{DQMLU}}$ ,  $\overline{\text{DQMUL}}$ ,  $\overline{\text{DQMUU}}$ ,  $\overline{\text{WE0}}$  to  $\overline{\text{WE3}}$  (for 64-bit bus) and CKE. All these signals except  $\overline{\text{CS1}}$  and  $\overline{\text{CS2}}$  are shared by all areas; they become valid and latched only when  $\overline{\text{CS1}}$  or  $\overline{\text{CS2}}$  is asserted except CKE.

This module supports burst read/write mode as the SDRAM operating mode. Data bus width can be 32 or 64 bits depending on the BW bit settings in the MIM register.

Since burst read/write access mode is used for SDRAM, 32-byte data is read even during a single read; similarly, 32-byte data is transferred even during a single write. However, none of  $\overline{\text{DQMLL}}$ ,  $\overline{\text{DQMLU}}$ ,  $\overline{\text{DQMUL}}$ ,  $\overline{\text{DQMUU}}$ , and  $\overline{\text{WE0}}$  to  $\overline{\text{WE3}}$  is asserted during unnecessary data transfer.

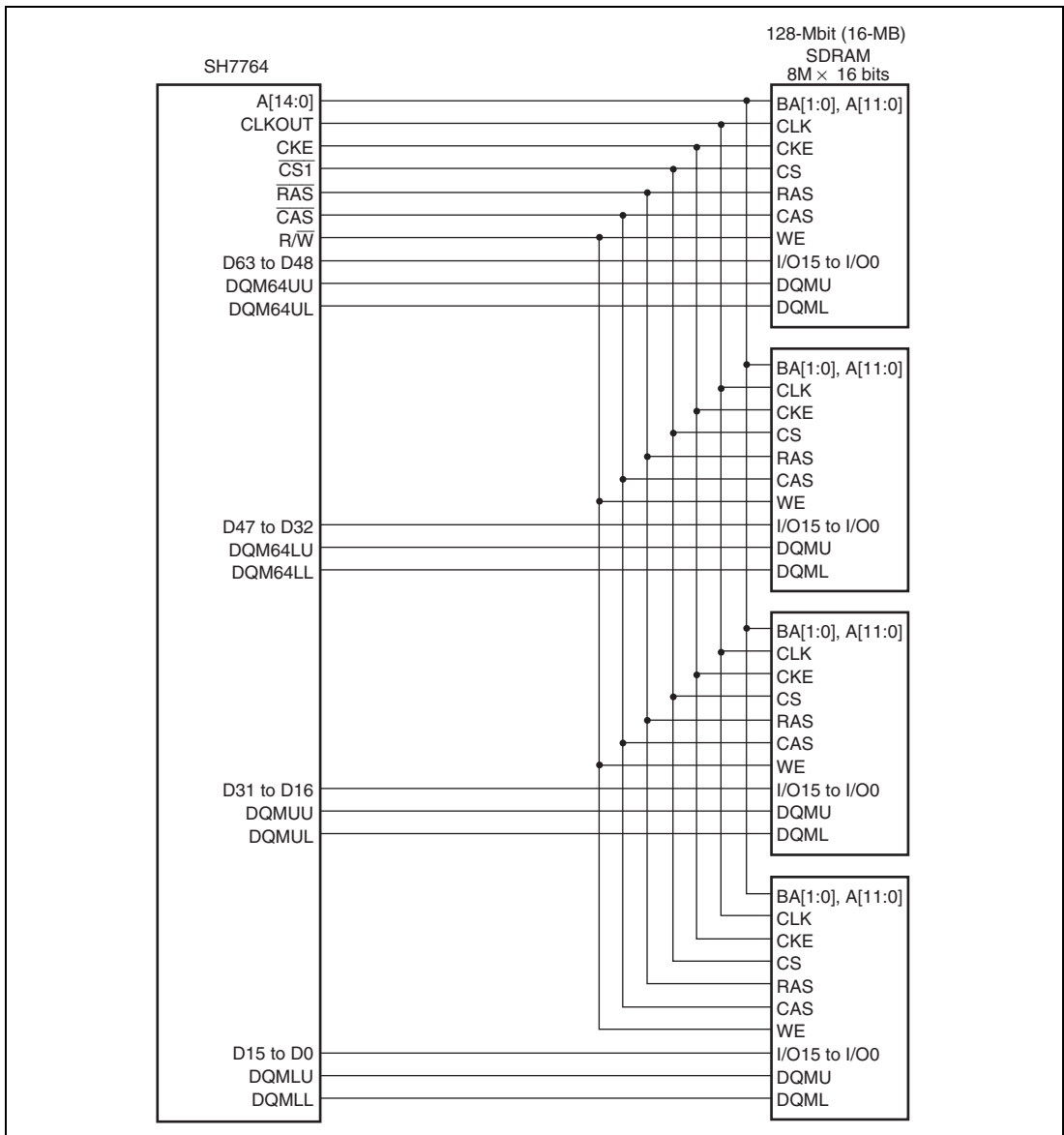
Table 11.17 shows the supported commands for SDRAM.

**Table 11.17 Supported Commands for SDRAM**

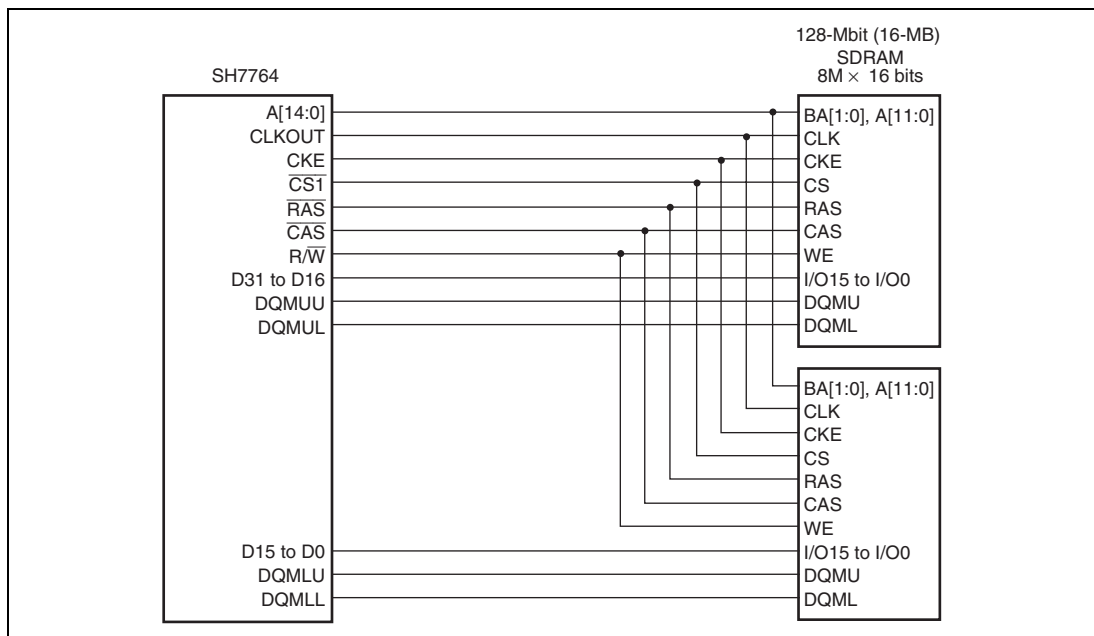
Function	Symbol	Pin									
		CKE n-1	CKE n	$\overline{\text{CSn}}$	$\overline{\text{RAS}}$	$\overline{\text{CAS}}$	$\text{R}/\overline{\text{W}}$	A [14:13]	A [12:11]	A [10]	A [9:0]
Device deselect	DESL	H	×	H	×	×	×	×	×	×	×
No operation	NOP	H	×	L	H	H	H	×	×	×	×
Read	READ	H	×	L	H	L	H	V	V	L	V
Read with auto precharge	READA	H	×	L	H	L	H	V	V	H	V
Write	WRITE	H	×	L	H	L	L	V	V	L	V
Write with auto precharge	WRITEA	H	×	L	H	L	L	V	V	H	V
Bank activate	ACT	H	×	L	L	H	H	V	V	V	V
Precharge select bank	PRE	H	×	L	L	H	L	×	×	L	×
Precharge select all bank	PALL	H	×	L	L	H	L	×	×	H	×
Auto refresh	CBR	H	H	L	L	L	H	×	×	×	×
Self-refresh entry from IDLE	SLFRSH	H	L	L	L	L	H	×	×	×	×
Self-refresh entry exit	SLFRSHX	L	H	H	×	×	×	×	×	×	×
Power-down entry	PWRDN	H	L	H	×	×	×	×	×	×	×
Power-down exit	PWRDNX	L	H	H	×	×	×	×	×	×	×
Mode register set	MRS	H	×	L	L	L	L	V	V	V	V



Figure 11.10 shows a connection example of SDRAMs with the configuration of 8 Mbytes  $\times$  16 bits when the 64-bit external bus is used. Figure 11.11 shows a similar example when the 32-bit external bus is used.



**Figure 11.10 Connection Example of Synchronous DRAMs for 64-Bit Data Bus (Area 1)**



**Figure 11.11 Connection Example of Synchronous DRAMs for 32-Bit Data Bus (Area 1)**

### 11.7.2 Address Multiplexing

Address multiplexing is performed so that the SDRAM is connected without the external address multiplexing circuit according to the setting of the BW[1:0] bits in MIM and the SPLIT[3:0] bits in SDRA. Table 11.18 shows the relationship between the bits to be multiplexed and the address bits to be output to the address pins.

Note that the address output to the A25 to A15 pins are not guaranteed.

**Table 11.18 SDRAM Bus Widths and Address Multiplexing (External Bus Width is 32 Bits)**

- When the external bus is 32 bits wide:

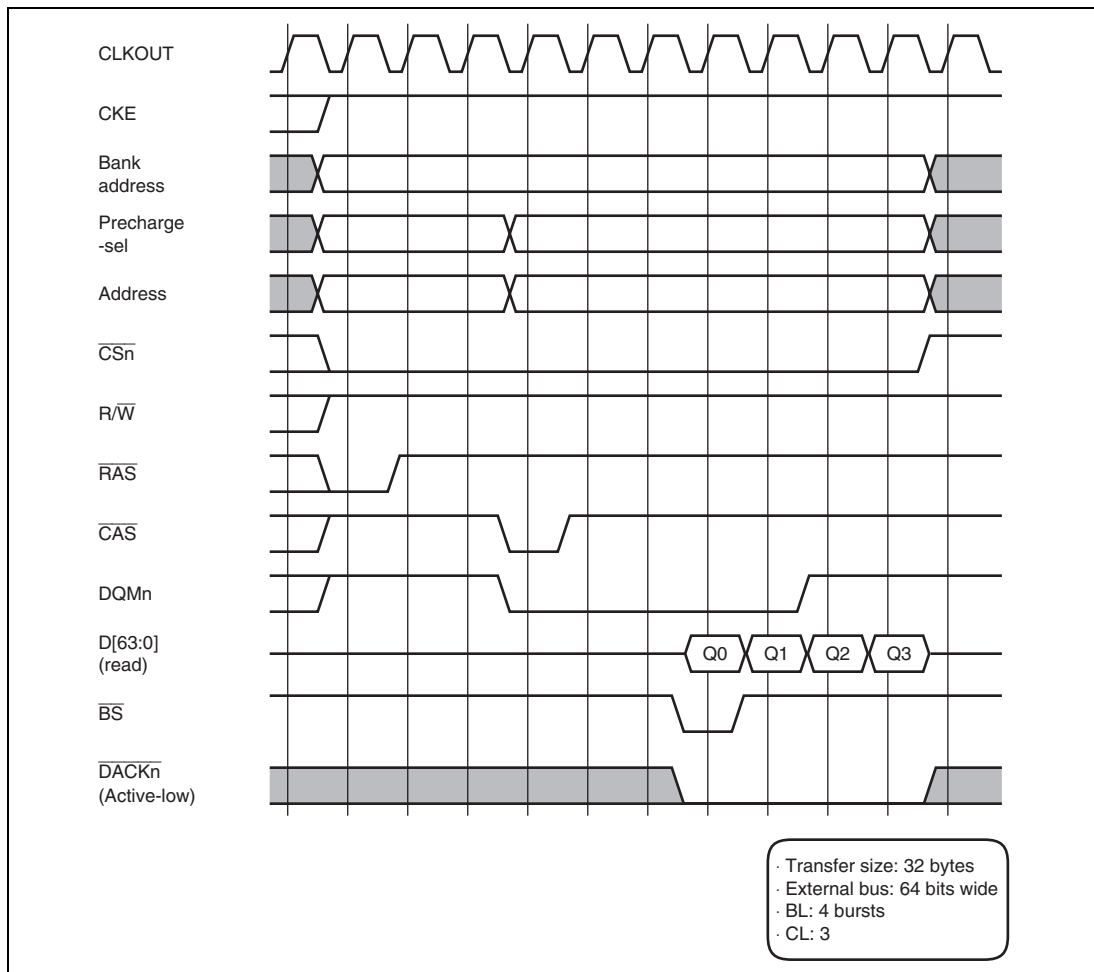
External Bus	SDRAM Address	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
32 bits	SH7764 Address	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
128 Mbits	Bank (2)	12	13													
(16 Mbytes)	Row (12)				11	24	23	22	21	20	19	18	17	16	15	14
8 M × 16	Column (9)							10	9	8	7	6	5	4	3	2
256 Mbits	Bank (2)	12	13													
(32 Mbytes)	Row (12)				11	24	23	22	21	20	19	18	17	16	15	14
8 M × 32	Column (9)							10	9	8	7	6	5	4	3	2
256 Mbits	Bank (2)	12	13													
(32 Mbytes)	Row (13)				11	25	24	23	22	21	20	19	18	17	16	15
16 M × 16	Column (9)							10	9	8	7	6	5	4	3	2

- When the external bus is 64 bits wide:

External Bus	SDRAM Address	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
64 bits	SH7764 Address	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
128 Mbits	Bank (2)	14	13													
(16 Mbytes)	Row (12)				12	25	24	23	22	21	20	19	18	17	16	15
8 M × 16	Column (9)							11	10	9	8	7	6	5	4	3
256 Mbits	Bank (2)	14	13													
(32 Mbytes)	Row (12)				12	25	24	23	22	21	20	19	18	17	16	15
8 M × 32	Column (9)							11	10	9	8	7	6	5	4	3

### 11.7.3 Burst Read

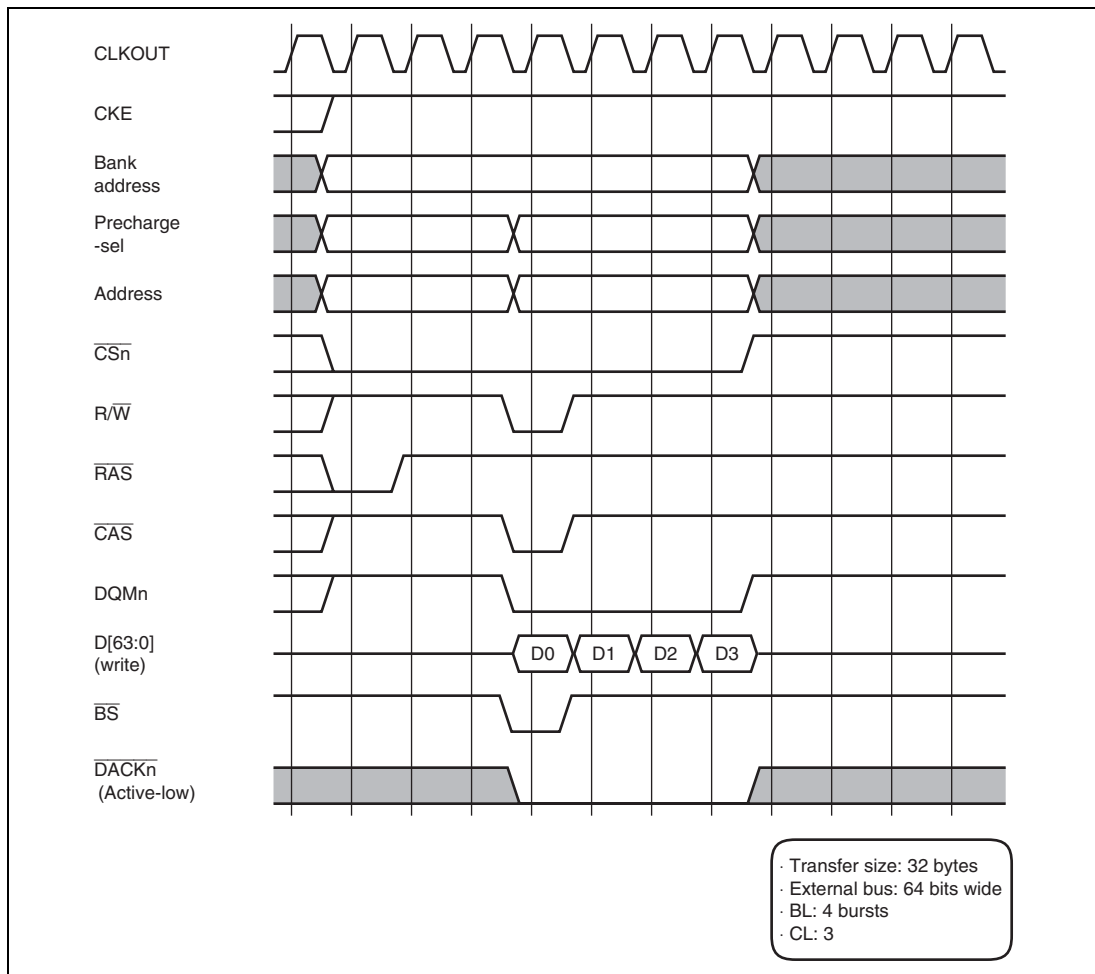
Figure 11.12 shows the burst read timing chart, in which it is assumed that data is 64 bits wide, bank close mode is used, and the burst length is four.



**Figure 11.12 Basic SDRAM Interface Timing (1) Burst Read**

### 11.7.4 Burst Write

Figure 11.13 shows the burst write timing chart, in which it is assumed that data is 64 bits wide, bank close mode is used, and the burst length is four.

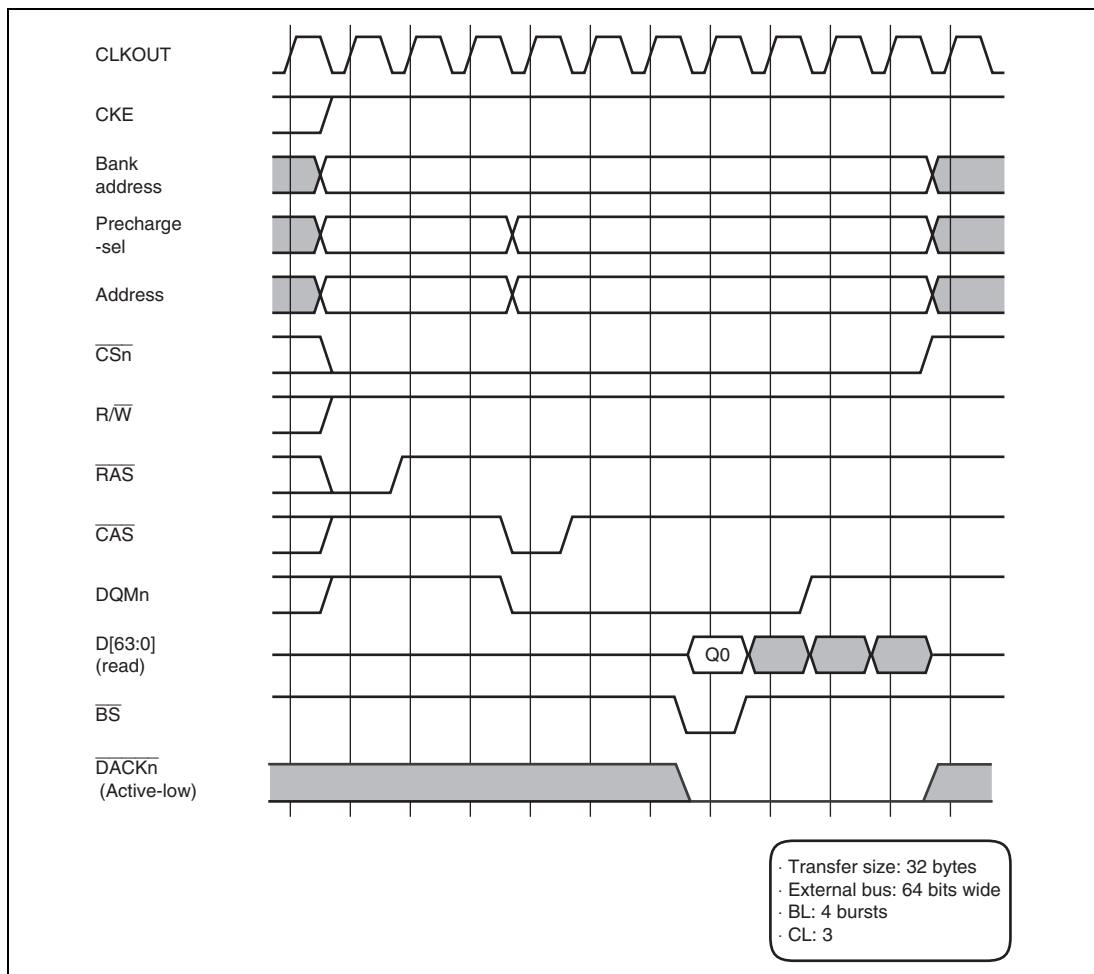


**Figure 11.13 Basic SDRAM Interface Timing (2) Burst Write**

### 11.7.5 Single Read

Figure 11.14 shows the single read timing chart, in which it is assumed that data is 64 bits wide, bank close mode is used, and the burst length is four. Even during single read, data is read out assuming that the burst length is four, like during burst read; the data size is then adjusted to the required read size in the MCU.

If there are unused cycles, the memory access time increases, reducing program execution speed and DMA transfer speed. To prevent this, it is important to use such a data structure that allows placing data on 32-byte boundaries thus enabling data transfer in units of 32 bytes.



**Figure 11.14 Basic SDRAM Interface Timing (3) Single Read**

### 11.7.6 Single Write

Figure 11.15 shows the single write timing chart, in which it is assumed that data is 64 bits wide, bank close mode is used, and the burst length is four. Even during single write, data is written assuming that the burst length is four, like during burst write; however, during the unnecessary data cycles, DQMn is asserted to mask the unnecessary data write.

If there are unused cycles, the memory access time increases, reducing program execution speed and DMA transfer speed. To prevent this, it is important to use such a data structure that allows placing data on 32-byte boundaries thus enabling data transfer in units of 32 bytes.

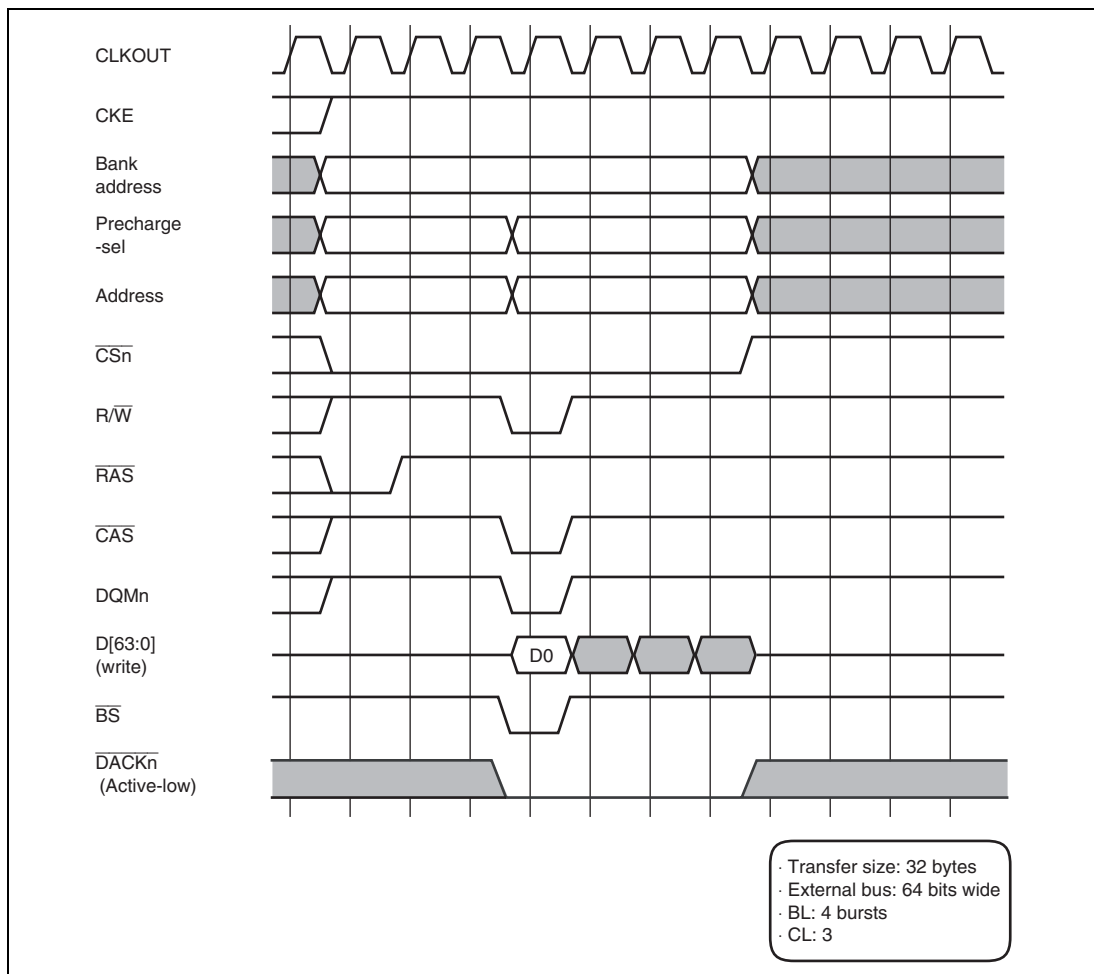


Figure 11.15 Basic SDRAM Interface Timing (4) Single Write

### 11.7.7 Bank Open Mode

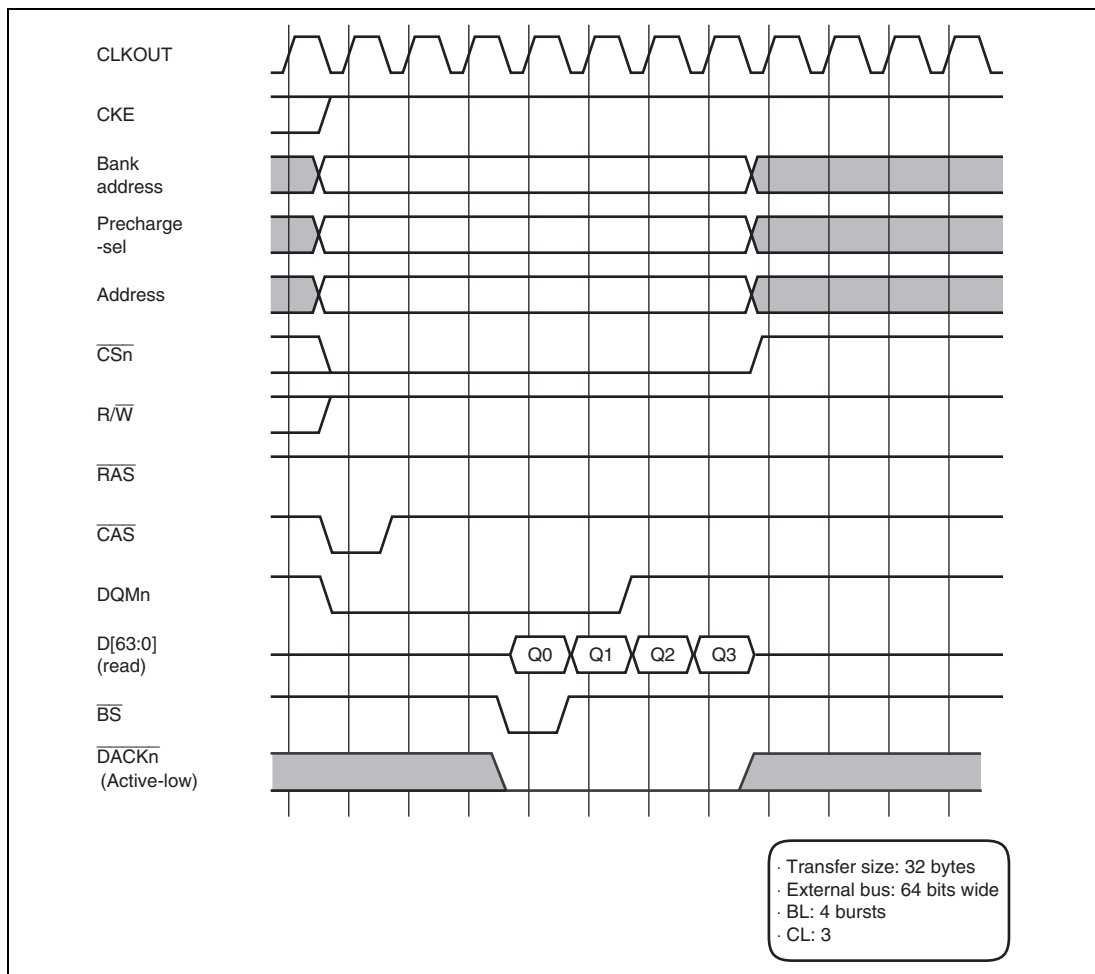
The SDRAM bank function can be used to support high-speed accesses to the same row address.

Specifically, when the BOMODE[1:0] bits in MIM are 00, read/write accesses are performed, using the commands without auto-precharge (READ/WRIT). In this case, precharge is not carried out when an access is completed. Therefore, when the next access is an access to the same row address in the same bank, the READ or WRIT command can be immediately issued without issuing the ACTV command. The SDRAM is internally divided into four banks, and a single row address can be held in the active state for each bank. On the other hand, when the next access is an access to the different row address, the PRE command is first issued to precharge the relevant bank. After precharge has been completed, the ACTV command is accessed followed by the READ or WRIT command. In the applications where the different row addresses are consecutively accessed, access time increases because precharge is started after an access request has been issued.

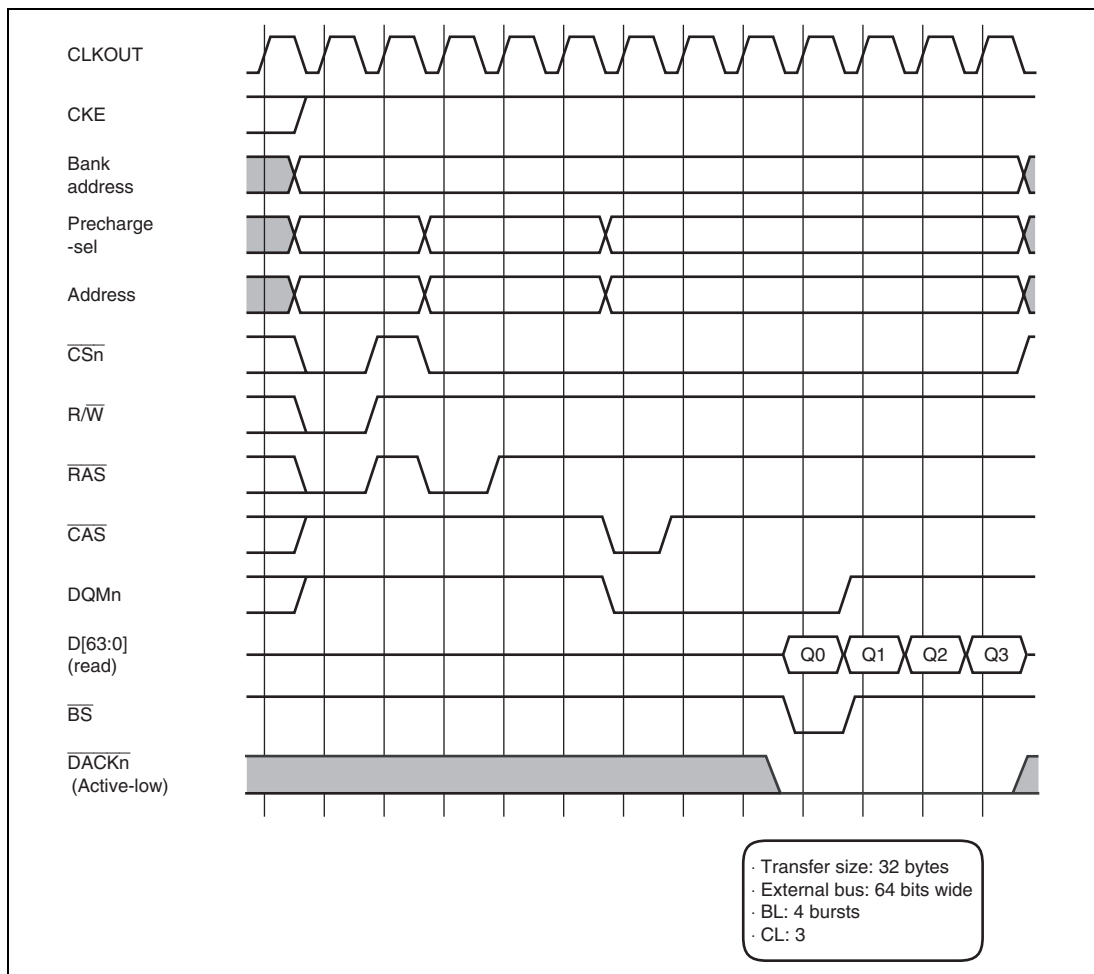
There is a limit on tRAS, i.e., the time duration for which each bank can be held in the active state. If there is no guarantee that this time limit can be kept by program execution when a cache hit does not occur and thus another row will be accessed, it is necessary to set auto-refreshing and set the refresh cycle to no more than the maximum value of tRAS. This allows keeping the limit value of the active time for each bank. When the auto-refresh function is not used, some measures should be taken through programming to prevent each bank from being held in the active state over the specified time.

Figure 11.16 shows the burst read timing for the same row address, and figure 11.17 shows the burst read timing for the different row addresses. Similarly, figure 11.18 shows the burst write timing for the same row address, and figure 11.19 shows the burst write timing for the different row addresses.

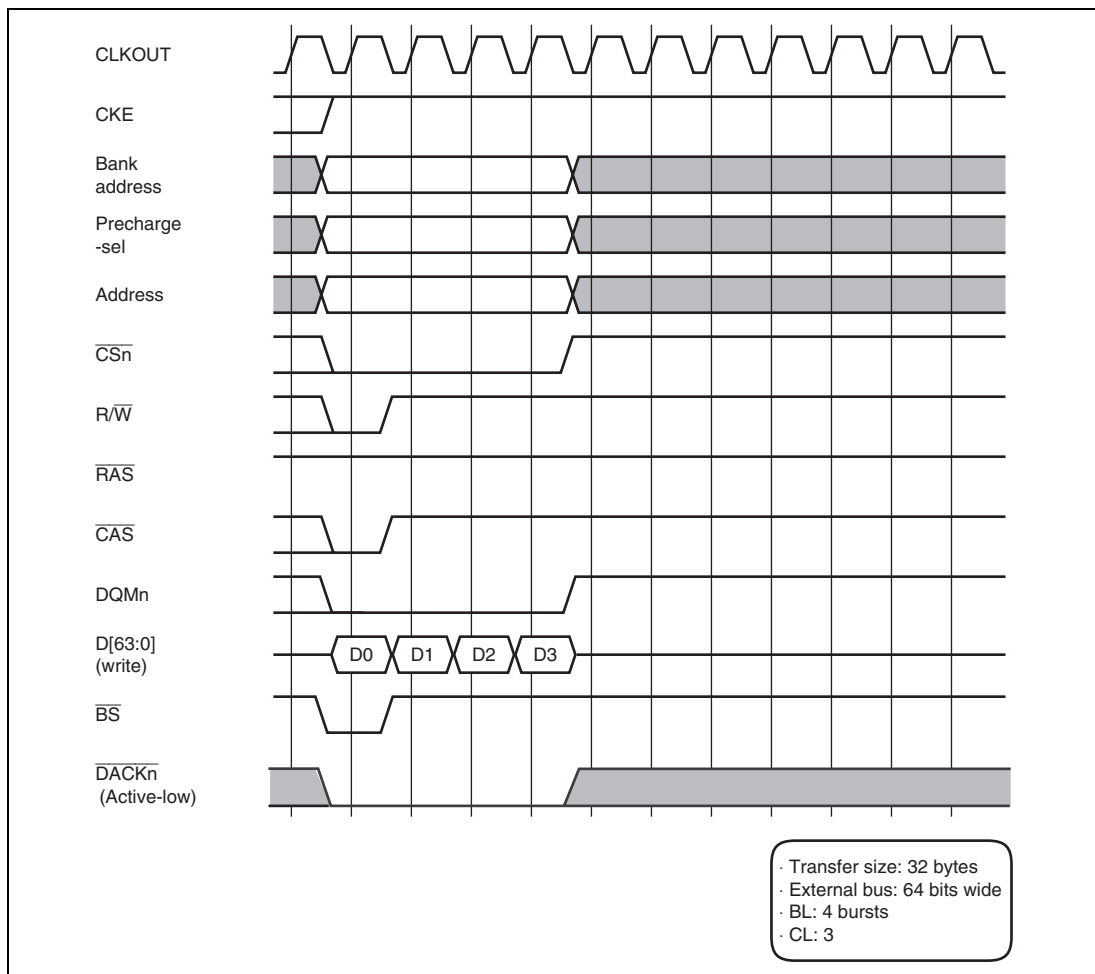




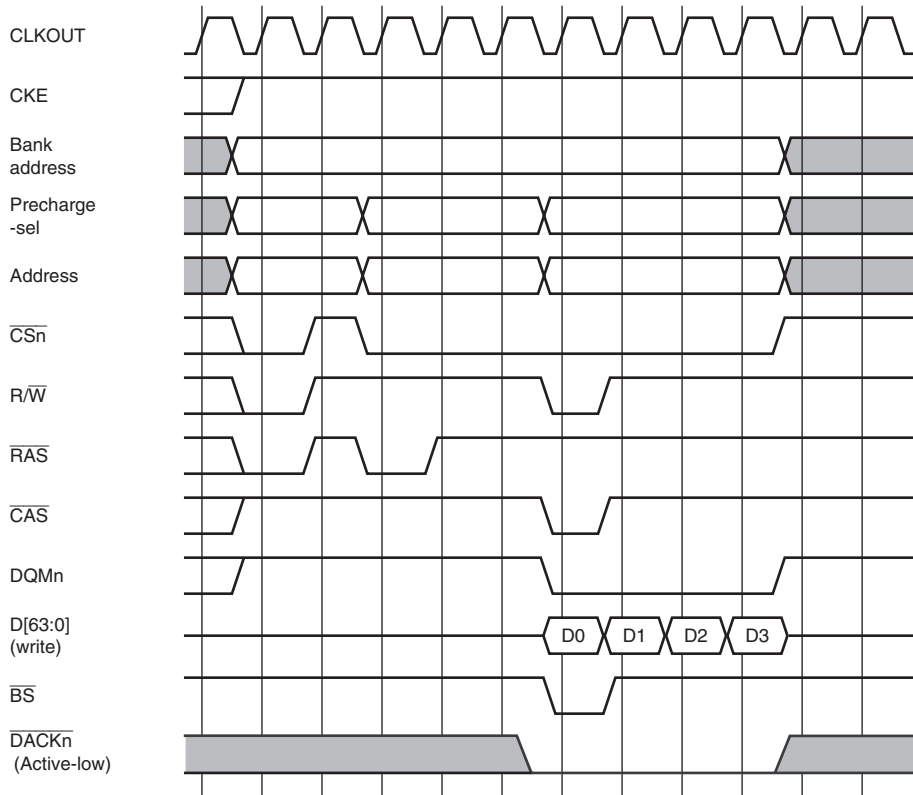
**Figure 11.16 Basic SDRAM Interface Timing (5) Burst Read Timing  
(Bank Open Mode; Same Row Address Accessed)**



**Figure 11.17 Basic SDRAM Interface Timing (6) Burst Read Timing  
(Bank Open Mode; Different Row Addresses Accessed)**



**Figure 11.18 Basic SDRAM Interface Timing (7) Burst Write Timing  
 (Bank Open Mode; Same Row Address Accessed)**



- Transfer size: 32 bytes
- External bus: 64 bits wide
- BL: 4 bursts
- CL: 3

**Figure 11.19 Basic SDRAM Interface Timing (8) Burst Write Timing**  
(Bank Open Mode; Different Row Addresses Accessed)

## 11.7.8 Refresh

### (1) Auto-Refresh

The auto-refresh interval is specified by the DRI[11:00] bits in MIM. If the DRE bit is set to 1 at the same time as the DRI bits are set, the time until the first auto-refresh is determined by the value of the DRI bits before the new setting was made. However, the second and subsequent auto-refresh intervals are determined by the value corresponding to the new setting for the DRI bits. To avoid this situation, clear the DRE bit to 0 when setting the DRI bits. When the DRE bit is subsequently set to 1, auto-refreshing proceeds with the specified interval from the first round. When writing 1 to the DRE bit, the previously written cycle number should be set to the DRI bits.

Note that when the MIM register is not set so that the counter for the auto-refresh function starts operating, the delay adjustment unit in the MCU does not work correctly, thus disabling correct SDRAM read access. Therefore, be sure to set the MIM register appropriately to enable the auto-refresh function before making a read access to SDRAM, as described in the SDRAM initialization sequence and the recovery sequence from the self-refresh state in this specifications document.

When SDRAM auto-refresh occurs during SRAM access by the LBSC, the ARBT module gives priority to auto-refresh and masks the next SRAM access request that has been already accepted. In this case, the next request is sent to the LBSC when the specified idle cycles have been passed after auto-refresh completion. This also applies to the case in which the next request is an SDRAM access.

### (2) Self-Refresh

[Transition to self-refresh state]

1. Confirm that the transaction to the memory controller is completed.
2. Through software control, set the SMS[2:0] bits in SCR to issue the PALL (precharge select all bank) command. This closes any SDRAM bank that was open. After that, use the SMS[2:0] bits in SCR to issue the CBR (auto-refresh) command to perform concentrated refresh on all memory rows (CBR).
3. To make the SDRAM enter the self-refresh state, set the DRE and RMODE bits in MIM in the SBSC to 1. In this case, the BW[1:0] bits should remain 10 and the DCE bit should remain 1.)
4. The memory controller automatically issues the self-refresh command and sets the external CKE pin low. The SDRAM then automatically enters power-down mode.
5. Whether the SDRAM has entered self-refresh mode can be checked by reading the SELFS bit in MIM.

[Recovery from self-refresh state]

1. Clear the DRE and RMODE bits in MIM to 0 to clear the self-refresh state. Here, the BW[1:0] bits should be set again.
2. Whether the SDRAM has recovered from the self-refresh state can be checked by reading the SELFS bit in MIM.
3. After the self-refresh state is cleared, wait for the time required by the SDRAM before accessing the SDRAM.
4. When access becomes possible, use the SMS[2:0] bits in SCR to issue the CBR (auto-refresh) command so that the concentrated refresh (CBR) is performed on all memory rows.
5. Use the SMS[2:0] bits in SCR to issue the PALL (precharge select all bank) command.
6. Use the SMS[2:0] bits in SCR to issue the CBR command.
7. Set MIM so that the counter for the auto-refresh function starts counting and thus drives auto-refreshing at a regular interval. After this, normal memory access is possible.

### 11.7.9 SDRAM Initialization Sequence

An example of the initialization sequence for the SDRAM is shown below. For details, see each memory manufacturer's datasheet.

1. Turn on the power supplies to the SDRAM in the order of VDD and VDDQ.
2. After the power supply and clock are stabilized, maintain the current state for at least 200  $\mu$ s.
3. Set the BW[1:0] and BOMODE[1:0] bits in MIM to enable the SDRAM controller and set the bus width and SDRAM access mode. Here, the DRE bit should remain 0 to prevent counter operation from enabling automatic auto-refresh operation.
4. Set the SPLIT[3:0] bits in SDRA to select the appropriate address multiplexing for the type of memory to be connected. Also set the STR register according to the timing specifications of the memory to be connected.
5. Use the SMS[2:0] bits in SCR to enable the CKE pin. This drives the external CKE pin high.
6. Use the SMS[2:0] bits in SCR to issue the PALL (precharge select all bank) command.
7. Use the SMS[2:0] bits in SCR to issue the CBR (auto-refresh) command eight times.
8. Use SDMR to issue the MRS command to determine the SDRAM operating mode.

## **11.8 Wait Cycles between Accesses**

### **11.8.1 Wait Cycles between Accesses to Area 0 or 3**

The specified number of idle cycles are inserted between consecutive accesses to area 0 or area 3, or between an access to area 0 or area 3 and the subsequent access to area 1 or area 2. The number of idle cycles to be inserted is determined by the CSn bus control register (CSnBCR) and bus control register (BCR).

Specifically, the bits IWW, IWRWD, IWRWS, IWRRD, and IWRRS in CSnBCR and the IRSD bits in BCR are used to specify the number of wait cycles to be inserted as the idle cycle. Here, at least the specified number of cycles are inserted.

Between an access to area 0 or area 3 and the subsequent access to area 1 or area 2, at least four idle cycles are inserted. In addition, when the access size is 8 bytes, 16 bytes, or 32 bytes, wait cycles are inserted every 4-byte access.

### **11.8.2 Wait Cycles between Accesses to Area 1 or 2**

With respect to the consecutive accesses to area 1 or 2, the same area can be consecutively accessed with an interval of one idle cycle (minimum) between the read and read commands or between the write and write commands. In case of the write command followed by read command or the read command followed by the write command, the same area is consecutively accessed with the specified interval between the commands. Here, the interval is specified by the WR or RW bits in the SDRAM timing register (STR).

Between consecutive accesses to area 1 and area 2, at least three idle cycles are inserted.

### **11.8.3 Wait Cycles between Access to Area 1 or 2 and the Subsequent Access to Area 0 or 3**

Between an access to area 1 or 2 and the subsequent access to area 0 or 3, at least two idle cycles are inserted.

## 11.9 Bus Arbitration

This module has two arbitration functions: one is arbitration of the accesses between the various internal modules, and the other is arbitration of the bus requests from the external devices.

### 11.9.1 Arbitration of Accesses between Internal Modules

This module arbitrates SDRAM or SRAM accesses between the CPU, various pixel bus modules, and LCDC. (SRAM cannot be accessed by the pixel bus modules or LCDC.) Since SDRAM accesses are often used for the applications requiring real-time operation such as reading out the image data for display, it is significant to assign the appropriate priority to the modules so that the requirements of the modules such as response time and bandwidth are satisfied. The policy of priority assignment is given below.

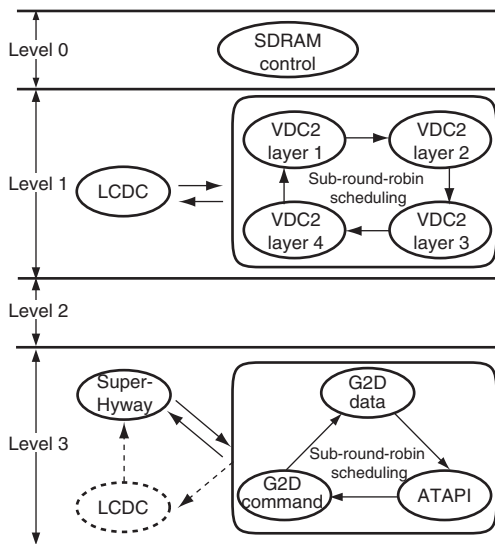
1. The highest priority (level 0) is given to SDRAM control such as refreshing and page management.  
Memory is refreshed according to the memory refresh interval specified separately.
2. A high priority (level 1) is given to the display controller (VDC2) and LCD controller (LCDC) to support transfer of the output data for display, which requires real-time operation.
3. A lower priority (level 2 or 3) is given to the other accesses.  
Either level can be selected for each module.

Figure 11.20 shows the arbitration of the access requests. In the figure, priority level 1 is given to the VDC2 and LCDC, and the round-robin method is used to arbitrate the accesses between them. Similarly, priority level 3 is given to the SuperHyway modules (CPU, DMAC, EtherC, and others), ATAPI, and G2D command/data; and the round-robin method is used to arbitrate the accesses between them. Access requests are arbitrated using the request signals that are being asserted at the arbitration timing. Figure 11.21 shows an arbitration example in which the priority levels of SuperHyway modules and G2D module are raised to level 2. Arbitration is carried out only between the modules indicated by the solid lines.

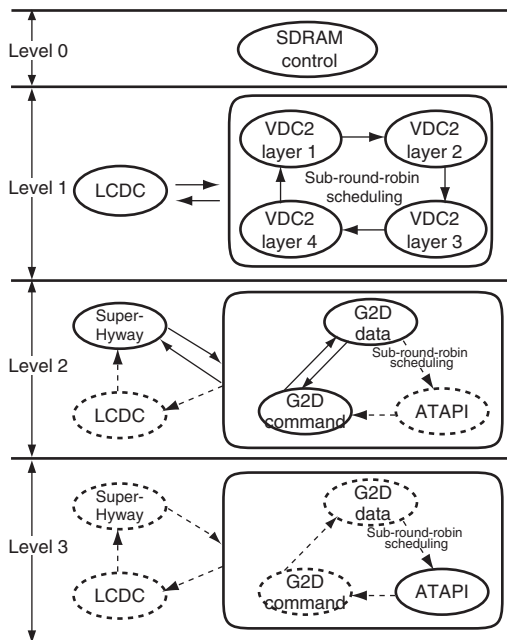
The request-masking function is provided to limit memory accesses during NMI interrupt processing. This function allows assigning relatively higher percentage of memory use by the CPU interrupt processing upon NMI interrupt generation. Requests from different modules can be masked separately through the request mask setting register (RQM) so that the optimum settings can be made according to the usage of NMI.

For the LCDC, any of priority levels 1, 2, and 3 can be selected through register setting.





**Figure 11.20 Arbitration of Access Requests (1)**



**Figure 11.21 Arbitration of Access Requests (2)**

The priority is determined hierarchically according to the priority level, round-robin scheduling for the modules with the same priority level, and sub-round-robin scheduling for the modules with the same priority level, in this order. The following describes each determination.

1. Determination according to the priority level

The priority is fixed: level 0 > level 1 > level 2 > level 3.

2. Determination according to the level-1 round-robin scheduling

The priority is determined according to the round-robin scheduling for the level-1 modules.

After reset: LCDC > pixel bus module

After the LCDC has been selected: Pixel bus > LCDC

After the pixel bus has been selected: LCDC > pixel bus

Note that the pixel bus refers to the modules selected according to the level-1 pixel bus sub-round-robin scheduling.

3. Determination according to the level-1 pixel bus sub-round-robin scheduling

When this determination is selected, the priority is also determined according to the round-robin scheduling for the level-1 pixel bus modules.

After reset: VDC2 (layer 1) > VDC2 (layer 2) > VDC2 (layer 3) > VDC2 (layer 4)

After the VDC2 (layer 1) has been selected: VDC2 (layer 2) > VDC2 (layer 3) > VDC2 (layer 4) > VDC2 (layer 1)

After the VDC2 (layer 2) has been selected: VDC2 (layer 3) > VDC2 (layer 4) > VDC2 (layer 1) > VDC2 (layer 2)

After the VDC2 (layer 3) has been selected: VDC2 (layer 4) > VDC2 (layer 1) > VDC2 (layer 2) > VDC2 (layer 3)

After the VDC2 (layer 4) has been selected: VDC2 (layer 1) > VDC2 (layer 2) > VDC2 (layer 3) > VDC2 (layer 4)

4. Determination according to the level-2 or level-3 round-robin scheduling

The priority is determined according to the round-robin scheduling for the level-2 or level-3 modules. Either priority level 2 or 3 can be given to the devices separately through the arbitration mode register.

When the same priority level is assigned to more than one device, the priority is determined according to the round-robin scheduling for each level.

The priority of the devices with the same priority level is as follows:

After reset: SuperHyway > pixel bus > LCDC

Note that the pixel bus refers to the devices selected according to the level-2 or level-3 pixel bus sub-round-robin scheduling.

After the pixel bus has been selected: LCDC > SuperHyway > pixel bus

After the LCDC has been selected: SuperHyway > pixel bus > LCDC

### 5. Determination according to the level-2 or level-3 pixel bus sub-round-robin scheduling

When this determination is selected, the priority is also determined according to the round-robin scheduling for the level-2 or level-3 pixel bus modules.

After reset: ATAPI > G2D command > G2D data

After the ATAPI has been selected: G2D command > G2D data > ATAPI

After the G2D command has been selected: G2D data > ATAPI > G2D command

After the G2D data has been selected: ATAPI > G2D command > G2D data

## 11.9.2 Multi-Step Arbitration

### (1) Three-Step Arbitration

The following three-step arbitration is applied to the memory accesses from a module.

#### (a) First-Step Arbitration

Arbitration is carried out according to the physical connection of the module to select the following five types of requests

A1: Level-1 pixel bus request. One is selected from among VDC2 layers 1, 2, 3, and 4.

A2: Level-2 pixel bus request. One is selected from among the ATAPI, G2D command, and G2D data.

A3: Level-3 pixel bus request. One is selected from among the ATAPI, G2D command, and G2D data.

A4: SuperHyway request. No arbitration is carried out since only one type of request is involved; however, the SuperHyway module is set to priority level 1 or 2 according to the arbitration mode register setting.

A5: LCDC request. No arbitration is carried out since only one type of request is involved; however, the LCDC is set to priority level 1, 2, or 3 according to the arbitration mode register setting.

#### (b) Second-Step Arbitration

B: One of A1 to A5 is selected based on the determination according to the priority level and the round-robin scheduling for the same priority level.

### (c) Third-Step Arbitration

C: Arbitration is carried out between the SDRAM control such as refreshing and B. The SDRAM control always takes priority.

### (2) Access Order after Arbitration

In the three-step arbitration, care should be taken about the order of access execution since some requests are selected from among multiple requests during the first- and second-step arbitration. Specifically, there are access-request queues corresponding to A1 to A5 so that such arbitration is carried out independently. The following gives a summary of queuing operations.

1. Memory control processing (C) is carried out.
2. The access (B) for which the priority has been determined and has been queued is executed.
3. One of A1 to A5 is selected (Ax) and queued as B.
4. The next arbitration is carried out between the modules corresponding to Ax selected in step 3 and the selected request (Ay) is queued.

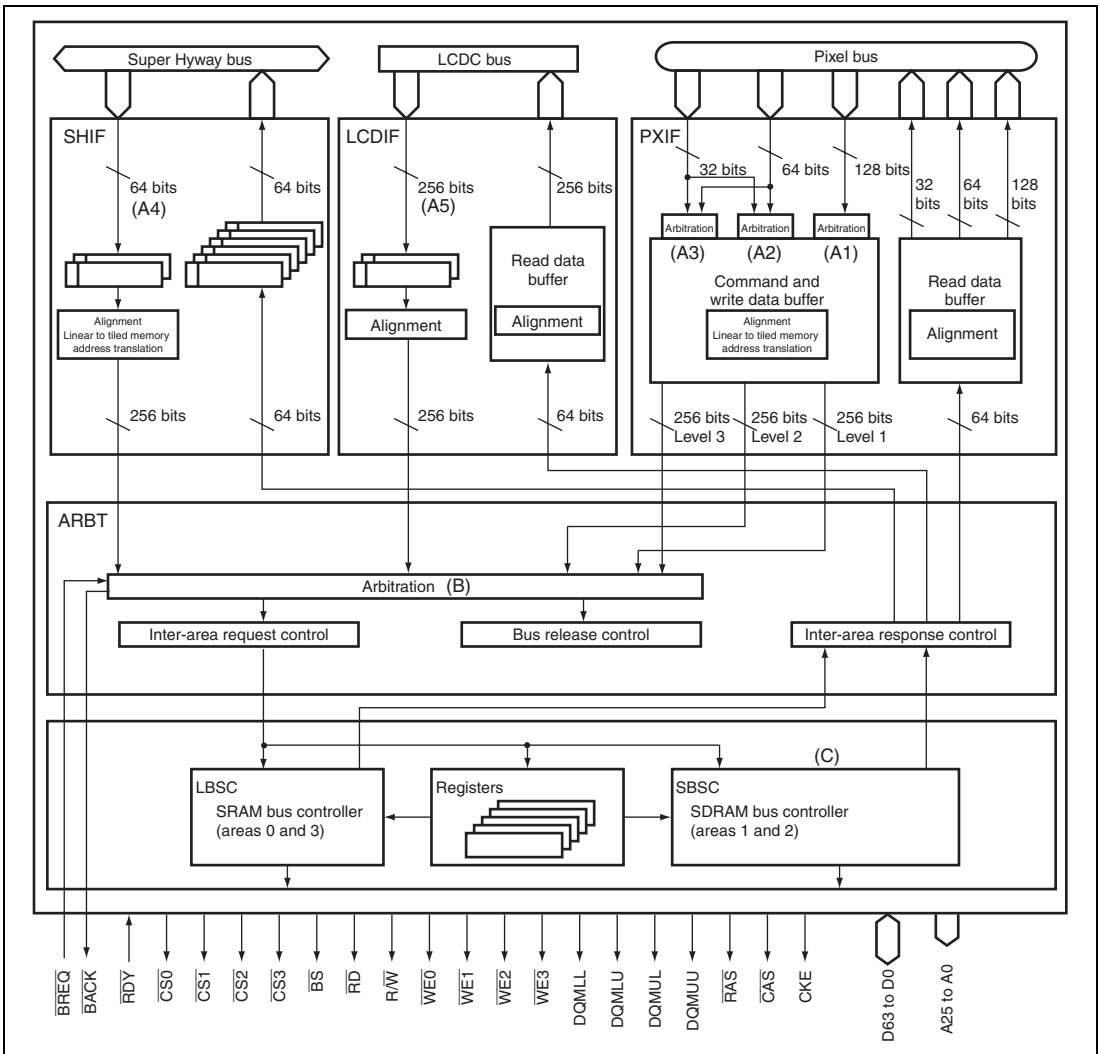
However, it should be noted that Ay is not necessarily queued as B in the next arbitration here; it depends on the result of the second-step arbitration (Az).

According to the above, the order of access execution is C, B, Ax, Az, ... Az, and Ay.

When A5 is set to the priority level other than level 1, all the level-1 access requests are grouped into A1. In addition, no requests with the other priority level are handled as A1 requests. Therefore, when Ay has priority level 1 (i.e., A1) and Az has the other priority level, Ay takes priority over Az and is selected, resulting in the execution order of C, B, Ax, and Ay (A1). Even if level-1 access requests do not survive the first-step arbitration and are not selected as A1, the execution order is C, B, Ax, A1, ... A1, and Ay (A1).

Although other level-1 access requests take priority, it is all the same that level-1 access requests are accepted after B and Ax for which the priority has been determined during the first- and second-step arbitration.

For the symbols (A1 to A5, B, and C) for the multi-step arbitration circuit used in the above descriptions, see figure 11.22 below.



**Figure 11.22 Block Diagram of MCU**  
(with Symbols Shown for Multi-Step Arbitration Circuit)

### 11.9.3 Bus Requests from External Devices

This module has the arbitration function in which the bus mastership is given to an external device when a bus request is issued from the external device.

In the normal state, this LSI has the bus mastership. Upon receiving the bus request from an external device, this LSI gives permission to use the bus and releases the bus. While the bus is released, all the signals connected to SRAM or SDRAM are driven to the high-impedance state except some signals. In the following descriptions, the external devices that issue a bus request are referred to as slaves.

In this LSI, there are several bus masters: the SuperHyway bus modules such as the CPU and DMAC, the pixel bus modules, and the LCDC. In addition, when refresh control is carried out for the connected SDRAM, the SDRAM refresh request can also be a bus master. When two or more internal bus masters issue a bus request, arbitration is carried out as described in section 11.9.1, Arbitration of Accesses between Internal Modules, and section 11.9.2, Multi-Step Arbitration. When any internal bus master and a slave issue a bus request simultaneously, the priority is given to the refresh requests, requests from the slaves, and requests from the internal bus masters, in this order.

When the bus mastership is transferred between a master and a slave, all the bus control signals are negated prior to bus release in order to prevent malfunction of the connected devices. Also, after the master or slave has received the bus mastership, it negates the bus control signals prior to driving the bus. Since both the bus master and slave, between which the bus mastership is transferred, drive the bus control signals to the same value, the conflicts between the output buffers can be avoided.

The bus mastership is transferred at the boundary of the bus cycles.

When the bus release request signal ( $\overline{\text{BREQ}}$ ) is asserted, this module processes all the requests having been already accepted, outputs the bus acknowledge signal ( $\overline{\text{BACK}}$ ), and then releases the bus. When  $\overline{\text{BREQ}}$  is negated, this module negates  $\overline{\text{BACK}}$  and resumes using the bus.

When a refresh request is issued while this LSI has the bus mastership, this LSI carries out refreshing operation immediately after completing the current bus cycle. However, when multiple bus cycles are generated because of data bus width being smaller than the access size, for example, when a longword access is made to the memory of 8-bit bus width, refreshing operation is suspended until all the multiple bus cycles have been completed. Refreshing operation is also suspended during 32-byte data transfer to cache file or for write-back.

Refreshing operation is also suspended in the bus-released state because refreshing operation is impossible in that state. When a refresh request is issued in the bus-released state, this LSI negates  $\overline{\text{BACK}}$  to request the external device currently having the bus mastership to release the bus. The external device should negate  $\overline{\text{BREQ}}$  when  $\overline{\text{BACK}}$  is negated. This returns the bus mastership to this LSI allowing this LSI to carry out necessary processing.

Since the bus mastership cannot be returned immediately after  $\overline{\text{BACK}}$  negation, refreshing operation may be suspended for a longer time than the time required when this LSI has the bus mastership. Due to this, the specified refresh interval may not be kept. Therefore, in the bus-released state, a refresh request is issued at the half interval of the interval that is set by the DRAM refresh interval bits (DRI[11:0] in MIM).

### 11.9.4 Bus Release and Recovery Sequences

This LSI has the bus mastership unless it receives a bus request from another device.

In response to assertion (low level) of the bus request ( $\overline{\text{BREQ}}$ ) from an external device, this LSI asserts (low level) the bus acknowledge ( $\overline{\text{BACK}}$ ) to release the bus immediately after completing all the accepted requests from internal bus masters. If there is no bus request for refreshing, this LSI negates (high level)  $\overline{\text{BACK}}$  and resumes using the bus upon negation (high level) of  $\overline{\text{BREQ}}$ , which indicates that the slave has released the bus.

If there is any bus request for refreshing in the bus-released state, this LSI first negates the bus acknowledge ( $\overline{\text{BACK}}$ ) and then resumes using the bus upon negation of  $\overline{\text{BREQ}}$ , which indicates that the slave has released the bus.

When releasing the bus, this LSI drives all the bus control signals related to bus interfacing to the high-impedance state except the SDRAM interface signal CKE, the bus arbitration signal  $\overline{\text{BACK}}$ , and the DMA transfer control signals DACK0, DACK1, DTEND0, and DTEND1.

In addition, this LSI issues the precharge command to the active banks of the SDRAM and releases the bus after the command has been completed.

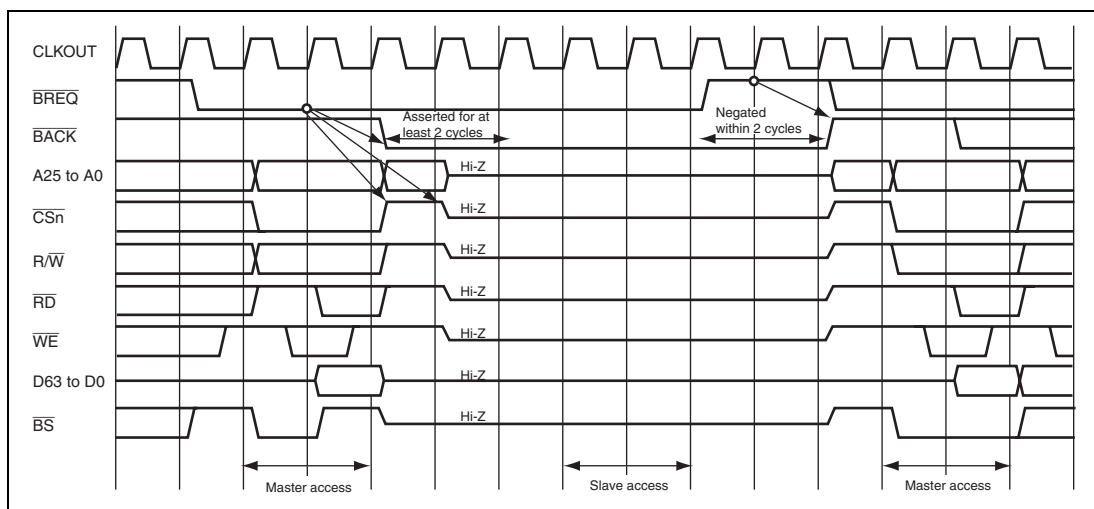
The specific bus release sequence is described below. First,  $\overline{\text{BACK}}$  is asserted in synchronization with the rising edge of the clock pulse, and in synchronization with the rising edge of the next clock pulse to the  $\overline{\text{BACK}}$  assertion, the address bus and data bus are driven to the high-impedance state. Simultaneously, the bus control signals ( $\overline{\text{BS}}$ ,  $\overline{\text{CSn}}$ ,  $\overline{\text{RAS}}$ ,  $\overline{\text{CAS}}$ ,  $\overline{\text{WEn}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{R/W}}$ , and  $\overline{\text{DQMn}}$ ) are driven to the high-impedance state. These bus control signals are negated at least one clock cycle before driven to the high-impedance state.  $\overline{\text{BREQ}}$  is sampled at the rising edge of the clock pulses.

The specific sequence of bus recovery from a slave is described below. On detecting  $\overline{\text{BREQ}}$  negation at the rising edge of the clock pulse,  $\overline{\text{BACK}}$  is immediately negated and the bus control signals begin to be driven simultaneously. The address bus also begins to be driven at the rising edge of the same clock pulse. The fastest timing at which bus access can be resumed is at the rising edge of the clock pulse that is one clock cycle after the cycle at which the bus control signals began to be driven.

Before starting refreshing operation or bus access execution after bus recovery,  $\overline{\text{BREQ}}$  should be negated for two or more clock cycles.

When a refresh request is issued while  $\overline{\text{BACK}}$  is asserted and the bus is released,  $\overline{\text{BACK}}$  is negated in order to request the slave to release the bus even while  $\overline{\text{BREQ}}$  is asserted. With the user-designed slave, multiple bus accesses may be generated consecutively to reduce the overhead due to arbitration. When the slave is to be connected such that the total time of the consecutive accesses exceeds the specified refresh interval, the slave should be designed so that it releases the bus as soon as  $\overline{\text{BACK}}$  negation is detected.

Also, when a bus access request is issued from an internal bus master while the bus is released, it is not accepted until the bus mastership is recovered. In this case, however,  $\overline{\text{BACK}}$  is not negated and a bus release is not requested. When the bus mastership is recovered in response to a memory refresh request issued, and any requests from internal bus masters are in queue for acceptance at that time, refreshing is immediately followed by execution of these bus accesses. Therefore, the bus may not be released until the accepted bus accesses have been completed even when the slave immediately issues the bus request again.



**Figure 11.23 Arbitration Sequence**



### 11.9.5 Cooperation between Master and Slave

In order to control the system resources by the master and slave without insistency, appropriate assignment of roles is significant. Assignment is also necessary when power-down operation is to be used.

In designing the applied system using this LSI, it is assumed that this LSI is responsible for all the controls including initialization and power-down operation.

This LSI does not accept a bus request from any slave after power-on reset until the  $\overline{\text{BREQ}}$  enable bit (BREQEN in BCR) is set to 1.

The  $\overline{\text{BREQ}}$  enable bit should be set to 1 after the memory has been initialized to prevent the slave from accessing the memory requiring initialization prior to use before the memory has been initialized.

## 11.10 Data Coherency

### (1) General Discussion on Memory Access

Memory accesses include read and write accesses and there are four combinations of them in terms of the order of the accesses.

#### (a) WAR (Write after Read)

The case should be taken into consideration in which a read access returns the data that is written by the write access subsequent to the read access. In other words, while a read request from a module is suspended on the bus, data may be written by another module, and the written data may be reflected as the read data.

In this case, where data should be read before damaged by the subsequent write. This case, where write should be initiated after read has been completed, can be managed through software or system. Through software or system, the written data can be prevented from being reflected as the read data. Therefore, no hardware preventive measures are provided.

#### (b) RAR (Read after Read)

No coherency-related problems occur since the same data is read.

**(c) WAW (Write after Write)**

The case should be taken into consideration in which a write access overwrites the data that is written by the write access subsequent to the write access. In other words, while a write request from a module is suspended on the bus, data may be written by another module, and then the suspended write may be reflected. This case, where data is written to the same address consecutively, can be managed through software or system. However, it should be guaranteed that the preceding write access has been reflected on the memory prior to the subsequent write access.

**(d) RAW (Read after Write)**

The case should be taken into consideration in which a preceding write is not reflected on the subsequent read data. In other words, while a write request from a module is suspended on the bus, data may be read by another module, and then the suspended write may be reflected. This case, where data is written to the same address consecutively, can be managed through software or system. However, it should be guaranteed that the preceding write access has been reflected on the memory prior to the subsequent read access.

From (a) to (d), some measures should be taken to check if a write access has been reflected on the memory, which is described in the following sections.

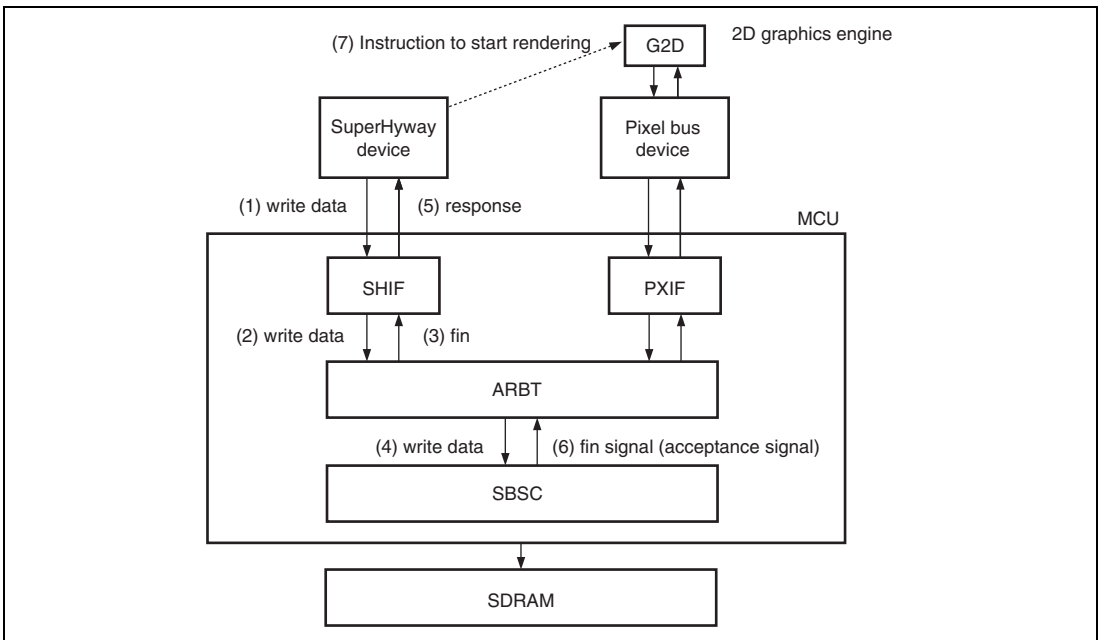
**(2) Confirming Reflection of Write Access****(a) Write Access by SuperHyway Bus Devices**

Execute the SYNCO instruction by the CPU to confirm that the write data has been reflected on the memory. This guarantees that the write data having been stored in the SuperHyway bus interface (SHIF) in the MCU has been reflected on the memory. An example is given below in which the CPU writes the display list to the main memory and instructs the 2D graphics engine to start rendering (figure 11.24). Note that no coherency-related problems occur when the CPU alone accesses the memory consecutively.

1. The CPU writes the display list (last write), and then executes the SYNCO instruction. The CPU stops until the ack signal is returned from the SuperHyway bus.
2. The SuperHyway bus interface (SHIF) in the MCU accepts the write data from the SuperHyway bus. At this point, the ack signal is not returned to the SuperHyway bus device.
3. The SHIF outputs the write data to the arbiter (ARBT).
4. At the same cycle as the step 3, the ARBT carries out arbitration. When the SHIF acquires the bus mastership as a result of arbitration, the ARBT returns the acceptance signal (fin).
5. The ARBT outputs the write data to the SBSC at the next cycle.
6. When data is accepted by the ARBT, the SHIF returns the response signal to the SuperHyway bus device.

7. On accepting the data from the ARBT, the SBSC returns the acceptance signal (fin) to the ARBT. On receiving the fin signal from the SBSC, the ARBT stops outputting the write data to the SBSC. Here, steps 5, 6, and 7 are carried out in parallel.
8. When the ack signal is returned to the CPU, the CPU executes the program (instruction to start rendering) subsequent to the SYNCO instruction.

As seen in the above sequence, it is guaranteed that the write data from the SuperHyway bus device has been accepted by the SBSC when the MCU returns the ack signal to the SuperHyway bus. Write or read requests accepted by the SBSC are processed in order of acceptance. Therefore, it is impossible that a read request that reaches the SBSC after a write request is executed before the write data is processed. With such a hardware mechanism provided, execution of the SYNCO instruction by the CPU guarantees that the write data has been accepted by the SBSC.



**Figure 11.24 Reflection of Data Written by SuperHyway Bus Device**

#### (b) Write Access by Pixel Bus Devices

The pixel bus devices receive the send signal for the write data (equivalent to the ack signal for the last write data) only after the write data has been accepted by the SBSC. By receiving this send signal for the write data, each pixel bus device can determine that the write data has reached the SBSC. Once the write data has reached the SBSC, the write data will never be passed by other write data.

### (3) Confirming Reflection of Software Reset

For a software reset, refer to the relevant sections for each module.

To reflect a software reset of the modules correctly on the SH7764, some measures should be taken to check if a write access has been reflected, similar to memory access. Therefore, after the module has entered the software reset state, the following processes should be carried out before it exits the state.

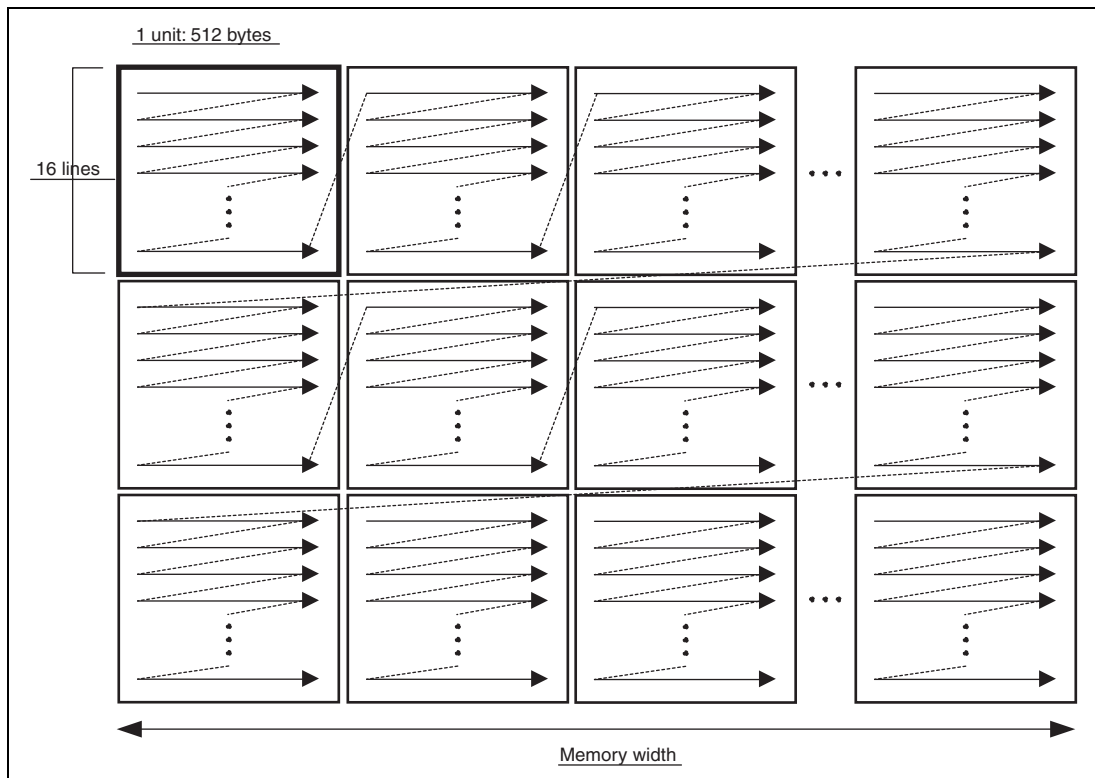
1. When the priority level of the CPU and that of the module to which a software reset is applied are the same, perform a dummy read three times to any SDRAM area.
2. When the priority level of the CPU is level 3 and that of the module to which a software reset is applied is level 2, perform a dummy read once to any SDRAM area.
3. When the priority level of the CPU is level 2 and that of the module to which a software reset is applied is level 3, terminate all the accesses to SDRAM from the level-2 and level-3 modules other than the software-reset-applied module.

## 11.11 Linear-to-Tiled Memory Address Translation

### (1) Tiled Memory Areas

This LSI can modify addresses in the specified range of the connected SDRAM. (For the accesses from the LCDC, linear addresses are not translated into tiled memory addresses.)

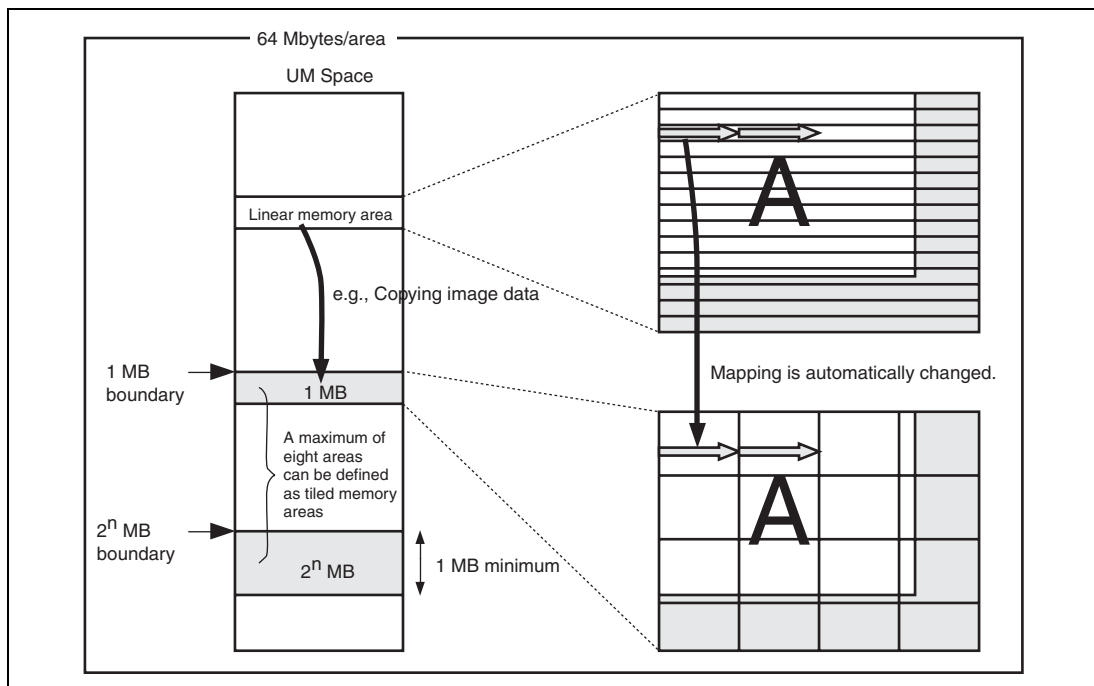
The areas with the modified addresses are called the tiled memory areas. The tiled memory areas are useful as graphics areas in which two-dimensional accesses occur frequently. In these areas, each graphic data area of  $32\text{B} \times 16$  lines is called a tile, and a tile unit is assigned the consecutive 512 bytes of memory. This increases the ratio of hitting at an SDRAM page when the locations to be accessed change frequently in the row direction.



**Figure 11.25 Data Arrangement in Tiled Memory Areas**

## (2) Linear-to-Tiled Memory Address Translation

Linear addresses are internally translated by the MCU into the addresses for tiled memory according to the settings of the registers LTC0 to LTC7 and LTAD0 to LTAD7. A maximum of eight areas can be defined as tiled memory areas on SDRAM. The minimum area size is 1Mbyte and the size can be enhanced to  $2^n$  Mbytes by masking the lower bits of the start address of the area. Each area is defined by the boundary addresses of the area of the specified size. Multiple tiled memory areas should not overlap each other. If overlapping occurs, correct address translation cannot be guaranteed. Also, a data transfer is prohibited if it involves the range extending over a linear address area and a tiled memory address area. They are undefined operations and memory contents cannot be guaranteed.



**Figure 11.26 Schematic of Linear-to-Tiled Memory Address Translation**

### (3) Method of Linear-to-Tiled Memory Address Translation

Table 11.19 shows the specifications for translating linear addresses into the tiled memory addresses.

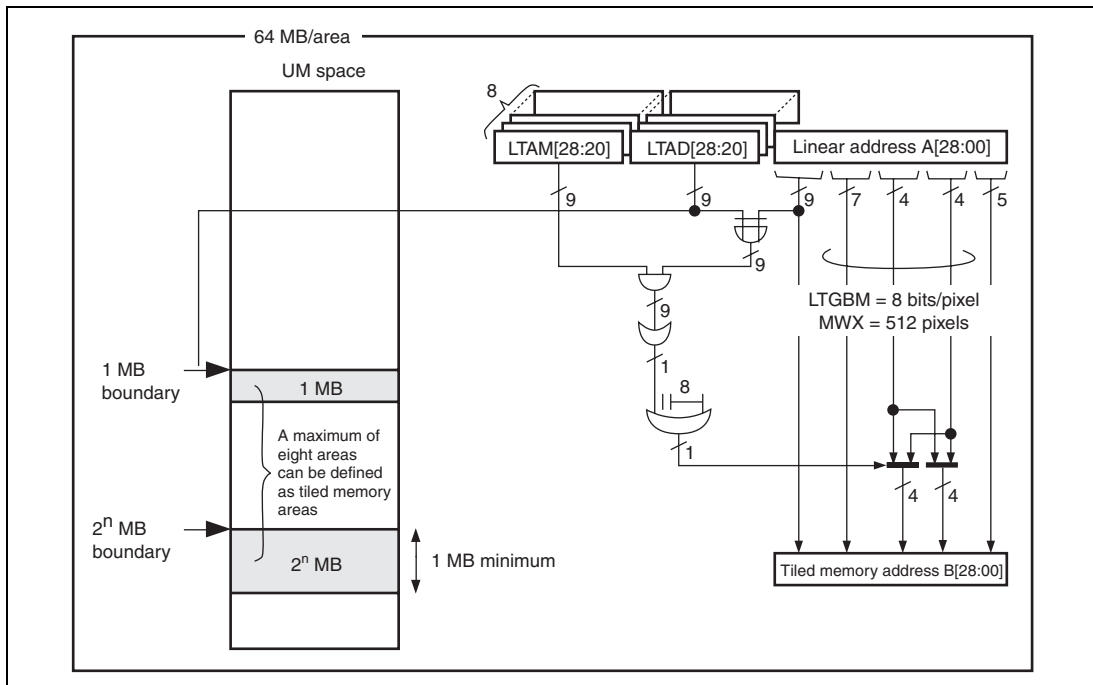
**Table 11.19 Correspondence between Linear Addresses and Tiled Memory Addresses**

LT-GBM	MWX	Linear Address																	
	Number of Pixels (Number of Tiles)	27 to 21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4 to 0
0	512 (16)	↑	↑	↑	↑	↑	↑	↑	↑	↑	8	7	6	5	12	11	10	9	↑
	1024 (32)	↑	↑	↑	↑	↑	↑	↑	↑	9	8	7	6	5	13	12	11	10	↑
	2048 (64)	↑	↑	↑	↑	↑	↑	↑	10	9	8	7	6	5	14	13	12	11	↑
	4096 (128)	↑	↑	↑	↑	↑	↑	11	10	9	8	7	6	5	15	14	13	12	↑
1	512 (16)	↑	↑	↑	↑	↑	↑	15	14	9	8	7	6	5	13	12	11	10	↑
	1024 (32)	↑	↑	↑	↑	↑	↑	↑	10	9	8	7	6	5	14	13	12	11	↑
	2048 (64)	↑	↑	↑	↑	↑	↑	11	10	9	8	7	6	5	15	14	13	12	↑
	4096 (128)	↑	↑	↑	↑	↑	12	11	10	9	8	7	6	5	16	15	14	13	↑

The LTGBM bit in the LTCn register specifies the pixel format of the image data in graphics bit mode. LTGBM = 0 specifies 8 bits/pixel and LTGBM = 1 specifies 16 bits/pixel. MWX in the table indicates the memory width in terms of the number of pixels.

The memory width can be 512, 1024, 2048, or 4096. Only formats of 8 bits/pixel and 16 bits/pixel are available for linear-to-tiled memory address translation.

Figure 11.27 shows the operation of linear-to-tiled memory address translation. In the figure, LTAM represents the linear-to-tiled memory address translation area start address mask registers and LTAD represents the linear-to-tiled memory address translation area start address registers.



**Figure 11.27** Operation of Linear-to-Tiled Memory Address Translation



## 11.12 Usage Notes

### 11.12.1 Refresh

In refresh standby mode and hardware standby mode, auto-refresh is unavailable. For the memory system requiring refresh operation, it is necessary to put the memory into the self-refresh state prior to transition to refresh standby mode. In hardware standby mode, neither self-refresh nor auto-refresh is available because all the pins are driven to the high-impedance state.

### 11.12.2 External Bus Arbitration

In refresh standby mode, the bus mastership is not released. For the system that carries out external bus arbitration, it is necessary to set the  $\overline{\text{BREQ}}$  enable bit (BREQEN in BCR) to 0 prior to transition to refresh standby mode. If transition is made to refresh standby mode with the  $\overline{\text{BREQ}}$  enable bit set to 1, correct operation is not guaranteed.



## Section 12 Direct Memory Access Controller (DMAC)

This LSI includes the direct memory access controller (DMAC).

The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

### 12.1 Features

- Six channels (two channels can receive an external request: channels 0 and 1)
- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, word (2 bytes), longword (4 bytes), 16 bytes, and 32 bytes
- Maximum transfer count: 16,777,216 transfers
- Address mode: Dual address mode
- Transfer requests:

External request (channels 0 and 1), on-chip peripheral module request (channels 0 to 5), or auto request can be selected.

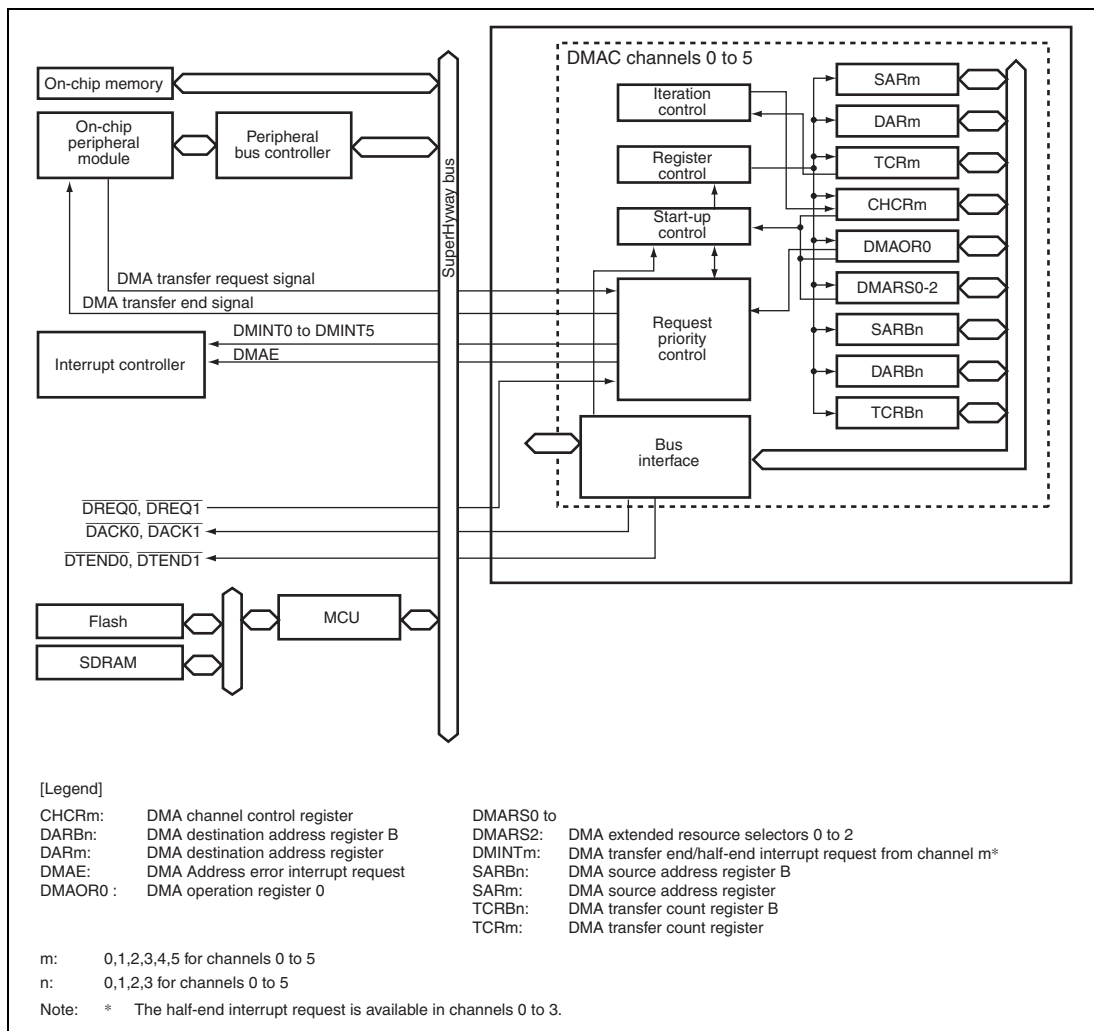
The following modules can issue an on-chip peripheral module request.

— SCIF0, SCIF1, SCIF2, USB, FLCTL, and SRC

- Selectable bus modes:  
Cycle steal mode (normal mode and intermittent mode) or burst mode can be selected.
- Selectable channel priority levels:  
The channel priority levels are selectable between fixed mode and round-robin mode.
- Interrupt request: An interrupt request can be generated to the CPU after half of the transfers ended, all transfers ended, or an address error occurred.
- External request detection: There are following four types of DREQn input detection.  
(n = 0, 1)
  - Low level detection (Initial value)
  - High level detection
  - Rising edge detection
  - Falling edge detection

- Active levels for both the DMA transfer request acceptance signal ( $\overline{\text{DACKn}}$ ) and DMA transfer end signal ( $\overline{\text{DTENDn}}$ ) can be set ( $n = 0, 1$ ).

Figure 12.1 shows the block diagram of the DMAC.



**Figure 12.1 Block Diagram of DMAC**

## 12.2 Input/Output Pins

The external pins for the DMAC are described below. Table 12.1 lists the configuration of the pins that are connected to external device. The DMAC has pins for two channels (channels 0 and 1) for external bus use.

**Table 12.1 Pin Configuration**

Channel	Function	Pin Name	I/O	Description
0	DMA transfer request	$\overline{\text{DREQ0}}^{*1}$	Input	DMA transfer request input from external device to channel 0
	DMA transfer request acknowledge	$\overline{\text{DACK0}}^{*2}$	Output	Strobe output from channel 0 to external device which has output, regarding DMA transfer request
	DMA transfer end notification	$\overline{\text{DTEND0}}^{*2}$	Output	DMA transfer end output 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
	DMA transfer request acknowledge	$\overline{\text{DACK1}}^{*2}$	Output	Strobe output from channel 1 to external device which has output, regarding DMA transfer request
	DMA transfer end notification	$\overline{\text{DTEND1}}^{*2}$	Output	DMA transfer end output from channel 1 to external device

Notes: 1. The initial value is detected at low level.

2. The initial value is low active.

## 12.3 Register Descriptions

Table 12.2 shows the configuration of registers of the DMAC. Table 12.3 shows the state of registers in each processing mode.

**Table 12.2 Register Configuration of DMAC**

Channel	Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size <sup>*3</sup>
0	DMA source address register 0	SAR0	R/W	H'FF60 8020	H'1F60 8020	32
	DMA destination address register 0	DAR0	R/W	H'FF60 8024	H'1F60 8024	32
	DMA transfer count register 0	TCR0	R/W	H'FF60 8028	H'1F60 8028	32
	DMA channel control register 0	CHCR0	R/W <sup>*1</sup>	H'FF60 802C	H'1F60 802C	32
1	DMA source address register 1	SAR1	R/W	H'FF60 8030	H'1F60 8030	32
	DMA destination address register 1	DAR1	R/W	H'FF60 8034	H'1F60 8034	32
	DMA transfer count register 1	TCR1	R/W	H'FF60 8038	H'1F60 8038	32
	DMA channel control register 1	CHCR1	R/W <sup>*1</sup>	H'FF60 803C	H'1F60 803C	32
2	DMA source address register 2	SAR2	R/W	H'FF60 8040	H'1F60 8040	32
	DMA destination address register 2	DAR2	R/W	H'FF60 8044	H'1F60 8044	32
	DMA transfer count register 2	TCR2	R/W	H'FF60 8048	H'1F60 8048	32
	DMA channel control register 2	CHCR2	R/W <sup>*1</sup>	H'FF60 804C	H'1F60 804C	32
3	DMA source address register 3	SAR3	R/W	H'FF60 8050	H'1F60 8050	32
	DMA destination address register 3	DAR3	R/W	H'FF60 8054	H'1F60 8054	32
	DMA transfer count register 3	TCR3	R/W	H'FF60 8058	H'1F60 8058	32
	DMA channel control register 3	CHCR3	R/W <sup>*1</sup>	H'FF60 805C	H'1F60 805C	32
0 to 5	DMA operation register	DMAOR	R/W <sup>*2</sup>	H'FF60 8060	H'1F60 8060	16
4	DMA source address register 4	SAR4	R/W	H'FF60 8070	H'1F60 8070	32
	DMA destination address register 4	DAR4	R/W	H'FF60 8074	H'1F60 8074	32
	DMA transfer count register 4	TCR4	R/W	H'FF60 8078	H'1F60 8078	32
	DMA channel control register 4	CHCR4	R/W <sup>*1</sup>	H'FF60 807C	H'1F60 807C	32
5	DMA source address register 5	SAR5	R/W	H'FF60 8080	H'1F60 8080	32
	DMA destination address register 5	DAR5	R/W	H'FF60 8084	H'1F60 8084	32
	DMA transfer count register 5	TCR5	R/W	H'FF60 8088	H'1F60 8088	32
	DMA channel control register 5	CHCR5	R/W <sup>*1</sup>	H'FF60 808C	H'1F60 808C	32

Channel	Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size*3
0	DMA source address register B0	SARB0	R/W	H'FF60 8120	H'1F60 8120	32
	DMA destination address register B0	DARB0	R/W	H'FF60 8124	H'1F60 8124	32
	DMA transfer count register B0	TCRB0	R/W	H'FF60 8128	H'1F60 8128	32
1	DMA source address register B1	SARB1	R/W	H'FF60 8130	H'1F60 8130	32
	DMA destination address register B1	DARB1	R/W	H'FF60 8134	H'1F60 8134	32
	DMA transfer count register B1	TCRB1	R/W	H'FF60 8138	H'1F60 8138	32
2	DMA source address register B2	SARB2	R/W	H'FF60 8140	H'1F60 8140	32
	DMA destination address register B2	DARB2	R/W	H'FF60 8144	H'1F60 8144	32
	DMA transfer count register B2	TCRB2	R/W	H'FF60 8148	H'1F60 8148	32
3	DMA source address register B3	SARB3	R/W	H'FF60 8150	H'1F60 8150	32
	DMA destination address register B3	DARB3	R/W	H'FF60 8154	H'1F60 8154	32
	DMA transfer count register B3	TCRB3	R/W	H'FF60 8158	H'1F60 8158	32
0, 1	DMA extended resource selector 0	DMARS0	R/W	H'FF60 9000	H'1F60 9000	16
2, 3	DMA extended resource selector 1	DMARS1	R/W	H'FF60 9004	H'1F60 9004	16
4, 5	DMA extended resource selector 2	DMARS2	R/W	H'FF60 9008	H'1F60 9008	16

- Note:
1. Writing 0 after read 1 of HE or TE bit of CHCR is possible to clear the flag.
  2. Writing 0 after read 1 of AE or NMIF bit of DMAOR is possible to clear the flag.
  3. Accessing with other access sizes is prohibited.

**Table 12.3 State of Registers in Each Operating Mode**

<b>Channel</b>	<b>Name</b>	<b>Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Module Standby</b>
0	DMA source address register 0	SAR0	Undefined	Retained	Retained
	DMA destination address register 0	DAR0	Undefined	Retained	Retained
	DMA transfer count register 0	TCR0	Undefined	Retained	Retained
	DMA channel control register 0	CHCR0	H'4000 0000	Retained	Retained
1	DMA source address register 1	SAR1	Undefined	Retained	Retained
	DMA destination address register 1	DAR1	Undefined	Retained	Retained
	DMA transfer count register 1	TCR1	Undefined	Retained	Retained
	DMA channel control register 1	CHCR1	H'4000 0000	Retained	Retained
2	DMA source address register 2	SAR2	Undefined	Retained	Retained
	DMA destination address register 2	DAR2	Undefined	Retained	Retained
	DMA transfer count register 2	TCR2	Undefined	Retained	Retained
	DMA channel control register 2	CHCR2	H'4000 0000	Retained	Retained
3	DMA source address register 3	SAR3	Undefined	Retained	Retained
	DMA destination address register 3	DAR3	Undefined	Retained	Retained
	DMA transfer count register 3	TCR3	Undefined	Retained	Retained
	DMA channel control register 3	CHCR3	H'4000 0000	Retained	Retained
0 to 5	DMA operation register	DMAOR	H'0000	Retained	Retained



Channel	Name	Abbreviation	Power-on Reset	Sleep	Module Standby
4	DMA source address register 4	SAR4	Undefined	Retained	Retained
	DMA destination address register 4	DAR4	Undefined	Retained	Retained
	DMA transfer count register 4	TCR4	Undefined	Retained	Retained
	DMA channel control register 4	CHCR4	H'4000 0000	Retained	Retained
5	DMA source address register 5	SAR5	Undefined	Retained	Retained
	DMA destination address register 5	DAR5	Undefined	Retained	Retained
	DMA transfer count register 5	TCR5	Undefined	Retained	Retained
	DMA channel control register 5	CHCR5	H'4000 0000	Retained	Retained
0	DMA source address register B0	SARB0	Undefined	Retained	Retained
	DMA destination address register B0	DARB0	Undefined	Retained	Retained
	DMA transfer count register B0	TCRB0	Undefined	Retained	Retained
1	DMA source address register B1	SARB1	Undefined	Retained	Retained
	DMA destination address register B1	DARB1	Undefined	Retained	Retained
	DMA transfer count register B1	TCRB1	Undefined	Retained	Retained
2	DMA source address register B2	SARB2	Undefined	Retained	Retained
	DMA destination address register B2	DARB2	Undefined	Retained	Retained
	DMA transfer count register B2	TCRB2	Undefined	Retained	Retained

Channel	Name	Abbreviation	Power-on Reset	Sleep	Module Stand by
3	DMA source address register B3	SARB3	Undefined	Retained	Retained
	DMA destination address register B3	DARB3	Undefined	Retained	Retained
	DMA transfer count register B3	TCRB3	Undefined	Retained	Retained
0, 1	DMA extended resource selector 0	DMARS0	H'0000	Retained	Retained
2, 3	DMA extended resource selector 1	DMARS1	H'0000	Retained	Retained
4, 5	DMA extended resource selector 2	DMARS2	H'0000	Retained	Retained

### 12.3.1 DMA Source Address Registers (SAR0 to SAR5)

SAR are 32-bit readable/writable registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary must be set for the source address value. The initial value is undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SAR															
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
	SAR															
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

### 12.3.2 DMA Source Address Registers (SARB0 to SARB3)

SARB are 32-bit readable/writable registers that specify the source address of a DMA transfer that is set in SAR again in repeat/reload mode. Data to be written from the CPU to SAR is also written to SARB. To set SARB address that differs from SAR address, write data to SARB after SAR.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary must be set for the source address value. The initial value is undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SARB															
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
	SARB															
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

### 12.3.3 DMA Destination Address Registers (DAR0 to DAR5)

DAR are 32-bit readable/writable registers that specify the destination address of a DMA transfer. During a DMA transfer, these registers indicate the next destination address.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary must be set for the destination address value. The initial value is undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DAR															
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

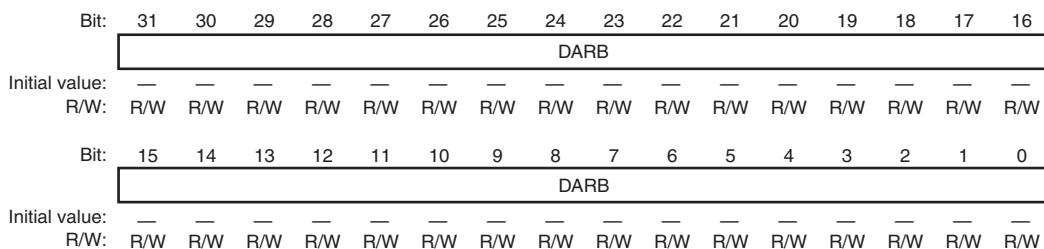
  

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAR															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 12.3.4 DMA Destination Address Registers (DARB0 to DARB3)

DARB are 32-bit readable/writable registers that specify the destination address of a DMA transfer that is set in DAR again in repeat/reload mode. Data to be written from the CPU to DAR is also written to DARB. To set DARB address that differs from DAR address, write data to DARB after DAR.

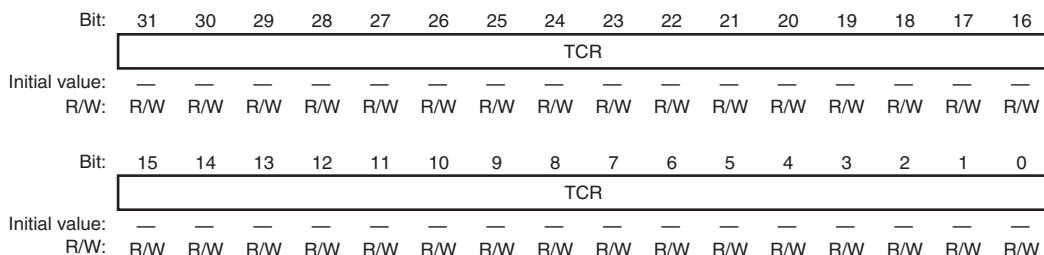
To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary must be set for the source address value. The initial value is undefined.



### 12.3.5 DMA Transfer Count Registers (TCR0 to TCR5)

TCR are 32-bit readable/writable registers that specify the DMA transfer count. The number of transfers is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

The upper eight bits of TCR (bits 31 to 24) are always read as 0, and the write value should always be 0. The initial value is undefined.



### 12.3.6 DMA Transfer Count Registers (TCRB0 to TCRB3)

TCRB are 32-bit readable/writable registers. Data to be written from the CPU to TCR is also written to TCRB. While the half-end function is used, TCRB are used as the initial value hold registers to detect HE. Also, TCRB specify the number of DMA transfers which are set in TCR in repeat mode. TCRB specify the number of DMA transfers and are used as transfer count counters in reload mode.

In reload mode, the lower 8 bits (bits 7 to 0) operate as transfer count counters, values of SAR and DAR are updated after the value of bits 7 to 0 became 0, and then the value of bits 23 to 16 of TCRB are loaded to bits 7 to 0. In bits 23 to 16, set the number of transferring until it reloads. In reload mode, set the same number of transfers in both bits 23 to 16 and 7 to 0, and clear bits 15 to 8 to H'00. Also, clear the HIE bit in CHCR to 0 and do not use the half-end function.

The upper eight bits of TCRB (bits 31 to 24) are always read as 0, and the write value should always be 0.

The initial value of TCRB is undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCRB															
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
	TCRB															
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

### 12.3.7 DMA Channel Control Registers (CHCR0 to CHCR5)

CHCR are 32-bit readable/writable registers that control the DMA transfer mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	LCKN	—	—	RPT[2:0]			—	DO	—	DVMD	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	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/(W)*	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM[1:0]		SM[1:0]		RS[3:0]			DL	DS	TB	TS[1:0]		IE	TE	DE	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

Note: Writing 0 is possible to clear 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  Specifies whether enable or disable the bus lock signal output when a load instruction is output in dual transfer mode. This bit is effective in cycle steal mode.  To disable the bus lock signal, the bus request from the bus master other than the DMAC could be received, and so improve the bus usage efficiency in total system.  Further, the initial value must be cleared to 0 in burst mode. And in the case of specifying the USB in an on-chip module request mode, the bit should be set to 1.  0: Bus lock signal output enabled 1: Bus lock signal output disabled
29, 28	—	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
27 to 25	RPT[2:0]	000	R/W	<p>DMA Setting Renewal Specify</p> <p>These bits are enabled in CHCR0 to CHCR3.</p> <p>000: Normal mode</p> <p>001: Repeat mode SAR/DAR/TCR used as repeat area</p> <p>010: Repeat mode DAR/TCR used as repeat area</p> <p>011: Repeat mode SAR/TCR used as repeat mode</p> <p>100: Reserved (setting prohibited)</p> <p>101: Reload mode SAR/DAR/TCR used as reload area</p> <p>110: Reload mode DAR/TCR used as reload area</p> <p>111: Reload mode SAR/TCR used as reload area</p>
24	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
23	DO	0	R/W	<p>DMA Overrun</p> <p>Selects whether <math>\overline{\text{DREQ}}</math> is detected by overrun 0 or by overrun 1. This bit is valid only in CHCR0 and CHCR1.</p> <p>0: Detects <math>\overline{\text{DREQ}}</math> by overrun 0</p> <p>1: Detects <math>\overline{\text{DREQ}}</math> by overrun 1</p>
22	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
21	DVMD	0	R/W	<p>Division Transfer Mode Specification</p> <p>Specifies the execution of the DMA transfer in 16-byte units between the FLCTL and external memory.</p> <p>When the FLCTL is not used, this bit should always be cleared to 0.</p>

Bit	Bit Name	Initial Value	R/W	Descriptions
20	TS[2]	0	R/W	<p>DMA Transfer Size Specify</p> <p>With TS1 and TS0, this bit specifies the DMA transfer size. When the transfer source or transfer destination is a register of an on-chip peripheral module with a transfer size set, a proper transfer size for the register should be set. For the transfer source or destination address specified by SAR or DAR, an address boundary should be set according to the transfer data size.</p> <p>TS[2:0]</p> <p>000: Byte units transfer</p> <p>001: Word (2-byte) units transfer</p> <p>010: Longword (4-byte) units transfer</p> <p>011: 16-byte units transfer</p> <p>100: 32-byte units transfer</p> <p>Other than above: Setting prohibited</p> <p>Note: To perform DMA transfer by selecting a peripheral module (other than FLCTL and USB) connected to the peripheral bus as source or destination, the transfer size set by TS[2:0] should be longword or less. If FLCTL is selected as source or destination, the transfer size can be set as up to 16 bytes. If USB is selected as source or destination, the transfer size can be set as up to 32 bytes. To perform 16-byte transfer while FLCTL is specified as source or destination, the DVMD bit should be set to 1.</p>

---



Bit	Bit Name	Initial Value	R/W	Descriptions
19	HE	0	R/(W)*	<p>Half End Flag</p> <p>After HIE (bit 18) is set to 1 and the number of transfers become half of TCR (1 bit shift to right) which is set before transfer starts, HE becomes 1.</p> <p>This bit is set to 1 when the TCR value is equal to <math>(\text{TCR set before transfer})/2</math>: TCR value is set to even number of times <math>(\text{TCR set before transfer} - 1)/2</math>: TCR value is set to odd number of times 8,388,608 (H'0080 0000): TCR value is set to the maximum number of times (H'0000 0000)</p> <p>The HE bit is not set when transfers are ended by an NMI interrupt or address error, or by clearing the DE bit or the DME bit in DMAOR before the number of transfers is decreased to half of the TCR value set preceding the transfer. The HE bit is kept set when the transfer ends by an NMI interrupt or address error, or clearing the DE bit (bit 0) or the DME bit in DMAOR after the HE bit is set to 1. To clear the HE bit, write 0 after reading 1 in the HE bit. This bit is valid only in CHCR0 to CHCR3.</p> <p>0: During the DMA transfer or DMA transfer has been interrupted  <math>\text{TCR} &gt; (\text{TCR set before transfer})/2</math>  [Clearing condition]  Writing 0 after HE = 1 is read.  1: <math>\text{TCR} = (\text{TCR set before transfer})/2</math></p>
18	HIE	0	R/W	<p>Half End Interrupt Enable</p> <p>Specifies whether an interrupt request is generated to the CPU when the number of transfers is decreased to half of the TCR value set preceding the transfer. When the HIE bit is set to 1 and the HE bit is set, an interrupt request is generated to the CPU. Clear this bit to 0 while reload mode is set. This bit is valid in CHCR0 to CHCR3.</p> <p>0: Interrupt request is disabled when  <math>\text{TCR} = (\text{TCR set before transfer})/2</math>  1: Interrupt request is enabled when  <math>\text{TCR} = (\text{TCR set before transfer})/2</math></p>

Bit	Bit Name	Initial Value	R/W	Descriptions
17	AM	0	R/W	<p>Acknowledge Mode</p> <p>Selects whether <math>\overline{\text{DACK}}</math> is output in data read cycle or in data write cycle.</p> <p>This bit is valid only in CHCR0 and CHCR1.</p> <p>0: <math>\overline{\text{DACK}}</math> output in read cycle</p> <p>1: <math>\overline{\text{DACK}}</math> output in write cycle</p>
16	AL	0	R/W	<p>Acknowledge Level</p> <p>Specifies whether the DACK and DTEND signal output is high active or low active.</p> <p>This bit is valid only in CHCR0 and CHCR1.</p> <p>0: Low-active output of DACK and DTEND</p> <p>1: High-active output of DACK and DTEND</p>
15, 14	DM[1:0]	00	R/W	<p>Destination Address Mode</p> <p>Specify whether the DMA destination address is incremented, decremented, or left fixed.</p> <p>00: Fixed destination address</p> <p>01: Destination address is incremented</p> <ul style="list-style-type: none"><li>+1 in byte units transfer</li><li>+2 in word units transfer</li><li>+4 in longword units transfer</li><li>+16 in 16-byte units transfer</li><li>+32 in 32-byte units transfer</li></ul> <p>10: Destination address is decremented</p> <ul style="list-style-type: none"><li>−1 in byte units transfer</li><li>−2 in word units transfer</li><li>−4 in longword units transfer</li></ul> <p>Setting prohibited in 16/32-byte units transfer</p> <p>11: Setting prohibited</p>

Bit	Bit Name	Initial Value	R/W	Descriptions
13, 12	SM[1:0]	00	R/W	<p>Source Address Mode</p> <p>Specify whether the DMA source address is incremented, decremented, or left fixed.</p> <p>00: Fixed source address</p> <p>01: Source address is incremented</p> <p>+1 in byte units transfer</p> <p>+2 in word units transfer</p> <p>+4 in longword units transfer</p> <p>+16 in 16-byte units transfer</p> <p>+32 in 32-byte units transfer</p> <p>10: Source address is decremented</p> <p>−1 in byte units transfer</p> <p>−2 in word units transfer</p> <p>−4 in longword units transfer</p> <p>Setting prohibited in 16/32-byte units transfer</p> <p>11: Setting prohibited</p>
11 to 8	RS[3:0]	0000	R/W	<p>Resource Select</p> <p>Specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state that the DMA enable bit (DE) is cleared to 0.</p> <p>0000: External request, dual address mode</p> <p>0100: Auto request</p> <p>1000: Selected by DMA extended resource selector (for on-chip modules)</p> <p>Other than above: Setting prohibited</p> <p>Note: External request specification is valid only in CHCR0 and CHCR1. None of the external request can be selected in CHCR2 to CHCR5. On-chip peripheral module request specification is valid in CHCR0 to CHCR5.</p>

Bit	Bit Name	Initial Value	R/W	Descriptions
7	DL	0	R/W	DREQ Level and DREQ Edge Select
6	DS	0	R/W	<p>Specify the detecting method of the DREQ pin input and the detecting level.</p> <p>These bits are valid only in CHCR0 and CHCR1.</p> <p>In channels 0 and 1, also, if the transfer request source is specified as an on-chip peripheral module or if an auto-request is specified, these bits are invalid.</p> <p>00: DREQ detected at low level (<math>\overline{\text{DREQ}}</math>)</p> <p>01: DREQ detected at falling edge</p> <p>10: DREQ detected at high level</p> <p>11: DREQ detected at rising edge</p>
5	TB	0	R/W	<p>Transfer Bus Mode</p> <p>Specifies the bus mode when DMA transfers data.</p> <p>0: Cycle steal mode</p> <p>1: Burst mode</p> <p>Burst mode cannot be used when the on-chip peripheral module is the transfer request source.</p>
4, 3	TS[1:0]	00	R/W	<p>DMA Transfer Size Specify</p> <p>See the description of TS[2] (bit 20).</p>
2	IE	0	R/W	<p>Interrupt Enable</p> <p>Specifies whether or not an interrupt request is generated to the CPU at the end of the DMA transfer. Setting this bit to 1 generates an interrupt request (DEI) to the CPU when the TE bit is set to 1. To confirm the completion of DMA transfer, issue the SYNCO instruction after the dummy read to unreserved space when generating an interrupt request to the CPU.</p> <p>0: Interrupt request is disabled.</p> <p>1: Interrupt request is enabled.</p>

Bit	Bit Name	Initial Value	R/W	Descriptions
1	TE	0	R/(W)*	<p>Transfer End Flag</p> <p>Shows that DMA transfer ends. The TE bit is set to 1 when data transfer ends when TCR becomes to 0.</p> <p>The TE bit is not set to 1 in the following cases.</p> <ul style="list-style-type: none"> <li>DMA transfer ends due to an NMI interrupt or DMA address error before TCR is cleared to 0.</li> <li>DMA transfer is ended by clearing the DE bit and DME bit in DMAOR.</li> </ul> <p>To clear the TE bit, the TE bit should be written to 0 after reading 1.</p> <p>Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled.</p> <p>0: During the DMA transfer or DMA transfer has been interrupted</p> <p>[Clearing condition]</p> <p>Writing 0 after TE = 1 read</p> <p>1: DMA transfer ends by the specified count (TCR = 0)</p>
0	DE	0	R/W	<p>DMA Enable</p> <p>Enables or disables the DMA transfer. In auto request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this time, all of the bits TE, NMIF, and AE in DMAOR must be 0. In an external request or peripheral module request, DMA transfer starts if DMA transfer request is generated by the devices or peripheral modules after setting the bits DE and DME to 1. In this case, however, all of the bits TE, NMIF, and AE must be 0, which is the same as in the case of auto request mode. Clearing the DE bit to 0 can terminate the DMA transfer.</p> <p>0: DMA transfer disabled</p> <p>1: DMA transfer enabled</p>

Note: \* Writing 0 is possible to clear the flag.

### 12.3.8 DMA Operation Register 0 (DMAOR0)

DMAOR0 is a 16-bit readable/writable register that specifies the priority level of channels at the DMA transfer. This register shows the DMA transfer status. DMAOR0 is a common register for channel 0 to 5.

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)*R/(W)*	R/W

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  Select either normal mode or intermittent mode in cycle steal mode.  It is necessary that all channel bus modes (for channels 0 to 5) are set to cycle steal mode to make valid intermittent mode.  00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Executes one DMA transfer in each of 16 clocks of an external bus clock. 11: Intermittent mode 64 Executes one DMA transfer in each of 64 clocks of an external bus clock.
11, 10	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
9, 8	PR[1:0]	00	R/W	<p>Priority Mode 1, 0</p> <p>Select the priority level between channels when there are transfer requests for multiple channels simultaneously.</p> <p>00: CH0 &gt; CH1 &gt; CH2 &gt; CH3 &gt; CH4 &gt; CH5</p> <p>01: CH0 &gt; CH2 &gt; CH3 &gt; CH1 &gt; CH4 &gt; CH5</p> <p>10: Setting prohibited</p> <p>11: Round-robin mode</p> <p>When round-robin mode is specified, do not mix the cycle steal mode and burst mode in channels 0 to 5 respectively.</p>
7 to 3	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
2	AE	0	R/(W)*	<p>Address Error Flag</p> <p>Indicates that an address error occurred during DMA transfer.</p> <p>This bit is set under following conditions:</p> <ul style="list-style-type: none"> <li>• The value set in SAR or DAR does not match to the transfer size boundary.</li> <li>• The transfer source or transfer destination is invalid space.</li> <li>• The transfer source or transfer destination is in module stop mode</li> </ul> <p>If this bit is set, DMA transfers in the corresponding channels (channels 0 to 5) are all disabled even if the DE bit in CHCR and the DME bit in DMAOR0 are set to 1.</p> <p>0: No DMAC address error</p> <p>[Clearing condition]</p> <p>Writing AE = 0 after AE = 1 read</p> <p>1: DMAC address error occurs</p>

Bit	Bit Name	Initial Value	R/W	Descriptions
1	NMIF	0	R/(W)*	<p><b>NMI Flag</b></p> <p>Indicates that an NMI interrupt occurred. If this bit is set, DMA transfer is disabled even if the DE bit in CHCR and the DME bit in DMAOR0 are set to 1.</p> <p>When the NMI is input, the DMA transfer in progress can be done in at least one transfer unit. When the DMAC is not in operational, the NMIF bit is set to 1 even if the NMI interrupt was input.</p> <p>DMA transfer is stopped when an NMI interrupt is input. After returning from the NMI interrupt routine, set all channels again, and then start the DMA transfer.</p> <p>0: No NMI interrupt [Clearing condition]</p> <p>Writing NMIF = 0 after NMIF = 1 read</p> <p>1: NMI interrupt occurs</p>
0	DME	0	R/W	<p><b>DMA Master Enable</b></p> <p>Enables or disables DMA transfers on all channels. If the DME bit and the DE bit in CHCR are set to 1, transfer is enabled. In this time, all of the bits TE in CHCR, NMIF, and AE in DMAOR must be 0. If this bit is cleared during transfer, transfers in all channels are terminated.</p> <p>To abort the DMA transfer when the on-chip peripheral module request mode is set for any of the channels specified by DMAOR0 (channels 0 to 5), clear the DE bit to 0 while the DMA transfer request from the corresponding peripheral module is cleared.</p> <p>0: Disables DMA transfers on all channels 1: Enables DMA transfers on all channels</p>

Note: \* Writing 0 is possible to clear the flag.



### 12.3.9 DMA Extended Resource Selectors (DMARS0 to DMARS2)

DMARS are 16-bit readable/writable registers that specify the DMA transfer sources from peripheral modules in each channel. DMARS0 specifies for channels 0 and 1, DMARS1 specifies for channels 2 and 3, and DMARS2 specifies for channels 4 and 5. This register can set the transfer request of SCIF0 to SCIF2, USB, FLCTL, SRC

When MID/RID other than the values listed in table 12.4 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits RS[3:0] has been set to B'1000 for CHCR0 to CHCR5 registers. Otherwise, even if DMARS has been set, transfer request source is not accepted. In addition, a transfer request from a peripheral module should not be assigned to multiple DMAC channels as a resource. Otherwise, correct operation cannot be guaranteed.

- DMARS0

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

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C1MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 1 (MID) See table 12.4.
9, 8	C1RID[1:0]	00	R/W	Transfer request register ID for DMA channel 1 (RID) See table 12.4.
7 to 2	C0MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 0 (MID) See table 12.4
1, 0	C0RID[1:0]	00	R/W	Transfer request register ID for DMA channel 0 (RID) See table 12.4.

- DMARS1

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

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C3MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 3 (MID) See table 12.4.
9, 8	C3RID[1:0]	00	R/W	Transfer request register ID0 for DMA channel 3 (RID) See table 12.4.
7 to 2	C2MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 2 (MID) See table 12.4.
1, 0	C2RID[1:0]	00	R/W R/W	Transfer request register ID for DMA channel 2 (RID) See table 12.4.

- DMARS2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C5MID[5:0]					C5RID[1:0]		C4MID[5:0]					C4RID[1:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C5MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 5 (MID) See table 12.4.
9, 8	C5RID[1:0]	00	R/W	Transfer request register ID for DMA channel 5 (RID) See table 12.4.
7 to 2	C4MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 4 (MID) See table 12.4.
1, 0	C4RID[1:0]	00	R/W	Transfer request register ID for DMA channel 4 (RID) See table 12.4.

**Table 12.4 Transfer Request Sources**

<b>Peripheral Module</b>	<b>Setting Value for One Channel (MID and RID)</b>	<b>MID</b>	<b>RID</b>	<b>Function</b>
SCIF0	H'21	B'0010 00	B'01	Transmit
	H'22	B'0010 00	B'10	Receive
SCIF1	H'29	B'0010 10	B'01	Transmit
	H'2A	B'0010 10	B'10	Receive
SCIF2	H'41	B'0100 00	B'01	Transmit
	H'42	B'0100 00	B'10	Receive
USB	H'45	B'0100 01	B'01	Transmit
	H'46	B'0100 01	B'10	Receive
FLCTL	H'83	B'1000 00	B'11	Transmit and receive the data section
	H'87	B'1000 01	B'11	Transmit and receive the control code section
SRC	H'C1	B'1100 00	B'01	Transfer data from SRCOD
	H'C2	B'1100 00	B'10	Transfer data to SRCID

## 12.4 Operation

When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. In bus mode, burst mode or cycle steal mode can be selected.

### 12.4.1 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated by external devices or on-chip peripheral modules that are neither the source nor the destination. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. The request mode is selected in the bits RS[3:0] in CHCR0 to CHCR5, and DMARS0 to DMARS2.

#### (1) Auto-Request Mode

When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits in CHCR0 to CHCR5 and the DME bit in DMAOR0 are set to 1, the transfer begins so long as the AE and NMIF bits in DMAOR0 are all 0.

#### (2) External Request Mode

In this mode, a transfer is performed at the request signal ( $\overline{\text{DREQ0}}$  and  $\overline{\text{DREQ1}}$ ) of an external device. This mode is valid only in channels 0 and 1. The setting of the external request mode with the RS bits in CHCRn (n = 0, 1) is shown in table 12.5. When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon a request at the DREQ input.

**Table 12.5 Setting External Request Mode with RS Bits**

CHCR				Address Mode	Source	Destination
RS3	RS2	RS1	RS0			
0	0	0	0	Dual address mode	Any	Any

Choose to detect  $\overline{\text{DREQ}}$  by either the edge or level of the signal input with the DL bit and DS bit in CHCRn (n=0, 1) as shown in table 12.6. The source of the transfer request does not have to be the data transfer source or destination.

**Table 12.6 Selecting External Request Detection with DL, DS Bits**

CHCRn (n=0, 1)		
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  $\overline{\text{DREQ}}$  is accepted, the  $\overline{\text{DREQ}}$  pin becomes request accept disabled state. After issuing acknowledge signal  $\overline{\text{DACK}}$  for the accepted  $\overline{\text{DREQ}}$ , the  $\overline{\text{DREQ}}$  pin again becomes request accept enabled state.

When  $\overline{\text{DREQ}}$  is used by level detection, there are following two cases by the timing to detect the next  $\overline{\text{DREQ}}$  after outputting  $\overline{\text{DACK}}$ .

- Overrun 0: Transfer is aborted after the same number of transfer has been performed as requests.
- Overrun 1: Transfer is aborted after transfers have been performed for (the number of requests plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

**Table 12.7 Selecting External Request Detection with DO Bit**

CHCR	
DO	External Request
0	Overrun 0 (initial value)
1	Overrun 1

### (3) On-Chip Peripheral Module Request Mode

In this mode, a transfer is performed at the transfer request signal of an on-chip peripheral module. Transfer request signals comprise the transmit data empty transfer request and receive data full transfer request from the SCIF0 to SCIF2, USB, FLCTL and SRC set by DMARS0/1/2.

When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon the input of a transfer request signal.

When a transmit data empty transfer request of the SCIF0 is set as the transfer request, the transfer destination must be the SCIF0's transmit data register. Likewise, when receive data full transfer request of the SCIF0 is set as the transfer request, the transfer source must be the SCIF0's receive data register. These conditions also apply to the SCIF1, SCIF2, USB, FLCTL and SRC.

**Table 12.8 Selecting On-Chip Peripheral Module Request Modes with Bits RS[3:0]**

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
RS[3:0]	MID	RID					
1000	001000	01	SCI F0 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR0	Cycle steal
		10	SCIF0 receiver	RXI (receive FIFO data full interrupt)	SCFRDR0	Any	Cycle steal
	001010	01	SCI F1 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR1	Cycle steal
		10	SCIF1 receiver	RXI (receive FIFO data full interrupt)	SCFRDR1	Any	Cycle steal
	010000	01	SCIF2 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR2	Cycle steal
		10	SCIF2 receiver	RXI (receive FIFO data full interrupt)	SCFRDR2	Any	Cycle steal
	010001	01	USB transmitter*	Transmit data empty request	Any	USB D1FIFO	Cycle steal/burst
			USB receiver*	Receive data is not read	USB D1FIFO	Any	Cycle steal/burst
		10	USB transmitter*	Transmit data empty request	Any	USB D0FIFO	Cycle steal/burst
			USB receiver*	Receive data full request	USB D0FIFO	Any	Cycle steal/burst

CHCR RS[3:0]	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
	MID	RID					
1000	100000	11	FLCTL data part transmit	Transmit FIFO data empty request	Any	FLDTFIFO	Cycle steal
			FLCTL data part receive	Receive FIFO data full request	FLDTFIFO	Any	Cycle steal
	100001	11	FLCTL management code part transmit	Transmit FIFO data empty request	Any	FLECFIFO	Cycle steal
			FLCTL management code part receive	Receive FIFO data full request	FLECFIFO	Any	Cycle steal
	110000	01	SRC SRCOD	SRCOD FIFO data full request	SRC	Any	Cycle steal
		10	SRC SRCID	SRCID FIFO data empty request	Any	SRC	Cycle steal

Note: Transmitter or receiver is selected by the USB setting.

### 12.4.2 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it transfers data according to a predetermined priority. Two modes (fixed mode and round-robin mode) are selected by the bits PR[1:0] in DMAOR0.

#### (1) Fixed Mode

In this mode, the priority levels among the channels remain fixed. There are two kinds of fixed modes as follows:

- CH0 > CH1 > CH2 > CH3 > CH4 > CH5
- CH0 > CH2 > CH3 > CH1 > CH4 > CH5

These are selected by the bits PR[1:0] in DMAOR0

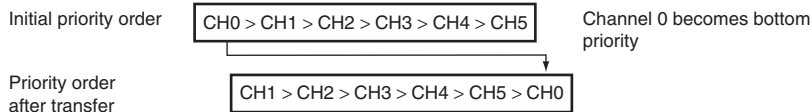
#### (2) Round-Robin Mode

In round-robin mode each time data of one transfer unit (word, byte, longword, 16-byte, or 32-byte unit) is transferred on one channel, the priority is rotated. The channel on which the transfer was just finished rotates to the bottom of the priority. The round-robin mode operation is shown in figure 12.2. The priority of round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 immediately after reset.

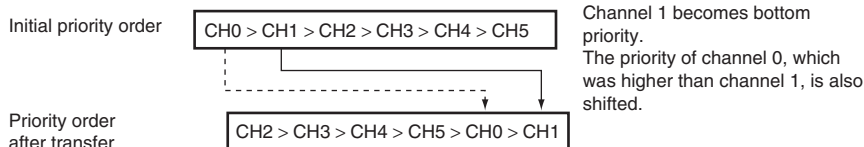
When round-robin mode is specified, do not mix the cycle steal mode and the burst mode in multiple channels' bus modes.



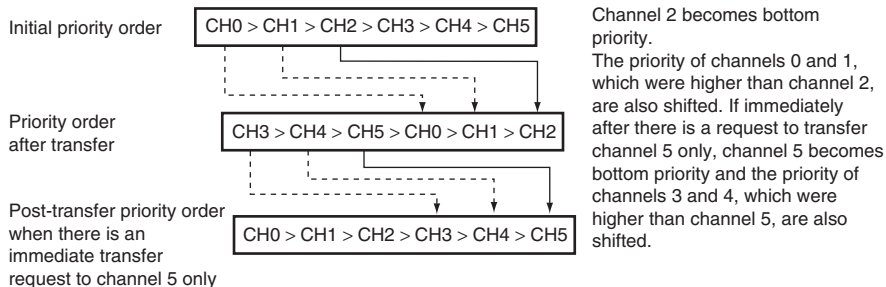
## (1) When channel 0 transfers



## (2) When channel 1 transfers



## (3) When channel 2 transfers



## (4) When channel 5 transfers

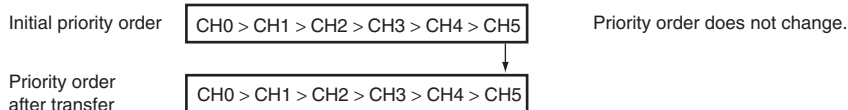
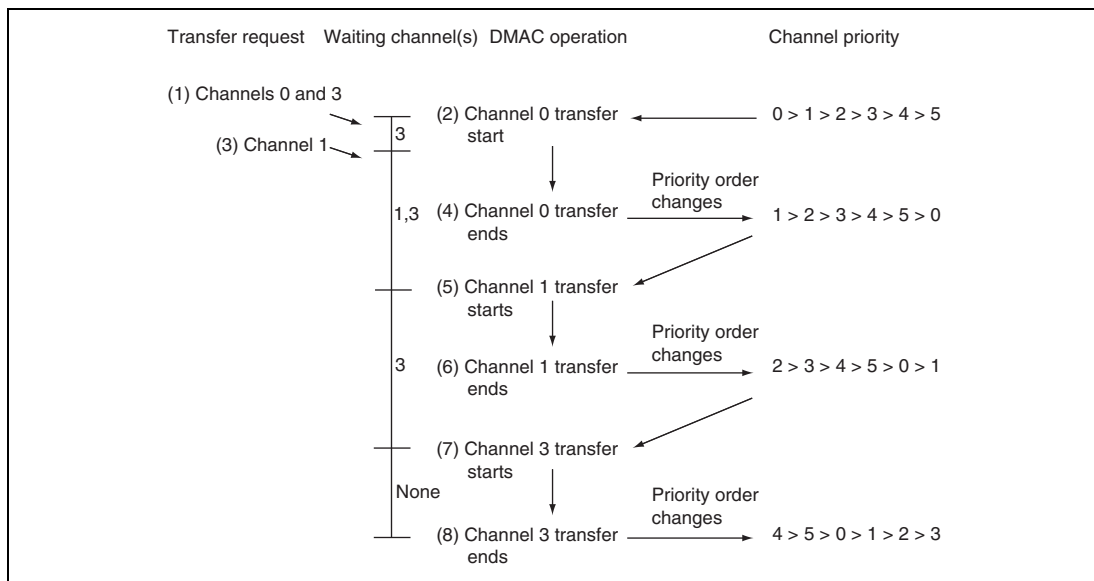
**Figure 12.2 Round-Robin Mode**

Figure 12.3 shows how the priority changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 becomes lowest priority.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 becomes lowest priority.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 shift downward in priority so that channel 3 becomes the lowest priority.



**Figure 12.3 Changes in Channel Priority in Round-Robin Mode**

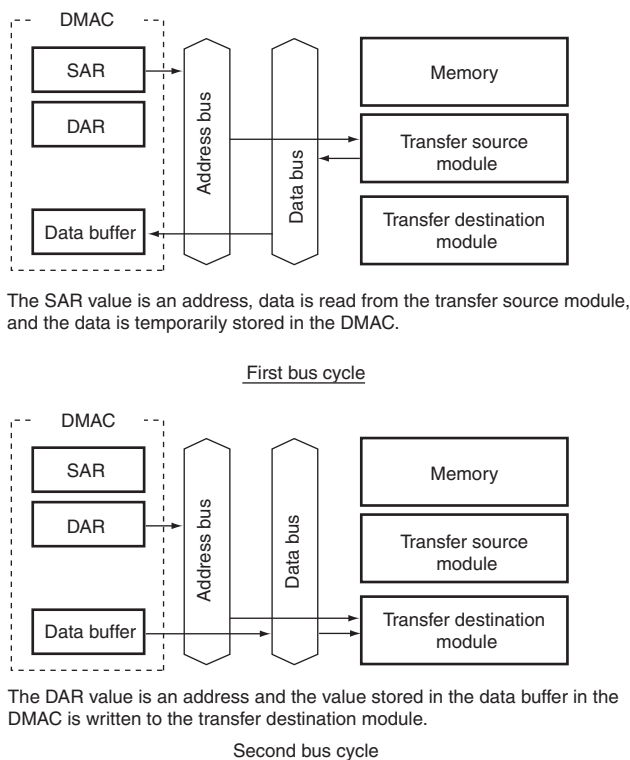
### 12.4.3 DMA Transfer Types

DMA transfer type is dual address mode transfer. A data transfer timing depends on the bus mode, which has cycle steal mode and burst mode.

#### (1) Dual Address Modes

In dual address mode, both the transfer source and destination are accessed by an address. The source and destination can be located externally or internally.

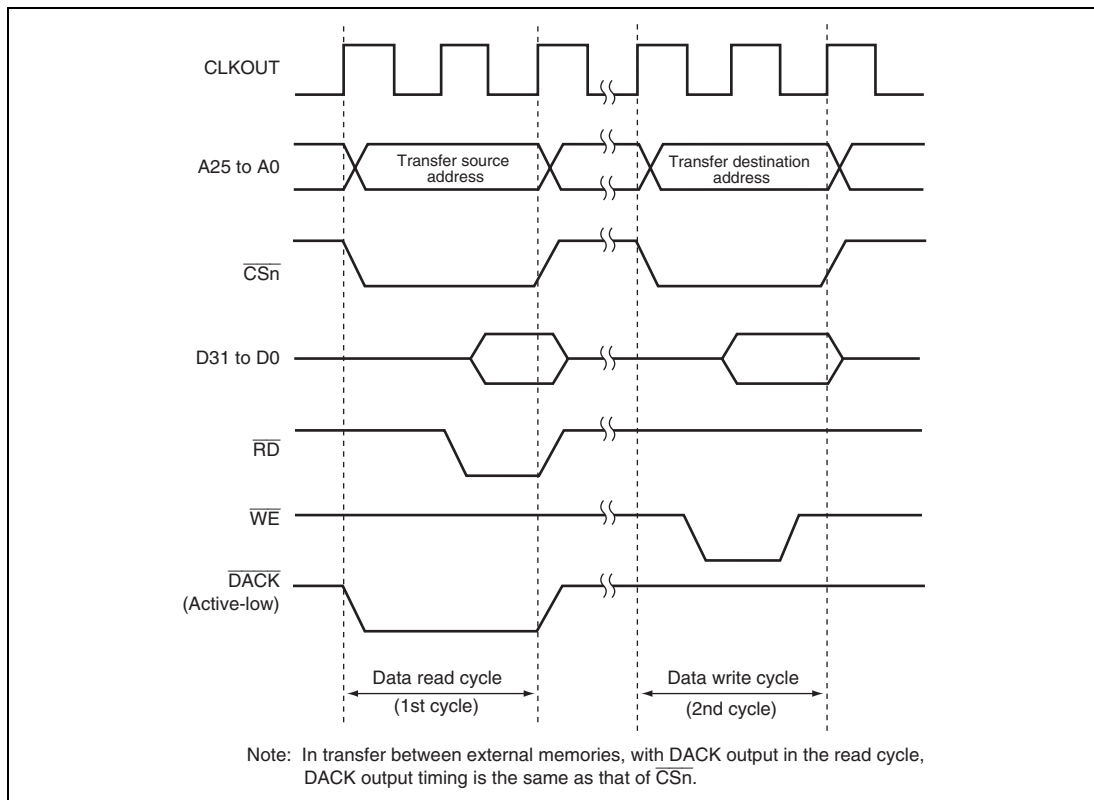
DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC. In the transfer between external memories as shown in figure 12.4, data is read to the DMAC from one external memory in a data read cycle, and then that data is written to the other external memory in a write cycle.



**Figure 12.4 Data Flow of Dual Address Mode**

Auto request, external request, and on-chip peripheral module request are available for the transfer request. DACK can be output in read cycle or write cycle in dual address mode. CHCR can specify whether the DACK is output in read cycle or write cycle.

Figure 12.5 shows an example of DMA transfer timing in dual address mode.



**Figure 12.5 Example of DMA Transfer Timing in Dual Address Mode**  
**(Source: Ordinary Memory, Destination: Ordinary Memory)**

## (2) Bus Modes

There are two bus modes: cycle steal mode and burst mode. Select the mode in the TB and LCKN bits in CHCR. And cycle steal mode has normal and intermittent modes that are specified by the CMS bits in DMAOR.

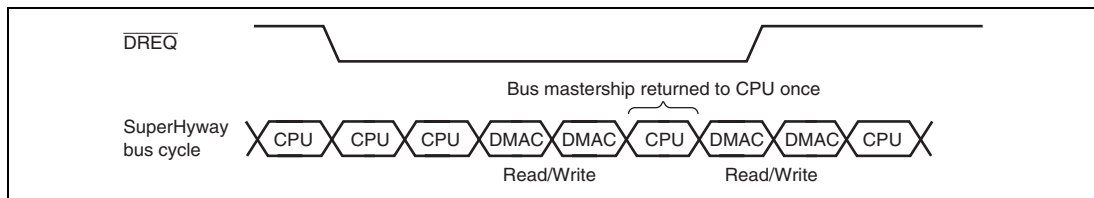
- Cycle-Steal Mode

- Normal mode1 (CHCR.LCKN = 0, CHCR.TB = 0)

In cycle-steal normal mode, the SuperHyway bus mastership is given to another bus master after a one-transfer unit (byte, word, longword, 16-byte, or 32-byte unit) DMA transfer. When the next transfer request occurs, the DMAC issues the next transfer request, the bus mastership is obtained from the other bus master and a transfer is performed for one-transfer unit. When that transfer ends, the bus mastership is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

In cycle-steal normal mode, transfer areas are not affected regardless of settings of the transfer request source, transfer source, and transfer destination.

Figure 12.6 shows an example of DMA transfer timing in cycle-steal normal mode. Transfer conditions shown in the figure are:

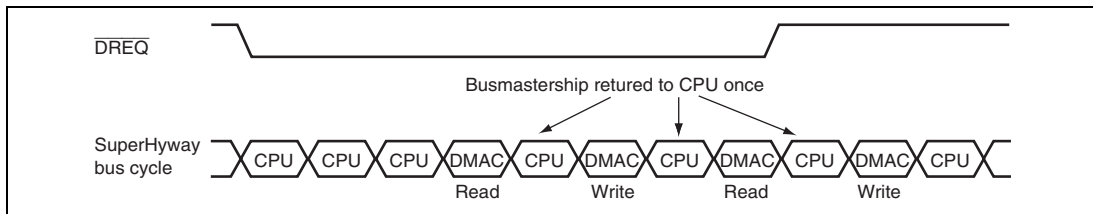


**Figure 12.6 DMA Transfer Timing Example in Cycle-Steal Normal Mode 1 (DREQ Low Level Detection)**

- Normal mode 2 (CHCR.LCKN = 1, CHCR.TB = 0)

In cycle steal normal mode 2, the DMAC does not keep the SuperHyway bus mastership, is to obtain the bus mastership in every one transfer unit of read or write cycle.

Figure 12.7 shows an example of DMA transfer timing in cycle steal normal mode 2.



**Figure 12.7 DMA Transfer Timing Example in Cycle-Steal Normal Mode 2 (DREQ Low Level Detection)**

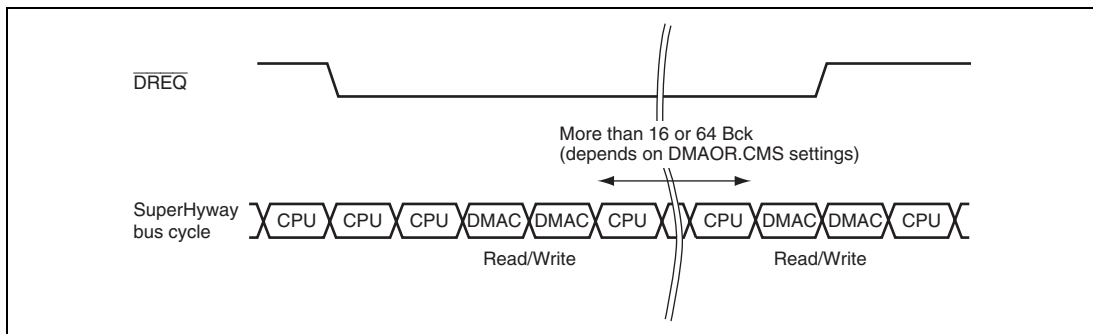
- Intermittent mode 16, intermittent mode 64 (CHCR.LCKN = 0 or 1, CHCR.TB = 0)

In intermittent mode of cycle steal, the DMAC returns the SuperHyway bus mastership to other bus master whenever a one-transfer unit (byte, word, longword, or 16-byte or 32-byte unit) is complete. If the next transfer request occurs after that, the DMAC issues the next transfer request after waiting for 16 or 64 clocks in Bck count, and obtains the bus mastership from other bus master. The DMAC then transfers data of one-transfer unit and returns the bus mastership to other bus master. These operations are repeated until the transfer end condition is satisfied. It is thus possible to make lower the ratio of bus occupation by DMA transfer than cycle-steal normal mode.

When the DMAC issues again the transfer request, DMA transfer can be postponed in case of entry updating due to cache miss.

This intermittent mode can be used for all transfer section; transfer request source, transfer source, and transfer destination. The bus modes, however, must be cycle steal mode in all channels.

Figure 12.8 shows an example of DMA transfer timing in cycle steal intermittent mode. Transfer conditions shown in the figure are:



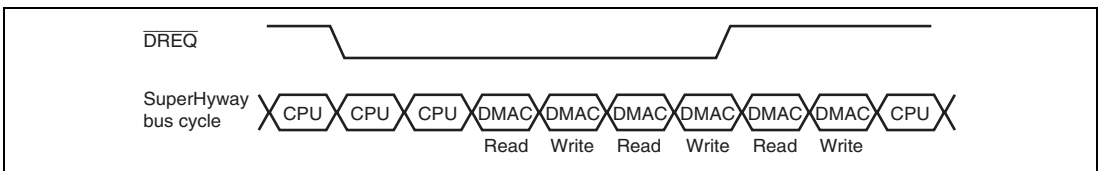
**Figure 12.8 Example of DMA Transfer Timing in Cycle Steal Intermittent Mode (DREQ Low Level Detection)**

- Burst Mode (LCKN = 0, TB = 1)

In burst mode, once the DMAC obtains the SuperHyway bus mastership, the transfer is performed continuously without releasing the bus mastership until the transfer end condition is satisfied. In external request mode with level detection of the DREQ pin, however, when the DREQ pin is not active, the bus mastership passes to the other bus master after the DMAC transfer request that has already been accepted ends, even if the transfer end conditions have not been satisfied.

Burst mode cannot be used when the on-chip peripheral module is the transfer request source.

Figure 12.9 shows DMA transfer timing in burst mode.



**Figure 12.9 DMA Transfer Timing Example in Burst Mode  
(DREQ Low Level Detection)**

### (3) DMA Transfer Matrix

Table 12.9 shows the DMA transfer matrix in auto-request mode and table 12.10 shows the DMA transfer matrix in external request mode, and table 12.11 shows the on-chip peripheral module request.

**Table 12.9 DMA Transfer Matrix in Auto-Request Mode**

Transfer Source	MCU Space	Transfer Destination	
		On-Chip Peripheral Module*	IL Memory
MCU space	Yes	Yes	Yes
On-chip peripheral module*	Yes	Yes	Yes
IL memory	Yes	Yes	Yes

[Legend]

Yes: Transfer is available.

Note: When the transfer source or destination is on-chip peripheral module register, the transfer size should be the same value of its access size.

**Table 12.10 DMA Transfer Matrix in External Request Mode (Only Channels 0 and 1)**

Transfer Source	MCU Space	Transfer Destination	
		On-Chip Peripheral Module*	IL Memory
MCU space	Yes	Yes	Yes
On-chip peripheral module*	Yes	Yes	Yes
IL memory	Yes	Yes	Yes

[Legend]

Yes: Transfer is available.

Note: When the transfer source or destination is on-chip peripheral module register, the transfer size should be the same value of its access size.



**Table 12.11 DMA Transfer Matrix in On-Chip Peripheral module Request Mode\*<sup>2</sup>**

Transfer Source	Transfer Destination		
	MCU Space	On-Chip Peripheral Module* <sup>1</sup>	IL Memory
MCU space	No	Yes	No
On-chip peripheral module* <sup>1</sup>	Yes	Yes	Yes
IL memory	No	Yes	No

[Legend]

Yes: Transfer is available.

No: Transfer is not available.

Notes: 1. When the transfer source or the destination is an on-chip peripheral module, the transfer size should be the same value of its register access size.

2. The transfer source or the transfer destination should be a register of request source in on-chip peripheral module request mode. This transfer is available only cycle steal mode.

#### (4) Bus Mode and Channel Priority

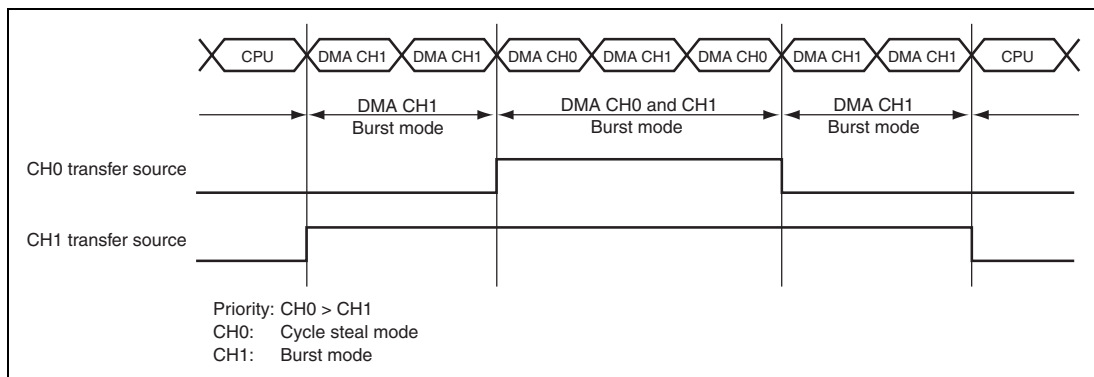
When the priority is set in fixed mode ( $CH0 > CH1$ ) and channel 1 is transferring in burst mode, if there is a transfer request to channel 0 with a higher priority, the transfer of channel 0 will begin immediately.

At this time, if channel 0 is also operating in burst mode, the channel 1 transfer will continue after the channel 0 transfer has completely finished.

When channel 0 is in cycle steal mode, channel 0 with a higher priority performs the transfer of one transfer unit and the channel 1 transfer is continuously performed without releasing the bus mastership. The bus mastership will then switch between the two in the order channel 0, channel 1, channel 0, and channel 1. In the bus status, the CPU cycle after data transfer in cycle steal mode is replaced with data transfer in burst mode. (Hereinafter, this bus status is referred to as burst mode prioritized execution.)

This example is shown in figure 12.10. When multiple channels are operating in burst modes, the channel with the highest priority is executed first.

When DMA transfer is executed in the multiple channels, the bus mastership will not be given to the bus master until all competing burst transfers are complete.



**Figure 12.10 Bus State when Multiple Channels are Operating**

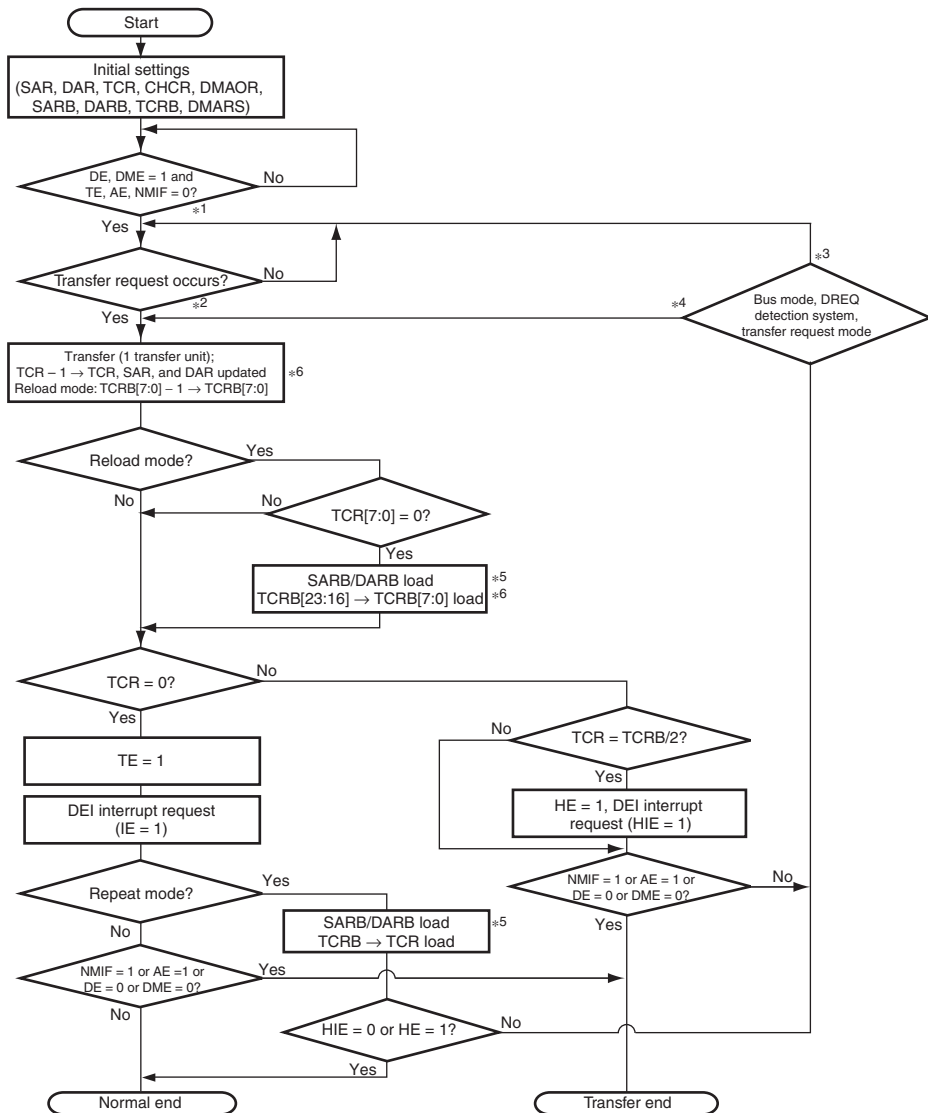
In round-robin mode, the priority changes according to the specification shown in figure 12.3. However, the channel in cycle steal mode cannot be mixed with the channel in burst mode.

#### 12.4.4 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA operation register (DMAOR), and DMA extended resource selectors (DMARS) are set, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request occurs while transfer is enabled, the DMAC transfers one transfer unit of data (depending on the TS0 and TS1 settings). In auto request mode, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented for each transfer. The actual transfer flows vary by address mode and bus mode.
3. When the specified number of transfer have been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
4. When an address error or an NMI interrupt is generated, the transfer is aborted. Transfers are also aborted when the DE bit in CHCR or the DME bit in DMAOR is changed to 0.

Figure 12.11 shows a flowchart of this procedure.



- Notes: 1. In repeat mode, a transfer request is accepted with TE = 1 when HIE = 1 and HE = 0 (half end interrupt is enable and clear the HE to 0 after HE is set to 1).
2. In auto-request mode, transfer starts when bits NMIF, AE, and TE are all 0 or bits TE and HIE are 1 and HE is 0 (in repeat mode), and bits DE and DME are set to 1.
3. DREQ is level detection (external request) in burst mode or cycle-steral mode.
4. DREQ is edge detection (external request) or auto request in burst mode.
5. Loading to SAR and DAR differs according to the operating conditions in each mode.

Figure 12.11 DMA Transfer Flowchart

### 12.4.5 Repeat Mode Transfer

In a repeat mode transfer, a DMA transfer is repeated without specifying the transfer settings every time before executing a transfer.

Using a repeat mode transfer with the half end function allows a double buffer transfer executed virtually. Following processings can be executed effectively by using a repeat mode transfer. As an example, operation of receiving data from the external memory and handling it is explained.

In the following example, handling 40-word data every data reception is explained.

1. DMAC settings

- Set address of the external memory in SAR
- Set address of an internal memory data store area in DAR
- Set TCR to H'50 (80 times)
- Satisfy the following settings of CHCR
  - Bits RPT[2:0] = B'010: Repeat mode (use DAR as a repeat area)
  - Bit HIE = B'1: TCR/2 interrupt generated
  - Bits DM[1:0] = B'01: DAR incremented
  - Bits SM[1:0] = B'00: SAR fixed
  - Bit IE = B'1: Interrupt enabled
  - Bit DE = B'1: DMA transfer enabled
- Set such as bits TB and TS[2:0] according to use conditions
- Set bits CMS[1:0] and PR[1:0] in DMAOR according to use conditions and set the DME bit to B'1

2. DMA transfer is executed.

3. TCR is decreased to half of its initial value and an interrupt is generated

After reading CHCR to confirm that the HE bit is set to 1 by an interrupt processing, clear the HE bit to 0 and handle 40-word data from the address set in DAR.

4. TCR is cleared to 0 and an interrupt is generated

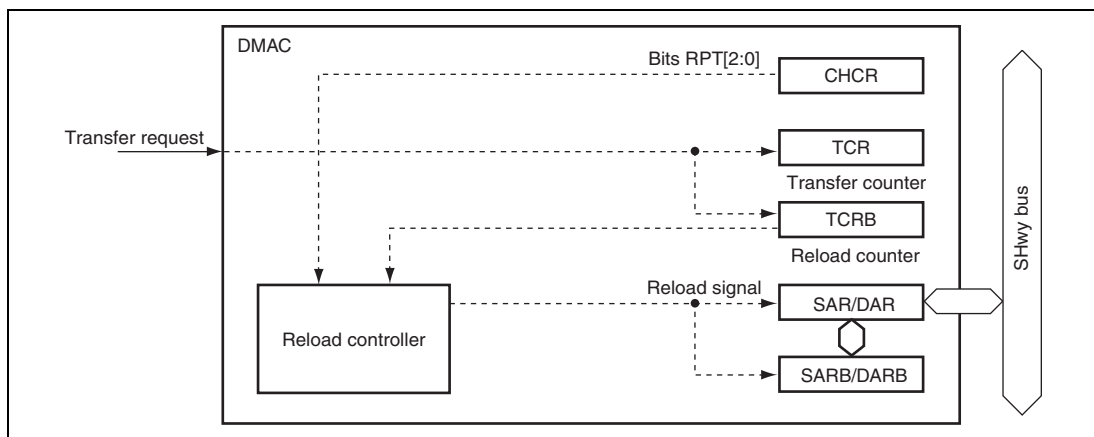
After reading CHCR to confirm that the TE bit is set to 1 by an interrupt processing, clear the TE bit to 0 and handle 40-word data from the address set in  $DAR + 40$ . After this operation, the value of DARB is copied to DAR in DMAC and initialized, and the value of TCRB is copied to TCR and initialized to 80.

5. Hereafter, steps 2 and 4 are repeated until the DME or DE bit is cleared to 0, or an NMI interrupt is generated.

As explained above, a repeat mode transfer enables sequential voice compression by changing buffer for storing data received consequentially and a data buffer for processing signals alternately.

### 12.4.6 Reload Mode Transfer

In a reload mode transfer, according to the settings of bits RPT[2:0] in CHCR, the value set in SARB/DARB is set to SAR/DAR and the value of bits TCRB[23:16] is set in bits TCRB[7:0] at each transfer set in the bits TCRB[7:0], and the transfer is repeated until TCR becomes 0 without specifying the transfer settings again. A reload mode transfer is effective when repeating data transfer with specific area. Figure 12.12 shows the operation of reload mode transfer.

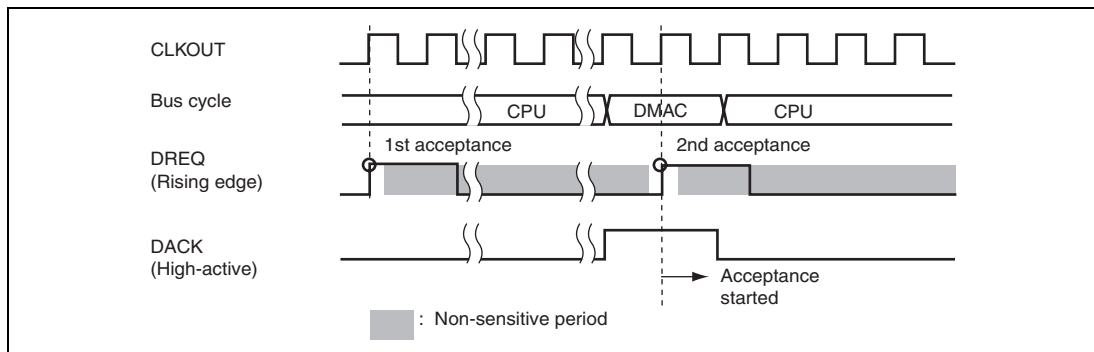


**Figure 12.12 Reload Mode Transfer**

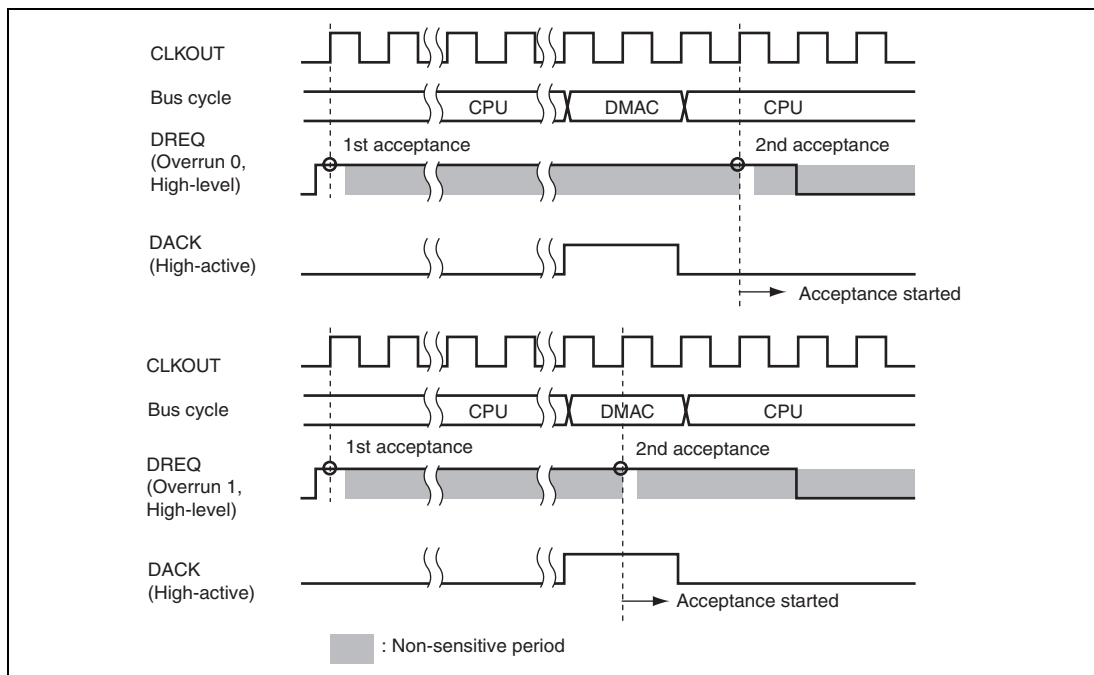
When a reload mode transfer is executed, TCRB is used as a reload counter. Set TCRB according to section 12.3.6, DMA Transfer Count Registers (TCRB0 to TCRB3).

### 12.4.7 DREQ Pin Sampling Timing

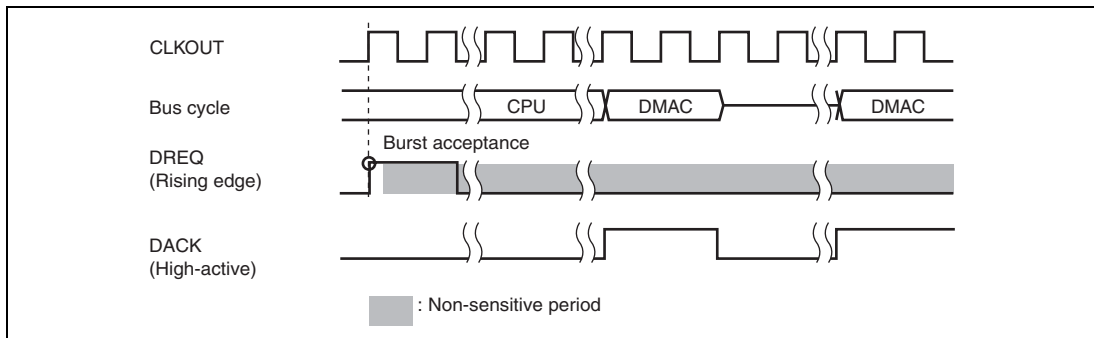
Figures 12.13 to 12.16 show the sample timing of the DREQ input in each bus mode, respectively.



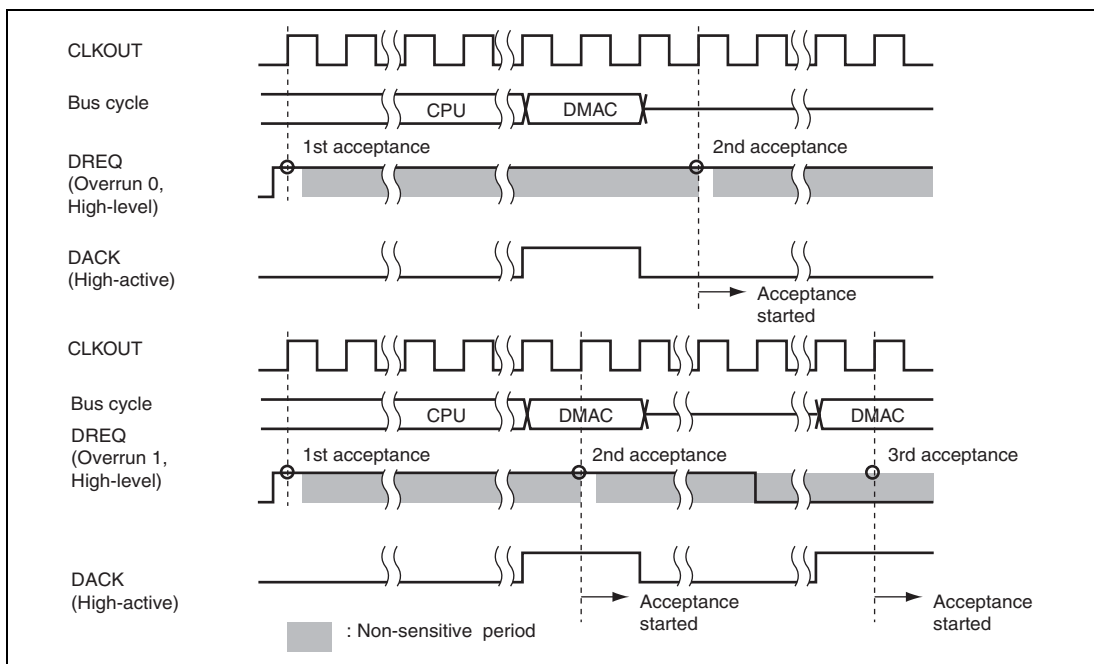
**Figure 12.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection**



**Figure 12.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection**

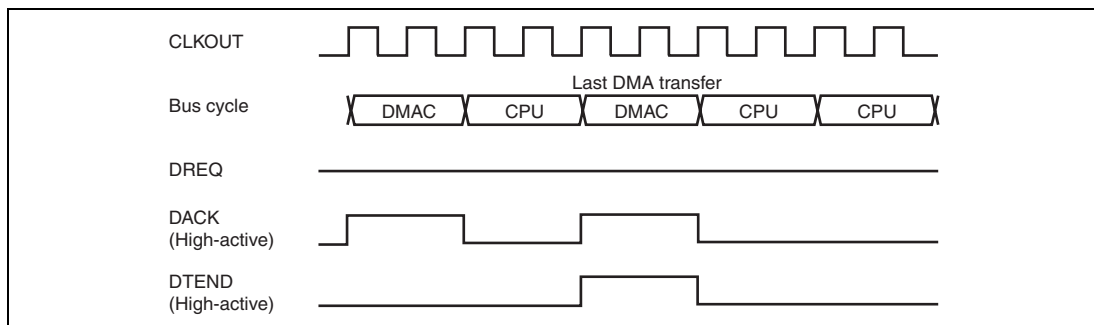


**Figure 12.15 Example of DREQ Input Detection in Burst Mode Edge Detection**



**Figure 12.16 Example of DREQ Input Detection in Burst Mode Level Detection**

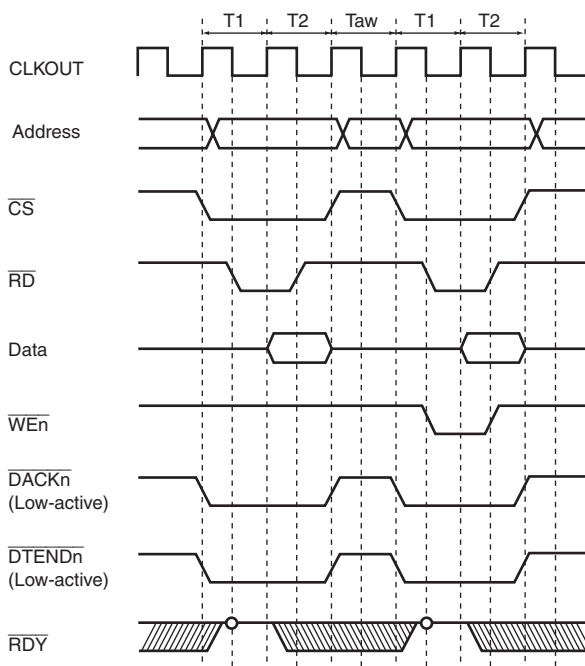
Figure 12.17 shows the timing of the DTEND output.



**Figure 12.17 DMA Transfer End Signal (Cycle Steal Mode Level Detection)**

Note that the DACK output and DTEND output are divided by 2 to align the data when an 8-bit or 16-bit external device is accessed in longword units, or when an 8-bit external device is accessed in word units. This example is shown in figure 12.18.





Note:  $\overline{DTEND}$  is asserted during the last transfer unit of the DMA transfer.  
 When the transfer unit is divided into several bus cycles and CS is negated between bus cycles,  $\overline{DTEND}$  is also divided.

**Figure 12.18 Example of BSC Ordinary Memory Access  
 (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)**

## 12.5 Usage Notes

Pay attentions to the following notes when the DMAC is used.

### 12.5.1 Module Stop

While DMAC is in operation, modules should not be stopped by setting MSTPCR (transition to the module standby state). When modules are stopped, transfer contents cannot be guaranteed.

### 12.5.2 Address Error

When a DMA address error is occurred, after execute the following procedure, and then start a transfer.

1. Dummy read for the below listed registers.
  - BCR (MCU)
  - INTCB3 (INTC)
  - IL memory
  - EESR (EtherC)
  - INTSTS0 (USB)
2. Issue the SYNCO instruction.
3. Set registers of all channels again.
  - If the AE bit in DMAOR is set to 1, channels 0 to 5 should be set again.

### 12.5.3 Notes on Burst Mode Transfer

During a burst mode transfer, following operation should not be executed until the transfer of corresponding channel has completed.

- Transition to sleep mode should not be made.
- Modules should not be stopped by setting the CPG register.

### **12.5.4 DACK Output Division and External Request**

The DMA transfer unit is divided into multiple bus cycles when a longword access is performed for an external device with 8-bit or 16-bit data bus, or when a word access is performed for an 8-bit external device.

Note that the DACK output is divided to align the data unit like the CS output when a setting is made so that a DMA transfer unit is divided into multiple bus cycles and the CS output is negated between bus cycles.

### **12.5.5 DMA Transfer to DMAC Prohibited**

Do not perform DMA transfer with the DMAC register specified as the transfer source or transfer destination.

### **12.5.6 NMI Interrupt**

When an NMI interrupt occurs, the DMA transfer is stopped. After returning from the NMI interrupt routine, set all channels again, and then restart the DMA transfer.



## Section 13 Interrupt Controller (INTC)

The interrupt controller (INTC) determines the priority of interrupt sources and controls interrupt requests to the CPU (SH-4A). The INTC has a register that sets the priority of each interrupt and interrupt requests are processed according to the priority set in this register by the user.

### 13.1 Features

SH-4 compatible specifications

- 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 levels for individual request sources.
- NMI noise canceller function  
An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception handling routine, the pin state can be checked, enabling it to be used as a noise canceller.
- NMI request masking when the block bit (BL) in the status register (SR) is set to 1  
Whether to mask NMI requests when the BL bit in SR is set to 1 can be selected.

Extended function for SH-4A

- Automatically updates the IMASK bit in SR according to the accepted interrupt level
- Thirty levels of on-chip module interrupt priority can be set  
By setting the interrupt priority registers (INT2PRI0 to INT2PRI12), the priorities of on-chip module interrupts can be selected from 30 levels for individual request sources.
- User-mode interrupt disabling function  
Specifying an interrupt mask level in the user interrupt mask level register (USERIMASK) disables interrupts which are not higher in priority than the specified mask level in user mode.

Figure 13.1 shows a block diagram of the INTC.

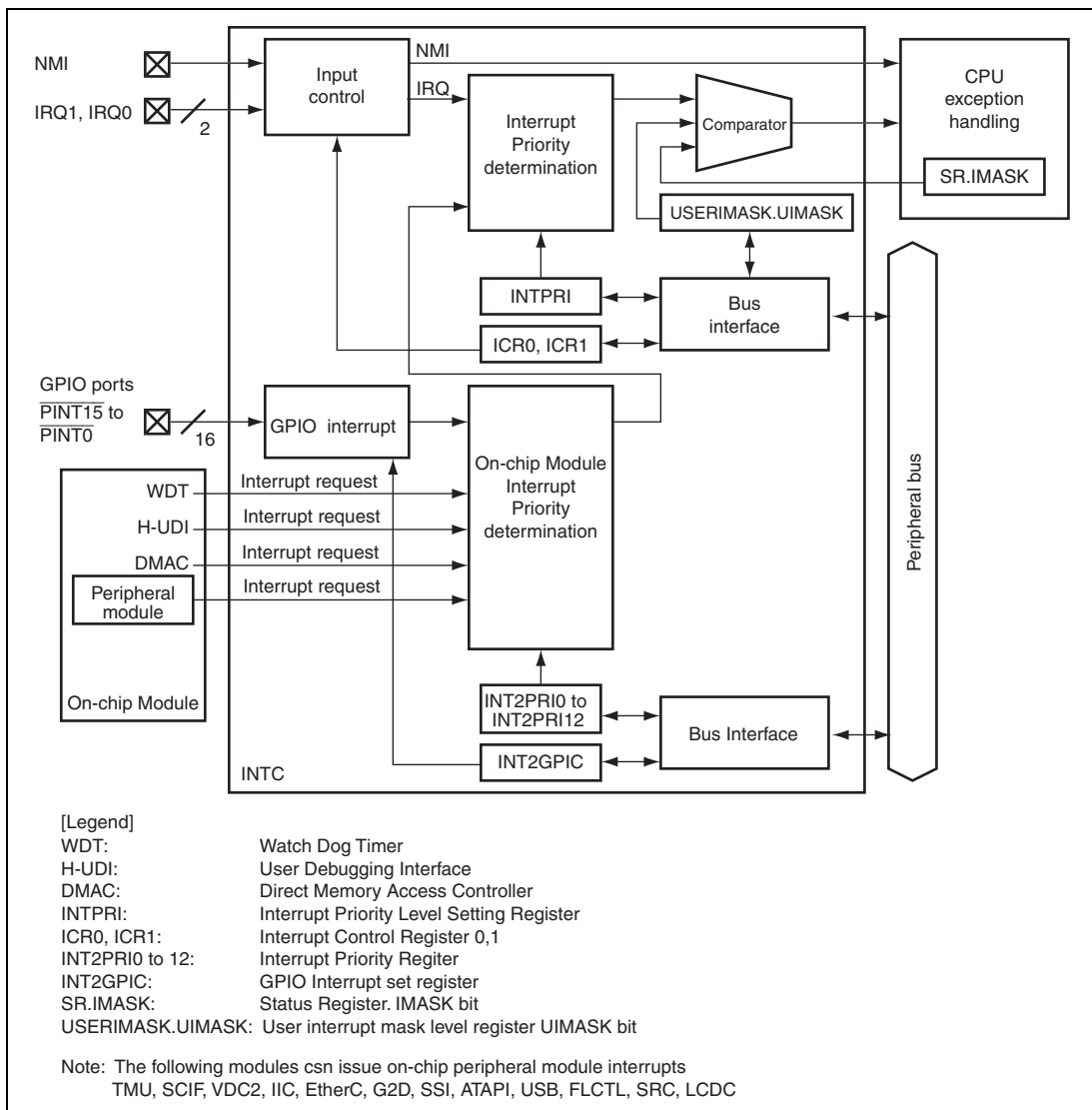


Figure 13.1 Block Diagram of INTC

### 13.1.1 Interrupt Method

The basic exception handling flow for the interrupt is as follows.

In interrupt 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 interrupt exception handling routine according to the 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 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.

1. The PC, SR and R15 contents are saved to 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.
4. The register bank (RB) bit in SR is set to 1.
5. 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 in the interrupt event register (INTEVT) of the exception source.
7. The processing is jumped to the start address of the interrupt exception handling routine, vector base register (VBR) + H'600.
8. The processing is branched to the vector address of the determined interrupt exception handling and the interrupt exception handling routine is started.

### 13.1.2 Interrupt Types in INTC

Table 13.1 shows an example of the interrupt types. The INTC supports both the external interrupts and on-chip module interrupts.

The external interrupts is the interrupt input from external pins, NMI and IRQ.

IRQ input can be selected as a level, or a rising or falling edge.

**Table 13.1 Interrupt Types**

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks
External interrupts	NMI	1	—	H'1C0	
	IRQ interrupt	2	Values set in INTPRI	H'240	IRQ0
				H'280	IRQ1
On-chip module interrupts	WDT	1	Setting value of INT2PRI0 to INT2PRI12	H'560	ITI
	TMU0	1		H'580	TUNI0
	TMU1	1		H'5A0	TUNI1
	TMU2	2		H'5C0	TUNI2
				H'5E0	TICPI2
				H'600	H-UDI
	LCDC	1		H'620	LCDCI
	DMAC	7(5/7)		H'640	DMINT0
				H'660	DMINT1
				H'680	DMINT2
				H'6A0	DMINT3
				H'6C0	DMAE
				H'700	ERI0
				H'720	RXI0
				H'740	BRI0
				H'760	TXI0
	DMAC	7(2/7)		H'780	DMINT4
				H'7A0	DMINT5
				H'860	VDCI
	IIC	1		H'8A0	IICI



Source		Number of Sources (Max.)	Priority	INTEVT	Remarks
On-chip module interrupts	EtherC	1	Setting value of INT2PRI0 to INT2PRI12	H'920	EINT
	G2D	1		H'980	G2DI
	SSI_A	4		H'A00	SSIDMA0
				H'A20	SSICH0
				H'A40	SSICH1
				H'A60	SSICH2
	SSI_B	4		H'AA0	SSIDMA1
				H'AC0	SSICH3
				H'AE0	SSICH4
				H'B00	SSICH5
	SCIF1	4		H'B80	ERI1
				H'BA0	RXI1
				H'BC0	BRI1
				H'BE0	TXI1
	ATAPI	1		H'C00	ATAI
	USB	1		H'C60	USBI
	FLCTL	4		H'D00	FLSTE
				H'D20	FLTEND
				H'D40	FLTRQ0
				H'D60	FLTRQ1
	TMU3	1		H'E00	TUNI3
	TMU4	1		H'E20	TUNI4
	TMU5	1		H'E40	TUNI5
	SRC	3		H'E80	SRC OVF
				H'EA0	SRC IDEI
				H'EC0	SRC ODFI
	SCIF2	4		H'F00	ERI2
				H'F20	RXI2
				H'F40	BRI2
				H'F60	TXI2

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks
On-chip module interrupts	GPIO	4	Setting value of INT2PRI0 to INT2PRI12	H'F80	CH0
				H'FA0	CH1
				H'FC0	CH2
				H'FE0	CH3

Notes: 1. ITI: Interval timer interrupt  
 TUNIO to TUNI5: TMU channel 0 to 5 under flow interrupt  
 TICPI2: TMU channel 2 input capture interrupt  
 DMINT0 to DMINT5: DMAC channel 0 to 5 transfer end interrupt  
 DMAE: DMAC address error interrupt (channel 0 to 5)  
 ERI0 to ERI2: SCIF channel 0 to 2 receive error interrupt  
 RXI0 to RXI2: SCIF channel 0 to 2 receive data full interrupt  
 BRI0 to BRI2: SCIF channel 0 to 2 break interrupt  
 TXI0 to TXI2: SCIF channel 0 to 2 transmission data empty interrupt

## 13.2 Input/Output Pins

Table 13.2 shows the pin configuration.

**Table 13.2 INTC Pin Configuration**

Pin Name	Function	I/O	Description
NMI	Nonmaskable interrupt input pin	Input	Nonmaskable interrupt request signal input
IRQ1, IRQ0	External interrupt input pin	Input	IRQ1 and IRQ0 interrupt request signal input
IRQOUT	Interrupt request output	Output	Notifies that an interrupt request has generated to the external devices.
PINT15 to PINT0	Port interrupt input pins	Input	Port interrupt request signal input

### 13.3 Register Descriptions

Table 13.3 shows the INTC register configuration. Table 13.4 shows the register states in each operating mode.

**Table 13.3 INTC Register Configuration**

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
Interrupt control register 0	ICR0	R/W	H'FFD0 0000	H'1FD0 0000	32
Interrupt control register 1	ICR1	R/W	H'FFD0 001C	H'1FD0 001C	32
Interrupt priority register	INTPRI	R/W	H'FFD0 0010	H'1FD0 0010	32
Interrupt source register	INTREQ	R/(W)	H'FFD0 0024	H'1FD0 0024	32
Interrupt mask register	INTMSK	R/W	H'FFD0 0044	H'1FD0 0044	32
Interrupt mask clear register	INTMSKCLR	R/W	H'FFD0 0064	H'1FD0 0064	32
NMI flag control register	NMIFCR	R/(W)	H'FFD0 00C0	H'1FD0 00C0	32
User interrupt mask level register	USERIMASK	R/W	H'FFD3 0000	H'1FD3 0000	32
Interrupt priority register 0	INT2PRI0	R/W	H'FFD4 0000	H'1FD4 0000	32
Interrupt priority register 1	INT2PRI1	R/W	H'FFD4 0004	H'1FD4 0004	32
Interrupt priority register 2	INT2PRI2	R/W	H'FFD4 0008	H'1FD4 0008	32
Interrupt priority register 3	INT2PRI3	R/W	H'FFD4 000C	H'1FD4 000C	32
Interrupt priority register 4	INT2PRI4	R/W	H'FFD4 0010	H'1FD4 0010	32
Interrupt priority register 5	INT2PRI5	R/W	H'FFD4 0014	H'1FD4 0014	32
Interrupt priority register 6	INT2PRI6	R/W	H'FFD4 0018	H'1FD4 0018	32
Interrupt priority register 7	INT2PRI7	R/W	H'FFD4 001C	H'1FD4 001C	32
Interrupt priority register 8	INT2PRI8	R/W	H'FFD4 00A0	H'1FD4 00A0	32
Interrupt priority register 9	INT2PRI9	R/W	H'FFD4 00A4	H'1FD4 00A4	32
Interrupt priority register 10	INT2PRI10	R/W	H'FFD4 00A8	H'1FD4 00A8	32
Interrupt priority register 11	INT2PRI11	R/W	H'FFD4 00AC	H'1FD4 00AC	32
Interrupt priority register 12	INT2PRI12	R/W	H'FFD4 00B0	H'1FD4 00B0	32
Interrupt source register 0 (mask state is not affected)	INT2A0	R	H'FFD4 0030	H'1FD4 0030	32
Interrupt source register 01 (mask state is not affected)	INT2A01	R	H'FFD4 00C0	H'1FD4 00C0	32

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
Interrupt source register 1 (mask state is affected)	INT2A1	R	H'FFD4 0034	H'1FD4 0034	32
Interrupt source register 11 (mask state is affected)	INT2A11	R	H'FFD4 00C4	H'1FD4 00C4	32
Interrupt mask register	INT2MSKR	R/W	H'FFD4 0038	H'1FD4 0038	32
Interrupt mask register 1	INT2MSKR1	R/W	H'FFD4 00D0	H'1FD4 00D0	32
Interrupt mask clear register	INT2MSKCR	W	H'FFD4 003C	H'1FD4 003C	32
Interrupt mask clear register 1	INT2MSKCR1	W	H'FFD4 00D4	H'1FD4 00D4	32
Individual module interrupt source register 0	INT2B0	R	H'FFD4 0040	H'1FD4 0040	32
Individual module interrupt source register 2	INT2B2	R	H'FFD4 0048	H'1FD4 0048	32
Individual module interrupt source register 3	INT2B3	R	H'FFD4 004C	H'1FD4 004C	32
Individual module interrupt source register 4	INT2B4	R	H'FFD4 0050	H'1FD4 0050	32
Individual module interrupt source register 5	INT2B5	R	H'FFD4 0054	H'1FD4 0054	32
Individual module interrupt source register 6	INT2B6	R	H'FFD4 0058	H'1FD4 0058	32
Individual module interrupt source register 7	INT2B7	R	H'FFD4 005C	H'1FD4 005C	32
GPIO interrupt set register	INT2GPIC	R/W	H'FFD4 0090	H'1FD4 0090	32

**Table 13.4 Register States in Each Operating Mode**

<b>Name</b>	<b>Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
Interrupt control register 0	ICR0	H'x000 0000	Retained	Retained
Interrupt control register 1	ICR1	H'0000 0000	Retained	Retained
Interrupt priority register	INTPRI	H'0000 0000	Retained	Retained
Interrupt source register	INTREQ	H'0000 0000	Retained	Retained
Interrupt mask register	INTMSK	H'FF00 0000	Retained	Retained
Interrupt mask clear register	INTMSKCLR	H'0000 0000	Retained	Retained
NMI flag control register	NMIFCR	H'x000 0000	Retained	Retained
User interrupt mask level register	USERIMASK	H'0000 0000	Retained	Retained
Interrupt priority register 0	INT2PRI0	H'0000 0000	Retained	Retained
Interrupt priority register 1	INT2PRI1	H'0000 0000	Retained	Retained
Interrupt priority register 2	INT2PRI2	H'0000 0000	Retained	Retained
Interrupt priority register 3	INT2PRI3	H'0000 0000	Retained	Retained
Interrupt priority register 4	INT2PRI4	H'0000 0000	Retained	Retained
Interrupt priority register 5	INT2PRI5	H'0000 0000	Retained	Retained
Interrupt priority register 6	INT2PRI6	H'0000 0000	Retained	Retained
Interrupt priority register 7	INT2PRI7	H'0000 0000	Retained	Retained
Interrupt priority register 8	INT2PRI8	H'0000 0000	Retained	Retained
Interrupt priority register 9	INT2PRI9	H'0000 0000	Retained	Retained
Interrupt priority register 10	INT2PRI10	H'0000 0000	Retained	Retained
Interrupt priority register 11	INT2PRI11	H'0000 0000	Retained	Retained
Interrupt priority register 12	INT2PRI12	H'0000 0000	Retained	Retained
Interrupt source register 0 (mask state is not affected)	INT2A0	H'xxxx xxxx	Retained	Retained
Interrupt source register 01 (mask state is affected)	INT2A01	H'xxxx xxxx	Retained	Retained
Interrupt source register 1 (mask state is affected)	INT2A1	H'0000 0000	Retained	Retained
Interrupt source register 11 (mask state is affected)	INT2A11	H'0000 0000	Retained	Retained
Interrupt mask register	INT2MSKR	H'FFFF FFFF	Retained	Retained
Interrupt mask register 1	INT2MSKR1	H'FFFF FFFF	Retained	Retained

Name	Abbreviation	Power-on Reset	Sleep	Standby
Interrupt mask clear register	INT2MSKCR	H'0000 0000	Retained	Retained
Interrupt mask clear register 1	INT2MSKCR1	H'0000 0000	Retained	Retained
Individual module interrupt source registers 0	INT2B0	H'xxxx xxxx	Retained	Retained
Individual module interrupt source registers 2	INT2B2	H'xxxx xxxx	Retained	Retained
Individual module interrupt source registers 3	INT2B3	H'xxxx xxxx	Retained	Retained
Individual module interrupt source registers 4	INT2B4	H'xxxx xxxx	Retained	Retained
Individual module interrupt source registers 5	INT2B5	H'xxxx xxxx	Retained	Retained
Individual module interrupt source registers 6	INT2B6	H'xxxx xxxx	Retained	Retained
Individual module interrupt source registers 7	INT2B7	H'xxxx xxxx	Retained	Retained
GPIO interrupt set register	INT2GPIC	H'0000 0000	Retained	Retained

### 13.3.1 Interrupt Control Register 0 (ICR0)

ICR0 sets the input signal detection mode of NMI pin, and indicates the input level to the NMI pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NMIL	MAI	—	—	—	—	NMIB	NMIE	—	—	—	—	—	—	—	—
Initial value:	—	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R/W	R	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	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	Undefined	R	<p>NMI Input Level</p> <p>Sets the signal level input to the NMI pin. Reading this bit allows the user to know the NMI pin level, and writing is invalid.</p> <p>0: Low level is input to the NMI pin</p> <p>1: High level is input to the NMI pin</p>
30	MAI	0	R/W	<p>MAI Interrupt Mask</p> <p>Specifies whether all interrupts are masked during the low level period of the NMI pin level regardless of the BL bit in SR of the CPU.</p> <p>0: Interrupts are enabled even if the NMI pin goes low</p> <p>1: Interrupts are disabled if the NMI pin goes low</p>
29 to 26	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
25	NMIB	0	R/W	<p>NMI Block Mode</p> <p>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.</p> <p>0: An NMI interrupt is held when the BL bit in SR is set to 1 (initial value)</p> <p>1: An NMI interrupt is not held when the BL bit in SR is set to 1</p> <p>Note: If interrupts are accepted with the BL bit in SR set to 1, previous exception information (SSR, SPC, SGR, and INTEVT) is lost.</p>
24	NMIE	0	R/W	<p>NMI Edge Select</p> <p>Selects whether an interrupt request signal to the NMI pin is detected at the rising edge or the falling edge.</p> <p>0: An interrupt request is detected at the falling edge of NMI input (initial value)</p> <p>1: An interrupt request is detected at the rising edge of NMI input</p>
23	—	1	R	<p>Reserved</p> <p>This bit is always read as 1. The write value should always be 1</p>
22 to 0	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>



### 13.3.2 Interrupt Control Register 1 (ICR1)

ICR1 is a 32-bit readable/writable register that specifies the individual input signal detection modes of external interrupt input pins IRQ1 and IRQ0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IRQ0S		IRQ1S		—	—	—	—	—	—	—	—	—	—	—	—
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:	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	0	R/W	IRQn Sense Select (n = 0, 1)
29, 28	IRQ1S	0	R/W	<p>Selects whether interrupt signals to the IRQ1 and IRQ0 pins are detected at the rising edge, falling edge, high level, or low level.</p> <p>00: Interrupt requests are detected at the falling edge of IRQn input</p> <p>01: Interrupt requests are detected at the rising edge of IRQn input</p> <p>10: Interrupt requests are detected at the low level of IRQn input</p> <p>11: Interrupt requests are detected at the high level of IRQn input</p>
27 to 0	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

**Note:** When the IRQ is set as level input (IRQnS[1] = 1), an interrupt source is held until the CPU accepts an interrupt (not always IRQ). Therefore even if an interrupt source is disabled before this LSI returns from sleep mode, it is guaranteed that processing is branched to the interrupt handler when this LSI returns from sleep mode. The held interrupt can be cleared by setting the corresponding interrupt mask bit (the IM bit in the interrupt mask register) to 1.

### 13.3.3 Interrupt Priority Register (INTPRI)

INTPRI is a 32-bit readable/writable register that sets the IRQ1 and IRQ0 interrupt priorities (levels 15 to 0).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IP0				IP1				—	—	—	—	—	—	—	—
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 28	IP0	H'0	R/W	Set priority of an independent interrupt request of IRQ0.
27 to 24	IP1	H'0	R/W	Set priority of an independent interrupt request of IRQ1.
23 to 0	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.

Interrupt priorities should be determined by setting a value from H'F to H'1 to each 4-bit field. If the value is larger, the priority is higher. When the value of H'0 is set to a field, a corresponding interrupt is masked (initial value).

### 13.3.4 Interrupt Source Register (INTREQ)

INTREQ is a 32-bit readable and writable with conditions register that indicates which IRQ [n] (n = 0, 1) interrupt is requested to the INTC.

Even if interrupts are masked by INTPRI and INTMSK, the INTREQ bits are not affected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IR0	IR1	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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

				Description	
Bit	Bit Name	Initial Value	R/W	At Edge Detection (ICR1.IRQnS = 00 or 01, n = 0, 1)	At Level Detection (ICR1.IRQnS = 10 or 11, n = 0, 1)
31	IR0	0	R/(W)	[When reading]	[When reading]
30	IR1	0	R/W	0: A corresponding IRQ interrupt request is not detected 1: A corresponding IRQ interrupt request is detected [When writing]* 0: Each bit is cleared by writing 0 after reading 1 1: Holds detected interrupt request Note: * Write 1 to the corresponding bit read as 0.	0: A corresponding IRQ interrupt pin is not asserted 1: A corresponding IRQ interrupt pin has asserted, but the CPU does not accept it yet Writing is ignored.
29 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	

### 13.3.5 Interrupt Mask Register (INTMSK)

INTMSK is 32-bit readable and writable with conditions registers that control mask settings for each IRQn (n = 0, 1) interrupt request. To clear the mask settings for interrupts, write 1 to the corresponding bits in INTMSKCLR. Writing 0 to bits in INTMSK is invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM00	IM01	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	IM00	1	R/W	Sets masking of an independent interrupt request of IRQ0. [When reading] 0: Interrupts are accepted 1: Interrupts are masked
30	IM01	1	R/W	Sets masking of an independent interrupt request of IRQ1. [When writing] 0: Invalid 1: Interrupts are masked
29 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

### 13.3.6 Interrupt Mask Clear Register (INTMSKCLR)

INTMSKCLR is 32-bit write-only registers that clear the mask settings for IRQn (n = 0, 1) interrupt requests. An undefined value is read.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IC00	IC01	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

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

Bit	Bit Name	Initial Value	R/W	Description	
31	IC00	Undefined	W	Clears masking of an independent interrupt request of IRQ0.	[When reading] An undefined value is read.
30	IC01	Undefined	W	Clears masking of an independent interrupt request of IRQ1.	[When writing] 0: Invalid
29 to 0	—	All Undefined	W	Reserved The write value should Always be 0.	1: Clears the corresponding interrupt mask (Interrupts are enabled)

### 13.3.7 NMI Flag Control Register (NMIFCR)

NMIFCR is a 32-bit readable and partially writable with conditions register that has an NMI flag (NMIFL bit) that can be read or cleared by software. The NMIFL bit is automatically set to 1 by hardware when an NMI interrupt is detected by the INTC. To clear the NMIFL bit, write 0 to the bit by software.

The NMIFL bit value does not affect NMI acceptance by the CPU. Although the NMI request detected by the INTC is cleared by CPU acceptance, 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:	—	0	0	0	0	0	0	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	Undefined	R	NMI Input Level Indicates the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified. 0: Low level is input to the NMI pin 1: High level is 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.

Bit	Bit Name	Initial Value	R/W	Description
16	NMIFL	0	R/(W)	<p>NMI Interrupt Request Signal Detection</p> <p>Indicates whether an NMI interrupt request signal has been detected. This bit is automatically set to 1 when the INTC detects an NMI interrupt request. Write 0 to clear the bit. Writing 1 is ignored.</p> <p>[When reading]</p> <p>1: NMI is detected</p> <p>0: NMI is not detected</p> <p>[When writing]</p> <p>0: The NMI flag is cleared</p> <p>1: Writing 1 is ignored</p>
15 to 0	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

### 13.3.8 User Interrupt Mask Level Register (USERIMASK)

USERIMASK is a 32-bit readable and writable with conditions register that sets the acceptable interrupt level. When addresses in area 7 are accessed using the MMU address translation function, USERIMASK can be accessed in user mode. Since only USERIMASK is allocated in the 64-Kbyte page (other INTC registers are allocated to a different area), it can be set to be accessed in user mode.

Interrupts whose priority levels are lower than the level set in the UIMASK bit are masked. If the value of H'F is set to the UIMASK bit, all interrupts other than the NMI are masked.

Interrupts whose priority levels are higher than the level set in the UIMASK bit 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 priority level set in the IMASK bit in SR is lower than that of the interrupt.

Even if interrupts are accepted, the UIMASK value is not changed.

USERIMASK is initialized to H'0000 0000 (all interrupts are enabled) when returning from a power-on reset.

To prevent incorrect writing, this register can only be written to with bits 31 to 24 set to H'A5.

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

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



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	H'00	R/W	To write a value to bits 7 to 4, write H'A5 to them. These bits are always read as 0.
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	0	R/W	Interrupt Mask Level Masks interrupts whose priority levels are lower than the level set in the UIMASK bit.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

#### Procedure for Using User Interrupt Mask Level Register

This function is used to save time by disabling interrupts whose priorities are low when a high priority interrupt is processed in the device driver.

Setting the interrupt mask level in USERIMASK disables interrupts having an equal or lower priority level than the specified mask level. This function can disable less-urgent interrupts in a task (such as device driver) operating in user mode to accelerate urgent processing.

USERIMASK is allocated to a different 64-Kbyte page than where the other INTC registers are allocated. When accessing this register in user mode, translate the address through the MMU. In the system that uses a multitasking OS, processes that can access USERIMASK must be controlled by using memory protection functions of the MMU. When terminating the task or switching to another task, be sure to clear USERIMASK to 0 before quitting the task. If the UIMASK bits are left set to a non-zero value, interrupts which are not higher in priority than the UIMASK level are held disabled, and correct operation may not be performed (for example, the OS cannot switch tasks).

An example of the usage procedure is shown below.

1. Classify interrupts to A and B as described below and set the A priority higher than the B priority.
  - A. Interrupts to be accepted in the device driver (interrupts to be used by the operating system: a timer interrupt etc.)
  - B. Interrupts to be disabled in the device driver

2. Make the MMU settings so that the address space including USERIMASK can only be accessed by the device driver in which interrupts should be disabled.
3. Branch to the device driver.
4. Set the UIMASK bit to mask B interrupts in the device driver that is operating in user mode.
5. Process interrupts with high priority in the device driver.
6. Clear the UIMASK bit to 0 to return from processing in the device driver.

### 13.3.9 On-Chip Module Interrupt Priority Registers (INT2PRI0 to INT2PRI12)

INT2PRI0 to INT2PRI12 are 32-bit readable/writable registers that set priorities (levels 31 to 0) of the on-chip peripheral module interrupts. INT2PRI0 to INT2PRI13 are initialized to H'0000 0000 by a reset.

INT2PRI0 to INT2PRI12 can set 30 priority levels (32 types of interrupt requests) to individual interrupt sources with five bits (interrupt requests are masked at H'00 and H'01).

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 13.5 shows the correspondence between interrupt request sources and bits in INT2PRI0 to INT2PRI12.

**Table 13.5 Interrupt Request Sources and INT2PRI0 to INT2PRI12**

Register	Bit			
	28 to 24	20 to 16	12 to 8	4 to 0
INT2PRI0	TMU0 (TUNI0)	TMU0 (TUNI1)	TMU0 (TUNI2)	TMU0 (TICPI2)
INT2PRI1	TMU1 (TUNI3)	TMU1 (TUNI4)	TMU1 (TUNI5)	Reserved
INT2PRI2	SCIF0	SCIF1	WDT	Reserved
INT2PRI3	H-UDI	DMAC	Reserved	Reserved
INT2PRI4	Reserved	G2D	SSI_A (SSIDMA0)	SSI_A (SSICH0)
INT2PRI5	SSI_A (SSICH1)	SSI_A (SSICH2)	Reserved	SSI_B
INT2PRI6	ATAPI	Reserved	FLCTL	SRC (OVF)
INT2PRI7	SCIF2	GPIO	Reserved	Reserved
INT2PRI8	Reserved	SRC (IDEI)	SRC (ODFI)	Reserved
INT2PRI9	LCDC	Reserved	Reserved	IIC
INT2PRI10	Reserved	Reserved	Reserved	Reserved
INT2PRI11	Reserved	Reserved	Reserved	Reserved
INT2PRI12	VDC2	Reserved	USB	EtherC

Note: If the value is larger, the priority is higher. Interrupt requests are masked at H'00 and H'01.  
For details, see the description above.

### 13.3.10 Interrupt Source Register 0 (Mask State is not affected) (INT2A0)

INT2A0 (mask state is not affected) is a 32-bit read-only register that indicates interrupt source modules. Even if interrupt masking is set in the interrupt mask register, INT2A0 indicates a source module in a corresponding bit (the corresponding interrupt is not generated). If source indication is not necessary depending on the state of the interrupt mask register, use INT2A1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SRC	FLCTL	—	ATAPI	SSI_B	—	SSI_A CH2	SSI_A CH1
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	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSI_A CH0	SSI_A DMA0	G2D	—	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	—	TMU1	TMU0
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	Function	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0.	Indicates interrupt sources for each peripheral module (INT2A0 is not affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated  Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A0 is not necessary.
25	GPIO	Undefined	R	Indicates GPIO interrupt source	
24	—	0	R	Reserved This bit is always read as 0.	
23	SRC	Undefined	R	Indicates SRCOVF interrupt source	
22	FLCTL	Undefined	R	Indicates FLCTL interrupt source	
21	—	0	R	Reserved This bit is always read as 0.	
20	ATAPI	Undefined	R	Indicates ATAPI interrupt source	
19	SSI_B	Undefined	R	Indicates SSI_B interrupt source	
18	—	0	R	Reserved This bit is always read as 0.	
17	SSI_ACH2	Undefined	R	Indicates SSI_A (SSICH2) interrupt source	
16	SSI_ACH1	Undefined	R	Indicates SSI_A (SSICH1) interrupt source	

Bit	Bit Name	Initial Value	R/W	Function	Description
15	SSI_ACH0	Undefined	R	Indicates SSI_A (SSICH0) interrupt source	Indicates interrupt sources for each peripheral module (INT2A0 is not affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A0 is not necessary.
14	SSI_A DMA0	Undefined	R	Indicates SSI_A (SSIDMA0) interrupt source	
13	G2D	Undefined	R	Indicates G2D interrupt source	
12 to 9	—	All 0	R	Reserved These bits are always read as 0.	
8	DMAC	Undefined	R	Indicates DMAC channels 0 to 5 interrupt source and address error interruption	
7	H-UDI	Undefined	R	Indicates H-UDI interrupt source	
6	—	0	R	Reserved This bit is always read as 0..	
5	WDT	Undefined	R	Indicates WDT interrupt source	
4	SCIF1	Undefined	R	Indicates SCIF1 interrupt source	
3	SCIF0	Undefined	R	Indicates SCIF0 interrupt source	
2	—	0	R	Reserved This bit is always read as 0.	
1	TMU1	Undefined	R	Indicates TMU1 interrupt source	
0	TMU0	Undefined	R	Indicates TMU0 interrupt source	

### 13.3.11 Interrupt Source Register 01 (Mask State is not affected) (INT2A01)

INT2A01 (mask state is not affected) is a 32-bit read-only register that indicates interrupt source modules. Even if interrupt masking is set in the interrupt mask register, INT2A01 indicates a source module in a corresponding bit (the corresponding interrupt is not generated). If source indication is not necessary depending on the state of the interrupt mask register, use INT2A11.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	—	—	—	—	—	VDC2	—	USB	EtherC
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
	—	—	—	—	—	—	—	—	LCDC	—	—	IIC	—	SRC ODFI	SRC IDEI	—
Initial value:	—	0	0	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	Function	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0.	<p>Indicates interrupt sources for each peripheral module (INT2A01 is not affected by the state of the interrupt mask register).</p> <p>0: No interrupts 1: Interrupts are generated</p> <p>Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A01 is not necessary. Writing to each bit is invalid. Each interrupt source is held in each on-chip module. However the SRCODFI or SRCIDEI bit is cleared to 0, if the Interrupt source for ODFI or IDEI of SRC is cleared.</p>
25	SCIF2	Undefined	R	Indicates SCIF2 interrupt source	
24 to 20	—	All 0	R	Reserved These bits are always read as 0.	
19	VDC2	Undefined	R	Indicates VDC2 interrupt source	
18	—	0	R	Reserved This bit is always read as 0.	
17	USB	Undefined	R	Indicates USB interrupt source	
16	EtherC	Undefined	R	Indicates EtherC interrupt source	
15 to 8	—	All 0	R	Reserved These bits are always read as 0.	
7	LCDC	Undefined	R	Indicates LCDC interrupt source	
6, 5	—	All 0	R	Reserved These bits are always read as 0.	
4	IIC	Undefined	R	Indicates IIC interrupt source	
3	—	0	R	Reserved This bit is always read as 0.	
2	SRCODFI	Undefined	R	Indicates SRCODFI interrupt source	
1	SRCIDEI	Undefined	R	Indicates SRCIDEI interrupt source	
0	—	0	R	Reserved This bit is always read as 0.	

### 13.3.12 Interrupt Source Register (Mask State is affected) (INT2A1)

INT2A1 (mask state is affected) is a 32-bit read-only register that indicates interrupt source modules. Note that if interrupt masking is set in the interrupt mask register, INT2A1 does not indicate a source module in a corresponding bit. To check whether interrupts are generated, regardless of the state of the interrupt mask register, use INT2A0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SRC	FLCTL	—	ATAPI	SSI_B	—	SSI_A CH2	SSI_A CH1
Initial value:	0	0	0	0	0	0	0	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
	SSI_A CH0	SSI_A DMA0	G2D	—	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	—	TMU1	TMU0
Initial value:	0	0	0	0	0	0	0	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	Function	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0.	Indicates interrupt sources for each peripheral module (INT2A1 is affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated  Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A1 is not necessary.
25	GPIO	0	R	Indicates GPIO interrupt source	
24	—	0	R	Reserved This bit is always read as 0.	
23	SRC	0	R	Indicates SRCOVF interrupt source	
22	FLCTL	0	R	Indicates FLCTL interrupt source	
21	—	0	R	Reserved This bit is always read as 0.	
20	ATAPI	0	R	Indicates ATAPI interrupt source	
19	SSI_B	0	R	Indicates SSI_B interrupt source	
18	—	0	R	Reserved This bit is always read as 0.	
17	SSI_ACH 2	0	R	Indicates SSI_A (SSICH2) interrupt source	
16	SSI_ACH 1	0	R	Indicates SSI_A (SSICH1) interrupt source	



Bit	Bit Name	Initial Value	R/W	Function	Description
15	SSI_ACH0	0	R	Indicates SSI_A (SSICH0) interrupt source	Indicates interrupt sources for each peripheral module (INT2A1 is affected by the state of the interrupt mask register).
14	SSI_ADM A0	0	R	Indicates SSI_A (SSIDMA0) interrupt source	
13	G2D	0	R	Indicates G2D interrupt source	
12 to 9	—	All 0	R	Reserved These bits are always read as 0.	
8	DMAC	0	R	Indicates interrupt source	Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A1 is not necessary.
7	H-UDI	0	R	Indicates H-UDI interrupt source	
6	—	0	R	Reserved This bit is always read as 0.	
5	WDT	0	R	Indicates WDT interrupt source	
4	SCIF1	0	R	Indicates SCIF1 interrupt source	0: No interrupts 1: Interrupts are generated
3	SCIF0	0	R	Indicates SCIF0 interrupt source	
2	—	0	R	Reserved This bit is always read as 0.	
1	TMU1	0	R	Indicates TMU1 interrupt source	
0	TMU0	0	R	Indicates TMU0 interrupt source	

### 13.3.13 Interrupt Source Register 11 (Mask State is affected) (INT2A11)

INT2A11 (mask state is affected) is a 32-bit read-only register that indicates interrupt source modules. Note that if interrupt masking is set in the interrupt mask register, INT2A11 does not indicate a source module in a corresponding bit. To check whether interrupts are generated, regardless of the state of the interrupt mask register, use INT2A01.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	—	—	—	—	—	VDC2	—	USB	EtherC
Initial value:	0	0	0	0	0	0	0	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
	—	—	—	—	—	—	—	—	LCDC	—	—	IIC	—	SRC ODFI	SRC IDEI	—
Initial value:	0	0	0	0	0	0	0	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	Function	Description
31 to 26	—	All 0	R	Reserved	Indicates interrupt sources for each peripheral module (INT2A11 is affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated  Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A11 is not necessary.
				These bits are always read as 0.	
25	SCIF2	0	R	Indicates SCIF2 interrupt source	
24 to 20	—	All 0	R	Reserved	
				These bits are always read as 0.	
19	VDC2	0	R	Indicates VDC2 interrupt source	
18	—	0	R	Reserved	
				This bit is always read as 0.	
17	USB	0	R	Indicates USB interrupt source	
16	EtherC	0	R	Indicates EtherC interrupt source	
15 to 8	—	All 0	R	Reserved	
				This bit is always read as 0.	
7	LCDC	0	R	Indicates LCDC interrupt source	
6, 5	—	All 0	R	Reserved	
				These bits are always read as 0.	
4	IIC	0	R	Indicates IIC interrupt source	

Bit	Bit Name	Initial Value	R/W	Function	Description
3	—	0	R	Reserved This bit is always read as 0.	Indicates interrupt sources for each peripheral module (INT2A11 is affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated  Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A11 is not necessary. Writing to each bit is invalid. Each interrupt source is held in each on-chip module. However the SRCODFI or SRCIDEI bit is cleared to 0, if the Interrupt source for ODFI or IDEI of SRC is cleared.
2	SRCODFI	0	R	Indicates SRCODFI interrupt source	
1	SRCIDEI	0	R	Indicates SRCIDEI interrupt source	
0	—	0	R	Reserved This bit is always read as 0.	

### 13.3.14 Interrupt Mask Register (INT2MSKR)

INT2MSKR is a 32-bit readable/writable register that sets masking for each source indicated in the interrupt source register. Interrupts whose corresponding bits in INT2MSKR are set to 1 are not notified to the CPU.

INT2MSKR is initialized to H'FFFF FFFF (mask state) by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SRC	FLCTL	—	ATAPI	SSI_B	—	SSI_A CH2	SSI_A CH1
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	R/W	R/W	R	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSI_A CH0	SSI_A DMA0	G2D	—	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	—	TMU1	TMU0
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	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 26	—	All 1	R	Reserved	Masks interrupts for each peripheral module.
				These bits are always read as 1. The write value should always be 1.	
25	GPIO	1	R/W	Masks GPIO interrupts	[When writing]
24	—	1	R	Reserved	0: Invalid
				This bit is always read as 1. The write value should always be 1.	1: Interrupts are masked
23	SRC	1	R/W	Masks SRCOVF interrupts	[When reading]
22	FLCTL	1	R/W	Masks FLCTL interrupts	0: No mask setting
21	—	1	R/W	Reserved	1: Mask setting
				This bit is always read as 1. The write value should always be 1.	
20	ATAPI	1	R/W	Masks ATAPI interrupts	
19	SSI_B	1	R/W	Masks SSI_B interrupts	
18	—	1	R	Reserved	
				This bit is always read as 1. The write value should always be 1.	
17	SSI_ACH2	1	R/W	Masks SSI_A (SSICH2) interrupts	

Bit	Bit Name	Initial Value	R/W	Function	Description
16	SSI_ACH1	1	R/W	Masks SSI_A (SSICH1) interrupts	Masks interrupts for each peripheral module. [When writing] 0: Invalid 1: Interrupts are masked [When reading]
15	SSI_ACH0	1	R/W	Masks SSI_A (SSICH0) interrupts	
14	SSI_ADMA0	1	R/W	Masks SSI_A (SSIDMA0) interrupts	
13	G2D	1	R/W	Masks G2D interrupts	
12 to 9	—	All 1	R/W	Reserved These bits are always read as 1. The write value should always be 1.	0: No mask setting 1: Mask setting
8	DMAC	1	R/W	Masks DMAC interrupts	
7	H-UDI	1	R/W	Masks H-UDI interrupts	
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.	
5	WDT	1	R/W	Masks WDT interrupts	
4	SCIF1	1	R/W	Masks SCIF1 interrupts	
3	SCIF0	1	R/W	Masks SCIF0 interrupts	
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.	
1	TMU1	1	R/W	Masks TMU1 interrupts	
0	TMU0	1	R/W	Masks TMU0 interrupts	

### 13.3.15 Interrupt Mask Register 1 (INT2MSKR1)

INT2MSKR1 is a 32-bit readable/writable register that sets masking for each source indicated in the interrupt source register. Interrupts whose corresponding bits in INT2MSKR1 are set to 1 are not notified to the CPU.

INT2MSKR1 is initialized to H'FFFF FFFF (mask state) by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	—	—	—	—	—	VDC2	—	USB	EtherC
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	R	R	R	R	R/W	R	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 26	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.	Masks interrupts for each peripheral module. [When writing]
25	SCIF2	1	R/W	Masks SCIF2 interrupts	0: Invalid
24 to 20	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.	1: Interrupts are masked [When reading]
19	VDC2	1	R/W	Masks VDC2 interrupts	0: No mask setting
18	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.	1: Mask setting
17	USB	1	R/W	Masks USB interrupts	
16	EtherC	1	R/W	Masks EtherC interrupts	
15 to 8	—	All 1	R	Reserved This bit is always read as 1. The write value should always be 1.	
7	LCDC	1	R/W	Masks LCDC interrupts	

Bit	Bit Name	Initial Value	R/W	Function	Description
6, 5	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.	Masks interrupts for each peripheral module. [When writing]
4	IIC	1	R/W	Masks IIC interrupts	0: Invalid
3	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.	1: Interrupts are masked [When reading]
2	SRCODFI	1	R/W	Masks SRCODFI interrupts	0: No mask setting
1	SRCIDEI	1	R/W	Masks SRCIDEI interrupts	1: Mask setting
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.	

### 13.3.16 Interrupt Mask Clear Register (INT2MSKCR)

INT2MSKCR is a 32-bit write-only register that clears any masking set in the interrupt mask register. Setting bits in this register to 1 clears the masking of the corresponding interrupt sources. Reading bits in this register is always 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SRC	FLCTL	—	ATAPI	SSI_B	—	SSI_A CH2	SSI_A CH1
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
	SSI_A CH0	SSI_A DMA0	G2D	—	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	—	TMU1	TMU0
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	Function	Description
31 to 26	—	All 0	W	Reserved The write value should always be 0	Clears interrupt masking for each peripheral module. [When writing]
25	GPIO	0	W	Clears GPIO interrupt masking	0: Invalid
24	—	0	W	Reserved The write value should always be 0	1: Interrupt mask is cleared
23	SRC	0	W	Clears SRCOVF interrupt masking	[When reading] Always 0
22	FLCTL	0	W	Clears FLCTL interrupt masking	
21	—	0	W	Reserved The write value should always be 0	
20	ATAPI	0	W	Clears ATAPI interrupt masking	
19	SSI_B	0	W	Clears SSI_B interrupt masking	
18	—	0	W	Reserved The write value should always be 0	



Bit	Bit Name	Initial Value	R/W	Function	Description
17	SSI_ACH2	0	W	Clears SSI_A (SSICH2) interrupt masking	Clears interrupt masking for each peripheral module. [When writing] 0: Invalid 1: Interrupt mask is cleared [When reading] Always 0
16	SSI_ACH1	0	W	Clears SSI_A (SSICH1) interrupt masking	
15	SSI_ACH0	0	W	Clears SSI_A (SSICH0) interrupt masking	
14	SSI_ADMA0	0	W	Clears SSI_A (SSIDMA0) interrupt masking	
13	G2D	0	W	Clears G2D interrupt masking	Always 0
12 to 9	—	All 0	W	Reserved The write value should always be 0	
8	DMAC	0	W	Clears DMAC interrupt masking	
7	H-UDI	0	W	Clears H-UDI interrupt masking	
6	—	0	W	Reserved The write value should always be 0	
5	WDT	0	W	Clears WDT interrupt masking	
4	SCIF1	0	W	Clears SCIF1 interrupt masking	
3	SCIF0	0	W	Clears SCIF0 interrupt masking	
2	—	0	W	Reserved The write value should always be 0	
1	TMU1	0	W	Clears TMU1 interrupt masking	
0	TMU0	0	W	Clears TMU0 interrupt masking	

### 13.3.17 Interrupt Mask Clear Register 1 (INT2MSKCR1)

INT2MSKCR1 is a 32-bit write-only register that clears any masking set in the interrupt mask register. Setting bits in this register to 1 clears the masking of the corresponding interrupt sources. Reading bits in this register is always 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	—	—	—	—	—	VDC2	—	USB	EtherC
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
	—	—	—	—	—	—	—	—	LCDC	—	—	IIC	—	SRC ODFI	SRC IDEI	—
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	Function	Description
31 to 26	—	All 0	W	Reserved The write value should always be 0	Clears interrupt masking for each peripheral module.
25	SCIF2	0	W	Clears SCIF2 interrupt masking	[When writing] 0: Invalid
24 to 20	—	All 0	W	Reserved The write value should always be 0	1: Interrupt mask is cleared
19	VDC2	0	W	Clears VDC2 interrupt masking	[When reading] Always 0
18	—	0	W	Reserved The write value should always be 0	
17	USB	0	W	Clears USB interrupt masking	
16	EtherC	0	W	Clears EtherC interrupt masking	
15 to 8	—	All 0	W	Reserved The write value should always be 0	
7	LCDC	0	W	Clears LCDC interrupt masking	
14 to 5	—	All 0	W	Reserved The write value should always be 0	

Bit	Bit Name	Initial Value	R/W	Function	Description
4	IIC	0	W	Clears IIC interrupt masking	Clears interrupt masking for each peripheral module.
3	—	0	W	Reserved The write value should always be 0	[When writing] 0: Invalid
2	SRCODFI	0	W	Clears SRCODFI interrupt masking	1: Interrupt mask is cleared
1	SRCIDEI	0	W	Clears SRCIDEI interrupt masking	[When reading] Always 0
0	—	0	W	Reserved The write value should always be 0	

### 13.3.18 On-Chip Module Interrupt Source Registers (INT2B0 and INT2B2 to INT2B7)

INT2B0 and INT2B2 to INT2B7 are 32-bit read-only registers that indicate detailed sources for interrupt source modules indicated in the interrupt source register. INT2B0 and INT2B2 to INT2B7 are not affected by the mask state of the interrupt mask register. When mask setting is made for individual detailed sources, set the interrupt mask register or interrupt enable register in the corresponding modules.

The initial value of these registers is undefined (reserve bit is always read as 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

**INT2B0:** Indicates detailed interrupt sources for the TMU.

Module	Bit	Source	Function	Description
TMU	31 to 7	—	These bits are always read as 0. The write value should always be 0.	Indicates TMU interrupt sources. This register indicates the TMU interrupt sources even if mask setting is made in the interrupt mask register for them.
	6	TUNI5	TMU channel 5 underflow interrupt	
	5	TUNI4	TMU channel 4 underflow interrupt	
	4	TUNI3	TMU channel 3 underflow interrupt	
	3	TICPI2	TMU channel 2 input capture interrupt	
	2	TUNI2	TMU channel 2 underflow interrupt	
	1	TUNI1	TMU channel 1 underflow interrupt	
	0	TUNI0	TMU channel 0 underflow interrupt	

**INT2B2:** Indicates detailed interrupt sources for the SCIF.

Module	Bit	Source	Function	Description
SCIF1	31 to 8	—	These bits are always read as 0. The write value should always be 0.	Indicates SCIF interrupt sources. This register indicates the SCIF interrupt sources even if mask setting is made in the interrupt mask register for them.
	7	TXI1	SCIF channel 1 transmit FIFO data empty interrupt	
	6	BRI1	SCIF channel 1 break interrupt or overrun error interrupt	
	5	RXI1	SCIF channel 1 receive FIFO data full interrupt or receive data ready interrupt	
	4	ERI1	SCIF channel 1 receive error interrupt	
SCIF0	3	TXI0	SCIF channel 0 transmit FIFO data empty interrupt	
	2	BRI0	SCIF channel 0 break interrupt or overrun error interrupt	
	1	RXI0	SCIF channel 0 receive FIFO data full interrupt or receive data ready interrupt	
	0	ERI0	SCIF channel 0 receive error interrupt	

**INT2B3:** Indicates detailed interrupt sources for the DMAC.

Module	Bit	Source	Function	Description
DMAC	31 to 13	—	These bits are always read as 0. The write value should always be 0.	Indicates DMAC interrupt sources. This register indicates DMAC interrupt sources even if mask setting is made in the interrupt mask register for them.
	12	DMAE	DMA channels 0 to 5 address error interrupt	
	11 to 6	—	These bits are always read as 0. The write value should always be 0.	
	5	DMINT5	Channel 5 DMA transfer end/half-end interrupt	
	4	DMINT4	Channel 4 DMA transfer end/half-end interrupt	
	3	DMINT3	Channel 3 DMA transfer end/half-end interrupt	

Module	Bit	Source	Function	Description
DMAC	2	DMINT2	Channel 2 DMA transfer end/half-end interrupt	Indicates DMAC interrupt sources. This register indicates DMAC interrupt sources even if mask setting is made in the interrupt mask register for them.
	1	DMINT1	Channel 1 DMA transfer end/half-end interrupt	
	0	DMINT0	Channel 0 DMA transfer end/half-end interrupt	

**INT2B4:** Indicates detailed interrupt sources for the SSI.

Module	Bit	Source	Function	Description
SSI	31 to 9	—	These bits are always read as 0. The write value should always be 0.	Indicates SSI interrupt sources. This register indicates the SSI interrupt sources even if mask setting is made in the interrupt mask register for them.
	8	SSICH5	SSI ch5 interrupt	
	7	SSICH4	SSI ch4 interrupt	
	6	SSICH3	SSI ch3 interrupt	
	5	SSIDMA1	SSI DMA1 interrupt	
	4	—	These bits are always read as 0. The write value should always be 0.	
	3	SSICH2	SSI ch2 interrupt	
	2	SSICH1	SSI ch1 interrupt	
	1	SSICH0	SSI ch0 interrupt	
	0	SSIDMA0	SSI DMA0 interrupt	

**INT2B5:** Indicates detailed interrupt sources for the FLCTL.

Module	Bit	Source	Function	Description
FLCTL	31 to 4	—	These bits are always read as 0. The write value should always be 0.	Indicates FLCTL interrupt sources. This register indicates FLCTL interrupt sources even if mask setting is made in the interrupt mask register for them.
	3	FLTRQ1	FLCTL FLECFIFO transfer request interrupt	
	2	FLTRQ0	FLCTL TLDFIFO transfer request interrupt	
	1	FLTEND	FLCTL transfer end interrupt	
	0	FLSTE	FLCTL status error or ready/busy timeout error interrupt	

**INT2B6:** Indicates detailed interrupt sources for the SCIF2.

Module	Bit	Source	Function	Description
SCIF2	31 to 4	—	These bits are always read as 0. The write value should always be 0.	Indicates SCIF2 interrupt sources. This register indicates the SCIF2 interrupt sources even if mask setting is made in the interrupt mask register for them.
	3	TXI2	SCIF channel 2 transmit FIFO data empty interrupt	
	2	BRI2	SCIF channel 2 break interrupt or overrun error interrupt	
	1	RXI2	SCIF channel 2 receive FIFO data full interrupt or receive data ready interrupt	
	0	ERI2	SCIF channel 2 receive error interrupt	

**INT2B7:** Indicates detailed interrupt sources for the GPIO.

Module	Bit	Source	Function	Description
GPIO	31 to 28	—	These bits are always read as 0. The write value should always be 0.	Indicates GPIO interrupt sources. This register indicates GPIO interrupt sources even if mask setting is made in the interrupt mask register for them.
	27	PINT15I	GPIO interrupt from $\overline{\text{PINT15}}$ pin	
	26	PINT14I	GPIO interrupt from $\overline{\text{PINT14}}$ pin	
	25	PINT13I	GPIO interrupt from $\overline{\text{PINT13}}$ pin	
	24	PINT12I	GPIO interrupt from $\overline{\text{PINT12}}$ pin	
	23 to 20	—	These bits are always read as 0. The write value should always be 0.	
	19	PINT11I	GPIO interrupt from $\overline{\text{PINT11}}$ pin	
	18	PINT10I	GPIO interrupt from $\overline{\text{PINT10}}$ pin	
	17	PINT9I	GPIO interrupt from $\overline{\text{PINT9}}$ pin	
	16	PINT8I	GPIO interrupt from $\overline{\text{PINT8}}$ pin	
	15 to 12	—	These bits are always read as 0. The write value should always be 0.	
	11	PINT7I	GPIO interrupt from $\overline{\text{PINT7}}$ pin	
	10	PINT6I	GPIO interrupt from $\overline{\text{PINT6}}$ pin	
	9	PINT5I	GPIO interrupt from $\overline{\text{PINT5}}$ pin	
	8	PINT4I	GPIO interrupt from $\overline{\text{PINT4}}$ pin	
	7 to 4	—	These bits are always read as 0. The write value should always be 0.	
	3	PINT3I	GPIO interrupt from $\overline{\text{PINT3}}$ pin	
	2	PINT2I	GPIO interrupt from $\overline{\text{PINT2}}$ pin	
	1	PINT1I	GPIO interrupt from $\overline{\text{PINT1}}$ pin	
	0	PINT0I	GPIO interrupt from $\overline{\text{PINT0}}$ pin	



### 13.3.19 GPIO Interrupt Set Register (INT2GPIC)

INT2GPIC enables interrupt requests input from the following pins: PA0 to PA7 and PB0 to PB7.

A GPIO interrupt is a low active and level-sense signal. Before enabling an interrupt request, set the corresponding pin as an input with the corresponding port control register (PTIO\_A, PTIO\_B). For the port control registers, see section 27, General Purpose I/O (GPIO).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PINT 15E	PINT 14E	PINT 13E	PINT 12E	—	—	—	—	PINT 11E	PINT 10E	PINT 9E	PINT 8E
Initial 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/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PINT 7E	PINT 6E	PINT 5E	PINT 4E	—	—	—	—	PINT 3E	PINT 2E	PINT 1E	PINT 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/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	Function	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	Enables a GPIO interrupt request for each pin. 0: Disables an interrupt request 1: Enables an interrupt request
27	PINT15E	0	R/W	Enables a GPIO interrupt request from PINT15 pin	
26	PINT14E	0	R/W	Enables a GPIO interrupt request from PINT14 pin	
25	PINT13E	0	R/W	Enables a GPIO interrupt request from PINT13 pin	
24	PINT12E	0	R/W	Enables a GPIO interrupt request from PINT12 pin	
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	
19	PINT11E	0	R/W	Enables a GPIO interrupt request from PINT11 pin	

Bit	Bit Name	Initial Value	R/W	Function	Description
18	PINT10E	0	R/W	Enables a GPIO interrupt request from $\overline{\text{PINT10}}$ pin	Enables a GPIO interrupt request for each pin. 0: Disables an interrupt request 1: Enables an interrupt request
17	PINT9E	0	R/W	Enables a GPIO interrupt request from PINT9 pin	
16	PINT8E	0	R/W	Enables a GPIO interrupt request from $\overline{\text{PINT8}}$ pin	
15 to 12	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.	
11	PINT7E	0	R/W	Enables a GPIO interrupt request from $\overline{\text{PINT7}}$ pin	
10	PINT6E	0	R/W	Enables a GPIO interrupt request from PINT6 pin	
9	PINT5E	0	R/W	Enables a GPIO interrupt request from $\overline{\text{PINT5}}$ pin	
8	PINT4E	0	R/W	Enables a GPIO interrupt request from $\overline{\text{PINT4}}$ pin	
7 to 4	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.	
3	PINT3E	0	R/W	Enables a GPIO interrupt request from $\overline{\text{PINT3}}$ pin	
2	PINT2E	0	R/W	Enables a GPIO interrupt request from $\overline{\text{PINT2}}$ pin	
1	PINT1E	0	R/W	Enables a GPIO interrupt request from $\overline{\text{PINT1}}$ pin	
0	PINT0E	0	R/W	Enables a GPIO interrupt request from PINT0 pin	

When GPIO ports are used as interrupt ports, if the GPIO detects an interrupt, the interrupt is notified to the INTC from the GPIO. However, it is indicated as a one-bit source in the INT2A0 or INT2A1 register of the INTC. Referring to the on-chip module interrupt source register INT2B7. Referring to the INTEVT code in the CPU can specify from which port group an interrupt is generated.

## 13.4 Interrupt Sources

There are three types of interrupt sources: NMI, IRQ, and on-chip modules. Each interrupt has a priority level (16 to 0), with level 16 as the highest and level 1 as the lowest. When level 0 is set, the interrupt is masked and interrupt requests are ignored.

### 13.4.1 NMI Interrupt

The NMI interrupt has the highest priority level of 16. It 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.

A setting can also be made to have the NMI interrupt accepted even if the BL bit is set to 1. Input from the NMI pin is edge-detected. The NMI edge select bit (NMIE) in ICR0 is used to select either rising or falling edge as the detection edge. When the NMIE bit in ICR0 is modified, the NMI interrupt is not detected for a maximum of six bus clock cycles after the modification. The IMASK value in SR is not affected by the accepted NMI interrupt.

### 13.4.2 IRQ Interrupts

The IRQnS[1:0] (n = 0, 1) bits in ICR1 is used to select either rising edge, falling edge, low level or high level detection.

A priority level (from 15 to 0) can be set for each input by writing to INTPRI.

When detects the IRQ interrupt request by low level or high level, the IRQ interrupt pin input level should be held until the CPU accepts the interrupt and starts interrupt exception handling.

When high or low level detection is selected, once the interrupt request has been detected, the INTC holds the interrupt request as the interrupt source in INTREQ even if the IRQ interrupt pin level may be changed and canceled. The interrupt source is being held until the CPU accepts any interrupt request (IRQ or not) or the corresponding interrupt mask bit is set to 1. Then clear the interrupt source that held in INTREQ after clearing the interrupt request in the exception handling routine. For details of clearing the interrupt request, see section 13.7.3, To Clear IRQ Interrupt Requests.

When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the level of the accepted interrupt. When the INTMU bit is cleared to 0, the IMASK value in SR is not affected by the accepted interrupt.

### 13.4.3 On-Chip Module Interrupts

On-chip module interrupts are interrupts generated by on-chip modules. Not every interrupt source is assigned a different interrupt vector, but sources are reflected in the interrupt event register (INTEVT), so it is easy to identify sources by using the INTEVT value as a branch offset in the exception handling routine.

A priority level from 31 to 0 can be set for each module by means of INT2PRI0 to INT2PRI12. The INTC rounds off the lowest one bit and sends 4-bit code to the CPU. In detail, see section 13.4.4, Interrupt Priority Level of On-Chip Module Interrupts.

When the INTMU bit in the CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the level 15 of the accepted NMI interrupt. When the INTMU bit in CPUOPM is cleared to 0, the IMASK value in SR is not affected by the accepted NMI interrupt.

Updating of the interrupt source flag and interrupt enable flag of a peripheral module should only be carried out when the BL bit in SR is set to 1 or while the corresponding interrupt does not occur by setting its mask bit. To prevent erroneous interrupt acceptance from an interrupt source that should have been updated, first read the on-chip peripheral register containing the relevant flag and wait the priority determination time, then clear the BL bit to 0. This will secure the necessary timing internally. When updating a number of flags, there is no problem if only the register containing the last flag updated is read from.

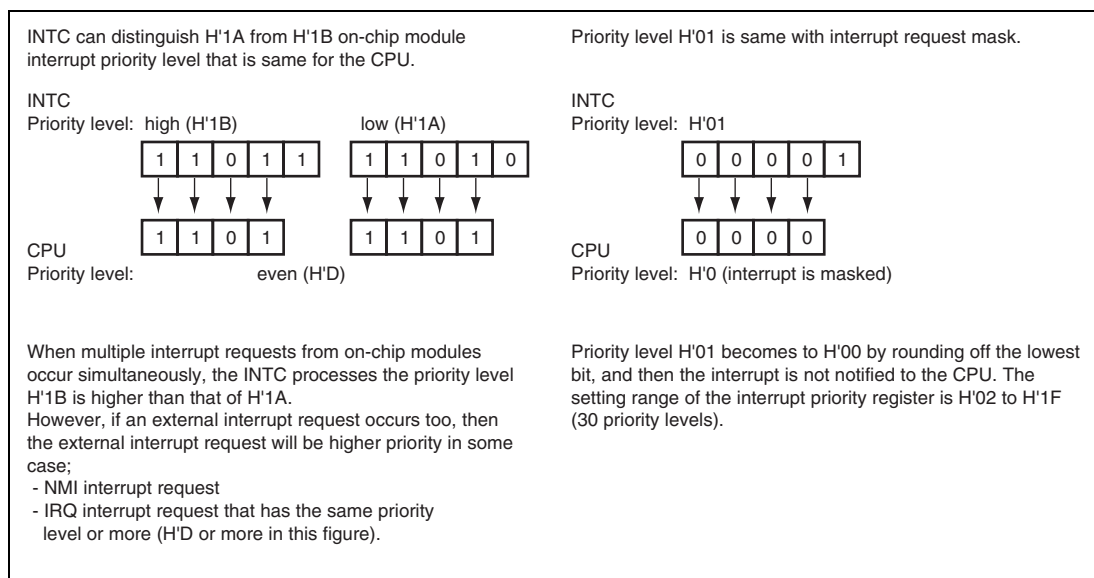
If flag updating is performed while the BL bit is cleared to 0, the program may jump to the interrupt handling routine when the INTEVT value is 0. In this case, interrupt processing is initiated due to the timing relationship between the flag update and interrupt request recognition within this LSI. Processing can be continued without any problem by executing an RTE instruction.

### 13.4.4 Interrupt Priority Level of On-Chip Module Interrupts

When an on-chip module interrupt is generated, the INTC outputs its interrupt exception code (INTEVT code) as individual source identification to the CPU. When the CPU accepts an interrupt, the corresponding INTEVT code is indicated in INTEVT. Even if the interrupt source register of the INTC is not read, the interrupt source can be identified by reading INTEVT of the CPU in the interrupt handler. Table 13.1 lists the source of on-chip module interrupt and the interrupt exception codes.

On-chip module interrupt, it can be set individual interrupt sources to 30 (5-bit) priority levels (see figure 13.2). The interrupt level receive interface consists of four bits and there are 15 priority levels (H'0 is interrupt request mask). The INTC consists of five bits in which one bit is extended

and determines the priorities of individual interrupt sources. The lowest one bit is then rounded off, the data is converted to 4-bit data, and the priority levels are notified. For example, two interrupt sources whose priority levels are set to H'1A and H'1B are both output as 4-bit priority level H'D. That is, the two interrupt sources have the same value. However, in terms of the INTEVT code that is notified when a conflict occurs between two interrupt sources, the INTEVT code that corresponds to the interrupt with a priority level of H'1B has priority. This is because the priority level of H'1B is higher than that of H'1A when comparing 5-bit data. When a conflict occurs between interrupts with the same priority level, the INTEVT code is notified according to the priority level shown in table 13.1.



**Figure 13.2 On-Chip Module Interrupt Priority**

### 13.4.5 Interrupt Exception Handling and Priority

Table 13.6 lists the codes for the interrupt event register (INTEVT), and the order of interrupt priority.

Each interrupt source is assigned a unique INTEVT code. The start address of the exception handling routine is common to each interrupt source. Therefore, to identify the interrupt source, branching is performed at the start of the exception handling routine using the INTEVT value. For instance, the INTEVT value is used as a branch offset.

The priority order of the on-chip modules is specified as desired by setting priority levels from 31 to 0 in INT2PRI0 to INT2PRI12. The priority order of the on-chip modules is set to 0 by a reset.

When the priorities for multiple interrupt sources are set to the same level and such interrupts are generated simultaneously, they are handled according to the default priority order shown in table 13.6.

Updating of INTPRI and INT2PRI0 to INT2PRI12 should only be carried out when the BL bit in SR is set to 1. To prevent erroneous interrupt acceptance, first read one of the interrupt priority level setting registers, then clear the BL bit to 0. This will secure the necessary timing internally.

**Table 13.6 Interrupt Exception Handling and Priority**

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
NMI	—	H'1C0	16	—	—	—	—	High
IRQ	IRQ0	H'240	INTPRI [31:28]	INTMSK[31] INTMSKCLR[31]	INTREQ [31]	—	High	↑ ↓ Low
	IRQ1	H'280	INTPRI [27:24]	INTMSK[30] INTMSKCLR[30]	INTREQ [30]	—	Low	
WDT	ITI*	H'560	INT2PRI 2[12:8]	INT2MSKR[5] INT2MSKCR[5]	INT2A0[5] INT2A1[5]	—	—	↑ ↓ Low
TMU0	TUNIO*	H'580	INT2PRI 0[28:24]	INT2MSKR[0] INT2MSKCR[0]	INT2A0[0] INT2A1[0]	INT2B0[0]	—	
TMU1	TUNI1*	H'5A0	INT2PRI 0[20:16]	—	—	INT2B0[1]	—	
TMU2	TUNI2*	H'5C0	INT2PRI 0[12:8]	—	—	INT2B0[2]	—	
	TICPI2*	H'5E0	INT2PRI 0[4:0]	—	—	INT2B0[3]	—	
H-UDI	H-UDI	H'600	INT2PRI 3[28:24]	INT2MSKR[7] INT2MSKCR[7]	INT2A0[7] INT2A1[7]	—	—	↑ ↓ Low
LCDC	LCDCI	H'620	INT2PRI 9[28:24]	INT2MSKR1[7] INT2MSKCR1[7]	INT2A01[7] INT2A11[7]	—	—	

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
DMAC	DMINT0*	H'640	INT2PRI3 [20:16]	INT2MSKR[8]	INT2A0[8]	INT2B3[0]	High	High
	DMINT1*	H'660		INT2MSKCR[8]	INT2A1[8]	INT2B3[1]	↕	
	DMINT2*	H'680				INT2B3[2]	↕	
	DMINT3*	H'6A0				INT2B3[3]	Low	
	DMAE*	H'6C0				INT2B3[12]		
SCIF0	ERIO*	H'700	INT2PRI2 [28:24]	INT2MSKR[3]	INT2A0[3]	INT2B2[0]	High	
	RXIO*	H'720		INT2MSKCR[3]	INT2A1[3]	INT2B2[1]	↕	
	BRIO*	H'740				INT2B2[2]	↕	
	TXIO*	H'760				INT2B2[3]	Low	
DMAC	DMINT4*	H'780	INT2PRI3 [20:16]	INT2MSKR[8]	INT2A0[8]	INT2B3[4]	High	Low
	DMINT5*	H'7A0		INT2MSKCR[8]	INT2A1[8]	INT2B3[5]	Low	
VDC2	VDCI	H'860	INT2PRI12 [28:24]	INT2MSKR1[19] INT2MSKCR1[19]	INT2A01[19] INT2A11 [19]	—		
IIC	IICI	H'8A0	INT2PRI9 [4:0]	INT2MSKR1[4] INT2MSKCR1[4]	INT2A01[4] INT2A11[4]	—		
EtherC	EINT	H'920	INT2PRI12 [4:0]	INT2MSKR1[16] INT2MSKCR1[16]	INT2A01[16] INT2A11[16]	—		
G2D	G2DI	H'980	INT2PRI4 [20:16]	INT2MSKR[13]	INT2A0[13]	—		
				INT2MSKCR[13]	INT2A1[13]			
SSI_A	SSIDMA0	H'A00	INT2PRI4 [12:8]	INT2MSKR[14]	INT2A0[14]	INT2B4[0]		
				INT2MSKCR[14]	INT2A1[14]			
	SSICH0	H'A20	INT2PRI4 [4:0]	INT2MSKR[15]	INT2A0[15]	INT2B4[1]		
				INT2MSKCR[15]	INT2A1[15]			
	SSICH1	H'A40	INT2PRI5 [28:24]	INT2MSKR[16]	INT2A0[16]	INT2B4[2]		
				INT2MSKCR[16]	INT2A1[16]			
	SSICH2	H'A60	INT2PRI5 [20:16]	INT2MSKR[17] INT2MSKCR[17]	INT2A0[17] INT2A1[17]	INT2B4[3]		Low

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
SSI_B	SSIDMA1	H'AA0	INT2PRI5 [4:0]	INT2MSKR[19]	INT2A0[19]	INT2B4[5]	High	High
	SSICH3	H'AC0		INT2MSKCR[19]	INT2A1[19]	INT2B4[6]	↑	
	SSICH4	H'AE0				INT2B4[7]	↓	
	SSICH5	H'B00				INT2B4[8]	Low	
SCIF1	ERI1*	H'B80	INT2PRI2 [20:16]	INT2MSKR[4]	INT2A0[4]	INT2B2[4]	High	
	RX11*	H'BA0		INT2MSKCR[4]	INT2A1[4]	INT2B2[5]	↑	
	BRI1*	H'BC0				INT2B2[6]	↓	
	TXI1*	H'BE0				INT2B2[7]	Low	
ATAPI	ATAI	H'C00	INT2PRI6 [28:24]	INT2MSKR[14]	INT2A0[14]	—		
				INT2MSKCR[14]	INT2A1[14]			
USB	USBI	H'C60	INT2PRI12 [12:8]	INT2MSKR1[17]	INT2A01 [17]	—		
				INT2MSKCR1 [17]	INT2A11 [17]			
FLCTL	FLSTE	H'D00	INT2PRI6 [12:8]	INT2MSKR[22]	INT2A0[22]	INT2B5[0]	High	
	FLTEND	H'D20		INT2MSKCR[22]	INT2A1[22]	INT2B5[1]	↑	
	FLTRQ0	H'D40				INT2B5[2]	↓	
	FLTRQ1	H'D60				INT2B5[3]	Low	
TMU3	TUNI3*	H'E00	INT2PRI1 [28:24]	INT2MSKR[1]	INT2A0[1]	INT2B0[4]		
TMU4	TUNI4*	H'E20		INT2MSKCR[1]	INT2A1[1]	INT2B0[5]		
TMU5	TUNI5*	H'E40	INT2PRI1 [12:8]			INT2B0[6]		
SRC	OVF	H'E80	INT2PRI6 [4:0]	INT2MSKR[23]	INT2A0[23]	—	High	
	IDEI	H'EA0		INT2MSKCR[23]	INT2A1[23]		↑	
	ODFI	H'EC0	INT2PRI8 [20:16]	INT2MSKR1[2]	INT2A01[2]	—	↓	
				INT2MSKCR1[2]	INT2A11[2]		Low	Low



Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
SCIF2	ERI2	H'F00	INT2PRI7 [28:24]	INT2MSKR1[25] INT2MSKCR1[25]	INT2A01 [25]	INT2B6[0]	<div> <div>High</div> <div>↑</div> <div>↓</div> <div>Low</div> </div>	<div> <div>High</div> <div>↑</div> <div>↓</div> <div>Low</div> </div>
	RX12	H'F20				INT2B6[1]		
	BRI2	H'F40			INT2A11 [25]	INT2B6[2]		
	TX12	H'F60				INT2B6[3]		
GPIO	CH0	H'F80	INT2PRI7 [20:16]	INT2MSKR[25] INT2MSKCR[25]	INT2A0[25]	INT2B7 [3:0]	<div> <div>High</div> <div>↑</div> <div>↓</div> <div>Low</div> </div>	<div> <div>High</div> <div>↑</div> <div>↓</div> <div>Low</div> </div>
	CH1	H'FA0			INT2A1[25]	INT2B7 [11:8]		
	CH2	H'FC0				INT2B7 [19:16]		
	CH3	H'FE0				INT2B7 [27:24]		

Note: \* ITI: Interval timer interrupt

TUNIO to TUNI5: TMU channel 0 to 5 under flow interrupt

TICPI2: TMU channel 2 input capture interrupt

DMINT0 to DMINT5: DMAC channel 0 to 5 transfer end interrupt

DMAE: DMAC address error interrupt (channel 0 to 5)

ERIO, ERI1, ERI2: SCIF channel 0, 1,2 receive error interrupt

RXIO, RXI1, RXI2: SCIF channel 0, 1,2 receive data full interrupt

BRI0, BRI1, BRI2: SCIF channel 0, 1,2 break interrupt

TXIO, TXI1, TXI1: SCIF channel 0, 1,2 transmission data empty interrupt

## 13.5 Operation

### 13.5.1 Interrupt Sequence

The sequence of interrupt operations is described below. Figure 13.3 is the flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the INTC.
2. The INTC selects the highest-priority interrupt from the interrupt requests sent, following the priority levels set in INTPRI and INT2PRI0 to INT2PRI12. Lower-priority interrupts are held pending. If two of these interrupts have the same priority level or if multiple interrupts occur within a single module, the interrupt with the highest priority is selected according to table 13.6.
3. The priority level of the interrupt selected by the INTC is compared with the interrupt mask level (IMASK) set in SR of the CPU. If the priority level is higher than the mask level, the INTC accepts the interrupt and sends an interrupt request signal to the CPU.
4. The CPU accepts an interrupt at a break in instructions.
5. The interrupt source code is set in the interrupt event register (INTEVT).
6. SR and program counter (PC) are saved to SSR and SPC, respectively. R15 is saved to SGR at this time.
7. The BL, MD, and RB bits in SR are set to 1.
8. Execution jumps to the start address of the interrupt exception handling routine (the sum of the value set in the vector base register (VBR) and H'0000 0600).

In the exception handling routine, execution may branch with the INTEVT value used as its offset in order to identify the interrupt source. This enables execution to branch to the handling routine for the individual interrupt source.

- Notes:
1. When the INTMU bit in the CPU operating mode register (CPUOPM.INTMU) is set to 1, the interrupt mask level (IMASK) in SR is automatically set to the level of the accepted interrupt. When the INTMU bit is cleared to 0, the IMASK value in SR is not affected by the accepted interrupt.
  2. The interrupt source flag should be cleared in the interrupt handler. To ensure that an interrupt source that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, wait for the time shown in table 13.7, and then clear the BL bit or execute an RTE instruction.
  3. For some interrupt sources, the interrupt mask setting (INTMSK, INT2MSKR or INT2MSKR1) for each interrupt source must be cleared by using INTMSKCLR, INT2MSKCR or INT2MSKCR1.

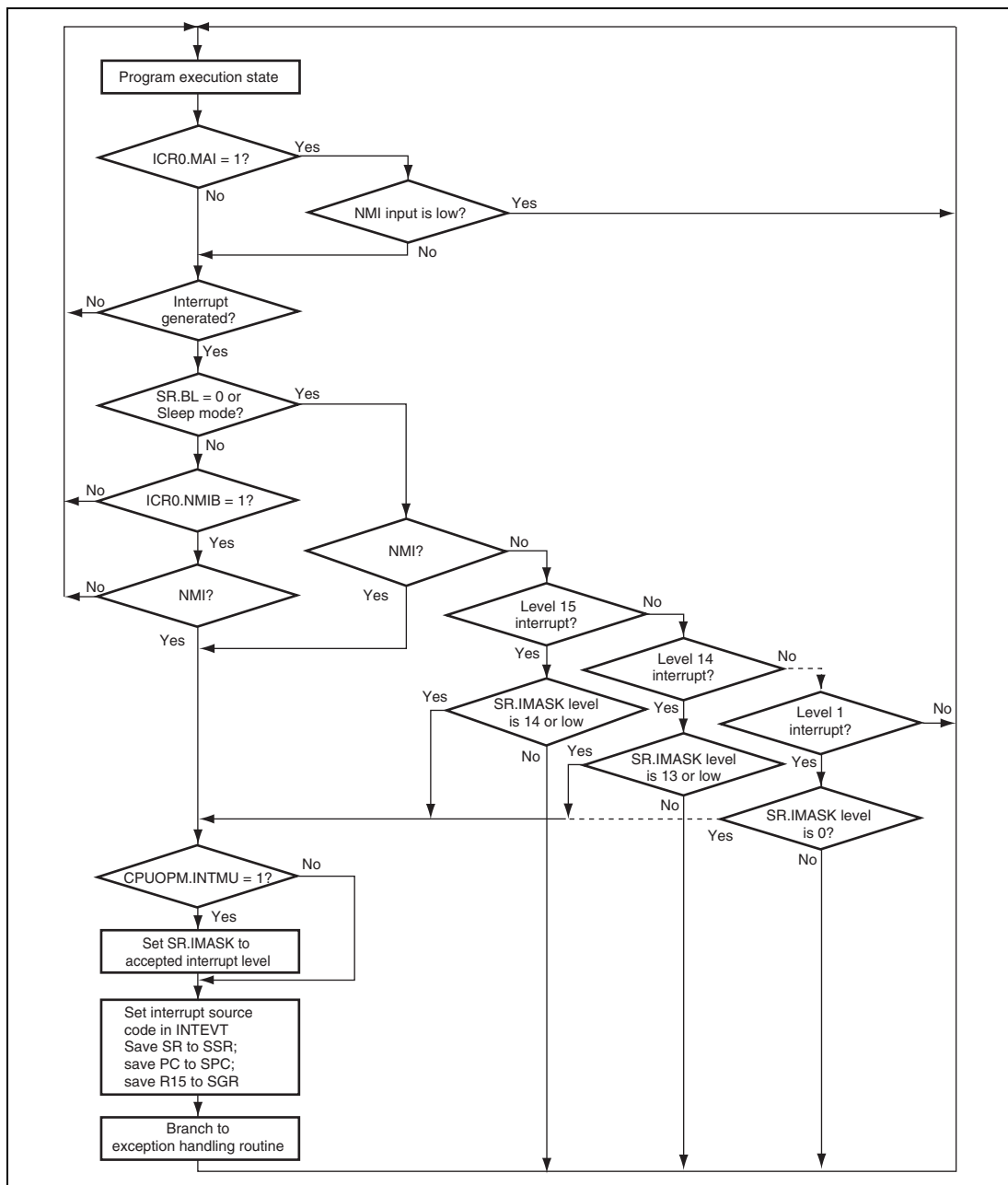


Figure 13.3 Interrupt Operation Flowchart

### 13.5.2 Multiple Interrupts

When handling multiple interrupts, an interrupt handling routine should include the following procedures:

1. To identify the interrupt source, branch to a specific interrupt handling routine for the interrupt source by using the INTEVT code as an offset.
2. Clear the interrupt source in each specific interrupt handling routine.
3. Save SSR and SPC to the stack.
4. Clear the BL bit in SR. When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the level of the accepted interrupt. When the INTMU bit in CPUOPM is cleared to 0, set the IMASK bit in SR by software to the accepted interrupt level.
5. Handle the interrupt as required.
6. Set the BL bit in SR to 1.
7. Restore SSR and SPC from memory.
8. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted if multiple interrupts occur after step 4. This reduces the interrupt response time for urgent processing.

### 13.5.3 Interrupt Masking by MAI Bit

Setting the MAI bit in ICR0 to 1 masks interrupts while the NMI signal is low regardless of the BL and IMASK bit settings in SR.

- Normal operation or sleep mode

All interrupts are masked while the NMI signal is low. Note that only NMI interrupts due to NMI signal input occur.

## 13.6 Interrupt Response Time

Table 13.7 shows the interrupt response time, which is the interval from when an interrupt request occurs until the interrupt exception handling is started and the start instruction of the exception handling routine is fetched.

**Table 13.7 Interrupt Response Time**

Item	Number of States				Remarks
	NMI	IRQ	Peripheral Module		
			Other than GPIO	GPIO	
Priority determination time	5Bcyc + 2Pcyc	4Bcyc + 2Pcyc	5Pcyc	7Pcyc	
Wait time until the CPU finishes the current sequence		S-1 (≥ 0) × 1cyc			
Interval from when interrupt exception handling begins (saving SR and PC) until a SHwy bus request is issued to fetch the start instruction of the exception handling routine		111cyc + 1Scyc			
Response Total time	(S + 10) 1cyc + 1Scyc + 5Bcyc + 2Pcyc	(S + 10) 1cyc + 1Scyc + 4Bcyc + 2Pcyc	(S + 10) 1cyc + 1Scyc + 5Pcyc	(S + 10) 1cyc + 1Scyc + 7Pcyc	
Minimum	291cyc + Sx1cyc	351cyc + Sx1cyc	311cyc + Sx1cyc*	391cyc+ Sx1cyc*	When 1cyc:Scyc: Bcyc:Pcyc = 4:2:1:1

**[Legend]**

1cyc: Period for one CPU clock cycle

Scyc: Period for one SHwy clock cycle

Bcyc: Period for one bus clock cycle

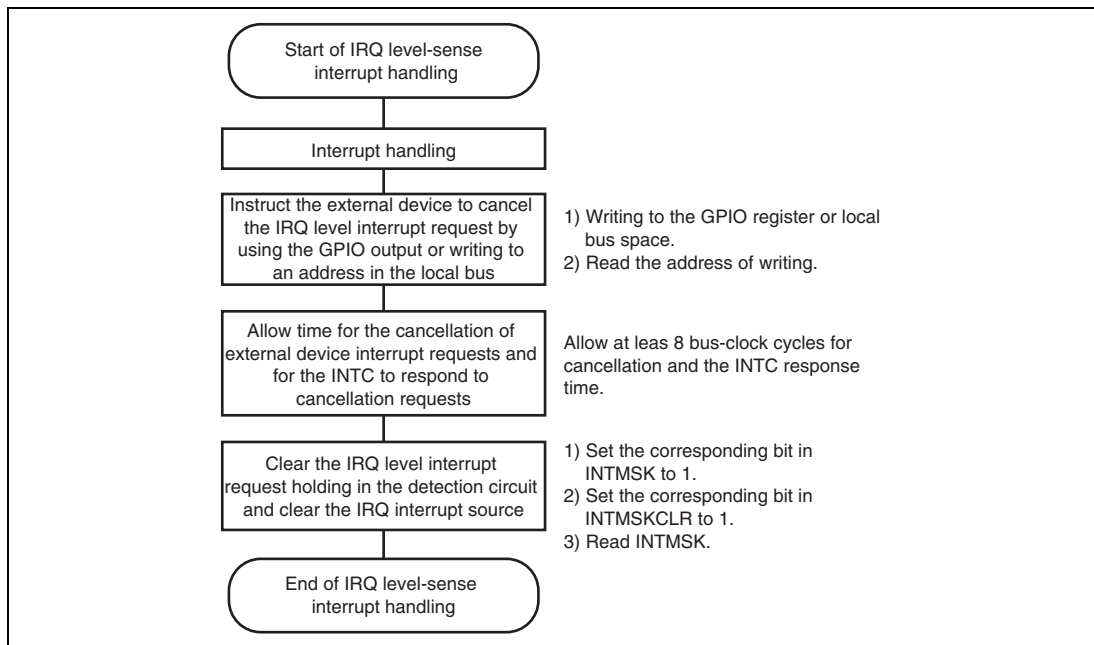
Pcyc: Period for one peripheral clock cycle

S: Number of instruction execution states

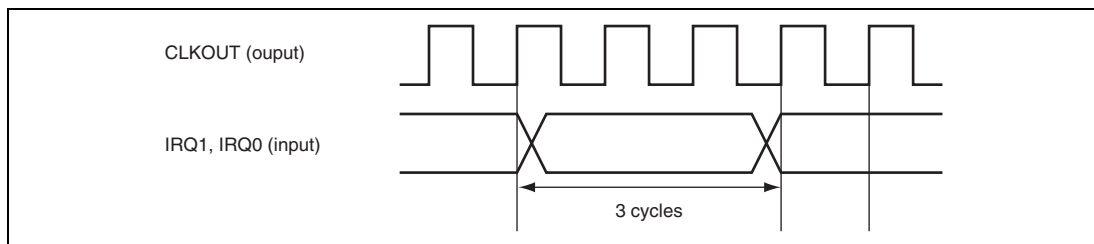
## 13.7 Usage Notes

### 13.7.1 To Clear Interrupt Request When Holding Function Selected

When an IRQ level-sense interrupt is generated and the holding function is in use, the interrupt request must be cleared in the interrupt handling routine after it has been accepted.



**Figure 13.4 Example of Interrupt Handling Routine**



**Figure 13.5 The time requested to detect interrupts from IRQ1 and IRQ0**

### 13.7.2 Notes on Setting IRQ1 and IRQ0 Pin Function

When switching the IRQ1 and IRQ0 pin function, it is possible that the INTC may hold an interrupt by mistake. Therefore, to prevent detecting unintentional interrupts, mask both all IRQ interrupts and then switch the IRQ1 and IRQ0 pin function.

**Table 13.8 Switching Sequence of IRQ1 and IRQ0 Pin Function**

Sequence	ITEM	PROCEDURE
1	IRQ interrupt request masking	Write 1 to all bits in INTMSK
2	IRQ0/DTEND1 pin setting to IRQ0 interrupt request input	Write 0 to the PTSEL_SI5 bit in the PTSEL_S in the GPIO
3	WDTOVF/IRQ1/AUDUCK/DACK1 pin setting to IRQ1 interrupt request input	Write 01 to the PTSEL_K7[1:0] bits in the PTSEL_K in GPIO
4	IRQ interrupt detection start	Write 1 to the corresponding bit in INTMSKCLR

### 13.7.3 To Clear IRQ Interrupt Requests

Clearing procedure of the interrupt held in the INTC is as follows

- **To clear IRQ level-sense interrupt requests**

To clear an IRQ level-sense interrupt request from the IRQ1 and IRQ0 pins, write 1 to the corresponding mask bit (IM01 and IM00) in INTMSK.

The IRQ interrupt requests detected by the INTC is not cleared even if 0 is written to a corresponding bit in INTPRI. The IRQ interrupt sources detected by the INTC (be cleared)

- **To clear IRQ edge-detection interrupt requests**

To clear an IRQ edge-detection interrupt request from the IRQ1 and IRQ0 pins, write 0 after reading 1 in the corresponding IRn (n = 0, 1) bit in INTREQ.

The IRQ interrupt requests detected by the INTC is not cleared even if 1 is written to a corresponding bit in INTMSK.





## Section 14 Timer Unit (TMU)

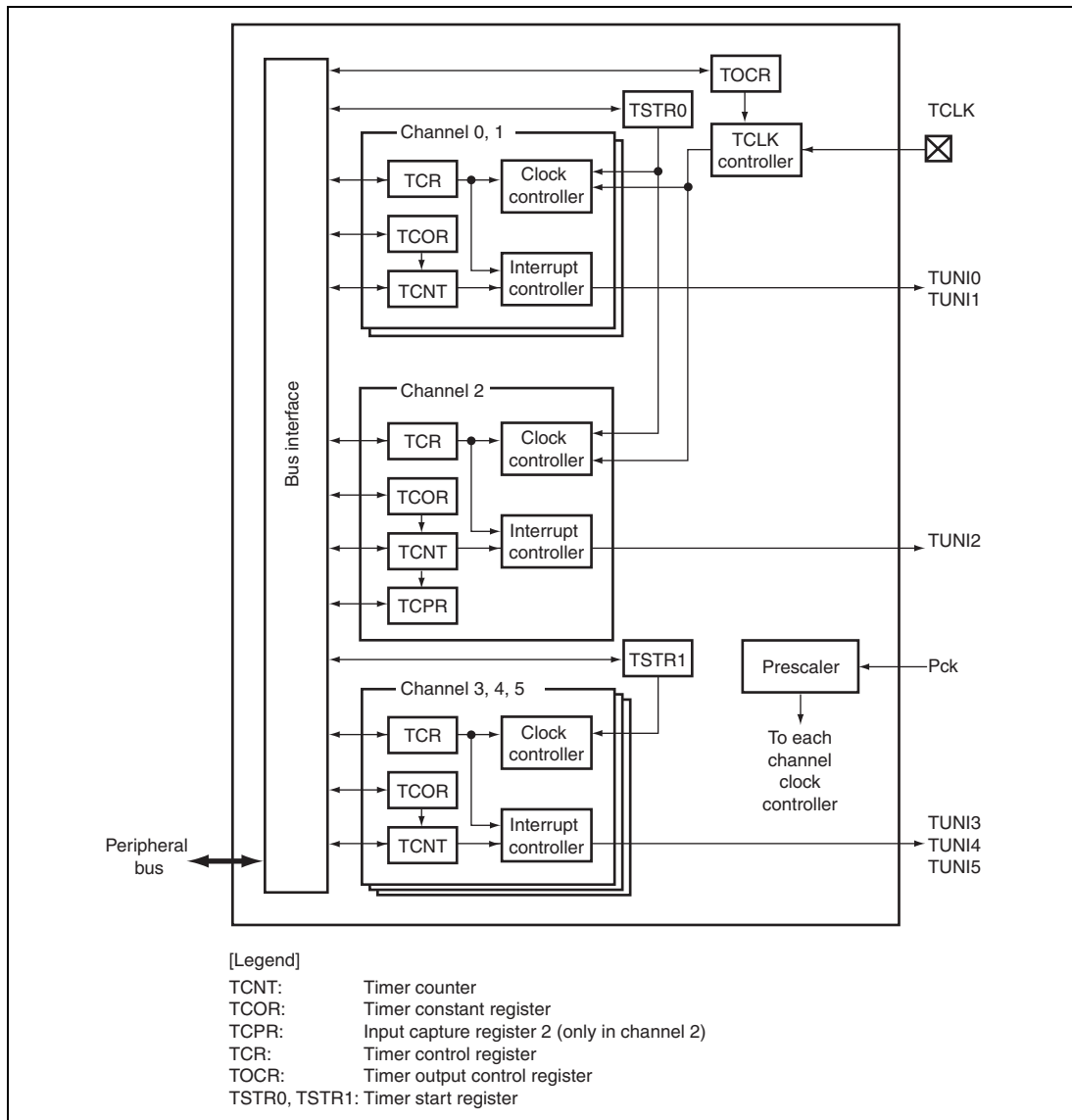
This LSI includes an on-chip 32-bit timer unit (TMU), which has six channels (channels 0 to 5).

### 14.1 Features

The TMU has the following features.

- Auto-reload type 32-bit down-counter provided for each channel
- Input capture function provided in 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 for each channel
- 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 six counter input clocks: Channel 0 to 2  
External clock (TCLK) and five peripheral clocks (Pck/4, Pck/16, Pck/64, Pck/256, and Pck/1024) (Pck is the peripheral clock).
- Selection of five counter input clocks: Channel 3 to 5  
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 14.1 shows a block diagram of the TMU.



**Figure 14.1 Block Diagram of TMU**

## 14.2 Input/Output Pins

Table 14.1 shows the TMU pin configuration.

**Table 14.1 Pin Configuration**

Pin Name	Function	I/O	Description
TCLK	Clock input	input	Channel 0, 1 and 2 external clock input pin/channel 2 input capture control input pin

## 14.3 Register Descriptions

Table 14.2 shows the TMU register configuration. Table 14.3 shows the register states in each processing mode.

**Table 14.2 Register Configuration**

Channel	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size
0, 1, 2 Common	Timer output control register	TOCR	R/W	H'FFD8 0000	H'1FD8 0000	8
	Timer start register 0	TSTR0	R/W	H'FFD8 0004	H'1FD8 0004	8
0	Timer constant register 0	TCOR0	R/W	H'FFD8 0008	H'1FD8 0008	32
	Timer counter 0	TCNT0	R/W	H'FFD8 000C	H'1FD8 000C	32
	Timer control register 0	TCR0	R/W	H'FFD8 0010	H'1FD8 0010	16
1	Timer constant register 1	TCOR1	R/W	H'FFD8 0014	H'1FD8 0014	32
	Timer counter 1	TCNT1	R/W	H'FFD8 0018	H'1FD8 0018	32
	Timer control register 1	TCR1	R/W	H'FFD8 001C	H'1FD8 001C	16
2	Timer constant register 2	TCOR2	R/W	H'FFD8 0020	H'1FD8 0020	32
	Timer counter 2	TCNT2	R/W	H'FFD8 0024	H'1FD8 0024	32
	Timer control register 2	TCR2	R/W	H'FFD8 0028	H'1FD8 0028	16
	Input capture register 2	TCPR2	R	H'FFD8 002C	H'1FD8 002C	32
3, 4, 5 Common	Timer start register 1	TSTR1	R/W	H'FFDC 0004	H'1FDC 0004	8
3	Timer constant register 3	TCOR3	R/W	H'FFDC 0008	H'1FDC 0008	32
	Timer counter 3	TCNT3	R/W	H'FFDC 000C	H'1FDC 000C	32
	Timer control register 3	TCR3	R/W	H'FFDC 0010	H'1FDC 0010	16
4	Timer constant register 4	TCOR4	R/W	H'FFDC 0014	H'1FDC 0014	32
	Timer counter 4	TCNT4	R/W	H'FFDC 0018	H'1FDC 0018	32
	Timer control register 4	TCR4	R/W	H'FFDC 001C	H'1FDC 001C	16
5	Timer constant register 5	TCOR5	R/W	H'FFDC 0020	H'1FDC 0020	32
	Timer counter 5	TCNT5	R/W	H'FFDC 0024	H'1FDC 0024	32
	Timer control register 5	TCR5	R/W	H'FFDC 0028	H'1FDC 0028	16

**Table 14.3 Register States in Each Processing Mode**

Channel	Register Name	Abbrev.	Power-on Reset	Sleep	Standby	Module Standby
0, 1, 2 Common	Timer output control register	TOCR	H'00	Retained	Retained	Retained
	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
3, 4, 5 Common	Timer start register 1	TSTR1	H'00	Retained	Retained	Retained
3	Timer constant register3	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

### 14.3.1 Timer Output Control Register (TOCR)

TOCR is an 8-bit read-only register that specifies whether external pin TCLK is used as the external clock or input capture control input pin.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCOE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
0	TCOE	0	R	Timer Clock Pin Control  Specifies whether timer clock pin TCLK is used as the external clock or input capture control input pin.  0: Timer clock pin (TCLK) is used as external clock input or input capture control input pin 1: Invalid

### 14.3.2 Timer Start Register (TSTR0, TSTR1)

TSTR is an 8-bit readable/writable register that specifies whether TCNT in each 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



### 14.3.3 Timer Constant Register (TCORn) (n = 0 to 5)

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
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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

### 14.3.4 Timer Counter (TCNTn) (n = 0 to 5)

The TCNT registers are 32-bit readable/writable registers. Each TCNT counts down on the input clock selected by the TPSC[2:0] 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
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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

### 14.3.5 Timer Control Registers (TCRn) (n = 0 to 5)

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, TCR4 and TCR5

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* <sup>1</sup>	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* <sup>2</sup>
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* <sup>2</sup>

Bit	Bit Name	Initial Value	R/W	Description
7, 6	ICPE[1:0]*1	00	R/W	<p><b>Input Capture Control</b></p> <p>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.</p> <p>The CKEG bits specify whether the rising edge or falling edge of the TCLK pin is used to set the TCNT2 value in TCPR2.</p> <p>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.</p> <p>00: Input capture function is not used.</p> <p>01: Setting prohibited</p> <p>10: Input capture function is used, but interrupt due to input capture (TICPI2) is not enabled.</p> <p>Data transfer request is sent to the DMAC in the event of input capture.</p> <p>11: Input capture function is used, and interrupt due to input capture (TICPI2) is enabled.</p>
5	UNIE	0	R/W	<p><b>Underflow Interrupt Control</b></p> <p>Controls enabling or disabling of interrupt generation when the UNF status flag is set to 1, indicating TCNT underflow.</p> <p>0: Interrupt due to underflow (TUNI) is disabled</p> <p>1: Interrupt due to underflow (TUNI) is enabled</p>
4, 3	CKEG[1:0]	00	R/W	<p><b>Clock Edge</b></p> <p>These bits select the external clock input edge when an external clock is selected or the input capture function is used.</p> <p>00: Count/input capture register set on rising edge</p> <p>01: Count/input capture register set on falling edge</p> <p>1X: Count/input capture register set on both rising and falling edges</p>

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	TPSC[2:0]	000	R/W	Timer Prescaler 2 to 0 These bits select the TCNT count clock. 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: Setting prohibited 110: Setting prohibited 111: Counts on external clock (TCLK) *3

Notes: X: Don't care

1. Reserved bit in channel 0 or 1 (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, 4, and 5.

### 14.3.6 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

## 14.4 Operation

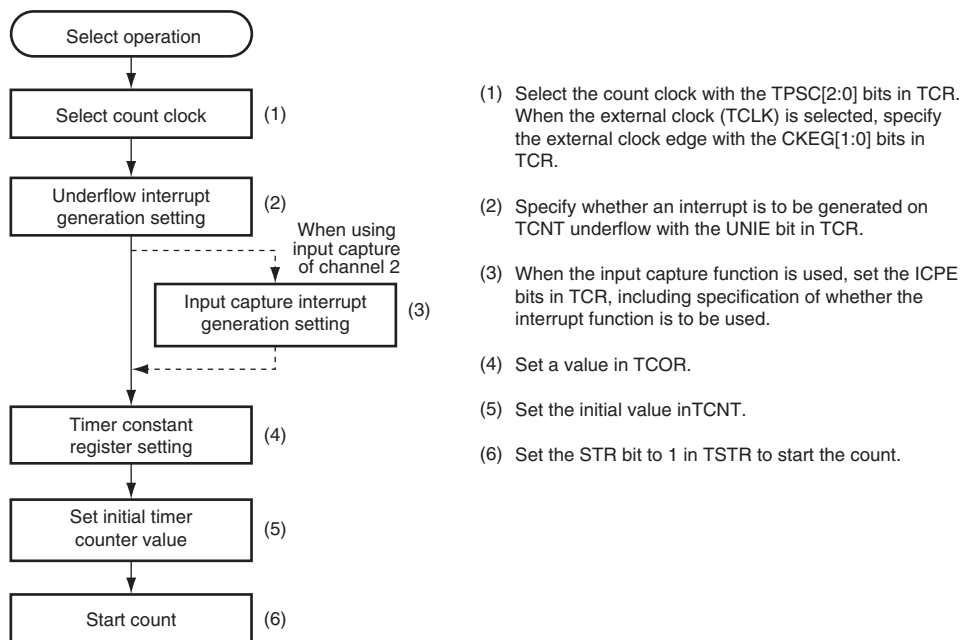
Each channel has a 32-bit timer counter (TCNT) and a 32-bit timer constant register (TCOR). Each TCNT performs count-down operation. The channels have an auto-reload function that allows cyclic count operations, and can also perform external event counting. Channel 2 also has an input capture function.

### 14.4.1 Counter Operation

When one of bits STR0 to STR2 in TSTR1 and TSTR0 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 Count Operation Setting Procedure

Figure 14.2 shows an example of the count operation setting procedure.

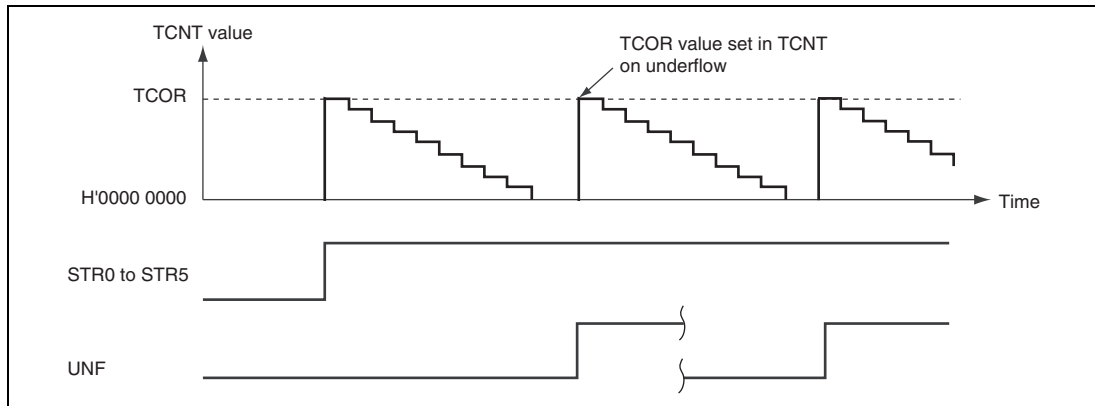


Note: When an interrupt is generated, clear the source flag in the interrupt handler.  
If the interrupt enabled state is set without clearing the flag, another interrupt will be generated.

**Figure 14.2 Example of Count Operation Setting Procedure**

## (2) Auto-Reload Count Operation

Figure 14.3 shows the TCNT auto-reload operation.



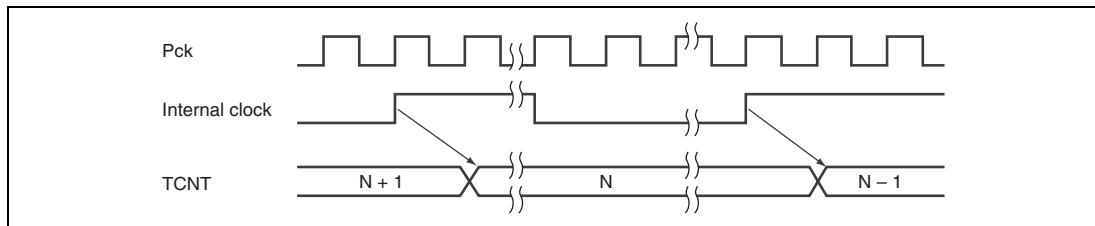
**Figure 14.3 TCNT Auto-Reload Operation**

## (3) TCNT Count Timing

- Operating on internal clock

Any of five 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 TPSC[2:0] bits in TCR.

Figure 14.4 shows the timing in this case.

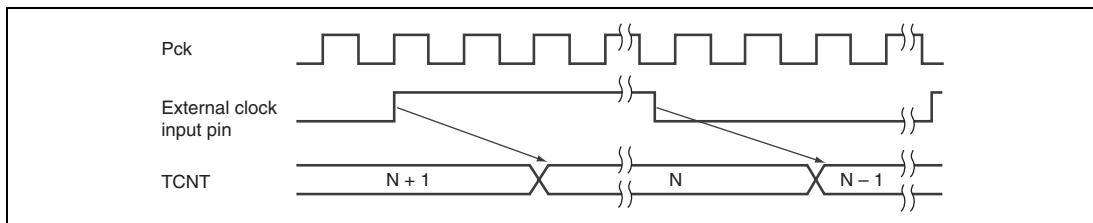


**Figure 14.4 Count Timing when Operating on Internal Clock**

- Operating on external clock

In channels 0, 1, and 2, the external clock pin (TCLK) input can be selected as the timer clock by means of the TPSC[2:0] bits in TCR. The detected edge (rising, falling, or both edges) can be selected with the CKEG[1:0] bits in TCR.

Figure 14.5 shows the timing for both-edge detection.



**Figure 14.5 Count Timing when Operating on External Clock**

### 14.4.2 Input Capture Function

Channel 2 has an input capture function.

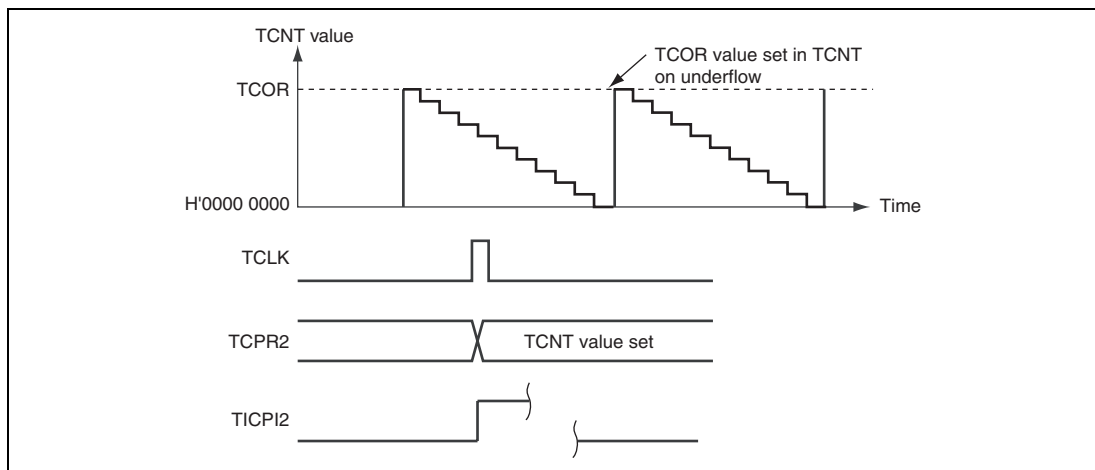
The procedure for using the input capture function is as follows:

1. Use bits TPSC[2:0] in TCR to set an internal clock as the timer operating clock.
2. Use bits IPCE[1:0] in TCR to specify use of the input capture function, and whether interrupts are to be generated when this function is used.
3. Use bits CKEG[1:0] in TCR 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.

Figure 14.6 shows the operation timing when the input capture function is used (with TCLK rising edge detection).





**Figure 14.6 Operation Timing when Using Input Capture Function**

## 14.5 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 (ICPE[1:0]) in TCR2 are both set to 11.

The TMU interrupt sources are summarized in Table 14.4.

**Table 14.4 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

## **14.6 Usage Notes**

### **14.6.1 Register Writes**

When writing to a TMU register, timer count operation must be stopped by clearing the start bit (STR5 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.

### **14.6.2 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.

### **14.6.3 External Clock Frequency**

Ensure that the external clock (TCLK) input frequency for channels 0, 1 and 2 does not exceed  $P_{ck}/4$ .



## Section 15 Serial Communication Interface with FIFO (SCIF)

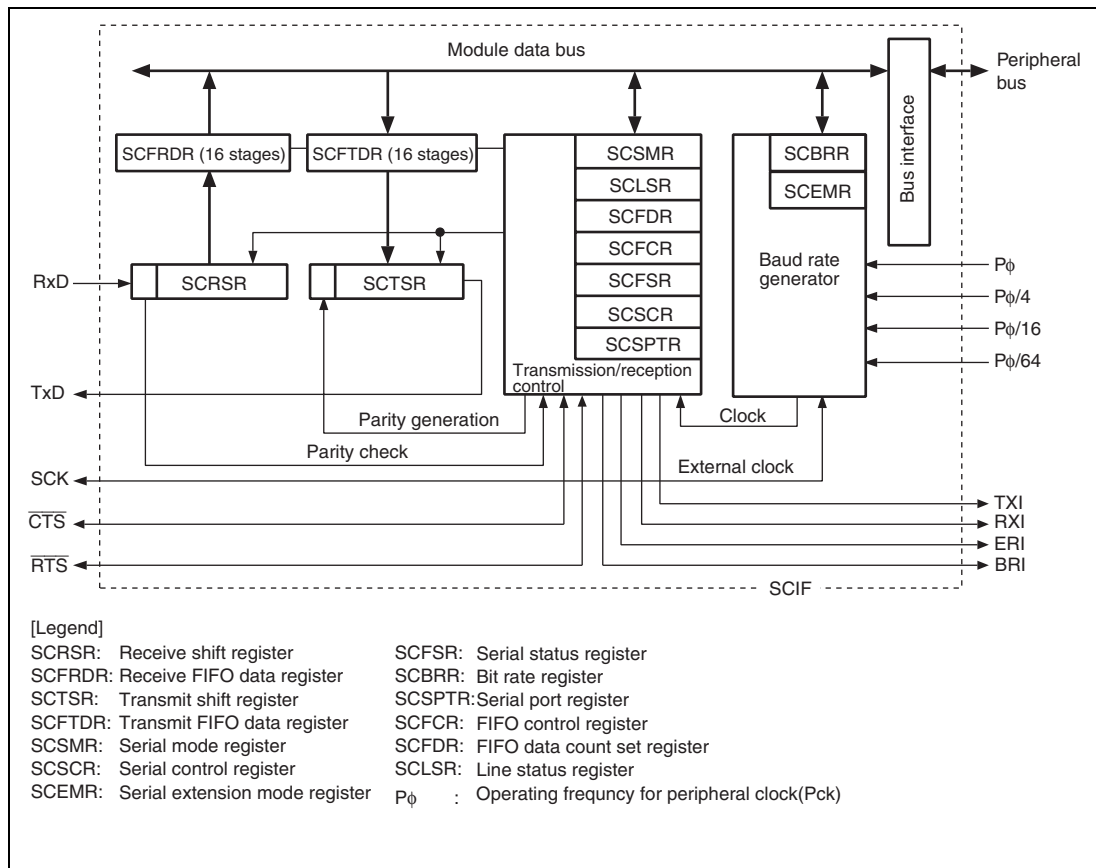
This LSI has a three-channel serial communication interface with FIFO (SCIF) that supports both asynchronous and clock synchronous serial communication. It also has 16-stage FIFO registers for both transmission and reception independently for each channel that enable this LSI to perform efficient high-speed continuous communication.

### 15.1 Features

- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, framing, and overrun errors
  - Break detection: Break is detected when a framing error is followed by at least one frame at the space 0 level (low level). It is also detected by reading the RxD level directly from the serial port register when a framing error occurs.
- Clock synchronous serial communication:
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a clock synchronous communication function. There is one serial data communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCIF can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)

- Four types of interrupts: Transmit-FIFO-data-empty interrupt, break interrupt, receive-FIFO-data-full interrupt, and receive-error interrupts are requested independently.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- In asynchronous mode, on-chip modem control functions ( $\overline{\text{RTS}}$  and  $\overline{\text{CTS}}$ ).
- The quantity of data in the transmit and receive FIFO data registers and the number of receive errors of the receive data in the receive FIFO data register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.
- In asynchronous mode, the base clock frequency can be either 16 or 8 times the bit rate.
- When an internal clock is selected as a clock source and the SCK pin is used as an input pin in asynchronous mode, either normal mode or double-speed mode can be selected for the baud rate generator.
- In asynchronous mode, the high-speed communication of 3 Mbps or higher can be supported.

Figure 15.1 shows a block diagram of the SCIF.



**Figure 15.1 Block Diagram of SCIF**

## 15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the SCIF.

**Table 15.1 Pin Configuration**

Channel	Pin Name	Symbol	I/O	Function
0 to 2	Serial clock pins	SCK0 to SCK2	I/O	Clock I/O
	Receive data pins	RxD0 to RxD2	Input	Receive data input
	Transmit data pins	TxD0 to TxD2	Output	Transmit data output
	Request to send pin	RTS0 to RTS2	I/O	Request to send
	Clear to send pin	CTS0 to CTS2	I/O	Clear to send



## 15.3 Register Descriptions

Figure 15.2 shows the SCIF register configuration. Figure 15.3 shows the register states in each processing mode. Since the register functions are the same in each channel, the channel number is omitted in the description below.

**Table 15.2 Register Configuration**

Channel	Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
0	Serial mode register_0	SCSMR_0	R/W	H'FFE0 0000	H'1FE0 0000	16
	Bit rate register_0	SCBRR_0	R/W	H'FFE0 0004	H'1FE0 0004	8
	Serial control register_0	SCSCR_0	R/W	H'FFE0 0008	H'1FE0 0008	16
	Transmit FIFO data register_0	SCFTDR_0	W	H'FFE0 000C	H'1FE0 000C	8
	Serial status register_0	SCFSR_0	R/(W)* <sup>1</sup>	H'FFE0 0010	H'1FE0 0010	16
	Receive FIFO data register_0	SCFRDR_0	R	H'FFE0 0014	H'1FE0 0014	8
	FIFO control register_0	SCFCR_0	R/W	H'FFE0 0018	H'1FE0 0018	16
	FIFO data count register_0	SCFDR_0	R	H'FFE0 001C	H'1FE0 001C	16
	Serial port register_0	SCSPTR_0	R/W	H'FFE0 0020	H'1FE0 0020	16
	Line status register_0	SCLSR_0	R/(W)* <sup>2</sup>	H'FFE0 0024	H'1FE0 0024	16
	Serial extension mode register_0	SCEMR_0	R/W	H'FFE0 0028	H'1FE0 0028	16
1	Serial mode register_1	SCSMR_1	R/W	H'FFE1 0000	H'FE1 0000	16
	Bit rate register_1	SCBRR_1	R/W	H'FFE1 0004	H'FE1 0004	8
	Serial control register_1	SCSCR_1	R/W	H'FFE1 0008	H'FE1 0008	16
	Transmit FIFO data register_1	SCFTDR_1	W	H'FFE1 000C	H'FE1 000C	8
	Serial status register_1	SCFSR_1	R/(W)* <sup>1</sup>	H'FFE1 0010	H'FE1 0010	16
	Receive FIFO data register_1	SCFRDR_1	R	H'FFE1 0014	H'FE1 0014	8
	FIFO control register_1	SCFCR_1	R/W	H'FFE1 0018	H'FE1 0018	16
	FIFO data count register_1	SCFDR_1	R	H'FFE1 001C	H'FE1 001C	16
	Serial port register_1	SCSPTR_1	R/W	H'FFE1 0020	H'FE1 0020	16
	Line status register_1	SCLSR_1	R/(W)* <sup>2</sup>	H'FFE1 0024	H'FE1 0024	16
	Serial extension mode register_1	SCEMR_1	R/W	H'FFE1 0028	H'FE1 0028	16

Channel	Register Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size
2	Serial mode register_2	SCSMR_2	R/W	H'FFE2 0000	H'1FE2 0000	16
	Bit rate register_2	SCBRR_2	R/W	H'FFE2 0004	H'1FE2 0004	8
	Serial control register_2	SCSCR_2	R/W	H'FFE2 0008	H'1FE2 0008	16
	Transmit FIFO data register_2	SCFTDR_2	W	H'FFE2 000C	H'1FE2 000C	8
	Serial status register_2	SCFSR_2	R/(W)* <sup>1</sup>	H'FFE2 0010	H'1FE2 0010	16
	Receive FIFO data register_2	SCFRDR_2	R	H'FFE2 0014	H'1FE2 0014	8
	FIFO control register_2	SCFCR_2	R/W	H'FFE2 0018	H'1FE2 0018	16
	FIFO data count register_2	SCFDR_2	R	H'FFE2 001C	H'1FE2 001C	16
	Serial port register_2	SCSPTR_2	R/W	H'FFE2 0020	H'1FE2 0020	16
	Line status register_2	SCLSR_2	R/(W)* <sup>2</sup>	H'FFE2 0024	H'1FE2 0024	16
	Serial extension mode register_2	SCEMR_2	R/W	H'FFE2 0028	H'1FE2 0028	16

Table 15.3 Register State in Each Operation Mode

Channel	Register Name	Abbreviation	Power-on Reset	Standby	Sleep	Module Standby
0	Serial mode register_0	SCSMR_0	H'0000	Retained	Retained	Retained
	Bit rate register_0	SCBRR_0	H'FF	Retained	Retained	Retained
	Serial control register_0	SCSCR_0	H'0000	Retained	Retained	Retained
	Transmit FIFO data register_0	SCFTDR_0	Undefined	Retained	Retained	Retained
	Serial status register_0	SCFSR_0	H'0060	Retained	Retained	Retained
	Receive FIFO data register_0	SCFRDR_0	Undefined	Retained	Retained	Retained
	FIFO control register_0	SCFCR_0	H'0000	Retained	Retained	Retained
	FIFO data count register_0	SCFDR_0	H'0000	Retained	Retained	Retained
	Serial port register_0	SCSPTR_0	H'0050	Retained	Retained	Retained
	Line status register_0	SCLSR_0	H'0000	Retained	Retained	Retained
	Serial extension mode register_0	SCEMR_0	H'0000	Retained	Retained	Retained

Channel	Register Name	Abbreviation	Power-on Reset	Standby	Sleep	Module Standby
1	Serial mode register_1	SCSMR_1	H'0000	Retained	Retained	Retained
	Bit rate register_1	SCBRR_1	H'FF	Retained	Retained	Retained
	Serial control register_1	SCSCR_1	H'0000	Retained	Retained	Retained
	Transmit FIFO data register_1	SCFTDR_1	Undefined	Retained	Retained	Retained
	Serial status register_1	SCFSR_1	H'0060	Retained	Retained	Retained
	Receive FIFO data register_1	SCFRDR_1	Undefined	Retained	Retained	Retained
	FIFO control register_1	SCFCR_1	H'0000	Retained	Retained	Retained
	FIFO data count register_1	SCFDR_1	H'0000	Retained	Retained	Retained
	Serial port register_1	SCSPTR_1	H'0050	Retained	Retained	Retained
	Line status register_1	SCLSR_1	H'0000	Retained	Retained	Retained
	Serial extension mode register_1	SCEMR_1	H'0000	Retained	Retained	Retained
2	Serial mode register_2	SCSMR_2	H'0000	Retained	Retained	Retained
	Bit rate register_2	SCBRR_2	H'FF	Retained	Retained	Retained
	Serial control register_2	SCSCR_2	H'0000	Retained	Retained	Retained
	Transmit FIFO data register_2	SCFTDR_2	Undefined	Retained	Retained	Retained
	Serial status register_2	SCFSR_2	H'0060	Retained	Retained	Retained
	Receive FIFO data register_2	SCFRDR_2	Undefined	Retained	Retained	Retained
	FIFO control register_2	SCFCR_2	H'0000	Retained	Retained	Retained
	FIFO data count register_2	SCFDR_2	H'0000	Retained	Retained	Retained
	Serial port register_2	SCSPTR_2	H'0050	Retained	Retained	Retained
	Line status register_2	SCLSR_2	H'0000	Retained	Retained	Retained
	Serial extension mode register_2	SCEMR_2	H'0000	Retained	Retained	Retained

### 15.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RxD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the receive FIFO data register (SCFRDR).

The CPU cannot read or write to SCRSR directly.

Bit:	7	6	5	4	3	2	1	0
	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

### 15.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a 16-byte FIFO register that stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into SCFRDR for storage. Continuous reception is possible until 16 bytes are stored. The CPU can read but not write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value is undefined.

When SCFRDR is full of receive data, subsequent serial data is lost.

Bit:	7	6	5	4	3	2	1	0
	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

### 15.3.3 Transmit Shift Register (SCTSR)

SCTSR transmits serial data. The SCIF loads transmit data from the transmit FIFO data register (SCFTDR) into SCTSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCIF automatically loads the next transmit data from SCFTDR into SCTSR and starts transmitting again.

The CPU cannot read from or write to SCTSR directly.

Bit:	7	6	5	4	3	2	1	0
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

### 15.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is a 16-byte FIFO register that stores data for serial transmission. When the SCIF detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCFTDR into SCTSR and starts serial transmission. Continuous serial transmission is performed until there is no transmit data left in SCFTDR. The CPU can write to SCFTDR at all times.

When SCFTDR is full of transmit data (16 bytes), no more data can be written. If writing of new data is attempted, the data is ignored.

Bit:	7	6	5	4	3	2	1	0
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

### 15.3.5 Serial Mode Register (SCSMR)

SCSMR specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read from and write to SCSMR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/ $\bar{A}$	0	R/W	Communication Mode Selects whether the SCIF operates in asynchronous or clock synchronous mode. 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data length in asynchronous mode. In the clock synchronous mode, the data length is always 8 bits, regardless of the CHR setting. 0: 8-bit data 1: 7-bit data* Note: * When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data register is not transmitted.

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	<p>Parity Enable</p> <p>Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting.</p> <p>0: Parity bit not added or checked 1: Parity bit added and checked*</p> <p>Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/<math>\bar{E}</math>) setting. Receive data parity is checked according to the even/odd (O/<math>\bar{E}</math>) mode setting.</p>
4	O/ $\bar{E}$	0	R/W	<p>Parity Mode</p> <p>Selects even or odd parity when parity bits are added and checked. The O/<math>\bar{E}</math> setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/<math>\bar{E}</math> setting is ignored in clock synchronous mode, or in asynchronous mode when parity addition and checking is disabled.</p> <p>0: Even parity*<sup>1</sup> 1: Odd parity*<sup>2</sup></p> <p>Notes: 1. If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined.</p> <p>2. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.</p>

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	<p>Stop Bit Length</p> <p>Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in clock synchronous mode because no stop bits are added.</p> <p>When receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character.</p> <p>0: One stop bit When transmitting, a single 1-bit is added at the end of each transmitted character.</p> <p>1: Two stop bits When transmitting, two 1 bits are added at the end of each transmitted character.</p>
2	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
1, 0	CKS[1:0]	00	R/W	<p>Clock Select</p> <p>Select the internal clock source of the on-chip baud rate generator. For further information on the clock source, bit rate register settings, and baud rate, see section 15.3.8, Bit Rate Register (SCBRR).</p> <p>00: Pch 01: Pch/4 10: Pch/16 11: Pch/64</p> <p>Note: Pch: Peripheral clock</p>



### 15.3.6 Serial Control Register (SCSCR)

SCSCR operates the SCIF transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TIE	RIE	TE	RE	REIE	-	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable  Enables or disables the transmit-FIFO-data-empty interrupt (TXI) requested when the serial transmit data is transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), when the quantity of data in the transmit FIFO register becomes less than the specified number of transmission triggers, and when the TDFE flag in the serial status register (SCFSR) is set to 1.  0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled  1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled*  Note: * The TXI interrupt request can be cleared by writing a greater quantity of transmit data than the specified transmission trigger number to SCFTDR and by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0.

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	<p>Receive Interrupt Enable</p> <p>Enables or disables the receive FIFO data full (RXI) interrupts requested when the RDF flag or DR flag in serial status register (SCFSR) is set to 1, receive-error (ERI) interrupts requested when the ER flag in SCFSR is set to 1, and break (BRI) interrupts requested when the BRK flag in SCFSR or the ORER flag in line status register (SCLSR) is set to 1.</p> <p>0: Receive FIFO data full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are disabled</p> <p>1: Receive FIFO data full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are enabled*</p> <p>Note: * RXI interrupt requests can be cleared by reading the DR or RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. ERI or BRI interrupt requests can be cleared by reading the ER, BR or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0.</p>
5	TE	0	R/W	<p>Transmit Enable</p> <p>Enables or disables the serial transmitter.</p> <p>0: Transmitter disabled</p> <p>1: Transmitter enabled*</p> <p>Note: * Serial transmission starts after writing of transmit data into SCFTDR. Select the transmit format in SCSMR and SCFCR and reset the transmit FIFO before setting TE to 1.</p>

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	<p>Receive Enable</p> <p>Enables or disables the serial receiver.</p> <p>0: Receiver disabled*<sup>1</sup></p> <p>1: Receiver enabled*<sup>2</sup></p> <p>Notes: 1. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, RDF, FER, PER, and ORER). These flags retain their previous values.</p> <p>2. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock is detected in clock synchronous mode. Select the receive format in SCSMR and SCFCR and reset the receive FIFO before setting RE to 1.</p>
3	REIE	0	R/W	<p>Receive Error Interrupt Enable</p> <p>Enables or disables the receive-error (ERI) interrupts and break (BRI) interrupts. The setting of REIE bit is valid only when RIE bit is set to 0.</p> <p>0: Receive-error interrupt (ERI) and break interrupt (BRI) requests are disabled</p> <p>1: Receive-error interrupt (ERI) and break interrupt (BRI) requests are enabled*</p> <p>Note: * ERI or BRI interrupt requests can be cleared by reading the ER, BR or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0. Even if RIE is set to 0, when REIE is set to 1, ERI or BRI interrupt requests are enabled. Set so If SCIF wants to inform INTC of ERI or BRI interrupt requests during DMA transfer.</p>

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKE[1:0]	00	R/W	<p>Clock Enable</p> <p>Select the SCIF clock source and enable or disable clock output from the SCK pin. Depending on CKE[1:0], the SCK pin can be used for serial clock output or serial clock input. If serial clock output is set in clock synchronous mode, set the <math>C/\bar{A}</math> bit in SCSMR to 1, and then set CKE[1:0].</p> <ul style="list-style-type: none"> <li>Asynchronous mode</li> </ul> <p>00: Internal clock, SCK pin used for input pin (input signal is ignored)</p> <p>01: Internal clock, SCK pin used for clock output (The output clock frequency is either 16 or 8 times the bit rate.)</p> <p>10: External clock, SCK pin used for clock input (The input clock frequency is either 16 or 8 times the bit rate.)</p> <p>11: Setting prohibited</p> <ul style="list-style-type: none"> <li>Clock synchronous mode</li> </ul> <p>00: Internal clock, SCK pin used for serial clock output</p> <p>01: Internal clock, SCK pin used for serial clock output</p> <p>10: External clock, SCK pin used for serial clock input</p> <p>11: Setting prohibited</p>

### 15.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. The upper 8 bits indicate the number of receive errors in the receive FIFO data register, and the lower 8 bits indicate the status flag indicating SCIF operating state.

The CPU can always read and write to SCFSR, but cannot write 1 to the status flags (ER, TEND, TDFE, BRK, RDF, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). The PER flag (bits 15 to 12 and bit 2) and the FER flag (bits 11 to 8 and bit 3) are read-only bits that cannot be written.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PER[3:0]				FER[3:0]				ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*

Note: \* Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PER[3:0]	0000	R	<p>Number of Parity Errors</p> <p>Indicate the quantity of data including a parity error in the receive data stored in the receive FIFO data register (SCFRDR). The value indicated by bits 15 to 12 after the ER bit in SCFSR is set, represents the number of parity errors in SCFRDR. When parity errors have occurred in all 16-byte receive data in SCFRDR, PER[3:0] shows 0000.</p>
11 to 8	FER[3:0]	0000	R	<p>Number of Framing Errors</p> <p>Indicate the quantity of data including a framing error in the receive data stored in SCFRDR. The value indicated by bits 11 to 8 after the ER bit in SCFSR is set, represents the number of framing errors in SCFRDR. When framing errors have occurred in all 16-byte receive data in SCFRDR, FER[3:0] shows 0000.</p>

Bit	Bit Name	Initial Value	R/W	Description
7	ER	0	R/(W)*	<p>Receive Error</p> <p>Indicates the occurrence of a framing error, or of a parity error when receiving data that includes parity.*<sup>1</sup></p> <p>0: Receiving is in progress or has ended normally</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"><li>• ER is cleared to 0 a power-on reset</li><li>• ER is cleared to 0 when the chip is when 0 is written after 1 is read from ER</li></ul> <p>1: A framing error or parity error has occurred.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"><li>• ER is set to 1 when the stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one data receive operation*<sup>2</sup></li><li>• ER is set to 1 when the total number of 1s in the receive data plus parity bit does not match the even/odd parity specified by the O/<math>\bar{E}</math> bit in SCSMR</li></ul> <p>Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the receive data is transferred to SCFRDR and the receive operation is continued. Whether or not the data read from SCFRDR includes a receive error can be detected by the FER and PER bits in SCFSR.</p> <p>2. In two stop bits mode, only the first stop bit is checked; the second stop bit is not checked.</p>

---

Bit	Bit Name	Initial Value	R/W	Description
6	TEND	1	R/(W)*	<p>Transmit End</p> <p>Indicates that when the last bit of a serial character was transmitted, SCFTDR did not contain valid data, so transmission has ended.</p> <p>0: Transmission is in progress</p> <p>[Clearing condition]</p> <ul style="list-style-type: none"> <li>TEND is cleared to 0 when 0 is written after 1 is read from TEND after transmit data is written in SCFTDR</li> </ul> <p>1: End of transmission</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>TEND is set to 1 when the chip is a power-on reset</li> <li>TEND is set to 1 when TE is cleared to 0 in the serial control register (SCSCR)</li> <li>TEND is set to 1 when SCFTDR does not contain receive data when the last bit of a one-byte serial character is transmitted</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
5	TDFE	1	R/(W)*	<p>Transmit FIFO Data Empty</p> <p>Indicates that data has been transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), the quantity of data in SCFTDR has become less than the transmission trigger number specified by the TTRG[1:0] bits in the FIFO control register (SCFCR), and writing of transmit data to SCFTDR is enabled.</p> <p>0: The quantity of transmit data written to SCFTDR is greater than the specified transmission trigger number</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>TDFE is cleared to 0 when data exceeding the specified transmission trigger number is written to SCFTDR after 1 is read from TDFE and then 0 is written</li> <li>TDFE is cleared to 0 when DMAC is activated by transmit FIFO data empty interrupt (TXI) and write data exceeding the specified transmission trigger number to SCFTDR</li> </ul> <p>1: The quantity of transmit data in SCFTDR is less than or equal to the specified transmission trigger number*</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>TDFE is set to 1 by a power-on reset</li> <li>TDFE is set to 1 when the quantity of transmit data in SCFTDR becomes less than or equal to the specified transmission trigger number as a result of transmission</li> </ul> <p>Note: * Since SCFTDR is a 16-byte FIFO register, the maximum quantity of data that can be written when TDFE is 1 is "16 minus the specified transmission trigger number". If an attempt is made to write additional data, the data is ignored. The quantity of data in SCFTDR is indicated by the upper 8 bits of SCFDR.</p>



Bit	Bit Name	Initial Value	R/W	Description
4	BRK	0	R/(W)*	<p>Break Detection</p> <p>Indicates that a break signal has been detected in receive data.</p> <p>0: No break signal received</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>BRK is cleared to 0 when the chip is a power-on reset</li> <li>BRK is cleared to 0 when software reads BRK after it has been set to 1, then writes 0 to BRK</li> </ul> <p>1: Break signal received*</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>BRK is set to 1 when data including a framing error is received, and a framing error occurs with space 0 in the subsequent receive data</li> </ul> <p>Note: * When a break is detected, transfer of the receive data (H'00) to SCFRDR stops after detection. When the break ends and the receive signal becomes mark 1, the transfer of receive data resumes.</p>
3	FER	0	R	<p>Framing Error Indication</p> <p>Indicates a framing error in the data read from the next receive FIFO data register (SCFRDR) in asynchronous mode.</p> <p>0: No receive framing error occurred in the next data read from SCFRDR</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>FER is cleared to 0 when the chip undergoes a power-on reset</li> <li>FER is cleared to 0 when no framing error is present in the next data read from SCFRDR</li> </ul> <p>1: A receive framing error occurred in the next data read from SCFRDR.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>FER is set to 1 when a framing error is present in the next data read from SCFRDR</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
2	PER	0	R	<p>Parity Error Indication</p> <p>Indicates a parity error in the data read from the next receive FIFO data register (SCFRDR) in asynchronous mode.</p> <p>0: No receive parity error occurred in the next data read from SCFRDR</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"><li>• PER is cleared to 0 when the chip undergoes a power-on reset</li><li>• PER is cleared to 0 when no parity error is present in the next data read from SCFRDR</li></ul> <p>1: A receive parity error occurred in the next data read from SCFRDR</p> <p>[Setting condition]</p> <ul style="list-style-type: none"><li>• PER is set to 1 when a parity error is present in the next data read from SCFRDR</li></ul>

---

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/(W)*	<p>Receive FIFO Data Full</p> <p>Indicates that receive data has been transferred to the receive FIFO data register (SCFRDR), and the quantity of data in SCFRDR has become more than the receive trigger number specified by the RTRG[1:0] bits in the FIFO control register (SCFCR).</p> <p>0: The quantity of transmit data written to SCFRDR is less than the specified receive trigger number</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• RDF is cleared to 0 by a power-on reset, standby mode</li> <li>• RDF is cleared to 0 when the SCFRDR is read until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number after 1 is read from RDF and then 0 is written</li> <li>• RDF is cleared to 0 when DMAC is activated by receive FIFO data full interrupt (RXI) and read SCFRDR until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number</li> </ul> <p>1: The quantity of receive data in SCFRDR is more than the specified receive trigger number</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>• RDF is set to 1 when a quantity of receive data more than the specified receive trigger number is stored in SCFRDR*</li> </ul> <p>Note: * As SCFTDR is a 16-byte FIFO register, the maximum quantity of data that can be read when RDF is 1 becomes the specified receive trigger number. If an attempt is made to read after all the data in SCFRDR has been read, the data is undefined. The quantity of receive data in SCFRDR is indicated by the lower 8 bits of SCFDR.</p>

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/(W)*	<p>Receive Data Ready</p> <p>Indicates that the quantity of data in the receive FIFO data register (SCFRDR) is less than the specified receive trigger number, and that the next data has not yet been received after the elapse of 15 ETU from the last stop bit in asynchronous mode. In clock synchronous mode, this bit is not set to 1.</p> <p>0: Receiving is in progress, or no receive data remains in SCFRDR after receiving ended normally</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>DR is cleared to 0 when the chip undergoes a power-on reset</li> <li>DR is cleared to 0 when all receive data are read after 1 is read from DR and then 0 is written.</li> <li>DR is cleared to 0 when all receive data are read after DMAC is activated by receive FIFO data full interrupt (RXI).</li> </ul> <p>1: Next receive data has not been received</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>DR is set to 1 when SCFRDR contains less data than the specified receive trigger number, and the next data has not yet been received after the elapse of 15 ETU from the last stop bit.*</li> </ul> <p>Note: * This is equivalent to 1.5 frames with the 8-bit, 1-stop-bit format. (ETU: elementary time unit)</p>

Note: \* Only 0 can be written to clear the flag after 1 is read.

### 15.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that is used with the CKS1 and CKS0 bits in the serial mode register (SCSMR) and the BGDM and ABCS bits in the serial extension mode register (SCEMR) to determine the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR. SCBRR is initialized to H'FF by a power-on reset. Each channel has independent baud rate generator control, so different values can be set in three channels.

Bit:	7	6	5	4	3	2	1	0
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
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

The SCBRR setting is calculated as follows:

- Asynchronous mode:

When baud rate generator operates in normal mode (when the BGDM bit of SCEMR is 0):

$$N = \frac{Pch}{64 \times 2^{2n-1} \times B} \times 10^6 - 1 \quad (\text{Operation on a base clock with a frequency of 16 times the bit rate})$$

$$N = \frac{Pch}{32 \times 2^{2n-1} \times B} \times 10^6 - 1 \quad (\text{Operation on a base clock with a frequency of 8 times the bit rate})$$

When baud rate generator operates in double speed mode (when the BGDM bit of SCEMR is 1):

$$N = \frac{Pch}{32 \times 2^{2n-1} \times B} \times 10^6 - 1 \quad (\text{Operation on a base clock with a frequency of 16 times the bit rate})$$

$$N = \frac{Pch}{16 \times 2^{2n-1} \times B} \times 10^6 - 1 \quad (\text{Operation on a base clock with a frequency of 8 times the bit rate})$$

- Clock synchronous mode:

$$N = \frac{Pch}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)

N: SCBRR setting for baud rate generator ( $0 \leq N \leq 255$ )  
(The setting must satisfy the electrical characteristics.)

Pch: Operating frequency for peripheral clock (MHz)

n: Baud rate generator clock source ( $n = 0, 1, 2, 3$ ) (for the clock sources and values of n, see table 15.4.)

**Table 15.4 SCSMR Settings**

n	Clock Source	SCSMR Settings	
		CKS[1]	CKS[0]
0	Pch	0	0
1	Pch/4	0	1
2	Pch/16	1	0
3	Pch/64	1	1

The bit rate error in asynchronous mode is given by the following formula:

When baud rate generator operates in normal mode (the BGDM bit of SCEMR is 0):

$$\text{Error (\%)} = \left\{ \frac{Pch \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100 \quad \text{(Operation on a base clock with a frequency of 16 times the bit rate)}$$

$$\text{Error (\%)} = \left\{ \frac{Pch \times 10^6}{(N + 1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100 \quad \text{(Operation on a base clock with a frequency of 8 times the bit rate)}$$

When baud rate generator operates in double speed mode (the BGDM bit of SCEMR is 1):

$$\text{Error (\%)} = \left\{ \frac{Pch \times 10^6}{(N + 1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100 \quad \text{(Operation on a base clock with a frequency of 16 times the bit rate)}$$

$$\text{Error (\%)} = \left\{ \frac{Pch \times 10^6}{(N + 1) \times B \times 16 \times 2^{2n-1}} - 1 \right\} \times 100 \quad \text{(Operation on a base clock with a frequency of 8 times the bit rate)}$$

Table 15.5 lists the sample SCBRR settings in asynchronous mode in which a base clock frequency is 16 times the bit rate (the ABCS bit in SCEMR is 0) and the baud rate generator operates in normal mode (the BGDM bit in SCEMR is 0), and table 15.6 lists the sample SCBRR settings in clock synchronous mode.

**Table 15.5 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0)**

Bit Rate (bit/s)	Pch (MHz)								
	45			50			54		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	199	-0.12	3	221	-0.02	3	239	-0.12
150	3	145	0.33	3	162	-0.15	3	175	-0.12
300	3	72	0.33	3	80	0.47	3	87	-0.12
600	2	145	0.33	2	162	-0.15	2	175	-0.12
1200	2	72	0.33	2	80	0.47	2	87	-0.12
2400	1	145	0.33	1	162	-0.15	1	175	-0.12
4800	1	72	0.33	1	80	0.47	1	87	-0.12
9600	0	145	0.33	0	162	-0.15	0	175	-0.12
19200	0	72	0.33	0	80	0.47	0	87	-0.12
31250	0	44	0.00	0	49	0.00	0	53	0.00
38400	0	36	-1.02	0	40	-0.76	0	43	-0.12

**Table 15.6 Bit Rates and SCBRR Settings (Clock Synchronous Mode)**

Bit Rate (bit/s)	Pch (MHz)					
	45		50		54	
	n	N	n	N	n	N
110	—	—	—	—	—	—
250	—	—	—	—	—	—
500	—	—	—	—	—	—
1k	3	175	3	194	3	210
2.5k	3	69	3	77	3	83
5k	2	140	2	155	2	168
10k	2	69	2	77	2	83
25k	1	112	1	124	1	134
50k	0	224	0	249	1	67
100k	0	112	0	124	0	134
250k	0	44	0	49	0	53
500k	0	22	0	24	0	26
1M	0	10	0	12	0	13
2M	0	5	0	5	0	6

[Legend]

—: Setting possible, but error occurs



Table 15.7 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Table 15.8 lists the maximum bit rates in asynchronous mode when the external clock input is used. Table 15.9 lists the maximum bit rates in clock synchronous mode when the external clock input is used (when  $t_{\text{Scyc}} = 12t_{\text{pcyc}}^*$ ).

Note: \* Make sure that the electrical characteristics of this LSI and that of a connected LSI are satisfied.

**Table 15.7 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)**

Pch (MHz)	Settings				Maximum Bit Rate (bits/s)
	BGDM	ABCS	n	N	
45	0	0	0	0	1406250
		1	0	0	2812500
	1	0	0	0	2812500
		1	0	0	5625000
50	0	0	0	0	1562500
		1	0	0	3125000
	1	0	0	0	3125000
		1	0	0	6250000
54	0	0	0	0	1687500
		1	0	0	3375000
	1	0	0	0	3375000
		1	0	0	6750000

**Table 15.8 Maximum Bit Rates with External Clock Input (Asynchronous Mode)**

Pch (MHz)	External Input Clock (MHz)	Settings	Maximum Bit Rate (bits/s)
		ABCS	
45	11.2500	0	703125
		1	1406250
50	12.500	0	781250
		1	1562500
54	13.5000	0	843750
		1	1687500

**Table 15.9 Maximum Bit Rates with External Clock Input**  
**(Clock Synchronous Mode,  $t_{\text{Secy}} = 12t_{\text{peyc}}$ )**

Pch (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
45	3.7500	3750000.0
50	4.1666	4166666.6
54	4.5000	4500000.0

### 15.3.9 FIFO Control Register (SCFCR)

SCFCR resets the quantity of data in the transmit and receive FIFO data registers, sets the trigger data quantity, and contains an enable bit for loop-back testing. SCFCR can always be read and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	RSTRG[2:0]			RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	RSTRG[2:0]	000	R/W	<p><math>\overline{\text{RTS}}</math> Output Active Trigger</p> <p>When the quantity of receive data in receive FIFO data register (SCFRDR) becomes more than the number shown below, RTS signal is set to high.</p> <p>000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14</p>
7, 6	RTRG[1:0]	00	R/W	<p>Receive FIFO Data Trigger</p> <ul style="list-style-type: none"> <li>Set the quantity of receive data which sets the receive data full (RDF) flag in the serial status register (SCFSR). The RDF flag is set to 1 when the quantity of receive data stored in the receive FIFO register (SCFRDR) is increased more than the set trigger number shown below.</li> <li>Asynchronous mode      • Clock synchronous mode</li> </ul> <p>00: 1                              00: 1 01: 4                              01: 2 10: 8                              10: 8 11: 14                              11: 14</p> <p>Note: In clock synchronous mode, to transfer the receive data using DMAC, set the receive trigger number to 1. If set to other than 1, CPU must read the receive data left in SCFRDR.</p>

Bit	Bit Name	Initial Value	R/W	Description
5, 4	TTRG[1:0]	00	R/W	<p>Transmit FIFO Data Trigger</p> <p>Set the quantity of remaining transmit data which sets the transmit FIFO data register empty (TDFE) flag in the serial status register (SCFSR). The TDFE flag is set to 1 when the quantity of transmit data in the transmit FIFO data register (SCFTDR) becomes less than the set trigger number shown below.</p> <p>00: 8 (8)*  01: 4 (12)*  10: 2 (14)*  11: 0 (16)*</p> <p>Note: * Values in parentheses mean the number of empty bytes in SCFTDR when the TDFE flag is set to 1.</p>
3	MCE	0	R/W	<p>Modem Control Enable</p> <p>Enables modem control signals <math>\overline{\text{CTS}}</math> and <math>\overline{\text{RTS}}</math>.</p> <p>For channels 0 to 2 in clock synchronous mode, MCE bit should always be 0.</p> <p>0: Modem signal disabled*  1: Modem signal enabled</p> <p>Note: * <math>\overline{\text{CTS}}</math> is fixed at active 0 regardless of the input value, and <math>\overline{\text{RTS}}</math> is also fixed at 0.</p>
2	TFRST	0	R/W	<p>Transmit FIFO Data Register Reset</p> <p>Disables the transmit data in the transmit FIFO data register and resets the data to the empty state.</p> <p>0: Reset operation disabled*  1: Reset operation enabled</p> <p>Note: * Reset operation is executed by a power-on reset.</p>
1	RFRST	0	R/W	<p>Receive FIFO Data Register Reset</p> <p>Disables the receive data in the receive FIFO data register and resets the data to the empty state.</p> <p>0: Reset operation disabled*  1: Reset operation enabled</p> <p>Note: * Reset operation is executed by a power-on reset.</p>

Bit	Bit Name	Initial Value	R/W	Description
0	LOOP	0	R/W	<p>Loop-Back Test</p> <p>Internally connects the transmit output pin (TxD) and receive input pin (RxD) and internally connects the RTS pin and CTS pin and enables loop-back testing.</p> <p>0: Loop back test disabled</p> <p>1: Loop back test enabled</p>

### 15.3.10 FIFO Data Count Set Register (SCFDR)

SCFDR is a 16-bit register which indicates the quantity of data stored in the transmit FIFO data register (SCFTDR) and the receive FIFO data register (SCFRDR).

It indicates the quantity of transmit data in SCFTDR with the upper 8 bits, and the quantity of receive data in SCFRDR with the lower 8 bits. SCFDR can always be read by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	T[4:0]					-	-	-	R[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
12 to 8	T[4:0]	00000	R	<p>T4 to T0 bits indicate the quantity of non-transmitted data stored in SCFTDR. H'00 means no transmit data, and H'10 means that SCFTDR is full of transmit data.</p>
7 to 5	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
4 to 0	R[4:0]	00000	R	<p>R4 to R0 bits indicate the quantity of receive data stored in SCFRDR. H'00 means no receive data, and H'10 means that SCFRDR full of receive data.</p>

### 15.3.11 Serial Port Register (SCSPTR)

SCSPTR controls input/output and data of pins multiplexed to SCIF function. Bits 7 and 6 can control input/output data of  $\overline{\text{RTS}}$  pin. Bits 5 and 4 can control input/output data of  $\overline{\text{CTS}}$  pin. Bits 3 and 2 can control input/output data of SCK pin. Bits 1 and 0 can input data from RxD pin and output data to TxD pin, so they control break of serial transmitting/receiving.

The CPU can always read and write to SCSPTR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
Initial value:	0	0	0	0	0	0	0	0	0	1	0	1	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	$\overline{\text{RTS}}$ Port Input/Output  Indicates input or output of the serial port $\overline{\text{RTS}}$ pin. When the $\overline{\text{RTS}}$ pin is actually used as a port outputting the RTSDT bit value, the MCE bit in SCFCR should be cleared to 0.  0: RTSDT bit value not output to $\overline{\text{RTS}}$ pin 1: RTSDT bit value output to $\overline{\text{RTS}}$ pin
6	RTSDT	1	R/W	$\overline{\text{RTS}}$ Port Data  Indicates the input/output data of the serial port $\overline{\text{RTS}}$ pin. Input/output is specified by the RTSIO bit. For output, the RTSDT bit value is output to the $\overline{\text{RTS}}$ pin. The $\overline{\text{RTS}}$ pin status is read from the RTSDT bit regardless of the RTSIO bit setting. However, $\overline{\text{RTS}}$ input/output must be set in the PFC.  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	<p><math>\overline{\text{CTS}}</math> Port Input/Output</p> <p>Indicates input or output of the serial port <math>\overline{\text{CTS}}</math> pin. When the <math>\overline{\text{CTS}}</math> pin is actually used as a port outputting the CTS<math>\overline{\text{DT}}</math> bit value, the MCE bit in SCFCR should be cleared to 0.</p> <p>0: CTS<math>\overline{\text{DT}}</math> bit value not output to <math>\overline{\text{CTS}}</math> pin 1: CTS<math>\overline{\text{DT}}</math> bit value output to <math>\overline{\text{CTS}}</math> pin</p>
4	CTS $\overline{\text{DT}}$	1	R/W	<p><math>\overline{\text{CTS}}</math> Port Data</p> <p>Indicates the input/output data of the serial port <math>\overline{\text{CTS}}</math> pin. Input/output is specified by the CTSIO bit. For output, the CTS<math>\overline{\text{DT}}</math> bit value is output to the <math>\overline{\text{CTS}}</math> pin. The <math>\overline{\text{CTS}}</math> pin status is read from the CTS<math>\overline{\text{DT}}</math> bit regardless of the CTSIO bit setting. However, <math>\overline{\text{CTS}}</math> input/output must be set in the PFC.</p> <p>0: Input/output data is low level 1: Input/output data is high level</p>
3	SCKIO	0	R/W	<p>SCK Port Input/Output</p> <p>Indicates input or output of the serial port SCK pin. When the SCK pin is actually used as a port outputting the SCK<math>\overline{\text{DT}}</math> bit value, the CK[1:0] bits in SCSCR should be cleared to 0.</p> <p>0: SCK<math>\overline{\text{DT}}</math> bit value not output to SCK pin 1: SCK<math>\overline{\text{DT}}</math> bit value output to SCK pin</p>
2	SCK $\overline{\text{DT}}$	0	R/W	<p>SCK Port Data</p> <p>Indicates the input/output data of the serial port SCK pin. Input/output is specified by the SCKIO bit. For output, the SCK<math>\overline{\text{DT}}</math> bit value is output to the SCK pin. The SCK pin status is read from the SCK<math>\overline{\text{DT}}</math> bit regardless of the SCKIO bit setting. However, SCK input/output must be set in the PFC.</p> <p>0: Input/output data is low level 1: Input/output data is high level</p>

Bit	Bit Name	Initial Value	R/W	Description
1	SPB2IO	0	R/W	<p>Serial Port Break Input/Output</p> <p>Indicates input or output of the serial port TxD pin. When the TxD pin is actually used as a port outputting the SPB2DT bit value, the TE bit in SCSCR should be cleared to 0.</p> <p>0: SPB2DT bit value not output to TxD pin</p> <p>1: SPB2DT bit value output to TxD pin</p>
0	SPB2DT	0	R/W	<p>Serial Port Break Data</p> <p>Indicates the input data of the RxD pin and the output data of the TxD pin used as serial ports. Input/output is specified by the SPB2IO bit. When the TxD pin is set to output, the SPB2DT bit value is output to the TxD pin. The RxD pin status is read from the SPB2DT bit regardless of the SPB2IO bit setting. However, RxD input and TxD output must be set in the PFC.</p> <p>0: Input/output data is low level</p> <p>1: Input/output data is high level</p>



### 15.3.12 Line Status Register (SCLSR)

The CPU can always read or write to SCLSR, but cannot write 1 to the ORER flag. This flag can be cleared to 0 only if it has first been read (after being set to 1).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ORER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*

Note: \* Only 0 can be written to clear the flag after 1 is read.

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)*	<p>Overrun Error</p> <p>Indicates the occurrence of an overrun error.</p> <p>0: Receiving is in progress or has ended normally*<sup>1</sup></p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• ORER is cleared to 0 when the chip is a power-on reset</li> <li>• ORER is cleared to 0 when 0 is written after 1 is read from ORER.</li> </ul> <p>1: An overrun error has occurred*<sup>2</sup></p> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>• ORER is set to 1 when the next serial receiving is finished while the receive FIFO is full of 16-byte receive data.</li> </ul> <p>Notes:</p> <ol style="list-style-type: none"> <li>1. Clearing the RE bit to 0 in SCSCR does not affect the ORER bit, which retains its previous value.</li> <li>2. The receive FIFO data register (SCFRDR) retains the data before an overrun error has occurred, and the next received data is discarded. When the ORER bit is set to 1, the SCIF cannot continue the next serial reception.</li> </ol>

### 15.3.13 Serial Extension Mode Register (SCEMR)

The CPU can always read from or write to SCEMR. Setting the BGDM bit in this register to 1 allows the baud rate generator in the SCIF operates in double-speed mode when asynchronous mode is selected (by setting the  $C/\bar{A}$  bit in SCSMR to 0) and an internal clock is selected as a clock source and the SCK pin is set as an input pin (by setting the  $CKE[1:0]$  bits in SCSCR to 00).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BGDM	-	-	-	-	-	-	ABCS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/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	BGDM	0	R/W	Baud Rate Generator Double-Speed Mode  When the BGDM bit is set to 1, the baud rate generator in the SCIF operates in double-speed mode. This bit is valid only when asynchronous mode is selected by setting the $C/\bar{A}$ bit in SCSMR to 0 and an internal clock is selected as a clock source and the SCK pin is set as an input pin by setting the $CKE[1:0]$ bits in SCSCR to 00. In other settings, this bit is invalid (the baud rate generator operates in normal mode regardless of the BGDM setting).  0: Normal mode 1: Double-speed mode
6 to 1	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
0	ABCS	0	R/W	Base Clock Select in Asynchronous Mode  This bit selects the base clock frequency within a bit period in asynchronous mode. This bit is valid only in asynchronous mode (when the $C/\bar{A}$ bit in SCSMR is 0).  0: Base clock frequency is 16 times the bit rate 1: Base clock frequency is 8 times the bit rate

## 15.4 Operation

### 15.4.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses.

The SCIF has a 16-stage FIFO buffer for both transmission and receptions, reducing the overhead of the CPU, and enabling continuous high-speed communication. Furthermore, each channel has RTS and CTS signals to be used as modem control signals.

The transmission format is selected in the serial mode register (SCSMR), as shown in table 15.10. The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCSCR), as shown in table 15.11.

#### (1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, receive FIFO data full, overrun errors, receive data ready, and breaks.
- The number of stored data bytes is indicated for both the transmit and receive FIFO registers.
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of on-chip baud rate generator.
  - When an external clock is selected, the external clock input must have a frequency 16 or 8 times the bit rate. (The on-chip baud rate generator is not used.)

**(2) Clock Synchronous Mode**

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of the on-chip baud rate generator, and outputs this clock to external devices as the synchronous clock.
  - When an external clock is selected, the SCIF operates on the input external synchronous clock not using the on-chip baud rate generator.

**Table 15.10 SCSMR Settings and SCIF Communication Formats**

SCSMR Settings					SCIF Communication Format		
Bit 7 C/ $\bar{A}$	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Parity Bit	Stop Bit Length
0	0	0	0	Asynchronous	8 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	x	x	x	Clock synchronous	8 bits	Not set	None

[Legend]

x: Don't care

**Table 15.11 SCSMR and SCSCR Settings and SCIF Clock Source Selection**

SCSMR		SCSCR	SCIF Transmit/Receive Clock	
Bit 7 C/ $\bar{A}$	Bit 1, 0 CKE[1:0]	Mode	Clock Source	SCK Pin Function
0	00	Asynchronous	Internal	SCIF does not use the SCK pin
	01			Outputs a clock with a frequency 16 or 8 times the bit rate
	10		External	Inputs a clock with frequency 16 or 8 times the bit rate
	11		Setting prohibited	
1	0x	Clock synchronous	Internal	Outputs the serial clock
	10		External	Inputs the serial clock
	11		Setting prohibited	

[Legend]

x: Don't care

Note: When using the baud rate generator in double-speed mode (BGMD = 1), select asynchronous mode by setting the C/A bit to 0, and select an internal clock as a clock source and the SCK pin is not used (the CKE[1:0] bits set to 00).

### 15.4.2 Operation in Asynchronous Mode

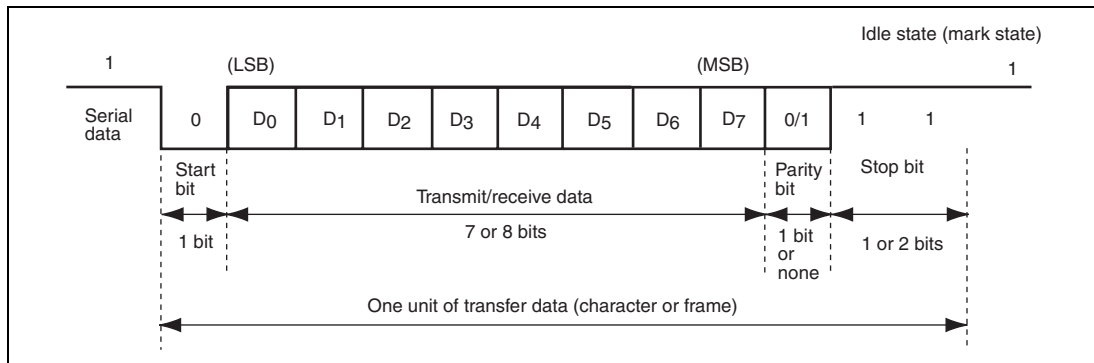
In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCIF are independent, so full duplex communication is possible. The transmitter and receiver are 16-byte FIFO buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 15.2 shows the general format of asynchronous serial communication.

In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCIF monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCIF synchronizes at the falling edge of the start bit. The SCIF samples each data bit on the eighth or fourth pulse of a clock with a frequency 16 or 8 times the bit rate. Receive data is latched at the center of each bit.



**Figure 15.2 Example of Data Format in Asynchronous Communication  
(8-Bit Data with Parity and Two Stop Bits)**

**(1) Transmit/Receive Formats**

Table 15.12 lists the eight communication formats that can be selected in asynchronous mode. The format is selected by settings in the serial mode register (SCSMR).

**Table 15.12 Serial Communication Formats (Asynchronous Mode)**

SCSMR Bits			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-bit data							STOP			
0	0	1	START	8-bit data							STOP	STOP		
0	1	0	START	8-bit data							P	STOP		
0	1	1	START	8-bit data							P	STOP	STOP	
1	0	0	START	7-bit data						STOP				
1	0	1	START	7-bit data						STOP	STOP			
1	1	0	START	7-bit data						P	STOP			
1	1	1	START	7-bit data						P	STOP	STOP		

[Legend]

START: Start bit

STOP: Stop bit

P: Parity bit

## (2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock. The clock source is selected by the  $C/\overline{A}$  bit in the serial mode register (SCSMR) and the CKE1 and CKE0 bits in the serial control register (SCSCR). For clock source selection, refer to table 15.11, SCSMR and SCSCR Settings and SCIF Clock Source Selection.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 or 8 times the desired bit rate.

When the SCIF operates on an internal clock, it can output a clock signal on the SCK pin. The frequency of this output clock is 16 or 8 times the desired bit rate.

## (3) Transmitting and Receiving Data

- SCIF Initialization (Asynchronous Mode)

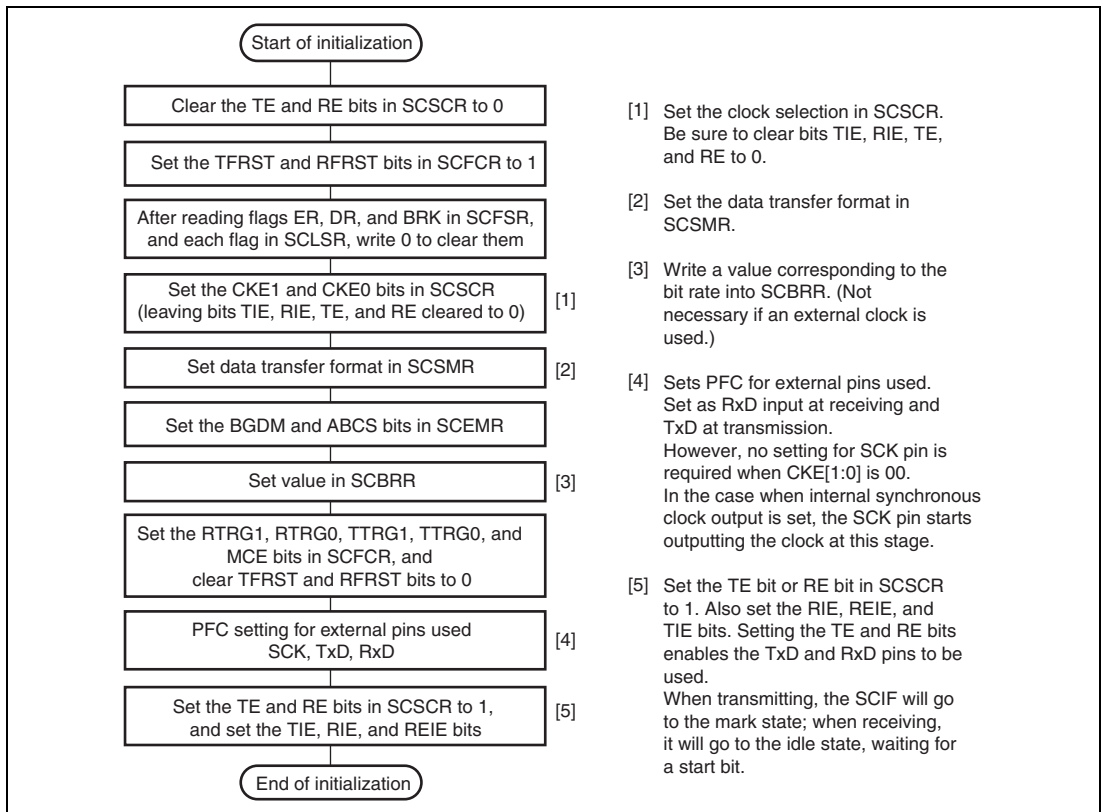
Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF as follows.

When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing TE and RE to 0, however, does not initialize the serial status register (SCFSR), transmit FIFO data register (SCFTDR), or receive FIFO data register (SCFRDR), which retain their previous contents. Clear TE to 0 after all transmit data has been transmitted and the TEND flag in the SCFSR is set. The TE bit can be cleared to 0 during transmission, but the transmit data goes to the Mark state after the bit is cleared to 0. Set the TFRST bit in SCFCR to 1 and reset SCFTDR before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCIF operation becomes unreliable if the clock is stopped.



Figure 15.3 shows a sample flowchart for initializing the SCIF.

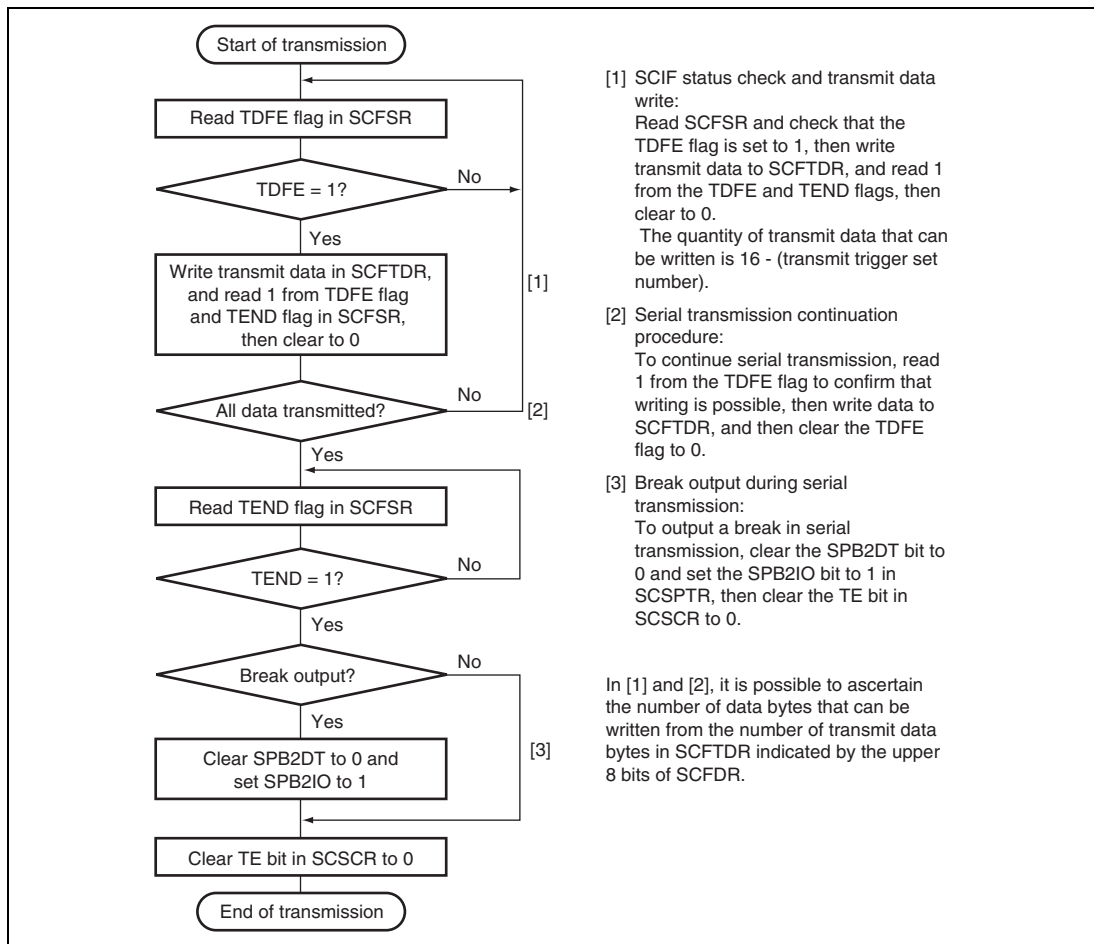


**Figure 15.3 Sample Flowchart for SCIF Initialization**

- Transmitting Serial Data (Asynchronous Mode)

Figure 15.4 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.



**Figure 15.4 Sample Flowchart for Transmitting Serial Data**

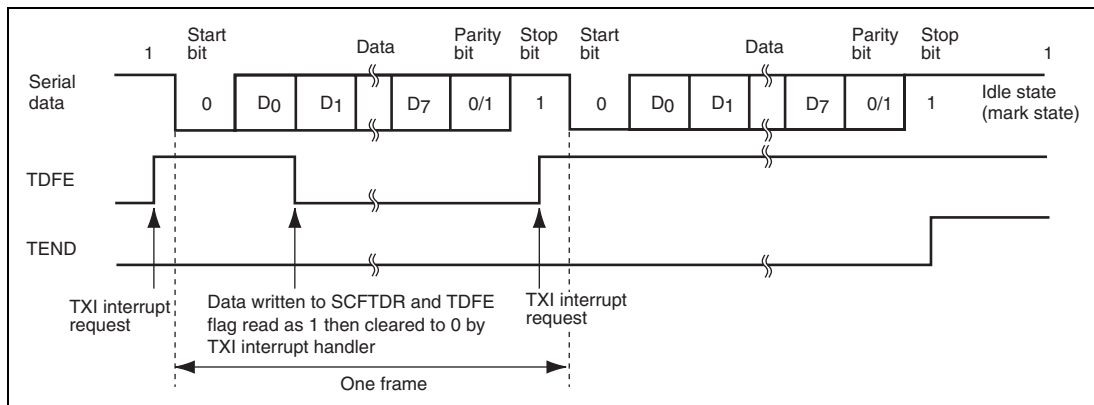
In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmitting. Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

The serial transmit data is sent from the TxD pin in the following order.

- A. Start bit: One-bit 0 is output.
  - B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
  - C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)
  - D. Stop bit(s): One or two 1 bits (stop bits) are output.
  - E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

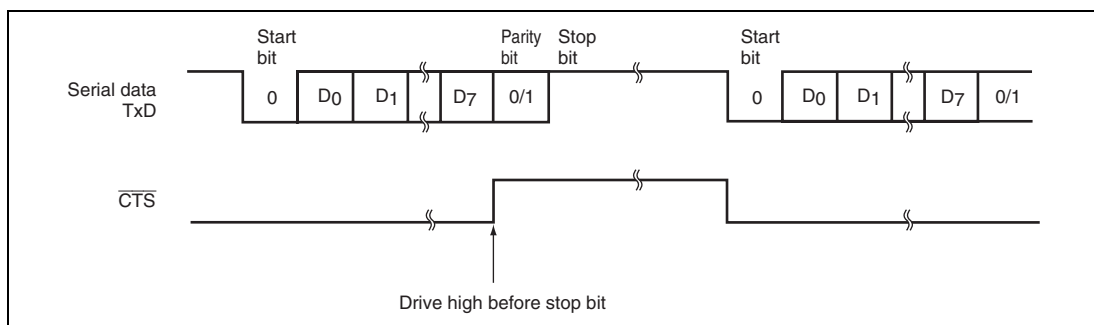
Figure 15.5 shows an example of the operation for transmission.



**Figure 15.5 Example of Transmit Operation  
(8-Bit Data, Parity, 1 Stop Bit)**

- When modem control is enabled, transmission can be stopped and restarted in accordance with the  $\overline{\text{CTS}}$  input value. When  $\overline{\text{CTS}}$  is set to 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When  $\overline{\text{CTS}}$  is set to 0, the next transmit data is output starting from the start bit.

Figure 15.6 shows an example of the operation when modem control is used.

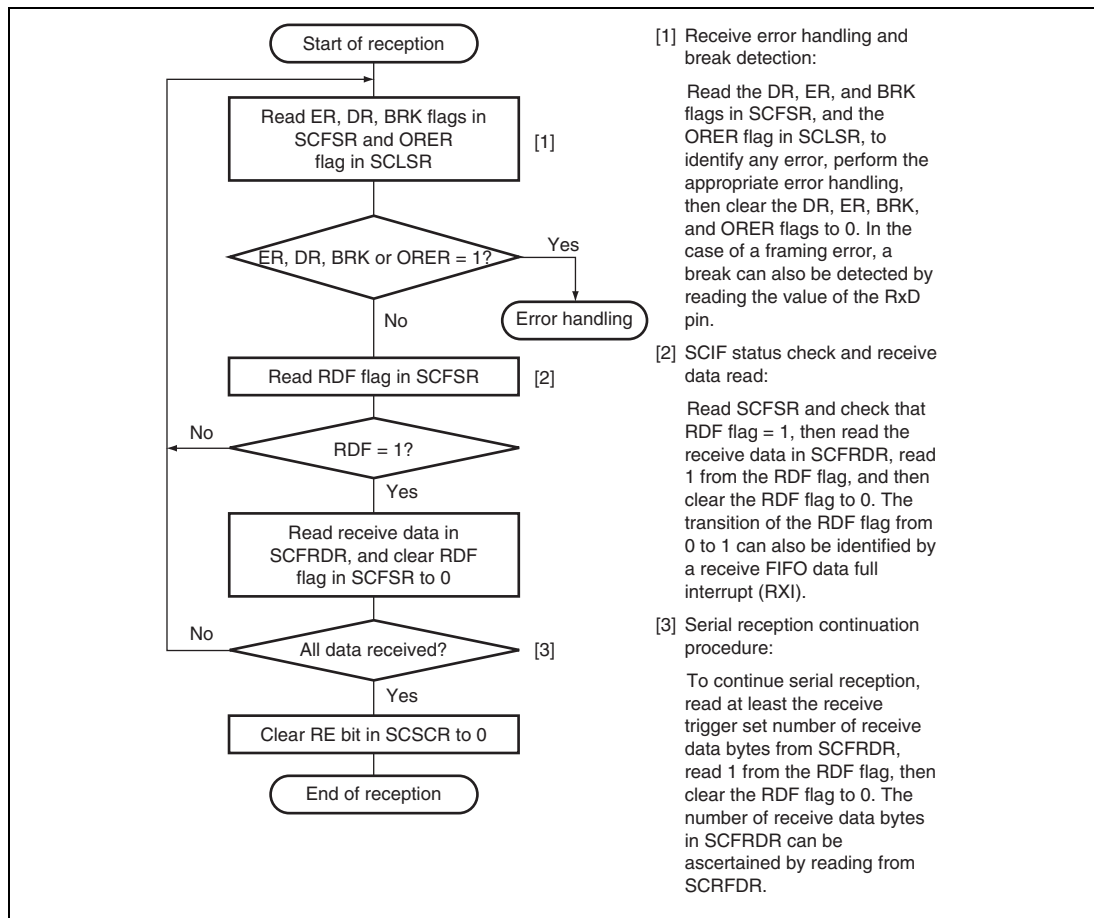


**Figure 15.6 Example of Operation Using Modem Control ( $\overline{\text{CTS}}$ )**

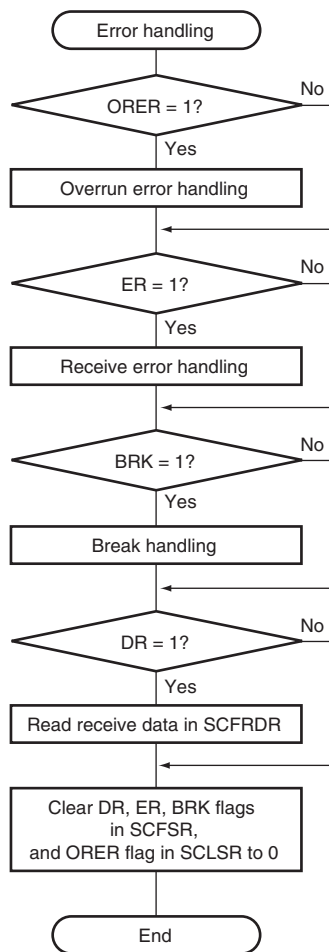
- Receiving Serial Data (Asynchronous Mode)

Figures 15.7 and 15.8 show sample flowcharts for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.



**Figure 15.7 Sample Flowchart for Receiving Serial Data**



- Whether a framing error or parity error has occurred in the receive data that is to be read from the receive FIFO data register (SCFRDR) can be ascertained from the FER and PER bits in the serial status register (SCFSR).
- When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.

**Figure 15.8 Sample Flowchart for Receiving Serial Data (cont)**

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.

After receiving these bits, the SCIF carries out the following checks.

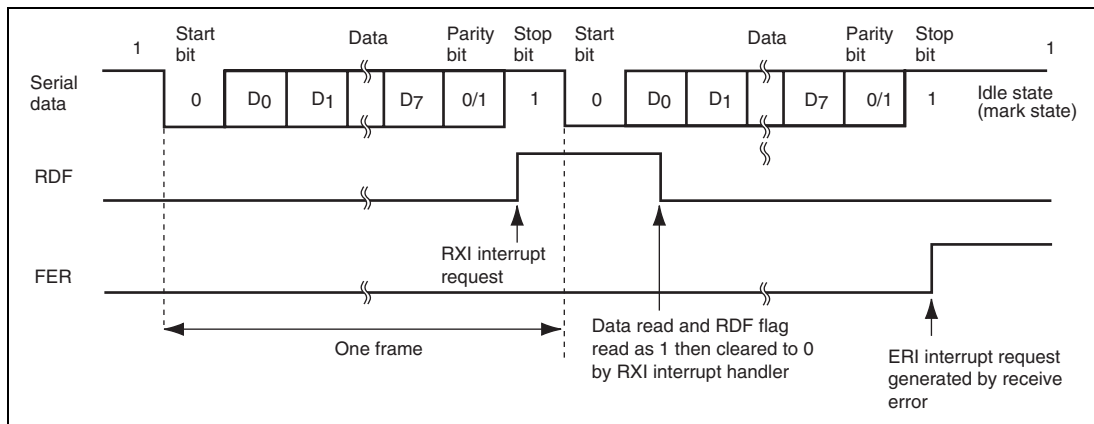
- A. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
- B. The SCIF checks whether receive data can be transferred from the receive shift register (SCRSR) to SCFRDR.
- C. Overrun check: The SCIF checks that the ORER flag is 0, indicating that the overrun error has not occurred.
- D. Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.

If all the above checks are passed, the receive data is stored in SCFRDR.

Note: When a parity error or a framing error occurs, reception is not suspended.

4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

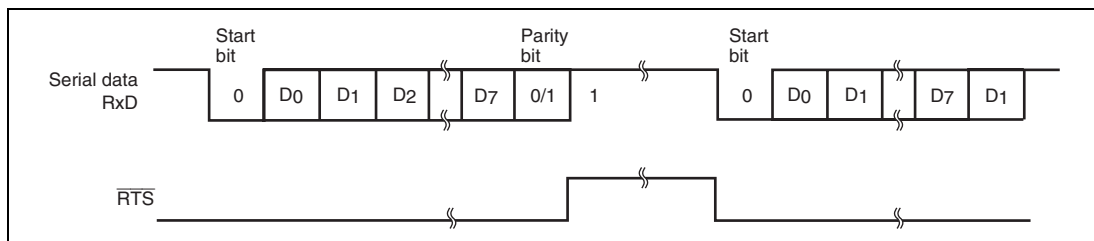
Figure 15.9 shows an example of the operation for reception.



**Figure 15.9 Example of SCIF Receive Operation  
(8-Bit Data, Parity, 1 Stop Bit)**

- When modem control is enabled, the  $\overline{\text{RTS}}$  signal is output when SCFRDR is empty. When  $\overline{\text{RTS}}$  is 0, reception is possible. When  $\overline{\text{RTS}}$  is 1, this indicates that SCFRDR exceeds the number set for the RTS output active trigger.

Figure 15.10 shows an example of the operation when modem control is used.



**Figure 15.10 Example of Operation Using Modem Control ( $\overline{\text{RTS}}$ )**

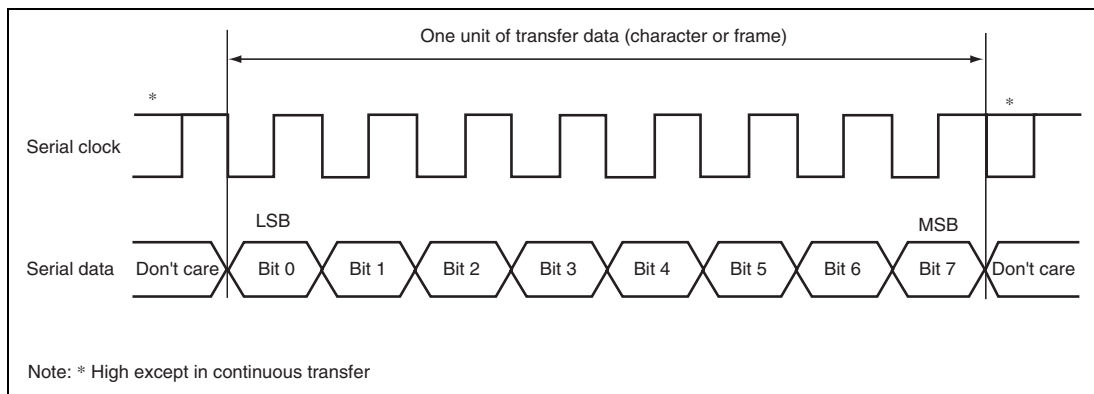


### 15.4.3 Operation in Clock Synchronous Mode

In clock synchronous mode, the SCIF transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCIF transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. The transmitter and receiver are also 16-byte FIFO buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 15.11 shows the general format in clock synchronous serial communication.



**Figure 15.11 Data Format in Clock Synchronous Communication**

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock.

In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB.

In clock synchronous mode, the SCIF receives data by synchronizing with the rising edge of the serial clock.

### (1) Transmit/Receive Formats

The data length is fixed at eight bits. No parity bit can be added.

### (2) Clock

An internal clock generated by the on-chip baud rate generator by the setting of the C/A bit in SCSMR and CKE[1:0] in SCSCR, or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock.

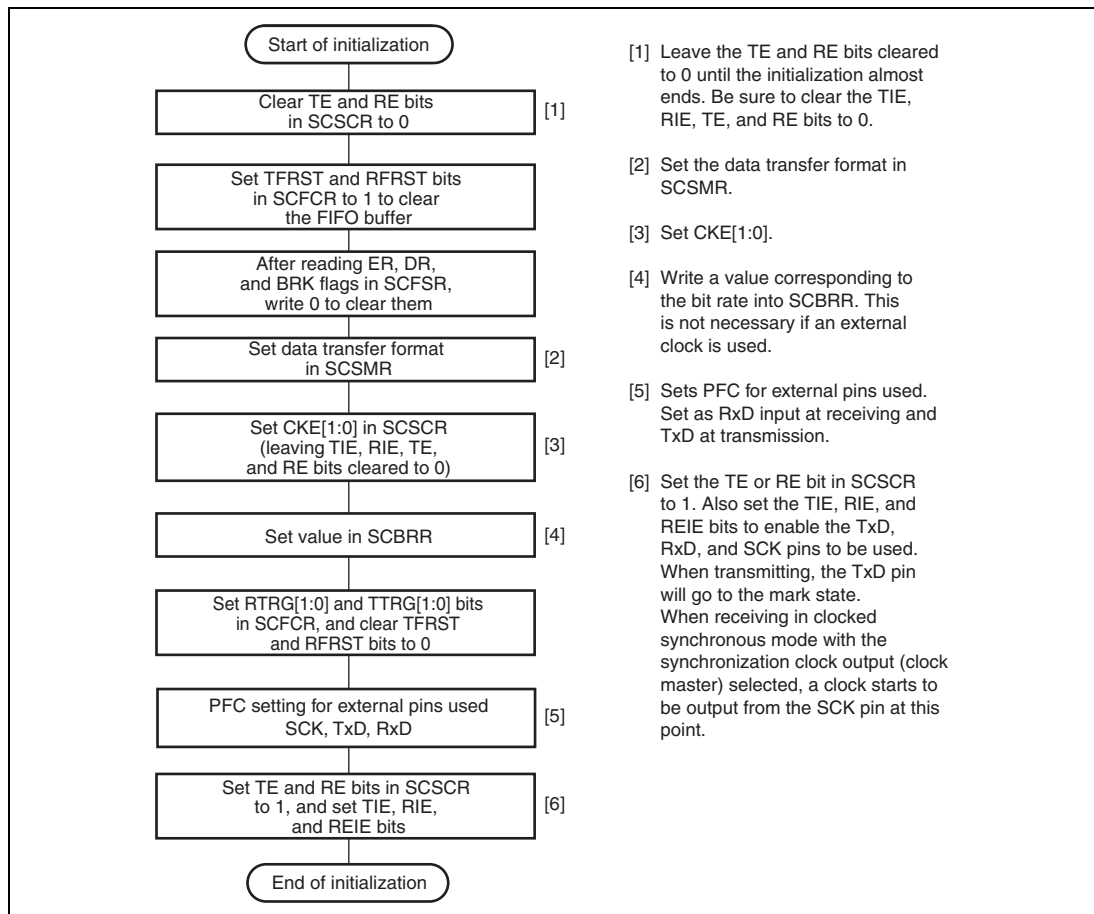
When the SCIF operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCIF is not transmitting or receiving, the clock signal remains in the high state. When only receiving, the clock signal outputs while the RE bit of SCSCR is 1 and the number of data in receive FIFO is more than the receive FIFO data trigger number.

### (3) Transmitting and Receiving Data

- SCIF Initialization (Clock Synchronous Mode)

Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDF, PER, FER, and ORER flags and receive data register (SCRDR), which retain their previous contents.

Figure 15.12 shows a sample flowchart for initializing the SCIF.

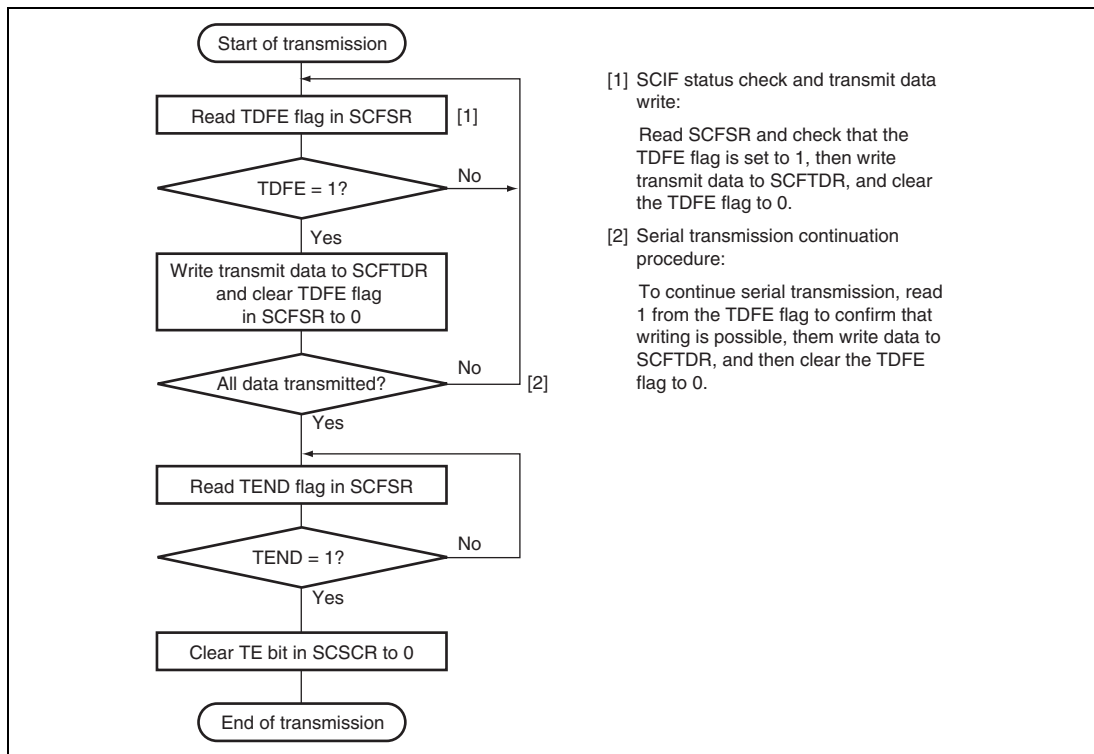


**Figure 15.12 Sample Flowchart for SCIF Initialization**

- Transmitting Serial Data (Clock Synchronous Mode)

Figure 15.13 shows a sample flowchart for transmitting serial data.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.



**Figure 15.13 Sample Flowchart for Transmitting Serial Data**

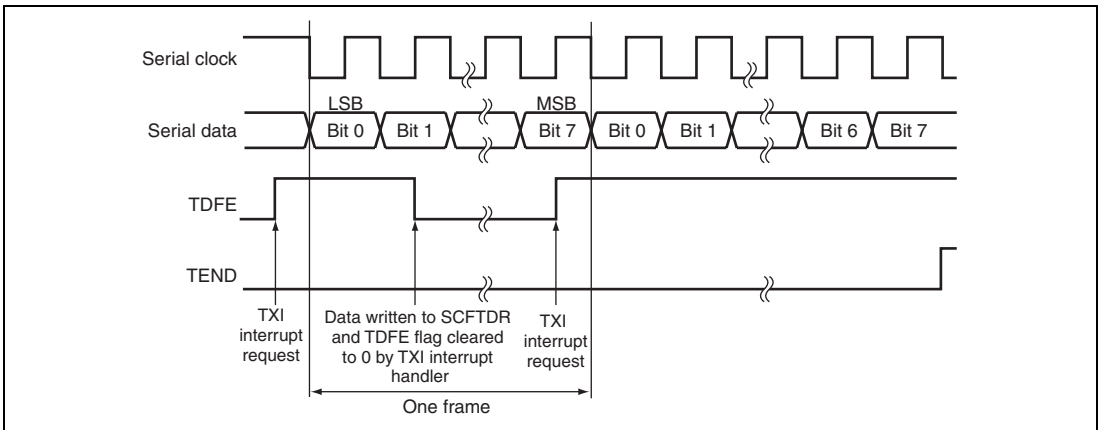
In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmitting. Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

If clock output mode is selected, the SCIF outputs eight synchronous clock pulses. If an external clock source is selected, the SCIF outputs data in synchronization with the input clock. Data is output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).

3. The SCIF checks the SCFTDR transmit data at the timing for sending the MSB (bit 7). If data is present, the data is transferred from SCFTDR to SCTSR, and then serial transmission of the next frame is started. If there is no data, the TxD pin holds the state after the TEND flag in SCFSR is set to 1 and the MSB (bit 7) is sent.
4. After the end of serial transmission, the SCK pin is held in the high state.

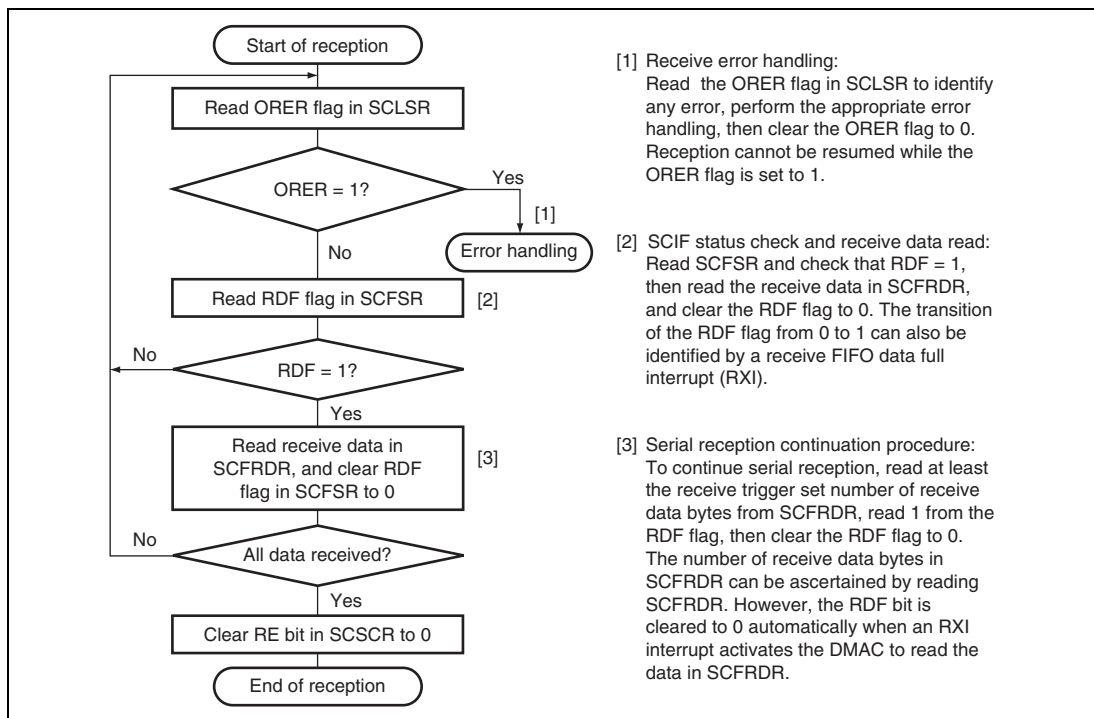
Figure 15.14 shows an example of SCIF transmit operation.



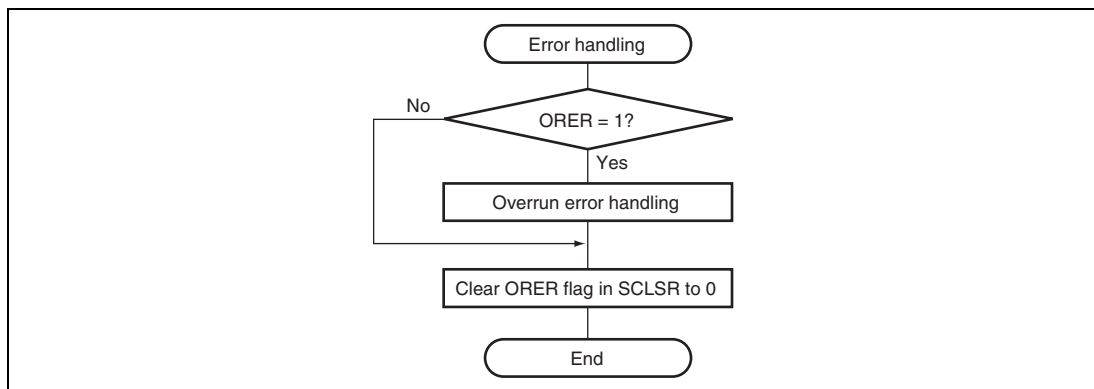
**Figure 15.14 Example of SCIF Transmit Operation**

- Receiving Serial Data (Clock Synchronous Mode)

Figures 15.15 and 15.16 show sample flowcharts for receiving serial data. When switching from asynchronous mode to clock synchronous mode without SCIF initialization, make sure that ORER, PER, and FER are cleared to 0.



**Figure 15.15 Sample Flowchart for Receiving Serial Data (1)**

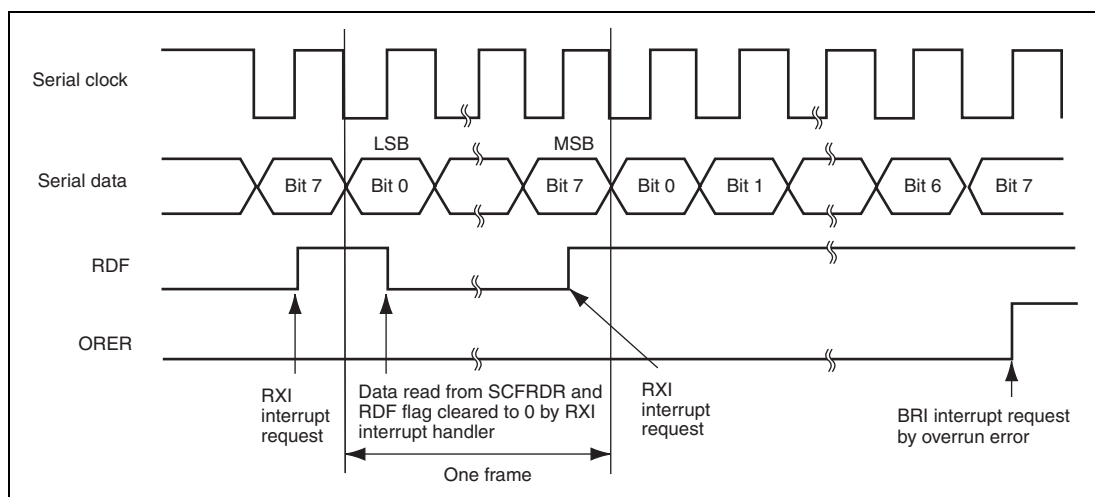


**Figure 15.16 Sample Flowchart for Receiving Serial Data (2)**

In serial reception, the SCIF operates as described below.

1. The SCIF synchronizes with serial clock input or output and starts the reception.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB. After receiving the data, the SCIF checks the receive data can be loaded from SCRSR into SCFRDR or not. If this check is passed, the RDF flag is set to 1 and the SCIF stores the received data in SCFRDR. If the check is not passed (overrun error is detected), further reception is prevented.
3. After setting RDF to 1, if the receive FIFO data full interrupt enable bit (RIE) is set to 1 in SCSCR, the SCIF requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) or the receive error interrupt enable bit (REIE) in SCSCR is also set to 1, the SCIF requests a break interrupt (BRI).

Figure 15.17 shows an example of SCIF receive operation.

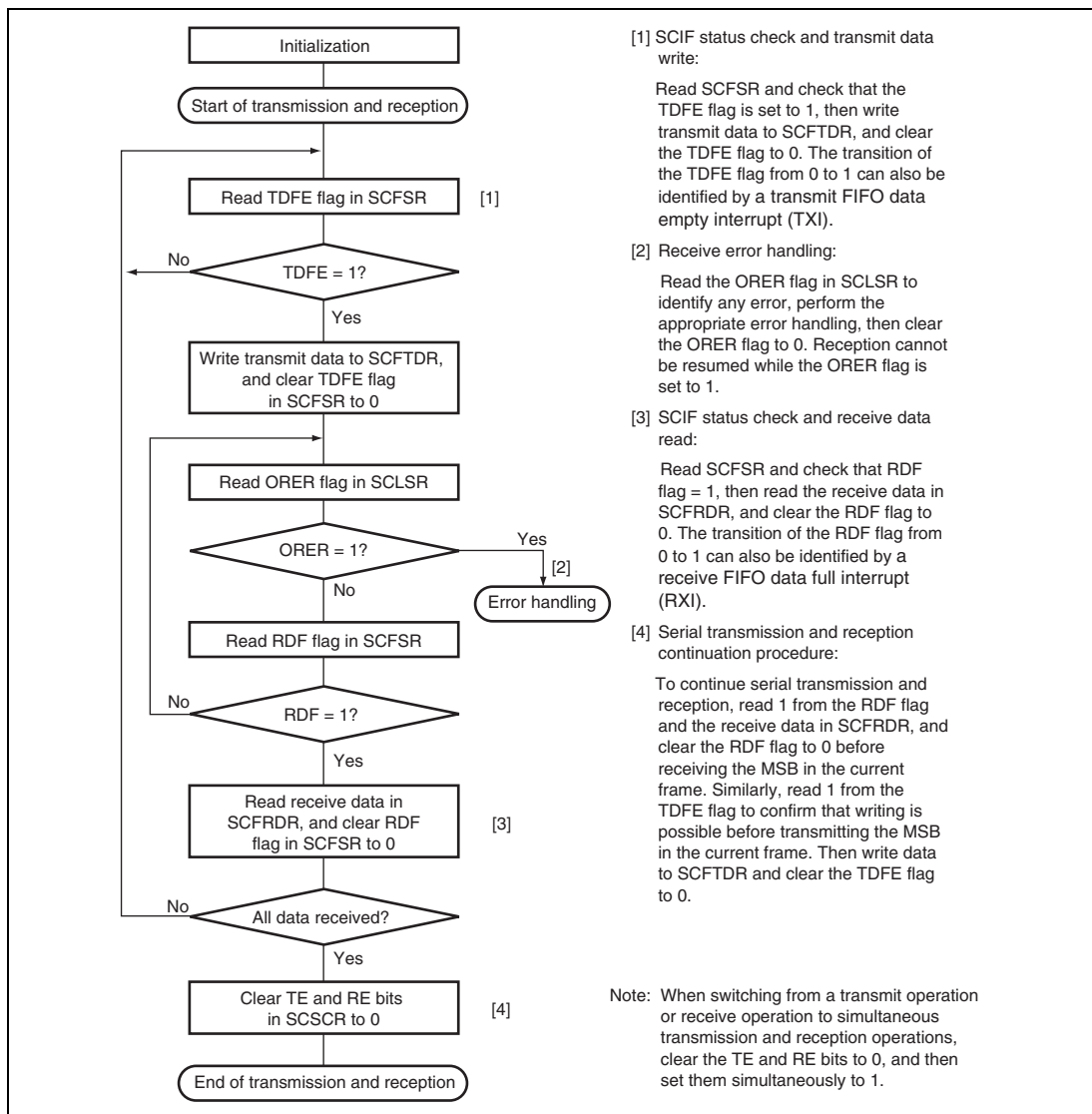


**Figure 15.17 Example of SCIF Receive Operation**

- Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)

Figure 15.18 shows a sample flowchart for transmitting and receiving serial data simultaneously.

Use the following procedure for the simultaneous transmission/reception of serial data, after enabling the SCIF for transmission/reception.



**Figure 15.18 Sample Flowchart for Transmitting/Receiving Serial Data**



## 15.5 SCIF Interrupts

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive FIFO data full (RXI), and break (BRI).

Table 15.13 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.


When a TXI request is enabled by the TIE bit and the TDFE flag in the serial status register (SCFSR) is set to 1, a TXI interrupt request is generated. The DMAC can be activated and data transfer performed by this TXI interrupt request. At this time, an interrupt request is not sent to the CPU.

When an RXI request is enabled by the RIE bit and the RDF flag or the DR flag in SCFSR is set to 1, an RXI interrupt request is generated. The DMAC can be activated and data transfer performed by this RXI interrupt request. At this time, an interrupt request is not sent to the CPU. The RXI interrupt request caused by the DR flag is generated only in asynchronous mode.

When the RIE bit is set to 0 and the REIE bit is set to 1, the SCIF requests only an ERI interrupt without requesting an RXI interrupt.

The TXI indicates that transmit data can be written, and the RXI indicates that there is receive data in SCFRDR.

**Table 15.13 SCIF Interrupt Sources**

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
BRI	Interrupt initiated by break (BRK) or overrun error (ORER)	Not possible	
ERI	Interrupt initiated by receive error (ER)	Not possible	
RXI	Interrupt initiated by receive FIFO data full (RDF) or data ready (DR)	Possible	
TXI	Interrupt initiated by transmit FIFO data empty (TDFE)	Possible	

## 15.6 Usage Notes

Note the following when using the SCIF.

### 15.6.1 SCFTDR Writing and TDFE Flag

The TDFE flag in the serial status register (SCFSR) is set when the number of transmit data bytes written in the transmit FIFO data register (SCFTDR) has fallen below the transmit trigger number set by bits TTRG[1:0] in the FIFO control register (SCFCR). After the TDFE flag is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to 0. TDFE flag clearing should therefore be carried out when SCFTDR contains more than the transmit trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from the upper 8 bits of the FIFO data count register (SCFDR).

### 15.6.2 SCFRDR Reading and RDF Flag

The RDF flag in the serial status register (SCFSR) is set when the number of receive data bytes in the receive FIFO data register (SCFRDR) has become equal to or greater than the receive trigger number set by bits RTRG[1:0] in the FIFO control register (SCFCR). After RDF flag is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes in SCFRDR exceeds the trigger number, the RDF flag will be set to 1 again if it is cleared to 0. The RDF flag should therefore be cleared to 0 after being read as 1 after reading the number of the received data in the receive FIFO data register (SCFRDR) which is less than the trigger number.

The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the FIFO data count register (SCFDR).

### 15.6.3 Break Detection and Processing

Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set.

Note that, although transfer of receive data to SCFRDR is halted in the break state, the SCIF receiver continues to operate.

### 15.6.4 Sending a Break Signal

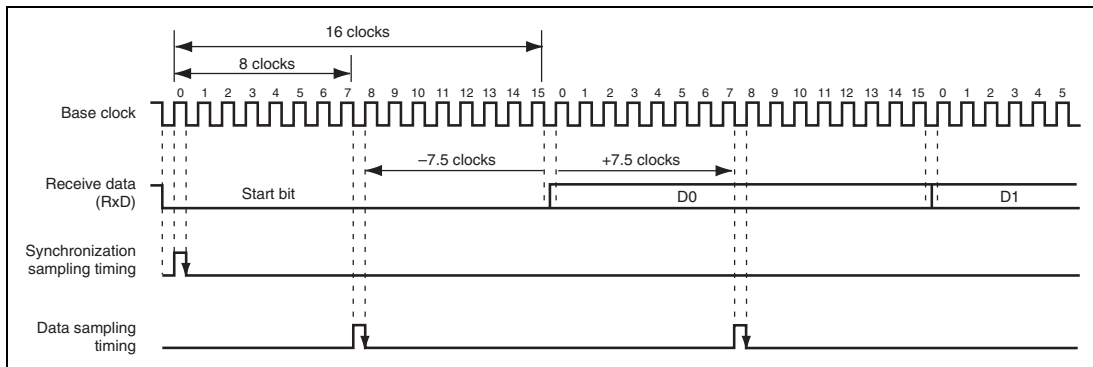
The I/O condition and level of the TxD pin are determined by the SPB2IO and SPB2DT bits in the serial port register (SCSPTR). This feature can be used to send a break signal.

Until TE bit is set to 1 (enabling transmission) after initializing, the TxD pin does not work. During the period, mark status is performed by the SPB2DT bit. Therefore, the SPB2IO and SPB2DT bits should be set to 1 (high level output).

To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating low level), 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 TxD pin.

### 15.6.5 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCIF operates on a base clock with a frequency 16 or 8 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 or fourth base clock pulse. When the SCIF operates on a base clock with a frequency 16 times the bit rate, the receive data is sampled at the timing shown in figure 15.19.



**Figure 15.19 Receive Data Sampling Timing in Asynchronous Mode  
(Operation on a Base Clock with a Frequency 16 Times the Bit Rate)**

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

**Equation 1:**

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

Where: M: Receive margin (%)

N: Ratio of clock frequency to bit rate (N = 16 or 8)

D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute deviation of clock frequency

From equation 1, if F = 0, D = 0.5 and N = 16, the receive margin is 46.875%, as given by equation 2.

**Equation 2:**

When D = 0.5 and F = 0:

$$\begin{aligned} M &= (0.5 - 1/(2 \times 16)) \times 100\% \\ &= 46.875\% \end{aligned}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

### 15.6.6 Selection of Base Clock in Asynchronous Mode

In this LSI, when asynchronous mode is selected, the base clock frequency within a bit period can be set to the frequency 16 or 8 times the bit rate by setting the ABCS bit in SCEMR.

Note that, however, if the base clock frequency 8 times the bit rate is used, receive margin is decreased as calculated using equation 1 in section 15.6.5, Receive Data Sampling Timing and Receive Margin (Asynchronous Mode).

If the desired bit rate can be set simply by setting SCBRR and the CKS1 and CKS0 bits in SCSMR, it is recommended to use the base clock frequency within a bit period 16 times the bit rate (by setting the ABCS bit in SCEMR to 0). If an internal clock is selected as a clock source and the SCK pin is not used, the bit rate can be increased without decreasing receive margin by selecting double-speed mode for the baud rate generator (setting the BGDM bit in SCEMR to 1).



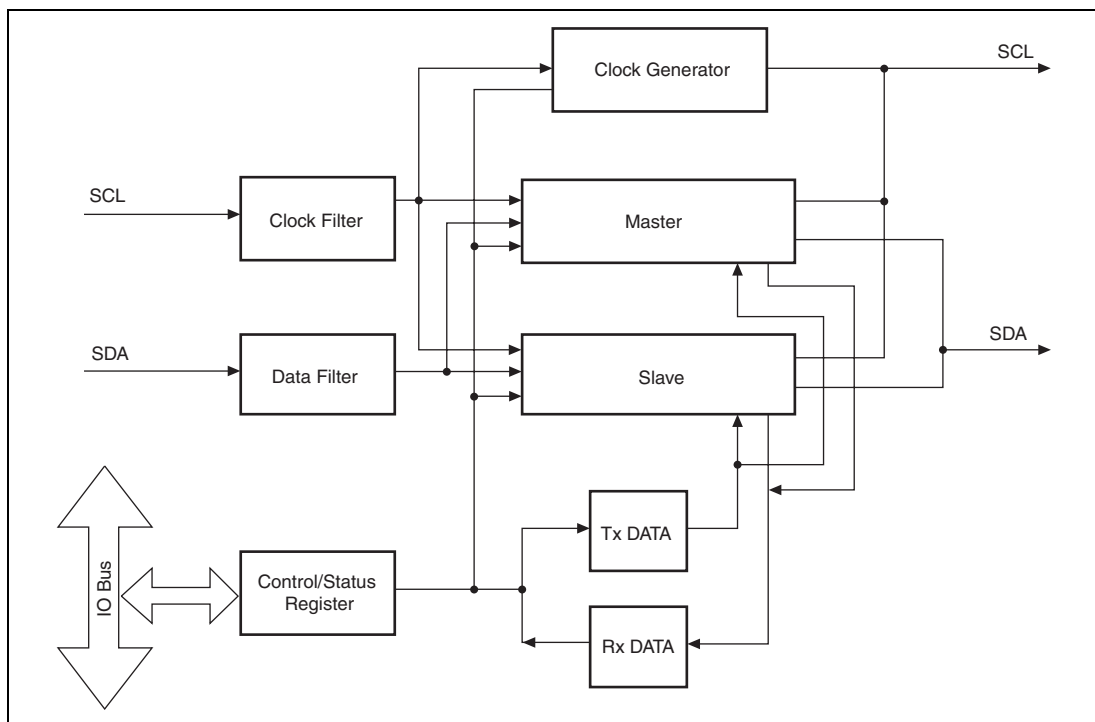
## Section 16 I<sup>2</sup>C Bus Interface (IIC)

### 16.1 Features

The I<sup>2</sup>C bus interface has the following features:

- Supports the Philips I<sup>2</sup>C bus interface
- Multi-master compatible
- Seven- or ten-bit address compatible master
- Seven-bit slave address
- Fast mode compatible
- Variable clock frequencies

Figure 16.1 shows a block diagram for the I<sup>2</sup>C bus interface.



**Figure 16.1 Block Diagram for I<sup>2</sup>C Bus Interface**

## 16.2 Input/Output Pins

Table 16.1 lists the pins used in the I<sup>2</sup>C bus interface.

**Table 16.1 Pin Configuration**

Pin Name	I/O	Description
SCL	I/O	I <sup>2</sup> C serial clock input/output pin*
SDA	I/O	I <sup>2</sup> C serial data input/output pin*

Note: \* The SCL and SDA pins are open drain pins (3.3 V).

## 16.3 Register Descriptions

Table 16.2 shows the IIC register configuration. Table 16.3 shows the register state in each operating mode.

**Table 16.2 Register Configuration**

Register Name	Abbreviation	R/W	Area P4 Address* <sup>1</sup>	Area 7 Address* <sup>1</sup>	Access Size
Slave control register	ICSCR	R/W	H'FFE7 0000	H'1FF7 0000	8
Master control register	ICMCR	R/W	H'FFE7 0004	H'1FF7 0004	8
Slave status register	ICSSR	R/(W)* <sup>2</sup>	H'FFE7 0008	H'1FF7 0008	8
Master status register	ICMSR	R/(W)* <sup>3</sup>	H'FFE7 000C	H'1FF7 000C	8
Slave interrupt enable register	ICSIER	R/W	H'FFE7 0010	H'1FF7 0010	8
Master interrupt enable register	ICMIER	R/W	H'FFE7 0014	H'1FF7 0014	8
Clock control register	ICCCR	R/W	H'FFE7 0018	H'1FF7 0018	8
Slave address register	ICSAR	R/W	H'FFE7 001C	H'1FF7 001C	8
Master address register	ICMAR	R/W	H'FFE7 0020	H'1FF7 0020	8
Receive data register	ICRXD	R/W	H'FFE7 0024	H'1FF7 0024	8
Transmit data register	ICTXD	R/W	H'FFE7 0024	H'1FF7 0024	8

Notes: 1. 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.

2. Only 0 can be written to bits 4 to 0 to clear the flags.

3. Only 0 can be written to bits 6 to 0 to clear the flags.



**Table 16.3 Register State in Each Operating Mode**

<b>Register Name</b>	<b>Abbreviation</b>	<b>Power-On Reset</b>	<b>Sleep</b>	<b>Standby</b>
Slave control register	ICSCR	H'00	Retained	Retained
Master control register	ICMCR	H'x0	Retained	Retained
Slave status register	ICSSR	H'00	Retained	Retained
Master status register	ICMSR	H'00	Retained	Retained
Slave interrupt enable register	ICSIER	H'00	Retained	Retained
Master interrupt enable register	ICMIER	H'00	Retained	Retained
Clock control register	ICCCR	H'00	Retained	Retained
Slave address register	ICSAR	H'00	Retained	Retained
Master address register	ICMAR	H'00	Retained	Retained
Receive data register	ICRXD	H'00	Retained	Retained
Transmit data register	ICTXD	H'00	Retained	Retained

### 16.3.1 Slave Control Register (ICSCR)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	SDBS	SIE	GCAE	FNA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved 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.  This bit is cleared by setting the MIE bit to 1.
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

Bit	Bit Name	Initial Value	R/W	Description
0	FNA	0	R/W	<p>Forced Non Acknowledgement</p> <p>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.</p> <p>Forced non acknowledgement is returned to the master during slave reception.</p> <p>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.</p>

### 16.3.2 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:	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
R/W:	R	R	R	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	<p>Reserved</p> <p>The write value should always be 0.</p>
6	GCAR	0	R	<p>General Call Address Received</p> <p>Indicates that the address received from the bus is a general call address (00H). This status bit does not cause an interrupt.</p> <p>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.</p>

Bit	Bit Name	Initial Value	R/W	Description
5	STM	0	R	<p>Slave Transmit Mode</p> <p>Indicates whether the current slave transmit mode is read or write. When this bit is set to 1, the mode is write. When this bit is set to 0, the mode is read. This status bit does not cause an interrupt.</p> <p>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.</p>
4	SSR	0	R/W*	<p>Slave Stop Received</p> <p>A stop condition has been output on the bus. This status bit becomes active after the rising edge of SDA during the stop bit.</p>
3	SDE	0	R/W*	<p>Slave Data Empty</p> <p>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.</p>
2	SDT	0	R/W*	<p>Slave Data Transmitted</p> <p>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.</p>

Bit	Bit Name	Initial Value	R/W	Description
1	SDR	0	R/W*	<p>Slave Data Received</p> <p>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.</p> <p>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.</p>
0	SAR	0	R/W*	<p>Slave Address Received</p> <p>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.</p>

Note: \* This bit can be read from or written to. Writing 0 clears this bit to 0 and writing 1 is ignored.

### 16.3.3 Slave Interrupt Enable Register (ICSIER)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	SSRE	SDEE	SDTE	SDRE	SARE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved 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.

### 16.3.4 Slave Address Register (ICSAR)

Bit:	7	6	5	4	3	2	1	0
	—	SADD0[6:0]						
Initial value:	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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved The write value should always be 0.
6 to 0	SADD0[6:0]	All 0	R/W	Slave Address This is the unique 7-bit address allocated to the slave on the I <sup>2</sup> C 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.

### 16.3.5 Master Control Register (ICMCR)

Bit:	7	6	5	4	3	2	1	0
	MDBS	FSCL	FSDA	OBPC	MIE	TSBE	FSB	ESG
Initial value:	0	—	—	0	0	0	0	0
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
7	MDBS	0	R/W	<p><b>Master Data Buffer Select</b></p> <p>This bit is used to select the data buffer. The double-buffer mode and single-buffer mode are available.</p> <p>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.</p> <p>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.</p> <p>0: Double-buffer mode 1: Single-buffer mode</p>
6	FSCL	Undefined	R/W	<p><b>Forced SCL</b></p> <p>This bit controls the status of the SCL pin (reading reflects the current level on the I<sup>2</sup>C bus). When the OBPC bit is set, this bit directly controls the SCL line on the bus.</p> <p>During a read cycle, the level on this bit (which includes the reset level) will change depending on the level on SCL since it reflects the level on the SCL.</p>
5	FSDA	Undefined	R/W	<p><b>Forced SDA</b></p> <p>This bit controls the status of the SDA pin (reading reflects the busy status level on the SDA). When the OBPC bit is set then this bit directly controls the SDA line on the bus.</p> <p>During a read cycle, the level of this bit (which includes the reset level) will show the busy status of the I<sup>2</sup>C bus (1 for busy; 0 for not busy).</p>



Bit	Bit Name	Initial Value	R/W	Description
4	OBPC	0	R/W	<p>Override Bus Pin Control</p> <p>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.</p>
3	MIE	0	R/W	<p>Master Interface Enable</p> <p>When this bit is set to 1, the master interface is enabled.</p>
2	TSBE	0	R/W	<p>Start Byte Transmission Enable</p> <p>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 I<sup>2</sup>C bus interfaces.</p>
1	FSB	0	R/W	<p>Forced Stop onto the Bus</p> <p>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.</p>
0	ESG	0	R/W	<p>Enable Start Generation</p> <p>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.</p>

### 16.3.6 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:	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
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
7	—	0	R	Reserved 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*	<p>Master Data Empty</p> <p>At the start of a byte data 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.</p> <p>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.</p>
2	MDT	0	R/W*	<p>Master Data Transmitted</p> <p>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.</p>
1	MDR	0	R/W*	<p>Master Data Received</p> <p>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.</p> <p>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.</p> <p>During master reception mode, the MDR bit is set at the same timing as the MAT bit set after transmission of the slave 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</p>
0	MAT	0	R/W*	<p>Master Address Transmitted</p> <p>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.</p>

Note: \* This bit can be read from or written to. Writing 0 clears this bit to 0 and writing 1 is ignored.

### 16.3.7 Master Interrupt Enable Register (ICMIER)

Bit:	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
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
7	—	0	R	Reserved 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.

### 16.3.8 Master Address Register (ICMAR)

Bit:	7	6	5	4	3	2	1	0
	SADD1[6:0]							STM1
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SADD1[6:0]	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.

### 16.3.9 Clock Control Register (ICCCR)

Bit:	7	6	5	4	3	2	1	0
	SCGD[5:0]						CDF[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	SCGD	All 0	R/W	<p>SCL Clock Generation Divider</p> <p>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:</p> <p>Equation 2 SCL rate calculation</p> $\text{SCLfreq} = \text{IICck} / (20 + (\text{SCGD} * 8))$ <p>IICck: I<sup>2</sup>C internal clock frequency</p> <p>Suggested settings for CDF and SCGD for various CPU speeds and the two I<sup>2</sup>C bus speeds are given in table 16.4.</p>
1, 0	CDF	All 0	R/W	<p>Clock Division Factor</p> <p>The internal clock used in most blocks in the I<sup>2</sup>C module is a divided peripheral clock. The internal I<sup>2</sup>C clock is generated from the peripheral clock using the CDF as the division ratio:</p> <p>Equation 1 I<sup>2</sup>C internal clock frequency calculation</p> $\text{IICck} = \text{Pck} / (1 + \text{CDF})$ <p>Pck: Peripheral clock</p> <p>The minimum time to ensure adequate setup and hold times on the SDA line relative to the SCL line on the bus.</p> <p>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 I<sup>2</sup>C specification.</p>

Note: CDF must be set so that the clock frequency (IICck) is lower than 20 MHz.

**Table 16.4 Suggested Settings for CDF and SCGD\***

Peripheral Clock Frequency	100 kHz		400 kHz	
	CDF	SCGD	CDF	SCGD
50 MHz	2	19	2	3
Error	- 3.10 %		- 5.30 %	

Note: \* These are suggested values for the SCL rate.

### 16.3.10 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 I<sup>2</sup>C bus, it is then loaded into RXD.

- Receive Data Register (ICRXD)

Bit:	7	6	5	4	3	2	1	0
	RXD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RXD[7:0]	All 0	R	Read—Receive Data Data received by master or slave.

- Transmit Data Register (ICTXD)

Bit:	7	6	5	4	3	2	1	0
	TXD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	TXD[7:0]	All 0	W	Write—Transmit Data Data transmitted by master or slave.

## 16.4 Operations

### 16.4.1 Data and Clock Filters

These blocks filter out glitches on signals coming from the I<sup>2</sup>C bus. Glitches up to one internal clock period in width are rejected (For details on the internal clock frequency see section 16.3.9, Clock Control Register (ICCCR)). This is for the faster I<sup>2</sup>C bit rate (400 KHz) but does not violate the slower I<sup>2</sup>C bus rate specification.

These blocks also resynchronizes bus signals with the internal clock.

### 16.4.2 Clock Generator

The clock generator has two functions. Firstly, it generates the SCL I<sup>2</sup>C 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.

### 16.4.3 Master/Slave Interfaces

These two interfaces run independently and in parallel. The master interface controls the transmission of address and data on the I<sup>2</sup>C bus. The slave interface monitors the I<sup>2</sup>C 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 I<sup>2</sup>C module. The interrupt source is either the master or the slave.

### 16.4.4 Software Status Interlocking

In order that the software interface to the I<sup>2</sup>C 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 I<sup>2</sup>C 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 I<sup>2</sup>C 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 I<sup>2</sup>C 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**

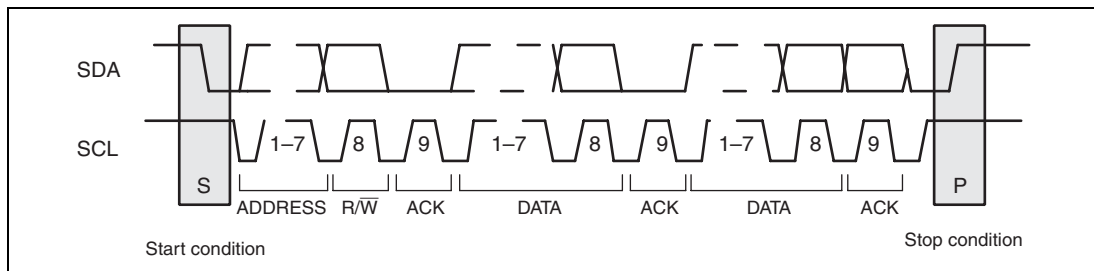
The SAR status bit is set when the slave identifies its address on the I<sup>2</sup>C bus. At this point the slave interface forces the SCL line low until the SAR status bit is reset.

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.

### 16.4.5 I<sup>2</sup>C Bus Data Format

Figure 16.2 shows a timing chart for the I<sup>2</sup>C bus interface. Table 16.5 describes the meaning of each symbol in figure 16.2.



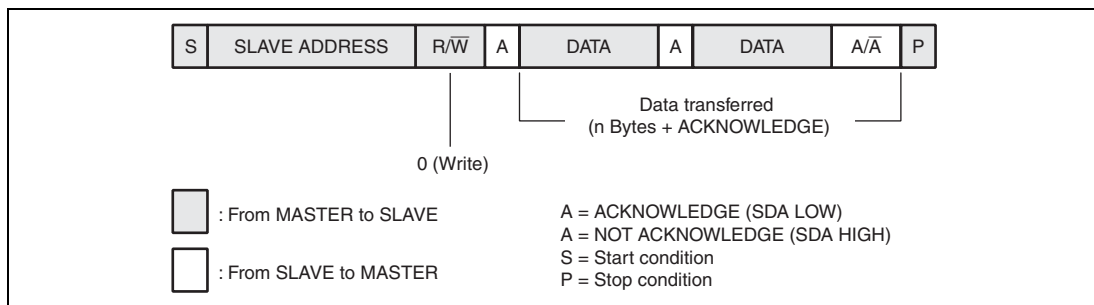
**Figure 16.2 I<sup>2</sup>C Bus Timing**

**Table 16.5 Description on Symbols of I<sup>2</sup>C Bus Data Format**

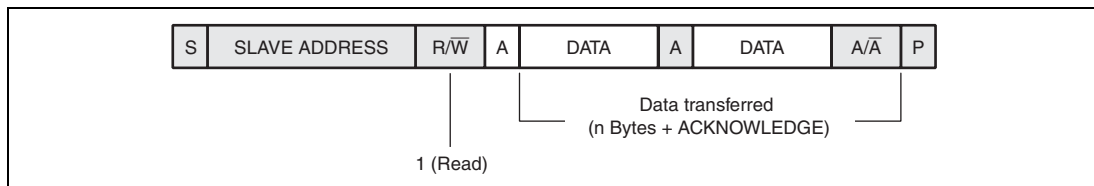
Symbol	Description
S	Indicates a start condition. A master device changes SDA from high to low while SCL is high level.
SLA	Indicates a slave address. A slave address is used when a master device selects a slave device.
R/W	Indicates the direction of data transmission. If the R/W bit is 1, the data flows 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.

### 16.4.6 7-Bit Address Format

Figure 16.3 shows the format of data transfer from a master to a slave device (master data transmit format). Figure 16.4 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.

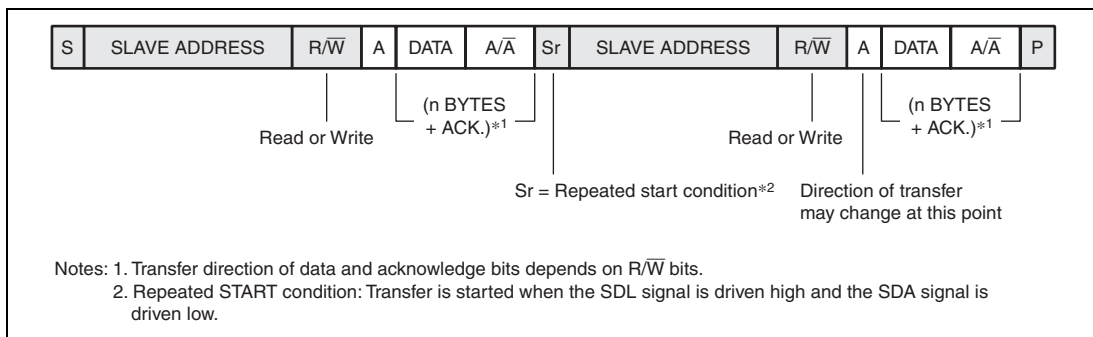


**Figure 16.3 Master Data Transmit Format**



**Figure 16.4 Master Data Receive Format**

Figure 16.5 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.



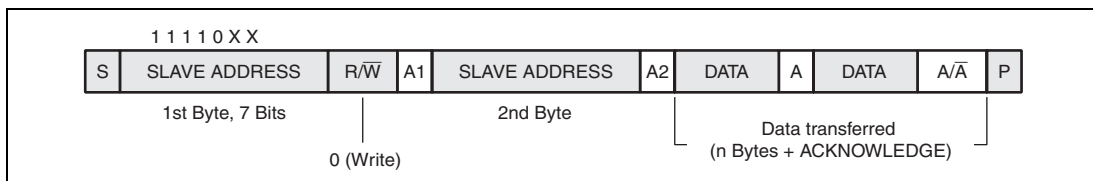
**Figure 16.5 Combination Transfer Format of Master Transfer**

### 16.4.7 10-Bit Address Format

Description is given below on the 10-bit address transfer format supported in master mode.

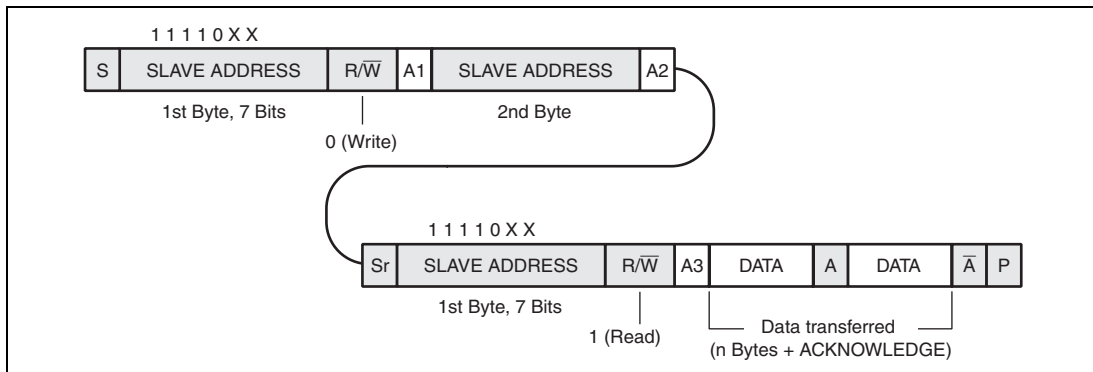
This format has three transfer methods as the 7-bit address transfer format.

Figure 16.6 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 16.6 10-Bit Address Data Transmit Format**

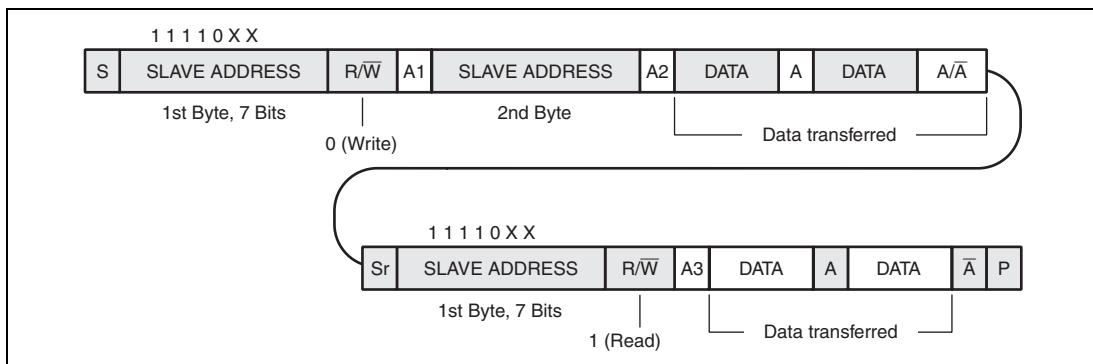
Figure 16.7 shows the data receive format. Two bytes of an address is transmitted a repeated START in the same way as in the data transmit format. Then, repeated START condition (Sr) is transmitted and the value set in the address register is output. At this time, STM1 must be set to 1 (receive mode). Data is transferred in the same way as in the 7-bit address data receive format.



**Figure 16.7 10-Bit Address Data Receive Format**

Figure 16.8 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 16.8 10-Bit Address Transmit/Receive Combined Format**

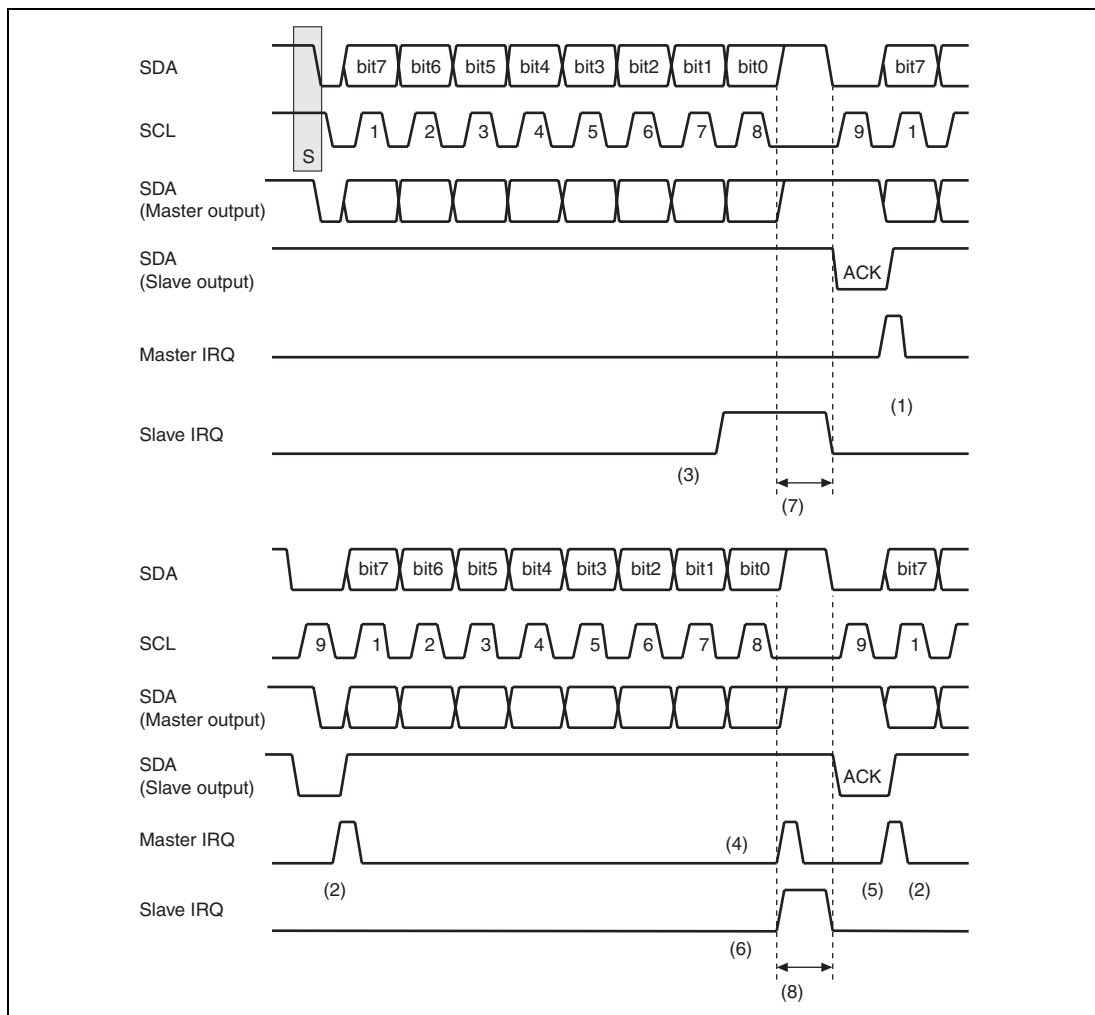
### 16.4.8 Master Transmit Operation

The transmit procedure and operation in master transmit mode are described below. Figure 16.9 shows the timing chart in master transmit mode. Setting the MDBS bit in the master control register allows the IIC 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 I<sup>2</sup>C 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 16.9. 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 16.9. 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 16.9). 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 16.9). An interrupt of MDT (bit 2) is generated at the eighth clock after 1-byte data transfer (at the timing of (4) in figure 16.9). 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 16.9). 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 16.9).
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 16.9) 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 IIC 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 16.9 are generated after the falling edge of the clock.



**Figure 16.9 Data Transmit Mode Operation Timing**

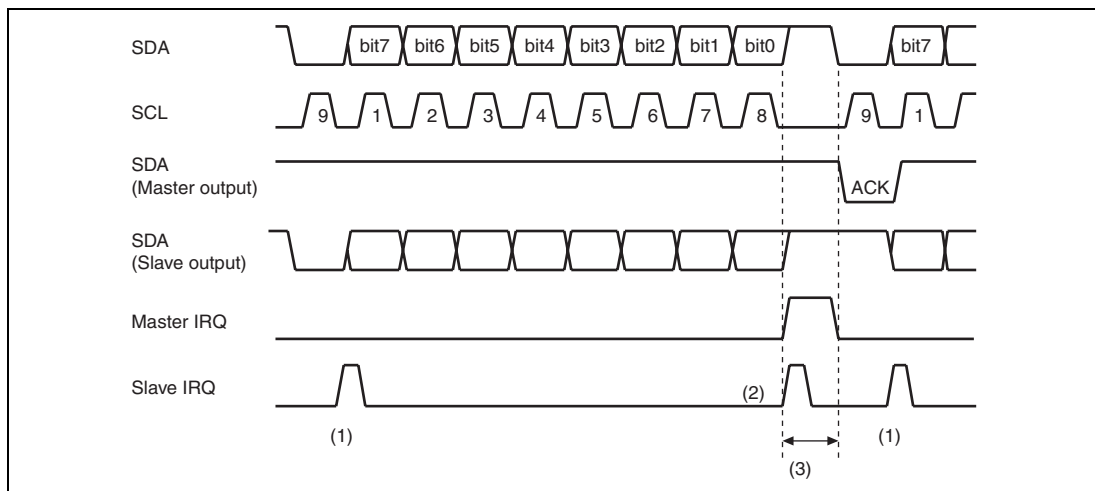
### 16.4.9 Master Receive Operation

The data receive procedure and operation in master receive mode are described below. Figure 16.10 shows the timing chart in master receive mode. Setting the MDBS bit in the master control register allows the IIC 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 11). 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, as shown at the timing of (3) in figure 16.10.
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 16.10) and an interrupt of the status SDE (bit 3) indicating data empty at the ninth clock (at the timing of (1) in figure 16.10). 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 IIC 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 16.10 are generated after the falling edge of the clock.



**Figure 16.10 Data Receive Mode Operation Timing**

## 16.5 Programming Examples

### 16.5.1 Master Transmitter

In order to set up the master interface to transmit a data packet on the I<sup>2</sup>C bus, follow the following procedure:

#### (1) Load Clock Control Register

1. SCL clock generation divider (SCGD) = H'03  
(SCL frequency of 400 kHz)
2. Clock division ratio (CDF) = H'2  
(The peripheral clock is 50 MHz and the IIC'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 = H'89  
(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 H'88 (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 H'8A, 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 H'8A.  
(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).

## 16.5.2 Master Receiver

To set up the master interface to receive a data packet on the I<sup>2</sup>C bus, follow the following procedure:

### (1) Load Clock Control Register

1. SCL clock generation divider (SCGD) = H'03  
(SCL frequency of 400 kHz).
2. Clock division ratio (CDF) = H'2  
(The peripheral clock is 50 MHz and the IIC'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 H'89  
(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 H'88  
(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 H'8A, 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, bit MDR 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 H'8A  
(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.

## **16.5.3 Master Transmitter - Restart - Master Receiver**

In order to set up the master interface to transmit a data packet on the I<sup>2</sup>C bus, issue a restart, then read byte data back from the slave, follow the following procedure:

### **(1) Load Clock Control Register**

1. Set the SCL clock generation divider (SCGD) to H'03  
(SCL frequency of 400 kHz).
2. Set the clock division (CDF) to H'2  
(The peripheral clock is 50 MHz and the IIC'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 H'89  
(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 H'88 (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 H'8A  
(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 MST in the master status register.
3. Reset the MST bit.



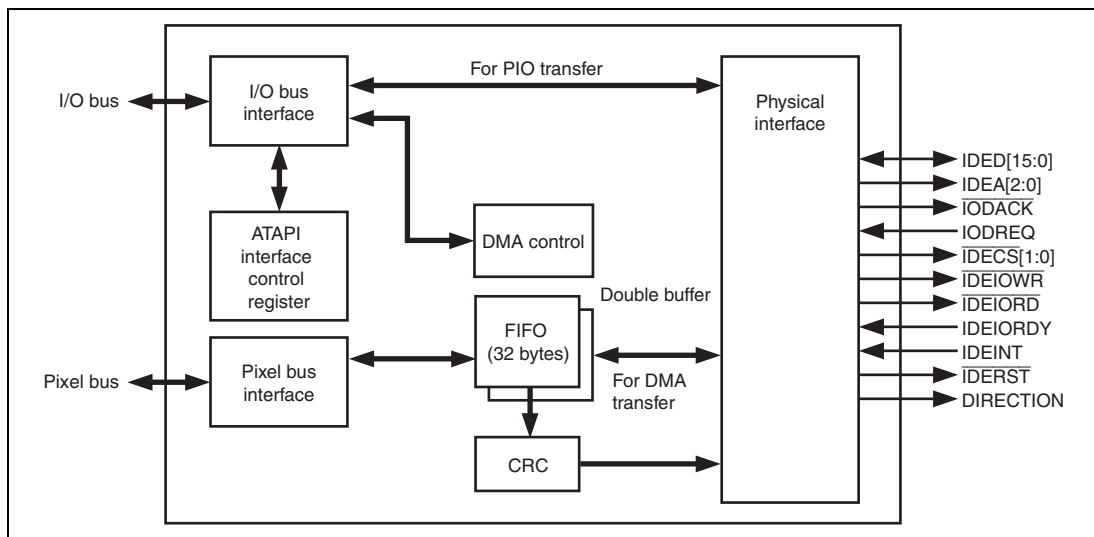
## Section 17 ATAPI

The ATAPI interface provides both the ATA and ATAPI physical interfaces. This device also supports both the ATA task and ATAPI packet commands.

### 17.1 Features

- Supporting primary channel
- Supporting master/slave
- Supporting 3.3V I/O interface
- Supporting PIO modes 0 to 4, the multiword DMA modes 0 to 2, and the Ultra DMA modes 0 to 4
- Supporting descriptor mode

Figure 17.1 shows a block diagram of the ATAPI.



**Figure 17.1 ATAPI Block Diagram**

## 17.2 Input/Output Pins

Table 17.1 summarizes the ATAPI's pin configuration.

The ATAPI has two pin groups: normal pin group and mirror pin group. The input/output operations of pins in both groups are always the same. The pin select register of the PFC is used to select the ATAPI pins. As pins in two groups have different input/output timing, mixed use of pins in two groups is not allowed.

**Table 17.1 Pin Configuration**

Pin Name		(ATAPI Specification)	Function	I/O
Normal I/O	Mirror I/O			
IDED15 to IDED0	IDED15_M to IDED0_M	(DD(15:0))	Bidirectional data bus	I/O
IDEA2 to IDEA0	IDEA2_M to IDEA0_M	(DA(2:0))	Address bus	Output
IODACK	IODACK_M	(DMACK)	Primary channel DMA acknowledge (active low)	Output
IODREQ	IODREQ_M	(DMARQ)	Primary channel DMA request (active high)	Input
IDECS1, IDECS0	IDECS1_M, IDECS0_M	(CS0, CS1)	Primary channel chip select (active low)	Output
IDEIOWR	IDEIOWR_M	(DIOW, STOP)	Primary channel disk write (active low)	Output
IDEIORD	IDEIORD_M	(DIOR, HDMARDY, HSTROBE)	Primary channel disk read (active low)	Output
IDEIORDY	IDEIORDY_M	(IORDY, DDMARDY, DSTROBE)	Primary channel ready signal (active high)	Input
IDEINT	IDEINT_M	(INTRQ)	Primary channel interrupt request* (active high)	Input
IDERST	IDERST_M	(RESET)	Primary channel ATAPI device reset (active low)	Output
DIRECTION	DIRECTION_M	—	External level shifter direction signal (0 when writing to the device)	Output

Note: \* The ATAPI interface treats the interrupt signal from the ATAPI device as a level-triggered input.



## 17.3 Register Description

The following ATAPI register set is allocated to the SH register map space.

**Table 17.2 ATA Task File Register Map**

(These registers are allocated to the ATAPI or ATA device, and are not allocated to this module.)

Address	Read Register	Write Register	Pin Address (IDECS[1:0], IDEA[2:0])	Access Size* <sup>1</sup> (Available Bit Size)	Register Location
			H: High Level L: Low Level @3.3V I/O		
H'FFF0 0000	Data	Data	HL-LLL/HH-XXX (X: Don't care)	32 (16)* <sup>2</sup>	Drive
H'FFF0 0004	Error	Function	HL-LLH	32 (8)* <sup>3</sup>	Drive
H'FFF0 0008	Sector count	Sector count	HL-LHL	32 (8)* <sup>3</sup>	Drive
H'FFF0 000C	Sector number	Sector number	HL-LHH	32 (8)* <sup>3</sup>	Drive
H'FFF0 0010	Cylinder low	Cylinder low	HL-HLL	32 (8)* <sup>3</sup>	Drive
H'FFF0 0014	Cylinder high	Cylinder high	HL-HLH	32 (8)* <sup>3</sup>	Drive
H'FFF0 0018	Device/head	Device/head	HL-HHL	32 (8)* <sup>3</sup>	Drive
H'FFF0 001C	Status	Command	HL-HHH	32 (8)* <sup>3</sup>	Drive
H'FFF0 0038	Alternate status	Device control	LH-HHL	32 (8)* <sup>3</sup>	Drive

Notes: 1. These registers must be accessed in longwords (32 bits) by the CPU. Byte or word accesses are prohibited.

2. Bits 15 to 0 of the data bus are used.
3. Bits 7 to 0 of the data bus are used.

**Table 17.3 ATAPI Packet Command Task File Register Map**

(These registers are allocated to the ATAPI or ATA device, and are not allocated to this module.)

Address	Read Register	Write Register	Pin Address (IDECS[1:0], IDEA[2:0])	Access Size* <sup>1</sup> (Available Bit Size)	Register Location
H'FFF0 0000	Data	Data	HL-LLL	32 (16)* <sup>2</sup>	Drive
H'FFF0 0004	Error	Function	HL-LLH	32 (8)* <sup>3</sup>	Drive
H'FFF0 0008	Interrupt source	—	HL-LHL	32 (8)* <sup>3</sup>	Drive
H'FFF0 000C	—	—	HL-LHH	32 (8)* <sup>3</sup>	Drive
H'FFF0 0010	Byte Count Low	Byte Count Low	HL-HLL	32 (8)* <sup>3</sup>	Drive
H'FFF0 0014	Byte Count High	Byte Count High	HL-HLH	32 (8)* <sup>3</sup>	Drive
H'FFF0 0018	Device select	Device select	HL-HHL	32 (8)* <sup>3</sup>	Drive
H'FFF0 001C	Status	Command	HL-HHH	32 (8)* <sup>3</sup>	Drive
H'FFF0 0038	Alternate Status	Device Control	LH-HHL	32 (8)* <sup>3</sup>	Drive

Notes: 1. These registers must be accessed in longwords (32 bits) by the CPU. Byte or word accesses are prohibited.

2. Bits 15 to 0 of the data bus are used.

3. Bits 7 to 0 of the data bus are used.

**Table 17.4 ATAPI Interface Control Register Map**

(These registers are allocated to this module.)

Address	Register Name	Abbreviation	Access Type	Register Access Size*
H'FFF0 0080	ATAPI control	ATAPI_CONTROL	R/W	32
H'FFF0 0084	ATAPI status	ATAPI_STATUS	R/W	32
H'FFF0 0088	Interrupt enable	ATAPI_INT_ENABLE	R/W	32
H'FFF0 008C	PIO timing	ATAPI_PIO_TIMING	R/W	32
H'FFF0 0090	Multiword DMA timing	ATAPI_MULTI_TIMING	R/W	32
H'FFF0 0094	Ultra DMA timing	ATAPI_ULTRA_TIMING	R/W	32
H'FFF0 0098	Descriptor table base address	ATAPI_DTB_ADR	R/W	32
H'FFF0 009C	DMA start address	ATAPI_DMA_START_ADR	R/W	32
H'FFF0 00A0	DMA transfer count	ATAPI_DMA_TRANS_CNT	R/W	32
H'FFF0 00A4	ATAPI control 2	ATAPI_CONTROL2	R/W	32
H'FFF0 00A8	Reserved		R	32
H'FFF0 00AC	Reserved		R	32
H'FFF0 00B0	ATAPI signal status	ATAPI_SIG_ST	R	32
H'FFF0 00BC	Byte swap	ATAPI_BYTE_SWAP	R/W	32

Note: \* These registers must be accessed in longwords (32 bits) by the CPU. Byte or word accesses are prohibited.

**[Legend]:**

Initial value: Register value after reset

—: Undefined value

R/W: Readable and writable bit. The write value can be read.

R/WC0: Readable and writable bit. When 0 is written, the bit is initialized. When 1 is written, it is ignored.

R: Read only register, only 0 should be written.

—/W: Write only bit. The read value is undefined.

All control/status registers are active high.

### 17.3.1 ATAPI Control (ATAPI\_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
	—	—	—	—	—	—	DTCD	—	RESET	M/S	BUSSEL	UDMAEN	DESE	R/W	STOP	START
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved
9	DTCD	0	R/W	<p>DTCD controls the operating mode for device termination occurring when the Ultra DMA operates. No abnormal termination occurs if the specified number of transfers has not been reached after the reception of device termination. Transfer will restart after waiting a DMARQ from the next device.</p> <p>Some of the existing ATA devices are those handling the pause the same way as device termination. If, therefore, the specified number of transfers has not been reached after the reception of device termination, no abnormal termination occurs and it is necessary to restart transfer after waiting a DMARQ from the next device. This operating mode is called the "device termination continuation mode."</p> <p>1: Suppressing the device termination continuation mode 0: Device termination continuation mode</p>
8	—	0	R	Reserved
7	RESET	0	R/W	<p>RESET is an ATAPI device reset. If this bit is set to 1 then the ATAPI reset signal is asserted. <math>\overline{IDERST}</math> is active low signal. When this bit is set to 1 then <math>\overline{IDERST}</math> is low. When this bit is set to '0' then <math>\overline{IDERST}</math> is high.</p>
6	M/S	0	R/W	<p>M/S selects an ATAPI device MASTER or SLAVE. 1 = MASTER, 0 = SLAVE.</p>
5	BUSSEL	1	R/W	<p>BUSSEL selects a pixel bus or I/O bus. 1 = Pixel bus, 0 = I/O bus.</p> <p>Note: Do not clear this bit to 0 and set to 1.</p>

Bit	Bit Name	Initial Value	R/W	Description
4	UDMAEN	0	R/W	UDMAEN is Ultra DMA enable. When Ultra DMA is used, set this bit to 1. Set it to '0' when using the multiword DMA or PIO mode.
3	DESE	0	R/W	DESE controls the descriptor table operation mode. 1. Validating the descriptor function. 0. Invalidating the descriptor function.
2	R/W	0	R/W	R/W is FIFO read/write. 1 = FIFO read (data-in operation at DMA transfer), 0 = FIFO write (data-out operation at DMA transfer).  Set this bit to 1 when reading data from the ATAPI device.  Set it to '0' when writing data to the ATAPI device.
1	STOP	0	R/W	STOP terminates a DMA transfer.  When writing 0: Ignored 1: Stop a data transfer  When reading 0: The stop command is not issued. 1: The data transfer stop command is issued.  It will become 0 when the next DMA starts.  Note: Transfer cannot be restarted from the address at which DMA transfer has been stopped.
0	START	0	R/W	START initiates a DMA transfer. If this bit set to 1 then the DMA transfer is started. 0 writing is ignored.  When writing 0: Ignored 1: DMA transfer start  When reading 0: DMA transfer is not active 1: Busy in DMA transfer  Note: Must not access the Task File Register while DMA is active.

### 17.3.2 ATAPI Status (ATAPI\_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
	—	—	—	—	—	—	—	SWERR	IFERR	DNEND	DEVTRM	DEVINT	TOUT	ERR	NEND	ACT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC0	R/WC0	R/WC0	R/WC0	R	R/WC0	R/WC0	R/WC0	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved
8	SWERR	0	R/WC0	<p>SWERR is a software error. It indicates that Task File register access is detected while DMA is active. It is prohibited. For example, this bit is set to 1 if PIO transfer is performed during the transfer of the Ultra DMA or multiword DMA. In this case, no output is sent to the outside of the LSI so that access is ignored</p> <p>Writing 0 resets this register.</p>
7	IFERR	0	R/WC0	<p>IFERR indicates that an ATAPI interface protocol error is detected. In other words,</p> <ol style="list-style-type: none"> <li>1. (IODREQ = 1) or (IDEIORDY = 0) when the ULTRA DMA data-in burst is in the host termination.</li> <li>2. IDEIORDY = 0 when the ULTRA DMA data-out burst is in the device termination.</li> <li>3. IDEIORDY = 0 when the ULTRA DMA data-out burst is initiated.</li> <li>4. (IODREQ = 1) or (IDEIORDY = 0) when the ULTRA DMA data-out burst is in the host termination.</li> </ol> <p>Writing 0 resets this register.</p>
6	DNEND	0	R/WC0	<p>DNEND indicates that all DMAs have been successfully terminated in the descriptor mode. Writing 0 resets this register.</p>
5	DEVTRM	0	R/WC0	<p>DEVTRM is set to 1 when the ATAPI device is terminated in the Ultra DMA mode before the number of DMA transfer bytes reach the value set in this ATAPI module. Writing 0 resets this register.</p>

Bit	Bit Name	Initial Value	R/W	Description
4	DEVINT	0	R	DEVINT is ATAPI device interrupt IDEINT status. This bit is read only. Since this bit does not hold its status in this module, if IDEINT becomes 0, this bit will also become 0. ATAPI interface treats the interrupt signal from the ATAPI device as a level-triggered input. According to the ATAPI standard, IDEINT will be negated by the ATAPI device within 400 ns of the negation of IDEIORD that reads the Status register to clear a pending interrupt.
3	TOUT	0	R/WC0	TOUT indicates that an IORDY timeout is detected. Timeout is detected if no response is returned (the IDEIORDY pin is at the Low level.) in 150 cycles or longer of Pixel Bus clock cycle time. Writing 0 resets this register.
2	ERR	0	R/WC0	ERR is set to 1 when a DMA abort is detected. ERR=1 occurs if: 1. The host brings DMA transfer to a forced stop. 2. DTCD=1 and ACT=0 because of device termination Writing 0 resets this register.
1	NEND	0	R/WC0	NEND is a DMA normal end. Writing 0 resets this register.
0	ACT	0	R	ACT indicates that the DMA is active. This bit is read only. This register is cleared when the DMA transfer is completed.  This bit should not be used as an interrupt source.

### 17.3.3 Interrupt Enable (ATAPI\_INT\_ENABLE)

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
	—	—	—	—	—	—	—	iSWERR	iIFERR	iDNEND	iDEVTRM	iDEVINT	iTOUT	iERR	iNEND	iACT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved
8	iSWERR	0	R/W	iSWERR is SWERR interrupt enable.
7	iIFERR	0	R/W	iIFERR is IFERR interrupt enable.
6	iDNEND	0	R/W	iDNEND is DNEND interrupt enable.
5	iDEVTRM	0	R/W	iDEVTRM is DEVTRM interrupt enable
4	iDEVINT	0	R/W	iDEVINT is DEVINT interrupt enable.
3	iTOUT	0	R/W	iTOUT is TOUT interrupt enable.
2	iERR	0	R/W	iERR is ERR interrupt enable.
1	iNEND	0	R/W	iNEND is NEND interrupt enable.
0	iACT	0	R/W	iACT is ACT interrupt enable. Since ACT is cleared automatically when a DMA transfer is completed, this bit should not be set to 1.

Note: Writing 1 to each bit enables the interrupt signal of the ATAPI status register bit.



### 17.3.4 PIO Timing Register (ATAPI\_PIO\_TIMING)

Set the machine cycle numbers to the following bits before the access to the ATAPI device.

The machine cycle is a pixel bus clock.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	pSDCT						pSDPW						pSDST	
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/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	pMDCT						pMDPW						pMDST	
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/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved
29 to 24	pSDCT	0	R/W	pSDCT sets the cycle time of the Slave ATAPI device.
23 to 19	pSDPW	0	R/W	pSDPW sets the $\overline{\text{IDEIORD}}/\text{IDEIOWR}$ pulse width of the Slave ATAPI device.
18 to 16	pSDST	0	R/W	pSDST sets the address setup time to $\overline{\text{IDEIORD}}/\text{IDEIOWR}$ for the slave ATAPI device in the PIO mode.
15, 14	—	All 0	R	Reserved
13 to 8	pMDCT	0	R/W	pMDCT sets the cycle time of the Master ATAPI device.
7 to 3	pMDPW	0	R/W	pMDPW sets the $\overline{\text{IDEIORD}}/\text{IDEIOWR}$ pulse width of the Master ATAPI device.
2 to 0	pMDST	0	R/W	pMDST sets the address setup time to $\overline{\text{IDEIORD}}/\text{IDEIOWR}$ for the master ATAPI device in the PIO mode.

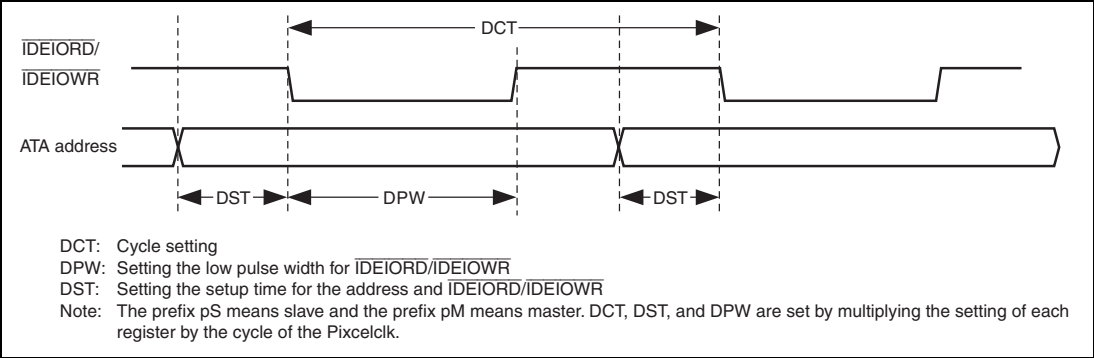


Figure 17.2 PIO Timing Register

PIO timing register value table (Master / Slave)

Pixel Bus Clock	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4
100 MHz	H'3DF0	H'28F6	H'22F4	H'134C	H'0D44

17.3.5 Multiword DMA Timing Register (ATAPI\_MULTI\_TIMING)

Set the machine cycle numbers to the following bits in this register before access to the ATAPI device.

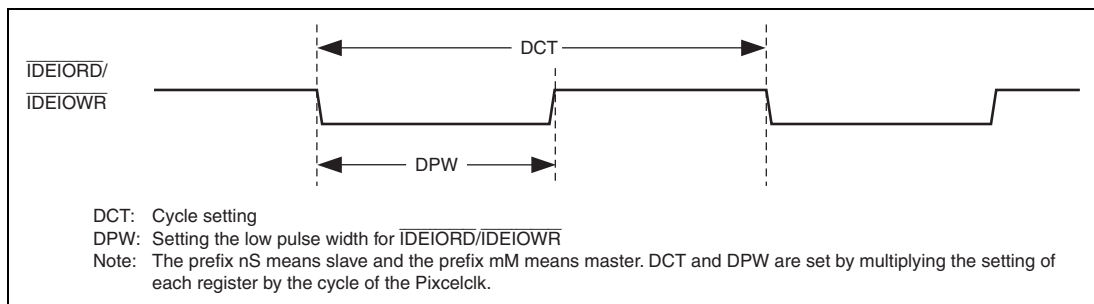
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	mSDCT						mSDPW				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	mMDCT						mMDPW				
Initial value:	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
31 to 27	—	All 0	R	Reserved
26 to 21	mSDCT	0	R/W	mSDCT sets the cycle time of the Slave ATAPI device.
20 to 16	mSDPW	0	R/W	mSDPW sets the IDEIORD/IDEIOWR pulse width of the Slave ATAPI device.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved
10 to 5	mMDCT	0	R/W	mMDCT sets the cycle time of the Master ATAPI device.
4 to 0	mMDPW	0	R/W	mMDPW sets the $\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ pulse width of the Master ATAPI device.



**Figure 17.3 Multiword DMA Timing Register**

Multiword DMA timing register value table

Pixel Bus Clock	Mode 0	Mode 1	Mode 2
100 MHz	H'0637	H'0209	H'01A8

### 17.3.6 Ultra DMA Timing Register (ATAPI\_ULTRA\_TIMING)

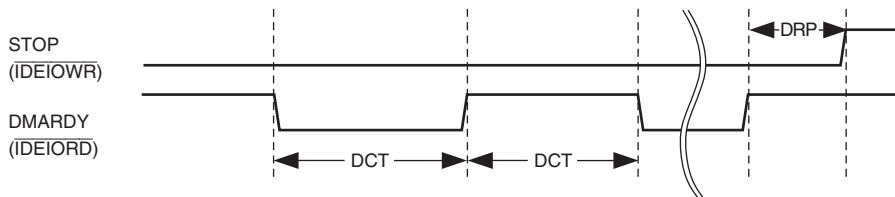
Set the machine cycle numbers to the following bits in this register before the access to the ATAPI device.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved
24 to 21	uSDCT	—	R/W	uSDCT sets the cycle time of the Slave ATAPI device.
20 to 16	uSDRP	0	R/W	uSDRP sets the time from negating DMARDY (Not IDEIORDY) until paused by the slave ATAPI device.
15 to 9	—	All 0	R	Reserved
8 to 5	uMDCT	0	R/W	uMDCT sets the cycle time of the Master ATAPI device.
4 to 0	uMDRP	0	R/W	uMDRP sets the time from negating DMARDY (Not IDEIORDY) until paused by the master ATAPI device.



DCT: Cycle setting

DRP: Setting the period from negating the DMARDY (IDEIORD) signal; used for data-in burst

Note: The prefix uS means slave and the prefix uM means master. DCT and DRP are set by multiplying the setting of each register by the cycle of the Pixelclk.

Figure 17.4 Ultra DMA Timing Register

## Ultra DMA timing register value table

Pixel bus clock	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4
100MHz	H'0191	H'010E	H'00CB	H'00AB	H'006B

## 17.3.7 Descriptor Table Base Address Register (ATAPI\_DTB\_ADR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DTBAA[2:0]			DTBA[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/W	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
	DTBA[15:2]														—	—
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	Description
31 to 29	—	All 0	R	Reserved
28 to 26	DTBAA [2:0]	0	R/W	DTBA shows the descriptor table base SDRAM area 001: SDRAM area 1 010: SDRAM area 2 Other than above: Setting prohibited.
25 to 2	DTBA [25:2]	0	R/W	DTBA shows the descriptor table base address. Bits 25 to 0 are used to set the descriptor table base address on a byte basis. Bits 3 and 2 must be set 0, and bits 1 and 0 are ignored because it is necessary to secure the boundary of 64-bit addresses in the descriptor table.
1, 0	—	All 0	R	Reserved

- Notes:
1. This register is valid only when bit 5 (BUSSEL) in the ATAPI Control Register is set to 1.
  2. This address will not change and will retain its setting even after the DMA becomes active.
  3. In the 32-bit address mode, bits 28 to 0 should contain the lower 29 bits of the specified 32-bit address.

### 17.3.8 Descriptor Table

The descriptor table consists of the termination flag, the descriptor DMA start address (DDSTA), and the descriptor DMA transfer count (DDTRC).

#### Descriptor Table Map in Memory

Address	Data description
DTBA	The first termination flag (bit 31=0) and DDSTAA/DDSTA
DTBA + 4	The first DDTRC
DTBA + 8	The second termination flag (bit 31=0) and DDSTAA/DDSTA
DTBA + 12	The second DDTRC
$DTBA + 8 \times (n - 1)$	The n-th termination flag (bit 31=1) and DDSTAA/DDSTA
$DTBA + 8 \times (n - 1) + 4$	The n-th DDTRC

### 17.3.9 Termination Flag and Descriptor DMA Start Address

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTEND	—	—	DDSTAA[2:0]			DDSTA[25:16]									
Initial value:	—	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31	DTEND	Undefined	R/W	DTEND controls the termination of a descriptor DMA operation.  1: Terminating the descriptor DMA operation (When DTEND is 1, the system recognizes that the descriptor table is the last one.)  0: Validating the descriptor table (When DTEND is 0, the system reads the DMA transfer count, transfers the DMA, and reads the next descriptor table.)
30 to 29	—	All 0	R	Reserved
28 to 26	DDSTAA [2:0]	Undefined	R/W	DDSTAA shows the DMA start SDRAM area in descriptor operation mode.  001: SDRAM area 1 010: SDRAM area 2  Other than above: Setting prohibited.
25 to 2	DDSTA [25:2]	Undefined	R/W	DDSTA shows the DMA start address in descriptor operation. Bits 25 to 0 are used to set the descriptor table start address on a byte basis.  Bits 4 to 2 must be set 0, and bits 1 and 0 are ignored because it is necessary to secure the boundary of 256-bit addresses in the DMA start address.
1, 0	—	All 0	R	Reserved

The valid flag and descriptor DMA start address should be set in the descriptor table base address + "m" in the memory, where the value of m is any multiple of 2 (such as 0, 2, 4, ...).

17.3.10 Descriptor DMA Transfer Count

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DDTRC[28:16]												
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
	DDTRC[15:1]															—
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

The descriptor DMA transfer count should be set in the descriptor table base address + "m" in the memory, where the value of m is any multiple number of 2 plus 1 (such as 1, 3, 5, ...).

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved
28 to 1	DDTRC[28:1]	Undefined	R/W	These bits set the DMA transfer count during descriptor operation.  Bits 28 to 0 are used to set the DMA transfer count on a byte basis.  Bit 0 is ignored because the ATAPI data bus is handled on a 16-bit basis (on a word basis).
0	—	0	R	Reserved



### 17.3.11 DMA Start Address Register (ATAPI\_DMA\_START\_ADR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DSTAA[2:0]			DSTA[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/W	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
	DSTA[15:2]														—	—
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	Description
31 to 29	—	All 0	R	Reserved
28 to 26	DSTAA[2:0]	0	R/W	DSTAA sets DMA start SDRAM area in descriptor operation mode. 001: SDRAM area 1 010: SDRAM area 2 Other than above: Setting prohibited.
25 to 2	DSTA[25:2]	0	R/W	DSTA sets a DMA start address that indicates the data transfer start address in the memory. Bits 25 to 0 are used to specify the DMA start address in byte. Since 256-bit address boundary must be kept for DMA start address, bits 4 to 2 must be set 0, and bits 1 and 0 are ignored.
1, 0	—	All 0	R	Reserved

- Notes:
1. This register is valid only when bit 5 (BUSSEL) in the ATAPI Control Register is 1.
  2. This address does not change and the set value is retained even after DMA activation.
  3. In the 32-bit address mode, bits 28 to 0 should contain the lower 29 bits of the specified 32-bit address.

### 17.3.12 DMA Transfer Count Register (ATAPI\_DMA\_TRANS\_CNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DTRC[28:16]												
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/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
	DTRC[15:1]															—
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 to 29	—	All 0	R	Reserved
28 to 1	DTRC[28:1]	0	R/W	DTRC sets the DMA transfer count. Bits 28 to 0 are used to set the DMA transfer count on a byte basis. Bit 0 is ignored because the ATAPI data bus is handled on a 16-bit basis (on a word basis).
0	—	0	R	Reserved

**Note:** This count value does not change and the set value is retained even after DMA activation.

### 17.3.13 ATAPI Control 2 (ATAPI\_CONTROL2)

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WORD SWAP	IFEN
Initial value:	0	0	0	0	0	0	0	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
1	WORDSWAP	0	R/W	<p>WORDSWAP controls the swapping of the upper 16-bit data and the lower 16-bit data when the 32-bit data bus is enabled in the pixel bus.</p> <p>0: Word swap is not executed. 32-bit Data on the pixel bus appears in a big endian format.</p> <p>1: Word swap is executed between the ATAPI interface and register/pixel bus interface. 32-bit data on the pixel bus appears in a little endian format.</p> <p>Note that word swap is only available on Data transfer when ATAPI Control Register[0]=1: DMA mode start. Other than DMA, all register accesses use longword access.</p>
0	IFEN	0	R/W	<p>IFEN controls an ATAPI interface enable.</p> <p>0: ATAPI interface disable</p> <p>1: ATAPI interface enable</p> <p>Note: At the value of 0, ATAPI interface I/O pins function as input, and output pins goes high impedance.</p>

### 17.3.14 ATAPI Signal Status Register (ATAPI\_SIG\_ST)

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDMARDY	DMARQ
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 2	—	All 0	R	Reserved
1	DDMARDY	Undefined	R	DDMARDY indicates ATAPI DDMARDY (Inverted IDEIORDY) signal status.
0	DMARQ	Undefined	R	DMARQ indicates ATAPI DMARQ (IODREQ) signal status.

### 17.3.15 Byteswap (ATAPI\_BYTE\_SWAP)

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BYTE 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
0	BYTESWAP	0	R/W	<p>BYTESWAP controls the swapping of the upper 8 bit data and the lower 8 bit data in the ATAPI interface.</p> <p>1: Byte swap is executed between the ATAPI interface and pixel bus.</p> <p>Note that byteswap is only available on Data transfer when ATAPI Control Register[0]=1: DMA mode start.</p>

17.3.16 ATAPI Data Bus Alignment

Data bus alignment on the IO bus side  
There is no difference between big and little endian settings.  
Physical bus width: 3

Bus		32-bit bus				16-bit bus				8-bit bus			
Access		31	16	8	0	31	16	8	0	31	16	8	0
Size	Address												
	4n												
	4n+1												
	4n+2												
Byte	4n+3												
		Not specified				Not specified				Not specified			
Word	4n												
	4n+2	Not specified				Not specified				Not specified			
Longword	4n												
	4n	B3	B2	B1	B0	Not specified				Not specified			

B3: 31 to 24, B2: 23 to 16, B1: 15 to 8, and B0: 7 to 0

Data bus alignment on the pixel bus side  
Bus width fixed at 32 bits; access size is longword

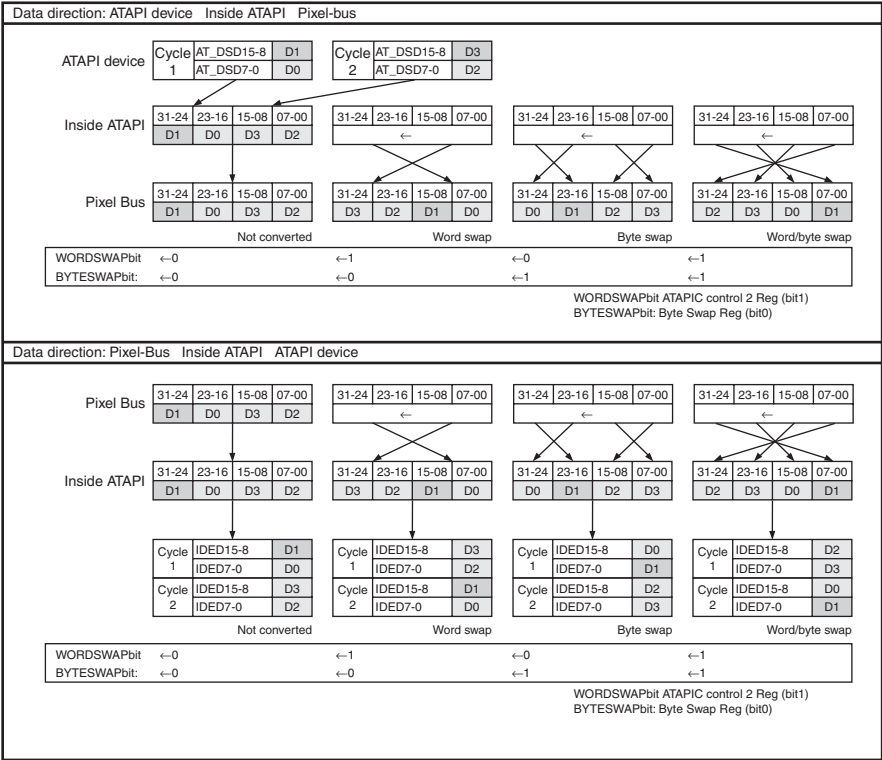


Figure 17.5 ATAPI Data Bus Alignment

## 17.4 Functional Description

This module supports a primary channel as a host. The master/slave configuration is also supported as defined in the ATAPI interface specification. The ATAPI interface's read/write FIFO buffers are designed to implement data transfer of up to 66 Mbyte/s in Ultra DMA and 16 Mbyte/s in multiword DMA modes. This module supports the 3.3-V I/O interface.

### 17.4.1 Data Transfer Modes

ATAPI interface control register supports the PIO transfer, multiword DMA transfer, and Ultra DMA transfer mode. It initiates transfer modes and sets a specific ATAPI interface timing which is different in each mode.

PIO modes 0 to 4, multiword DMA modes 0 to 2 and Ultra DMA mode 0 to 4 (up to 66 Mbyte/s) are supported.

For both multiword DMA and Ultra DMA data transfers, the pixel bus can be used while the I/O bus can only be used for the PIO transfer.

**Table 17.5 Data Transfer Modes**

Internal Operation and Internal Register	Data Transfer Mode		
	PIO Data Transfer	DMA Data Transfer between ATA Device and Pixel Bus	
		Multiword DMA	Ultra DMA
FIFO operation	Bypass*	Used	Used
BUSSEL bit in control register	Don't care	1	1
UDMAEN bit in control register	Don't care	0	1
START/STOP bit in control register	Not used	Used	Used

Note: \* The CPU accesses the ATA device in PIO mode. For DMA transfer in this table, data is transferred between the ATAPI device and the memory.

### 17.4.2 Descriptor Function

A DMA transfer for which continuous memory areas are specified can be used in this module. Set individual DMA start addresses and DMA transfer counts in the descriptor table.

## 17.5 Operating Procedure

### 17.5.1 Initialization

#### (1) Setting of Interface Enable Bit

Set the IFEN bit of the ATAPI control 2 register to 1.

#### (2) Setting of Timing Registers

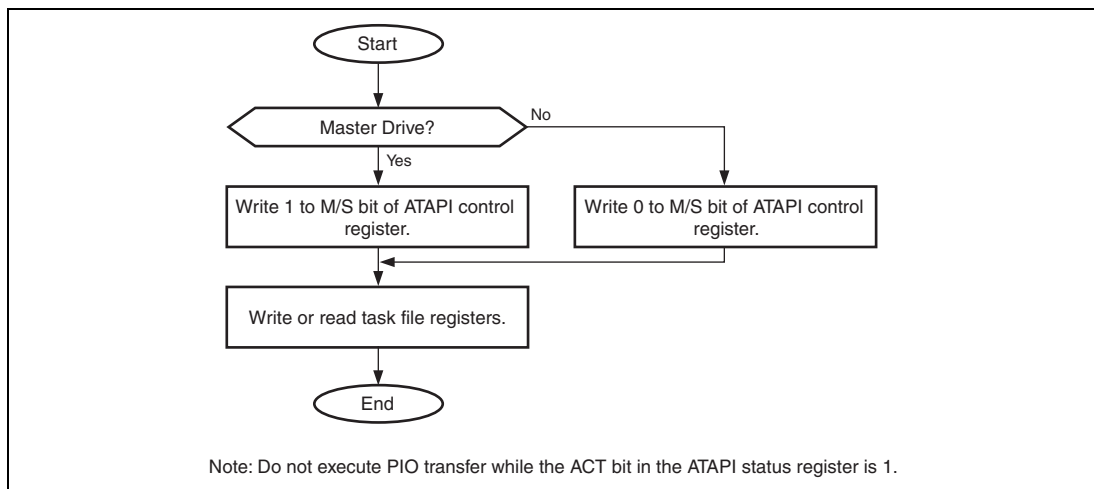
Write appropriate values to the following registers.

For details, refer to register descriptions.

- PIO timing register
- Multiword DMA timing register
- Ultra DMA timing register

### 17.5.2 Procedure in PIO Transfer Mode

- Case Not Using FIFO



**Figure 17.6 Procedure in PIO Transfer Mode**



### 17.5.3 Procedure in Multiword DMA Transfer Mode

- Transfer to and from memory via pixel bus by polling

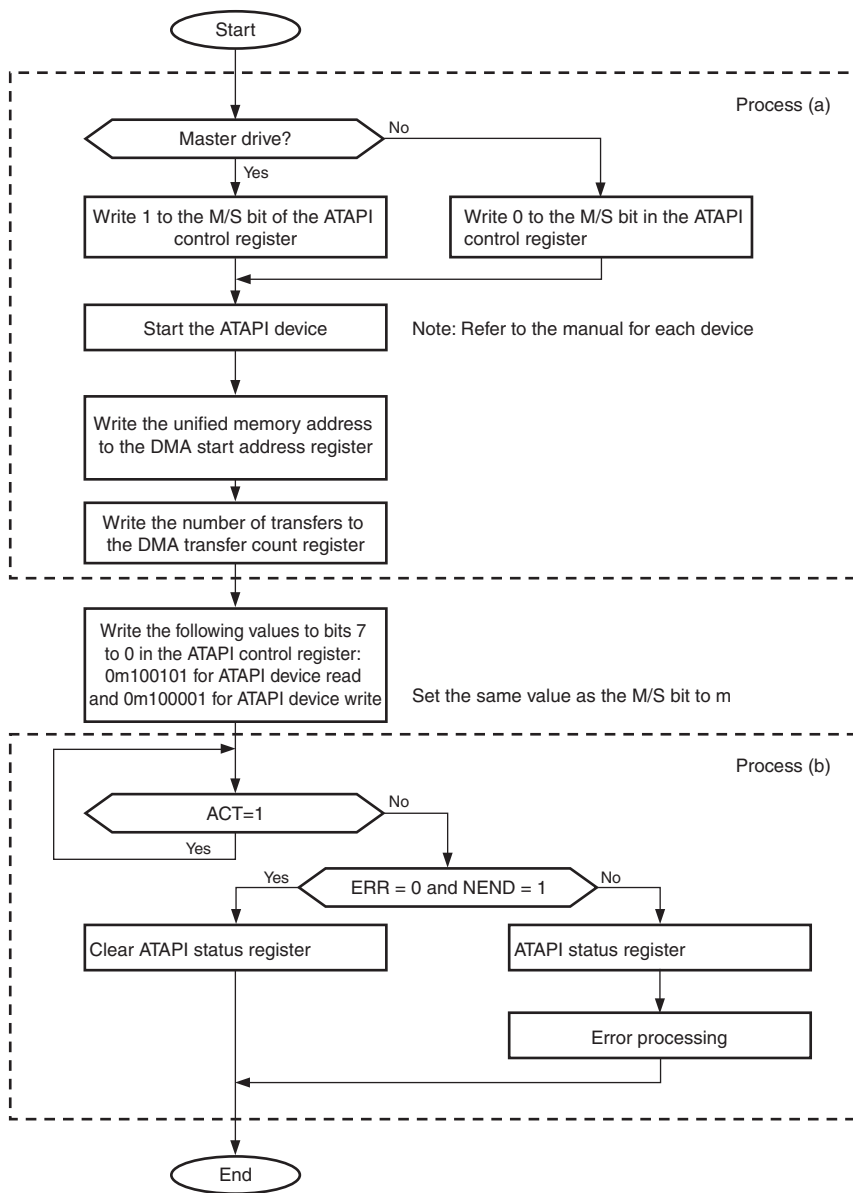
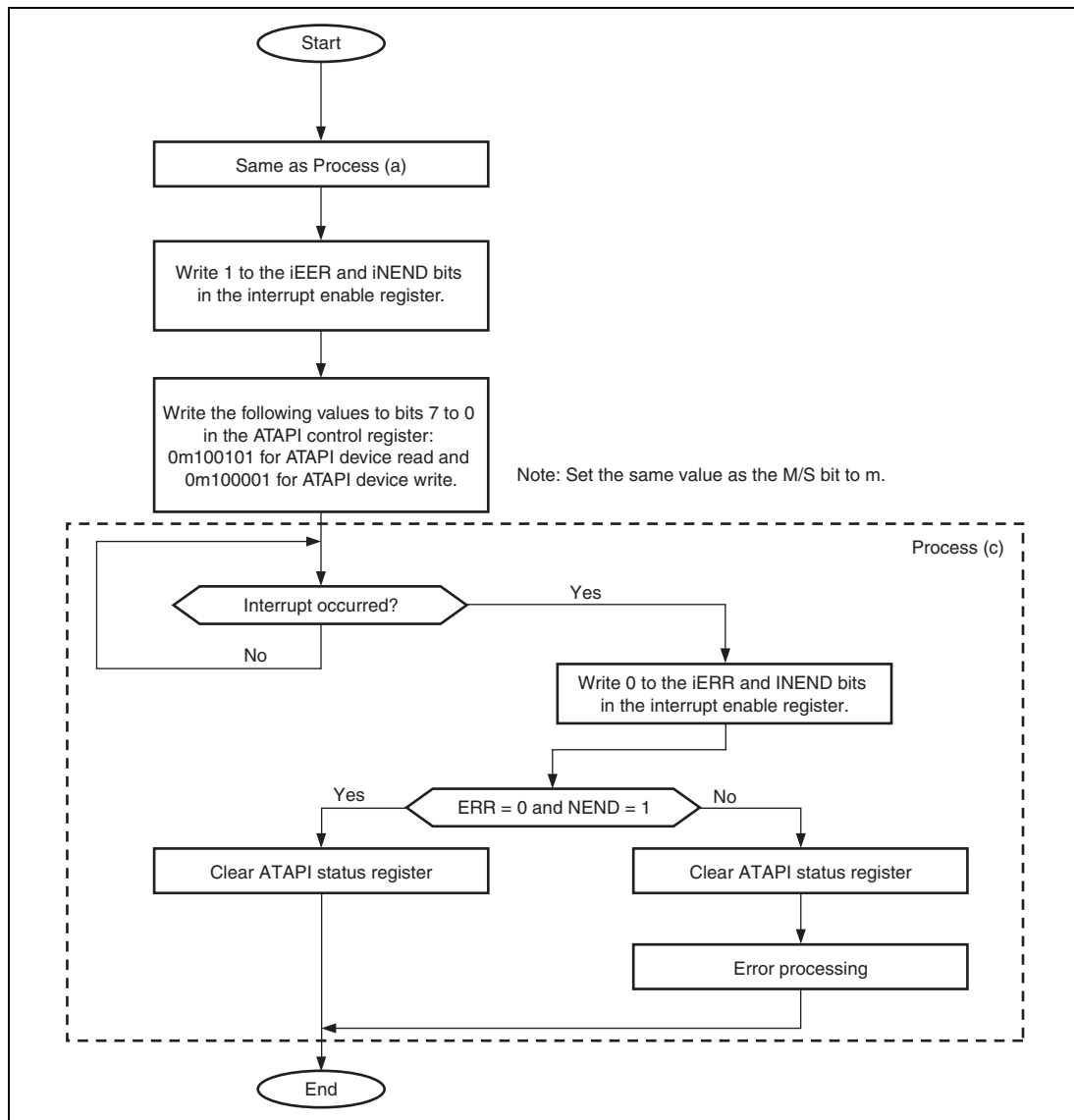


Figure 17.7 Transfer to and from Memory via Pixel Bus by Polling

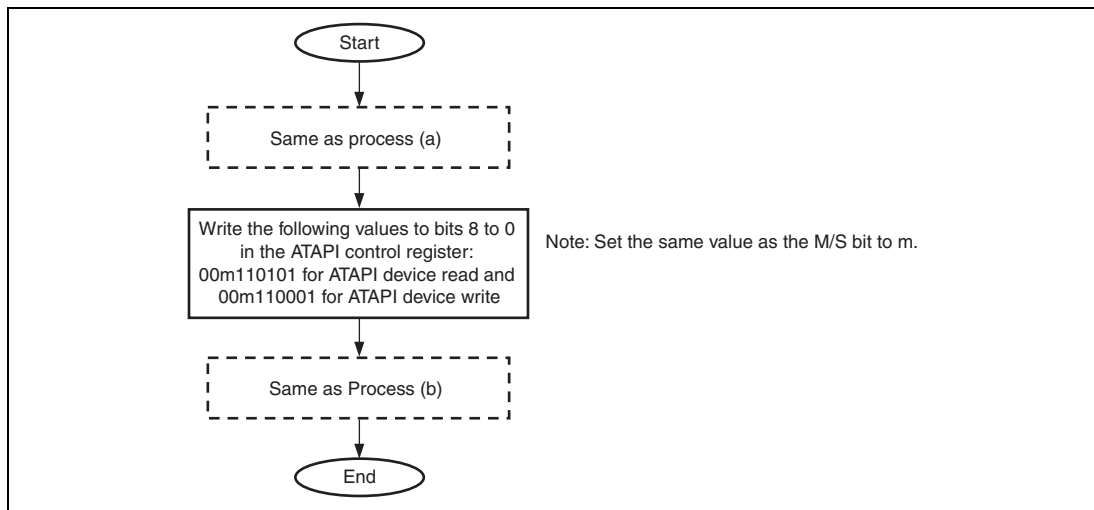
- Transfer to and from memory via pixel bus by interrupt



**Figure 17.8 Transfer to and from Memory via Pixel Bus by Interrupt**

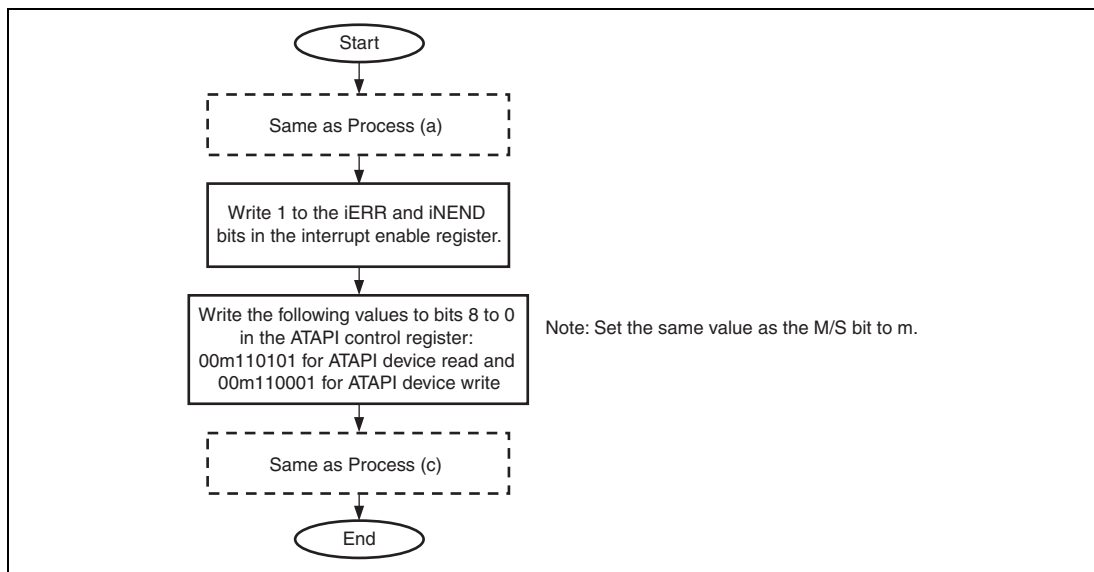
### 17.5.4 Procedure in Ultra DMA Transfer Mode

- Transfer to and from memory via pixel bus by polling



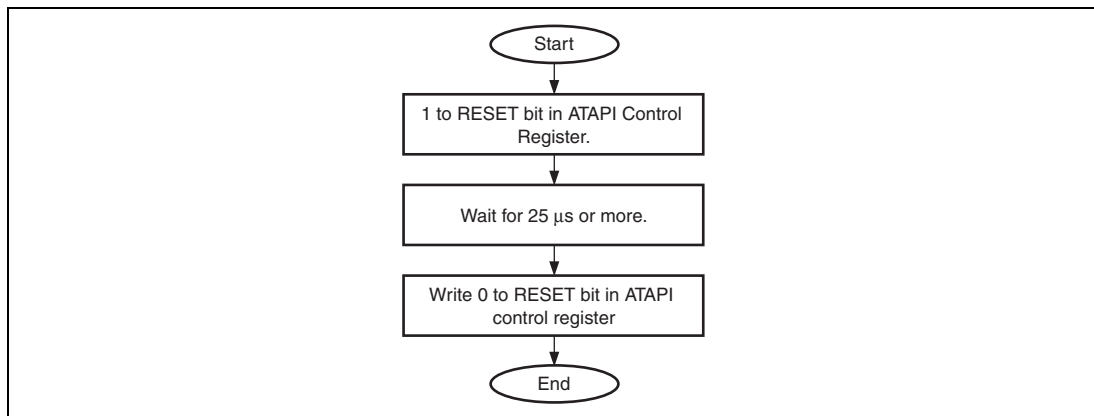
**Figure 17.9 Transfer to and from Memory via Pixel Bus by Polling**

- Transfer to and from graphics memory via pixel bus by interrupt



**Figure 17.10 Transfer to and from Memory via Pixel Bus by Interrupt**

### 17.5.5 Procedure in Hardware Reset for ATAPI Device



**Figure 17.11 Procedure in Hardware Reset for ATAPI Device**



## Section 18 Serial Sound Interface (SSI)

The serial sound interface (SSI) is a module designed to send or receive audio data interface to or from a variety of devices offering Compatibility with Philips and other formats. It also provides additional modes for other common formats, as well as support for a multi-channel mode.

### 18.1 Features

#### 18.1.1 SSI Module Configuration

The SSI module consists of the following blocks:

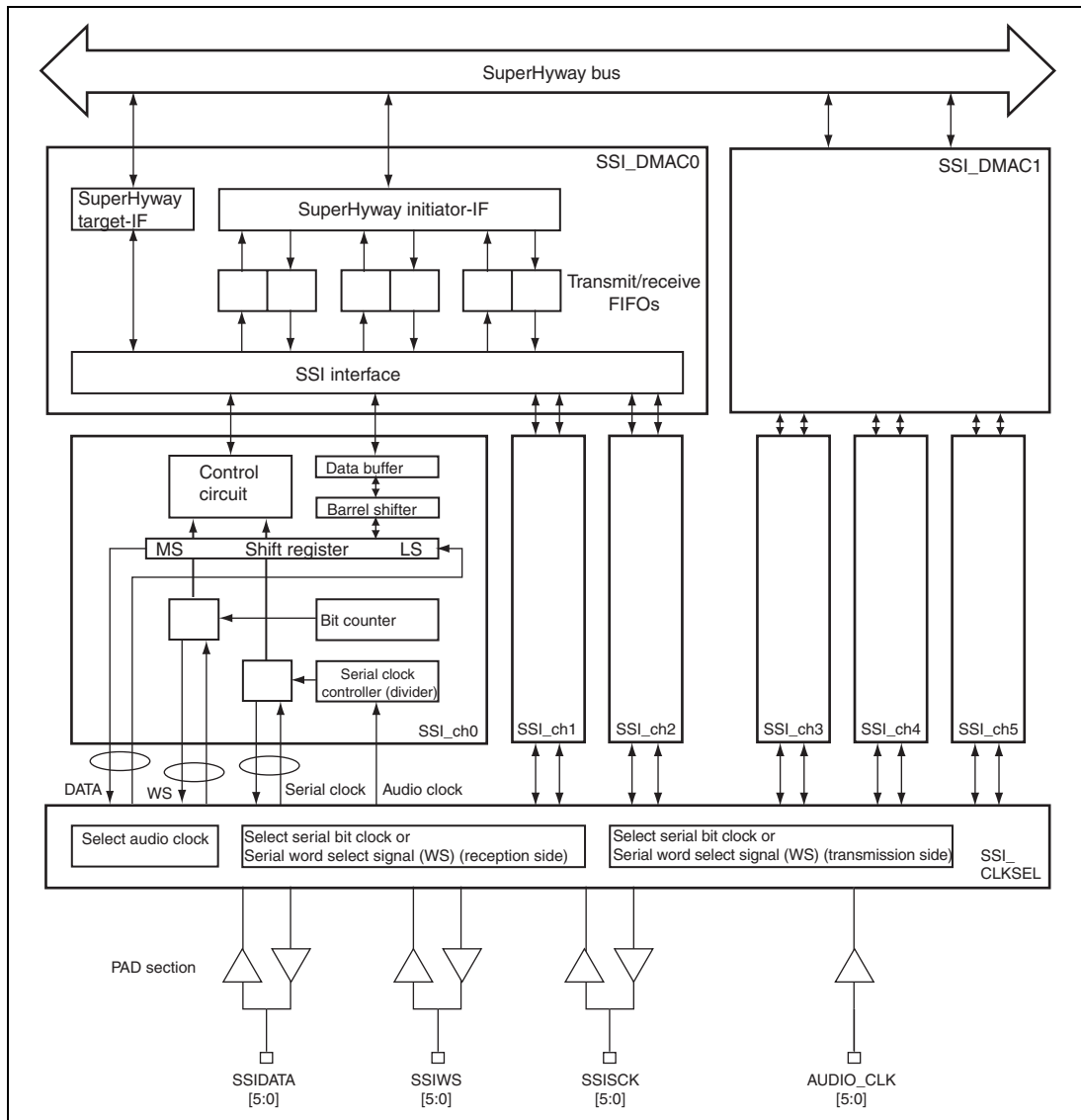
- SSI\_DMAC0 and SSI\_DMAC1
- SSI\_CH0, SSI\_CH1, SSI\_CH2, SSI\_CH3, SSI\_CH4, and SSI\_CH5
- SSI\_CLKSEL

#### 18.1.2 SSI Features

The SSI has the following features:

- Number of channels: Six channel
- Operating modes: Non-compressed mode  
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 can be used.
- Asynchronous transfer between the buffer and the shift register
- Division ratios of the serial bus interface clock can be selected.
- Data transmission and reception can be controlled by the SSI\_DMAC0/1 or interrupts.
- Audio clock can be selected for each channel.
- Serial bit clock or serial word select signal can be selected for each channel.

Figure 18.1 is a block diagram of the SSI.



**Figure 18.1 Block Diagram of SSI**



## 18.2 Input/Output Pins

Table 18.1 lists the pin configurations relating to the SSI.

**Table 18.1 Pin Configuration**

Pin Name	Function	I/O	Description
AUDIO_CLK[5:0]	Audio clock	Input	Divider input clock for SSI_CH0 to SSI_CH5 (oversampling clock)
SSISCK[5:0]	Serial bit clock	I/O	Serial bit clock for SSI_CH0 to SSI_CH5
SSIWS[5:0]	Word select	I/O	Serial word select signal for SSI_CH0 to SSI_CH5
SSIDATA[5:0]	Serial data	I/O	Serial data for SSI_CH0 to SSI_CH5

## 18.3 Register Descriptions

Table 18.2 shows the SSI\_DMAC0 register configuration. Table 18.3 shows the register state in each operating mode.

**Table 18.2 SSI\_DMAC0 Register Configuration**

Channel	Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
0	DMA mode register 0	SSIDMMR0	R/W	H'FF40 1000	H'1F40 1000	32
	RDMA transfer source address register 0	SSIRDMADR0	R/W	H'FF40 1008	H'1F40 1018	32
	RDMA transfer word count register 0	SSIRDMCNTR0	R/W	H'FF40 1010	H'1F40 1010	32
	WDMA transfer destination address register 0	SSIWDMADR0	R/W	H'FF40 1018	H'1F40 1018	32
	WDMA transfer word count register 0	SSIWDMCNTR0	R/W	H'FF40 1020	H'1F40 1020	32
	DMA control register 0	SSIDMCOR0	R/W	H'FF40 1028	H'1F40 1028	32
	Transmit suspension block counter 0	SSISTPBLCNT0	R/W	H'FF40 1030	H'1F40 1030	32
	Transmit suspension transfer data register 0	SSISTPDR0	R/W	H'FF40 1038	H'1F40 1038	32

Channel	Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
0	Block count source register 0	SSIBLCNTSR0	R/W	H'FF40 1040	H'1F40 1040	32
	Block counter 0	SSIBLCNT0	R	H'FF40 1048	H'1F40 1048	32
	n-times block transfer interrupt count source register 0	SSIBLNCNTSR0	R/W	H'FF40 1050	H'1F40 1050	32
	n-times block counter 0	SSIBLNCNT0	R	H'FF40 1058	H'1F40 1058	32
1	DMA mode register 1	SSIDMMR1	R/W	H'FF40 1060	H'1F40 1060	32
	RDMA transfer source address register 1	SSIRDMADR1	R/W	H'FF40 1068	H'1F40 1068	32
	RDMA transfer word count register 1	SSIRDMCNR1	R/W	H'FF40 1070	H'1F40 1070	32
	WDMA transfer destination address register 1	SSIWDMADR1	R/W	H'FF40 1078	H'1F40 1078	32
	WDMA transfer word count register 1	SSIWDMCNR1	R/W	H'FF40 1080	H'1F40 1080	32
	DMA control register 1	SSIDMCOR1	R/W	H'FF40 1088	H'1F40 1088	32
	Transmit suspension block counter 1	SSISTPBLCNT1	R/W	H'FF40 1090	H'1F40 1090	32
	Transmit suspension transfer data register 1	SSISTPDR1	R/W	H'FF40 1098	H'1F40 1098	32
	Block count source register 1	SSIBLCNTSR1	R/W	H'FF40 10A0	H'1F40 10A0	32
	Block counter 1	SSIBLCNT1	R	H'FF40 10A8	H'1F40 10A8	32
	n-times block transfer interrupt count source register 1	SSIBLNCNTSR1	R/W	H'FF40 10B0	H'1F40 10B0	32
	n-times block counter 1	SSIBLNCNT1	R	H'FF40 10B8	H'1F40 10B8	32
2	DMA mode register 2	SSIDMMR2	R/W	H'FF40 10C0	H'1F40 10C0	32
	RDMA transfer source address register 2	SSIRDMADR2	R/W	H'FF40 10C8	H'1F40 10C8	32
	RDMA transfer word count register 2	SSIRDMCNR2	R/W	H'FF40 10D0	H'1F40 10D0	32

Channel	Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
2	WDMA transfer destination address register 2	SSIWDMADR2	R/W	H'FF40 10D8	H'1F40 10D8	32
	WDMA transfer word count register 2	SSIWDMCNTR2	R/W	H'FF40 10E0	H'1F40 10E0	32
	DMA control register 2	SSIDMCOR2	R/W	H'FF40 10E8	H'1F40 10E8	32
	Transmit suspension block counter 2	SSISTPBLCNT2	R/W	H'FF40 10F0	H'1F40 10F0	32
	Transmit suspension transfer data register 2	SSISTPDR2	R/W	H'FF40 10F8	H'1F40 10F8	32
	Block count source register 2	SSIBLCNTR2	R/W	H'FF40 1100	H'1F40 1100	32
	Block counter 2	SSIBLCNT2	R	H'FF40 1108	H'1F40 1108	32
	n-times block transfer interrupt count source register 2	SSIBLNCNTR2	R/W	H'FF40 1110	H'1F40 1110	32
	n-times block counter 2	SSIBLNCNT2	R	H'FF40 1118	H'1F40 1118	32
Common to 0 to 2	DMA operation register 0	SSIDMAOR0	R/W	H'FF40 1180	H'1F40 1180	32
	Interrupt status register 0	SSIDMINTSR0	R/W	H'FF40 1188	H'1F40 1188	32
	Interrupt mask register 0	SSIDMINTMR0	R/W	H'FF40 1190	H'1F40 1190	32

**Table 18.3 SSI\_DMACH0 Register State in Each Operating Mode**

Channel	Register Name	Abbreviation	Power-On Reset	Sleep	Standby
0	DMA mode register 0	SSIDMMR0	H'0000 0000	Retained	Retained
	RDMA transfer source address register 0	SSIRDMADR0	H'0000 0000	Retained	Retained
	RDMA transfer word count register 0	SSIRDMCNTR0	H'0000 0000	Retained	Retained
	WDMA transfer destination address register 0	SSIWDMADR0	H'0000 0000	Retained	Retained
	WDMA transfer word count register 0	SSIWDMCNTR0	H'0000 0000	Retained	Retained
	DMA control register 0	SSIDMCOR0	H'0000 0000	Retained	Retained
	Transmit suspension block counter 0	SSISTPBLCNT0	H'0000 0000	Retained	Retained
	Transmit suspension transfer data register 0	SSISTPDR0	H'0000 0000	Retained	Retained
	Block count source register 0	SSIBLCNTSR0	H'0000 0000	Retained	Retained
	Block counter 0	SSIBLCNT0	H'0000 0000	Retained	Retained
	n-times block transfer interrupt count source register 0	SSIBLNCNTSR0	H'0000 0000	Retained	Retained
	n-times block counter 0	SSIBLNCNT0	H'0000 0000	Retained	Retained
1	DMA mode register 1	SSIDMMR1	H'0000 0000	Retained	Retained
	RDMA transfer source address register 1	SSIRDMADR1	H'0000 0000	Retained	Retained
	RDMA transfer word count register 1	SSIRDMCNTR1	H'0000 0000	Retained	Retained
	WDMA transfer destination address register 1	SSIWDMADR1	H'0000 0000	Retained	Retained
	WDMA transfer word count register 1	SSIWDMCNTR1	H'0000 0000	Retained	Retained
	DMA control register 1	SSIDMCOR1	H'0000 0000	Retained	Retained
	Transmit suspension block counter 1	SSISTPBLCNT1	H'0000 0000	Retained	Retained

Channel	Register Name	Abbreviation	Power-On Reset	Sleep	Standby
1	Transmit suspension transfer data register 1	SSISTPDR1	H'0000 0000	Retained	Retained
	Block count source register 1	SSIBLCNTR1	H'0000 0000	Retained	Retained
	Block counter 1	SSIBLCNT1	H'0000 0000	Retained	Retained
	n-times block transfer interrupt count source register 1	SSIBLNCNTR1	H'0000 0000	Retained	Retained
	n-times block counter 1	SSIBLNCNT1	H'0000 0000	Retained	Retained
2	DMA mode register 2	SSIDMMR2	H'0000 0000	Retained	Retained
	RDMA transfer source address register 2	SSIIRDMADR2	H'0000 0000	Retained	Retained
	RDMA transfer word count register 2	SSIIRDMCNTR2	H'0000 0000	Retained	Retained
	WDMA transfer destination address register 2	SSIWDMADR2	H'0000 0000	Retained	Retained
	WDMA transfer word count register 2	SSIWDMCNTR2	H'0000 0000	Retained	Retained
	DMA control register 2	SSIDMCOR2	H'0000 0000	Retained	Retained
	Transmit suspension block counter 2	SSISTPBLCNT2	H'0000 0000	Retained	Retained
	Transmit suspension transfer data register 2	SSISTPDR2	H'0000 0000	Retained	Retained
	Block count source register 2	SSIBLCNTR2	H'0000 0000	Retained	Retained
	Block counter 2	SSIBLCNT2	H'0000 0000	Retained	Retained
	n-times block transfer interrupt count source register 2	SSIBLNCNTR2	H'0000 0000	Retained	Retained
	n-times block counter 2	SSIBLNCNT2	H'0000 0000	Retained	Retained
Common to 0 to 2	DMA operation register 0	SSIDMAOR0	H'0000 0000	Retained	Retained
	Interrupt status register 0	SSIDMINTSR0	H'0101 0101	Retained	Retained
	Interrupt mask register 0	SSIDMINTMR0	H'1F1F 1F1F	Retained	Retained

Table 18.4 shows the SSI\_DMACH1 register configuration. Table 18.5 shows the register state in each operating mode.

**Table 18.4 SSI\_DMACH1 Register Configuration**

Channel	Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
3	DMA mode register 3	SSIDMMR3	R/W	H'FF50 1000	H'1F50 1000	32
	RDMA transfer source address register 3	SSIRDMDADR3	R/W	H'FF50 1008	H'1F50 1008	32
	RDMA transfer word count register 3	SSIRDMCNTR3	R/W	H'FF50 1010	H'1F50 1010	32
	WDMA transfer destination address register 3	SSIWDMADR3	R/W	H'FF50 1018	H'1F50 1018	32
	WDMA transfer word count register 3	SSIWDMCNTR3	R/W	H'FF50 1020	H'1F50 1020	32
	DMA control register 3	SSIDMCOR3	R/W	H'FF50 1028	H'1F50 1028	32
	Transmit suspension block counter 3	SSISTPBLCNT3	R/W	H'FF50 1030	H'1F50 1030	32
	Transmit suspension transfer data register 3	SSISTPDR3	R/W	H'FF50 1038	H'1F50 1038	32
	Block count source register 3	SSIBLCNTR3	R/W	H'FF50 1040	H'1F50 1040	32
	Block counter 3	SSIBLCNT3	R	H'FF50 1048	H'1F50 1048	32
	n-times block transfer interrupt count source register 3	SSIBLNCNTR3	R/W	H'FF50 1050	H'1F50 1050	32
	n-times block counter 3	SSIBLNCNT3	R	H'FF50 1058	H'1F50 1058	32
4	DMA mode register 4	SSIDMMR4	R/W	H'FF50 1060	H'1F50 1060	32
	RDMA transfer source address register 4	SSIRDMDADR4	R/W	H'FF50 1068	H'1F50 1068	32
	RDMA transfer word count register 4	SSIRDMCNTR4	R/W	H'FF50 1070	H'1F50 1070	32
	WDMA transfer destination address register 4	SSIWDMADR4	R/W	H'FF50 1078	H'1F50 1078	32

Channel	Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
4	WDMA transfer word count register 4	SSIWDMCNR4	R/W	H'FF50 1080	H'1F50 1080	32
	DMA control register 4	SSIDMCOR4	R/W	H'FF50 1088	H'1F50 1088	32
	Transmit suspension block counter 4	SSISTPBLCNT4	R/W	H'FF50 1090	H'1F50 1090	32
	Transmit suspension transfer data register 4	SSISTPDR4	R/W	H'FF50 1098	H'1F50 1098	32
	Block count source register 4	SSIBLCNTR4	R/W	H'FF50 10A0	H'1F50 10A0	32
	Block counter 4	SSIBLCNT4	R	H'FF50 10A8	H'1F50 10A8	32
	n-times block transfer interrupt count source register 4	SSIBLNCNTR4	R/W	H'FF50 10B0	H'1F50 10B0	32
	n-times block counter 4	SSIBLNCNT4	R	H'FF50 10B8	H'1F50 10B8	32
5	DMA mode register 5	SSIDMMR5	R/W	H'FF50 10C0	H'1F50 10C0	32
	RDMA transfer source address register 5	SSIRDMAADR5	R/W	H'FF50 10C8	H'1F50 10C8	32
	RDMA transfer word count register 5	SSIRDMCNR5	R/W	H'FF50 10D0	H'1F50 10D0	32
	WDMA transfer destination address register 5	SSIWDMADR5	R/W	H'FF50 10D8	H'1F50 10D8	32
	WDMA transfer word count register 5	SSIWDMCNR5	R/W	H'FF50 10E0	H'1F50 10E0	32
	DMA control register 5	SSIDMCOR5	R/W	H'FF50 10E8	H'1F50 10E8	32
	Transmit suspension block counter 5	SSISTPBLCNT5	R/W	H'FF50 10F0	H'1F50 10F0	32
	Transmit suspension transfer data register 5	SSISTPDR5	R/W	H'FF50 10F8	H'1F50 10F8	32
	Block count source register 5	SSIBLCNTR5	R/W	H'FF50 1100	H'1F50 1100	32
	Block counter 5	SSIBLCNT5	R	H'FF50 1108	H'1F50 1108	32
	n-times block transfer interrupt count source register 5	SSIBLNCNTR5	R/W	H'FF50 1110	H'1F50 1110	32
	n-times block counter 5	SSIBLNCNT5	R	H'FF50 1118	H'1F50 1118	32

Channel	Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
Common to 3 to 5	DMA operation register 1	SSIDMAOR1	R/W	H'FF50 1180	H'1F50 1180	32
	Interrupt status register 1	SSIDMINTSR1	R/W	H'FF50 1188	H'1F50 1188	32
	Interrupt mask register 1	SSIDMINTMR1	R/W	H'FF50 1190	H'1F50 1190	32

Table 18.5 SSI\_DMACH1 Register State in Each Operating Mode

Channel	Register Name	Abbreviation	Power-On Reset	Sleep	Standby
3	DMA mode register 3	SSIDMMR3	H'0000 0000	Retained	Retained
	RDMA transfer source address register 3	SSIRDMADR3	H'0000 0000	Retained	Retained
	RDMA transfer word count register 3	SSIRDMCNR3	H'0000 0000	Retained	Retained
	WDMA transfer destination address register 3	SSIWDMADR3	H'0000 0000	Retained	Retained
	WDMA transfer word count register 3	SSIWDMCNR3	H'0000 0000	Retained	Retained
	DMA control register 3	SSIDMCOR3	H'0000 0000	Retained	Retained
	Transmit suspension block counter 3	SSISTPBLCNT3	H'0000 0000	Retained	Retained
	Transmit suspension transfer data register 3	SSISTPDR3	H'0000 0000	Retained	Retained
	Block count source register 3	SSIBLCNTSR3	H'0000 0000	Retained	Retained
	Block counter 3	SSIBLCNT3	H'0000 0000	Retained	Retained
	n-times block transfer interrupt count source register 3	SSIBLNCNTSR3	H'0000 0000	Retained	Retained
	n-times block counter 3	SSIBLNCNT3	H'0000 0000	Retained	Retained
4	DMA mode register 4	SSIDMMR4	H'0000 0000	Retained	Retained
	RDMA transfer source address register 4	SSIRDMADR4	H'0000 0000	Retained	Retained



Channel	Register Name	Abbreviation	Power-On Reset	Sleep	Standby
4	RDMA transfer word count register 4	SSIRDMCNTR4	H'0000 0000	Retained	Retained
	WDMA transfer destination address register 4	SSIWDMADR4	H'0000 0000	Retained	Retained
	WDMA transfer word count register 4	SSIWDMCNTR4	H'0000 0000	Retained	Retained
	DMA control register 4	SSIDMCOR4	H'0000 0000	Retained	Retained
	Transmit suspension block counter 4	SSISTPBLCNT4	H'0000 0000	Retained	Retained
	Transmit suspension transfer data register 4	SSISTPDR4	H'0000 0000	Retained	Retained
	Block count source register 4	SSIBLCNTR4	H'0000 0000	Retained	Retained
	Block counter 4	SSIBLCNT4	H'0000 0000	Retained	Retained
	n-times block transfer interrupt count source register 4	SSIBLNCNTR4	H'0000 0000	Retained	Retained
	n-times block counter 4	SSIBLNCNT4	H'0000 0000	Retained	Retained
5	DMA mode register 5	SSIDMMR5	H'0000 0000	Retained	Retained
	RDMA transfer source address register 5	SSIRDMADR5	H'0000 0000	Retained	Retained
	RDMA transfer word count register 5	SSIRDMCNTR5	H'0000 0000	Retained	Retained
	WDMA transfer destination address register 5	SSIWDMADR5	H'0000 0000	Retained	Retained
	WDMA transfer word count register 5	SSIWDMCNTR5	H'0000 0000	Retained	Retained
	DMA control register 5	SSIDMCOR5	H'0000 0000	Retained	Retained
	Transmit suspension block counter 5	SSISTPBLCNT5	H'0000 0000	Retained	Retained
	Transmit suspension transfer data register 5	SSISTPDR5	H'0000 0000	Retained	Retained
	Block count source register 5	SSIBLCNTR5	H'0000 0000	Retained	Retained

Channel	Register Name	Abbreviation	Power-On Reset	Sleep	Standby
5	Block counter 5	SSIBLCNT5	H'0000 0000	Retained	Retained
	n-times block transfer interrupt count source register 5	SSIBLNCNTR5	H'0000 0000	Retained	Retained
	n-times block counter 5	SSIBLNCNT5	H'0000 0000	Retained	Retained
Common to 3 to 5	DMA operation register 1	SSIDMAOR1	H'0000 0000	Retained	Retained
	Interrupt status register 1	SSIDMINTSR1	H'0101 0101	Retained	Retained
	Interrupt mask register 1	SSIDMINTMR1	H'1F1F 1F1F	Retained	Retained

Table 18.6 shows the register configuration of SSI\_CH0 to SSI\_CH5. Table 18.7 shows the register state in each operating mode.

**Table 18.6 Register Configuration of SSI\_CH0 to SSI\_CH5**

Channel	Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
0	Control register 0	SSICR0	R/W	H'FF40 2000	H'1F40 2000	32
	Status register 0	SSISR0	R/W*	H'FF40 2004	H'1F40 2004	32
	Transmit data register 0	SSITDR0	R/W	H'FF40 2008	H'1F40 2008	32
	Receive data register 0	SSIRDR0	R	H'FF40 200C	H'1F40 200C	32
1	Control register 1	SSICR1	R/W	H'FF40 3000	H'1F40 3000	32
	Status register 1	SSISR1	R/W*	H'FF40 3004	H'1F40 3004	32
	Transmit data register 1	SSITDR1	R/W	H'FF40 3008	H'1F40 3008	32
	Receive data register 1	SSIRDR1	R	H'FF40 300C	H'1F40 300C	32
2	Control register 2	SSICR2	R/W	H'FF40 4000	H'1F40 4000	32
	Status register 2	SSISR2	R/W*	H'FF40 4004	H'1F40 4004	32
	Transmit data register 2	SSITDR2	R/W	H'FF40 4008	H'1F40 4008	32
	Receive data register 2	SSIRDR2	R	H'FF40 400C	H'1F40 400C	32
3	Control register 3	SSICR3	R/W	H'FF50 2000	H'1F50 2000	32
	Status register 3	SSISR3	R/W*	H'FF50 2004	H'1F50 2004	32
	Transmit data register 3	SSITDR3	R/W	H'FF50 2008	H'1F50 2008	32
	Receive data register 3	SSIRDR3	R	H'FF50 200C	H'1F50 200C	32
4	Control register 4	SSICR4	R/W	H'FF50 3000	H'1F50 3000	32
	Status register 4	SSISR4	R/W*	H'FF50 3004	H'1F50 3004	32
	Transmit data register 4	SSITDR4	R/W	H'FF50 3008	H'1F50 3008	32
	Receive data register 4	SSIRDR4	R	H'FF50 300C	H'1F50 300C	32
5	Control register 5	SSICR5	R/W	H'FF50 4000	H'1F50 4000	32
	Status register 5	SSISR5	R/W*	H'FF50 4004	H'1F50 4004	32
	Transmit data register 5	SSITDR5	R/W	H'FF50 4008	H'1F50 4008	32
	Receive data register 5	SSIRDR5	R	H'FF50 400C	H'1F50 400C	32

Note: \* Only bits 26 and 27 can be read and written to. Other bits are read-only. For details, see section 18.3.17, Status Registers 0 to 5 (SSISR0 to SSISR5).

**Table 18.7 Register State in Each Operating Mode for SSI\_CH0 to SSI\_CH5**

Channel	Register Name	Abbreviation	Power-On Reset	Sleep	Standby
0	Control register 0	SSICR0	H'0000 0000	Retained	Retained
	Status register 0	SSISR0	H'0210 A003	Retained	Retained
	Transmit data register 0	SSITDR0	H'0000 0000	Retained	Retained
	Receive data register 0	SSIRDR0	H'0000 0000	Retained	Retained
1	Control register 1	SSICR1	H'0000 0000	Retained	Retained
	Status register 1	SSISR1	H'0210 A003	Retained	Retained
	Transmit data register 1	SSITDR1	H'0000 0000	Retained	Retained
	Receive data register 1	SSIRDR1	H'0000 0000	Retained	Retained
2	Control register 2	SSICR2	H'0000 0000	Retained	Retained
	Status register 2	SSISR2	H'0210 A003	Retained	Retained
	Transmit data register 2	SSITDR2	H'0000 0000	Retained	Retained
	Receive data register 2	SSIRDR2	H'0000 0000	Retained	Retained
3	Control register 3	SSICR3	H'0000 0000	Retained	Retained
	Status register 3	SSISR3	H'0210 A003	Retained	Retained
	Transmit data register 3	SSITDR3	H'0000 0000	Retained	Retained
	Receive data register 3	SSIRDR3	H'0000 0000	Retained	Retained
4	Control register 4	SSICR4	H'0000 0000	Retained	Retained
	Status register 4	SSISR4	H'0210 A003	Retained	Retained
	Transmit data register 4	SSITDR4	H'0000 0000	Retained	Retained
	Receive data register 4	SSIRDR4	H'0000 0000	Retained	Retained
5	Control register 5	SSICR5	H'0000 0000	Retained	Retained
	Status register 5	SSISR5	H'0210 A003	Retained	Retained
	Transmit data register 5	SSITDR5	H'0000 0000	Retained	Retained
	Receive data register 5	SSIRDR5	H'0000 0000	Retained	Retained

### 18.3.1 DMA Mode Registers 0 to 5 (SSIDMMR0 to SSIDMMR5)

SSIDMMR0 to SSIDMMR5 is a 32-bit readable/writable register that set the operation mode for SSI\_DMACH other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMACH (DMRST bit in SSIDMACOR0 to SSIDMACOR3).

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RDMBSZ[1:0]	WDMBSZ[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	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved  These bits are always read as a 0. The write value should always be 0.
18	RDSAM	0	R/W	RDMA Transfer Source Address Mode  Increments or decrements the transfer source address during RDMA transfer.  0: Increments the transfer source address (+4). 1: Decrements the transfer source address (−4).
17	—	0	R	Reserved  This bit is always read as 0. The write value should always be 0.
16	WDDAM	0	R/W	WDMA Transfer Destination Address Mode  Increments or decrements the transfer destination address during WDMA transfer.  0: Increments the transfer destination address (+4). 1: Decrements the transfer destination address (−4).
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, 6	RDMBSZ[1:0]	00	R/W	RDMA Maximum Burst Size These bits set the maximum burst size during RDMA data transfer. 00: 1 burst (8 bytes) 01: 2 bursts (16 bytes) 10: 4 bursts (32 bytes) 11: Setting prohibited
5, 4	WDMBSZ[1:0]	00	R/W	WDMA Maximum Burst Size These bits set the maximum burst size during WDMA data transfer. 00: 1 burst (8 bytes) 01: 2 bursts (16 bytes) 10: 4 bursts (32 bytes) 11: Setting prohibited
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

### 18.3.2 RDMA Transfer Source Address Registers 0 to 5 (SSIRDMADR0 to SSIRDMADR5)

SSIRDMADR0 to SSIRDMADR5 is a 32-bit readable/writable register that set the data transfer source address during RDMA transfer other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAC (DMRST bit in SSIDMACOR0 to SSIDMACOR3). To be written to this register, DMEN bit in SSIDMCOR0 to SSIDMCOR5 must be 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDMADR[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
	RDMADR[15:3]													—	—	—
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	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	RDMADR[31:3]	All 0	R/W	RDMA Transfer Source Address These bits set the data transfer source memory address during RDMA transfer.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.).

### 18.3.3 RDMA Transfer Word Count Registers 0 to 5 (SSIRDMCNTR0 to SSIRDMCNTR5)

SSIRDMCNTR0 to SSIRDMCNTR5 is a 32-bit readable/writable register that set the data transfer word count (number of bytes) during RDMA transfer other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAR (DMRST bit in SSIDMACOR0 to SSIDMACOR3). To be written to this register, DMEN bit in SSIDMCOR0 to SSIDMCOR5 must be 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDMCNT[31:16]															
Bit:	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
	RDMCNT[15:4]												—	—	—	—
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 4	RDMCNT[31 :4]	All 0	R/W	<p>RDMA Transfer Count</p> <p>These bits set the data transfer count during RDMA transfer. The following transfer count should be selected according to the RDMA maximum burst size.</p> <p>1 burst: <math>8 \times n</math> [H'08 <math>\times</math> n] (bytes)</p> <p>2 bursts: <math>16 \times n</math> [H'10 <math>\times</math> n] (bytes)</p> <p>4 bursts: <math>32 \times n</math> [H'20 <math>\times</math> n] (bytes)</p>
3 to 0	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.).</p>



### 18.3.4 WDMA Transfer Destination Address Registers 0 to 5 (SSIWDMADR0 to SSIWDMADR5)

SSIWDMADR0 to SSIWDMADR5 is a 32-bit readable/writable register that set the transfer destination memory address during WDMA transfer other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAC (DMRST bit in SSIDMACOR0 to SSIDMACOR3). To be written to this register, DMEN bit in SSIDMCOR0 to SSDMCOR5 must be 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDMADR[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
	WDMADR[15:3]													—	—	—
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	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	WDMADR [31:3]	All 0	R/W	WDMA Transfer Source Address These bits set the data transfer destination memory address during WDMA transfer.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

### 18.3.5 WDMA Transfer Word Count Registers 0 to 5 (SSIWDMCNTR0 to SSIWDMCNTR5)

SSIWDMCNTR0 to SSIWDMCNTR5 is a 32-bit readable/writable register that set data transfer word count (number of bytes) during WDMA transfer other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAC (DMRST bit in SSIDMACOR0 to SSIDMACOR3). To be written to this register, DMEN bit in SSIDMCOR0 to SSDMCOR5 must be 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDMCNT[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
	WDMCNT[15:4]												—	—	—	—
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 4	WDMCNT [31:4]	All 0	R/W	<b>WDMA Transfer Count</b> These bits set the data transfer count during WDMA transfer. The following transfer count should be selected according to the WDMA maximum burst size. 1 burst: $8 \times n$ [H'08 $\times$ n] (bytes) 2 bursts: $16 \times n$ [H'10 $\times$ n] (bytes) 4 bursts: $32 \times n$ [H'20 $\times$ n] (bytes)
3 to 0	—	All 0	R	<b>Reserved</b> These bits are always read as 0. The write value should always be 0.

### 18.3.6 DMA Control Registers 0 to 5 (SSIDMCOR0 to SSIDMCOR5)

SSIDMCOR0 to SSIDMCOR5 is a 32-bit readable/writable register that controls the operations and stop for SSI\_DMAC0/1 and selects serial bit clocks other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAC (DMRST bit in SSIDMACOR0 to SSIDMACOR3).

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SCKSOP[2:0]			SCKSIP[2:0]			SCKS[2:0]			—	RPT MD	TRMD	DMEN
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	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	DMRST	0	W	SSI_DMAC0/1 Software Reset Resets the SSI_DMAC0/1 of corresponding channel among SSI_CH0 to SSI_CH5, and stops data transfer. This bit is always read as 0. 0: The software reset is disabled. 1: The software reset is enabled.
30	TXRST	0	R/W	Transmit FIFO Buffer Reset Resets the transmit FIFO buffer. 0: The transmit FIFO buffer reset is disabled. 1: The transmit FIFO buffer reset is enabled.
29	RXRST	0	R/W	Receive FIFO Buffer Reset Resets the receive FIFO buffer. 0: The receive FIFO buffer reset is disabled. 1: The receive FIFO buffer reset is enabled.
28 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	SCKSOP [2:0]	000	R/W	<p>Serial Bit Clock and Serial Word Select Signal Output Selection</p> <p>These bits select the serial bit clock and serial word select signal to be output from SSI_CH0 through SSI_CH5 to external devices.</p> <ul style="list-style-type: none"> <li>In SSIDMCOR0:               <ul style="list-style-type: none"> <li>000: SSISCK0/SSIWS0 = serial bit clock or word select signal of SSI_CH0 are output.</li> <li>001: SSISCK0/SSIWS0 = serial bit clock or word select signal of SSI_CH1 are output.</li> <li>010: SSISCK0/SSIWS0 = serial bit clock or word select signal of SSI_CH2 are output.</li> <li>011: Setting prohibited</li> <li>100: SSISCK0/SSIWS0 = serial bit clock or word select signal of SSI_CH3 are output.</li> <li>101: SSISCK0/SSIWS0 = serial bit clock or word select signal of SSI_CH4 are output.</li> <li>110: SSISCK0/SSIWS0 = serial bit clock or word select signal of SSI_CH5 are output.</li> <li>111: Setting prohibited</li> </ul> </li> <li>In SSIDMCOR1:               <ul style="list-style-type: none"> <li>000: SSISCK1/SSIWS1 = serial bit clock or word select signal of SSI_CH1 are output.</li> <li>001: SSISCK1/SSIWS1 = serial bit clock or word select signal of SSI_CH2 are output.</li> <li>010: Setting prohibited</li> <li>011: SSISCK1/SSIWS1 = serial bit clock or word select signal of SSI_CH3 are output.</li> <li>100: SSISCK1/SSIWS1 = serial bit clock or word select signal of SSI_CH4 are output.</li> <li>101: SSISCK1/SSIWS1 = serial bit clock or word select signal of SSI_CH5 are output.</li> <li>110: Setting prohibited</li> <li>111: SSISCK1/SSIWS1 = serial bit clock or word select signal of SSI_CH0 are output.</li> </ul> </li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	SCKSOP [2:0]	000	R/W	<ul style="list-style-type: none"> <li>In SSIDMCOR2:               <ul style="list-style-type: none"> <li>000: SSISCK2/SSIWS2 = serial bit clock or word select signal of SSI_CH2 are output.</li> <li>001: Setting prohibited</li> <li>010: SSISCK2/SSIWS2 = serial bit clock or word select signal of SSI_CH3 are output.</li> <li>011: SSISCK2/SSIWS2 = serial bit clock or word select signal of SSI_CH4 are output.</li> <li>100: SSISCK2/SSIWS2 = serial bit clock or word select signal of SSI_CH5 are output.</li> <li>101: Setting prohibited</li> <li>110: SSISCK2/SSIWS2 = serial bit clock or word select signal of SSI_CH0 are output.</li> <li>111: SSISCK2/SSIWS2 = serial bit clock or word select signal of SSI_CH1 are output.</li> </ul> </li> <li>In SSIDMCOR3:               <ul style="list-style-type: none"> <li>000: SSISCK3/SSIWS3 = serial bit clock or word select signal of SSI_CH3 are output.</li> <li>001: SSISCK3/SSIWS3 = serial bit clock or word select signal of SSI_CH4 are output.</li> <li>010: SSISCK3/SSIWS3 = serial bit clock or word select signal of SSI_CH5 are output.</li> <li>011: Setting prohibited</li> <li>100: SSISCK3/SSIWS3 = serial bit clock or word select signal of SSI_CH0 are output.</li> <li>101: SSISCK3/SSIWS3 = serial bit clock or word select signal of SSI_CH1 are output.</li> <li>110: SSISCK3/SSIWS3 = serial bit clock or word select signal of SSI_CH2 are output.</li> <li>111: Setting prohibited</li> </ul> </li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	SCKSOP [2:0]	000	R/W	<ul style="list-style-type: none"> <li>In SSIDMCOR4:               <ul style="list-style-type: none"> <li>000: SSISCK4/SSIWS4 = serial bit clock or word select signal of SSI_CH4 are output.</li> <li>001: SSISCK4/SSIWS4 = serial bit clock or word select signal of SSI_CH5 are output.</li> <li>010: Setting prohibited</li> <li>011: SSISCK4/SSIWS4 = serial bit clock or word select signal of SSI_CH0 are output.</li> <li>100: SSISCK4/SSIWS4 = serial bit clock or word select signal of SSI_CH1 are output.</li> <li>101: SSISCK4/SSIWS4 = serial bit clock or word select signal of SSI_CH2 are output.</li> <li>110: Setting prohibited</li> <li>111: SSISCK4/SSIWS4 = serial bit clock or word select signal of SSI_CH3 are output.</li> </ul> </li> <li>In SSIDMCOR5:               <ul style="list-style-type: none"> <li>000: SSISCK5/SSIWS5 = serial bit clock or word select signal of SSI_CH5 are output.</li> <li>001: Setting prohibited</li> <li>010: SSISCK5/SSIWS5 = serial bit clock or word select signal of SSI_CH0 are output.</li> <li>011: SSISCK5/SSIWS5 = serial bit clock or word select signal of SSI_CH1 are output.</li> <li>100: SSISCK5/SSIWS5 = serial bit clock or word select signal of SSI_CH2 are output.</li> <li>101: Setting prohibited</li> <li>110: SSISCK5/SSIWS5 = serial bit clock or word select signal of SSI_CH3 are output.</li> <li>111: SSISCK5/SSIWS5 = serial bit clock or word select signal of SSI_CH4 are output.</li> </ul> </li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
9 to 7	SCKSIP [2:0]	000	R/W	<p>Serial Bit Clock and Serial Word Select Signal Output Selection</p> <p>These bits select the serial bit clock and serial word select signal to be input from external devices to SSI_CH0 through SSI_CH5.</p> <ul style="list-style-type: none"> <li>In SSIDMCOR0:               <ul style="list-style-type: none"> <li>000: The serial bit clock or word select signal of SSI_CH0 = SSISCK0/SSIWS0 are input.</li> <li>001: The serial bit clock or word select signal of SSI_CH0 = SSISCK1/SSIWS1 are input.</li> <li>010: The serial bit clock or word select signal of SSI_CH0 = SSISCK2/SSIWS2 are input.</li> <li>011: Setting prohibited</li> <li>100: The serial bit clock and word select signal of SSI_CH0 = SSISCK3/SSIWS3 are input.</li> <li>101: The serial bit clock or word select signal of SSI_CH0 = SSISCK4/SSIWS4 are input.</li> <li>110: The serial bit clock or word select signal of SSI_CH0 = SSISCK5/SSIWS5 are input.</li> <li>111: Setting prohibited</li> </ul> </li> <li>In SSIDMCOR1:               <ul style="list-style-type: none"> <li>000: The serial bit clock or word select signal of SSI_CH1 = SSISCK1/SSIWS1 are input.</li> <li>001: The serial bit clock or word select signal of SSI_CH1 = SSISCK2/SSIWS2 are input.</li> <li>010: Setting prohibited</li> <li>011: The serial bit clock or word select signal of SSI_CH1 = SSISCK3/SSIWS3 are input.</li> <li>100: The serial bit clock or word select signal of SSI_CH1 = SSISCK4/SSIWS4 are input.</li> <li>101: The serial bit clock or word select signal of SSI_CH1 = SSISCK5/SSIWS5 are input.</li> <li>110: Setting prohibited</li> <li>111: The serial bit clock or word select signal of SSI_CH1 = SSISCK0/SSIWS0 are input.</li> </ul> </li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
9 to 7	SCKSIP [2:0]	000	R/W	<ul style="list-style-type: none"> <li>In SSIDMCOR2:               <ul style="list-style-type: none"> <li>000: The serial bit clock or word select signal of SSI_CH2 = SSISCK2/SSIWS2 are input.</li> <li>001: Setting prohibited</li> <li>010: The serial bit clock or word select signal of SSI_CH2 = SSISCK3/SSIWS3 are input.</li> <li>011: The serial bit clock or word select signal of SSI_CH2 = SSISCK4/SSIWS4 are input.</li> <li>100: The serial bit clock or word select signal of SSI_CH2 = SSISCK5/SSIWS5 are input.</li> <li>101: Setting prohibited</li> <li>110: The serial bit clock or word select signal of SSI_CH2 = SSISCK0/SSIWS0 are input.</li> <li>111: The serial bit clock or word select signal of SSI_CH2 = SSISCK1/SSIWS1 are input.</li> </ul> </li> <li>In SSIDMCOR3:               <ul style="list-style-type: none"> <li>000: The serial bit clock or word select signal of SSI_CH3 = SSISCK3/SSIWS3 are input.</li> <li>001: The serial bit clock or word select signal of SSI_CH3 = SSISCK4/SSIWS4 are input.</li> <li>010: The serial bit clock or word select signal of SSI_CH3 = SSISCK5/SSIWS5 are input.</li> <li>011: Setting prohibited</li> <li>100: The serial bit clock or word select signal of SSI_CH3 = SSISCK0/SSIWS0 are input.</li> <li>101: The serial bit clock or word select signal of SSI_CH3 = SSISCK1/SSIWS1 are input.</li> <li>110: The serial bit clock or word select signal of SSI_CH3 = SSISCK2/SSIWS2 are input.</li> <li>111: Setting prohibited</li> </ul> </li> </ul>



Bit	Bit Name	Initial Value	R/W	Description
9 to 7	SCKSIP [2:0]	000	R/W	<ul style="list-style-type: none"> <li>In SSIDMCOR4:               <ul style="list-style-type: none"> <li>000: The serial bit clock or word select signal of SSI_CH4 = SSISCK4/SSIWS4 are input.</li> <li>001: The serial bit clock or word select signal of SSI_CH4 = SSISCK5/SSIWS5 are input.</li> <li>010: Setting prohibited</li> <li>011: The serial bit clock or word select signal of SSI_CH4 = SSISCK0/SSIWS0 are input.</li> <li>100: The serial bit clock or word select signal of SSI_CH4 = SSISCK1/SSIWS1 are input.</li> <li>101: The serial bit clock or word select signal of SSI_CH4 = SSISCK2/SSIWS2 are input.</li> <li>110: Setting prohibited</li> <li>111: The serial bit clock or word select signal of SSI_CH4 = SSISCK3/SSIWS3 are input.</li> </ul> </li> <li>In SSIDMCOR5:               <ul style="list-style-type: none"> <li>000: The serial bit clock or word select signal of SSI_CH5 = SSISCK5/SSIWS5 are input.</li> <li>001: Setting prohibited</li> <li>010: The serial bit clock or word select signal of SSI_CH5 = SSISCK0/SSIWS0 are input.</li> <li>011: The serial bit clock or word select signal of SSI_CH5 = SSISCK1/SSIWS1 are input.</li> <li>100: The serial bit clock or word select signal of SSI_CH5 = SSISCK2/SSIWS2 are input.</li> <li>101: Setting prohibited</li> <li>110: The serial bit clock or word select signal of SSI_CH5 = SSISCK3/SSIWS3 are input.</li> <li>111: The serial bit clock or word select signal of SSI_CH5 = SSISCK4/SSIWS4 are input.</li> </ul> </li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	SCKS[2:0]	000	R/W	<p>Audio Clock Selection</p> <p>These bits select the audio clock to be input from external devices to SSI_CH0 to SSI_CH5.</p> <ul style="list-style-type: none"> <li>In SSIDMCOR0:               <ul style="list-style-type: none"> <li>000: The SSI_CH0 audio clock = AUDIO_CLK0 is input.</li> <li>001: The SSI_CH0 audio clock = AUDIO_CLK1 is input.</li> <li>010: The SSI_CH0 audio clock = AUDIO_CLK2 is input.</li> <li>011: Setting prohibited</li> <li>100: The SSI_CH0 audio clock = AUDIO_CLK3 is input.</li> <li>101: The SSI_CH0 audio clock = AUDIO_CLK4 is input.</li> <li>110: The SSI_CH0 audio clock = AUDIO_CLK5 is input.</li> <li>111: Setting prohibited</li> </ul> </li> <li>In SSIDMCOR1:               <ul style="list-style-type: none"> <li>000: The SSI_CH1 audio clock = AUDIO_CLK1 is input.</li> <li>001: The SSI_CH1 audio clock = AUDIO_CLK2 is input.</li> <li>010: Setting prohibited</li> <li>011: The SSI_CH1 audio clock = AUDIO_CLK3 is input.</li> <li>100: The SSI_CH1 audio clock = AUDIO_CLK4 is input.</li> <li>101: The SSI_CH1 audio clock = AUDIO_CLK5 is input.</li> <li>110: Setting prohibited</li> <li>111: The SSI_CH1 audio clock = AUDIO_CLK0 is input.</li> </ul> </li> </ul>

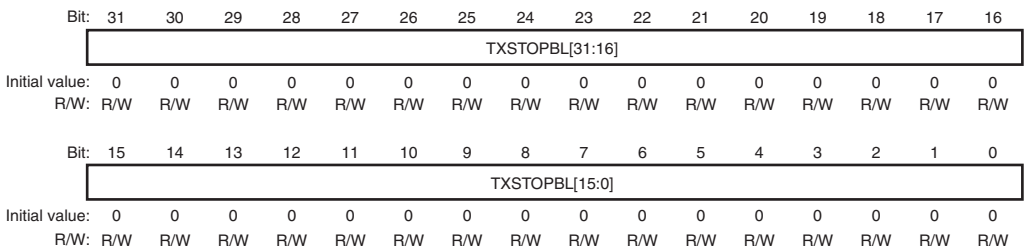
Bit	Bit Name	Initial Value	R/W	Description
6 to 4	SCKS[2:0]	000	R/W	<ul style="list-style-type: none"> <li>In SSIDMCOR2:               <ul style="list-style-type: none"> <li>000: The SSI_CH2 audio clock = AUDIO_CLK2 is input.</li> <li>001: Setting prohibited</li> <li>010: The SSI_CH2 audio clock = AUDIO_CLK3 is input.</li> <li>011: The SSI_CH2 audio clock = AUDIO_CLK4 is input.</li> <li>100: The SSI_CH2 audio clock = AUDIO_CLK5 is input.</li> <li>101: Setting prohibited</li> <li>110: The SSI_CH2 audio clock = AUDIO_CLK0 is input.</li> <li>111: The SSI_CH2 audio clock = AUDIO_CLK1 is input.</li> </ul> </li> <li>In SSIDMCOR3:               <ul style="list-style-type: none"> <li>000: The SSI_CH3 audio clock = AUDIO_CLK3 is input.</li> <li>001: The SSI_CH3 audio clock = AUDIO_CLK4 is input.</li> <li>010: The SSI_CH3 audio clock = AUDIO_CLK5 is input.</li> <li>011: Setting prohibited</li> <li>100: The SSI_CH3 audio clock = AUDIO_CLK0 is input.</li> <li>101: The SSI_CH3 audio clock = AUDIO_CLK1 is input.</li> <li>110: The SSI_CH3 audio clock = AUDIO_CLK2 is input.</li> <li>111: Setting prohibited</li> </ul> </li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	SCKS[2:0]	000	R/W	<ul style="list-style-type: none"> <li>In SSIDMCOR4:               <ul style="list-style-type: none"> <li>000: The SSI_CH4 audio clock = AUDIO_CLK4 is input.</li> <li>001: The SSI_CH4 audio clock = AUDIO_CLK5 is input.</li> <li>010: Setting prohibited</li> <li>011: The SSI_CH4 audio clock = AUDIO_CLK0 is input.</li> <li>100: The SSI_CH4 audio clock = AUDIO_CLK1 is input.</li> <li>101: The SSI_CH4 audio clock = AUDIO_CLK2 is input.</li> <li>110: Setting prohibited</li> <li>111: The SSI_CH4 audio clock = AUDIO_CLK3 is input.</li> </ul> </li> <li>In SSIDMCOR5:               <ul style="list-style-type: none"> <li>000: The SSI_CH5 audio clock = AUDIO_CLK5 is input.</li> <li>001: Setting prohibited</li> <li>010: The SSI_CH5 audio clock = AUDIO_CLK0 is input.</li> <li>011: The SSI_CH5 audio clock = AUDIO_CLK1 is input.</li> <li>100: The SSI_CH5 audio clock = AUDIO_CLK2 is input.</li> <li>101: Setting prohibited</li> <li>110: The SSI_CH5 audio clock = AUDIO_CLK3 is input.</li> <li>111: The SSI_CH5 audio clock = AUDIO_CLK4 is input.</li> </ul> </li> </ul>
3	—	0	R	Reserved This bit is always read as an undefined value. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	RPTMD	0	R/W	Repeat Mode Selects either normal mode or repeat mode. 0: Normal mode 1: Repeat mode
1	TRMD	0	R/W	Transmission/Reception Mode Selection Selects either transmission mode or reception mode of SSI_DMACH0/1 of corresponding SSI channel from 0 to 5 (SSI_CH0 to SSI_CH5). 0: Reception mode 1: Transmission mode
0	DMEN	0	R/W	SSI-DMAC Enable Enables or disables the SSI_DMACH0/1 operation of corresponding SSI channel from 0 to 5 (SSI_CH0 to SSI_CH5). 0: The SSI_DMACH0/1 operation is disabled. 1: The SSI_DMACH0/1 operation is enabled.

### 18.3.7 Transmit Suspension Block Counters 0 to 5 (SSISTPBLCNT0 to SSISTPBLCNT5)

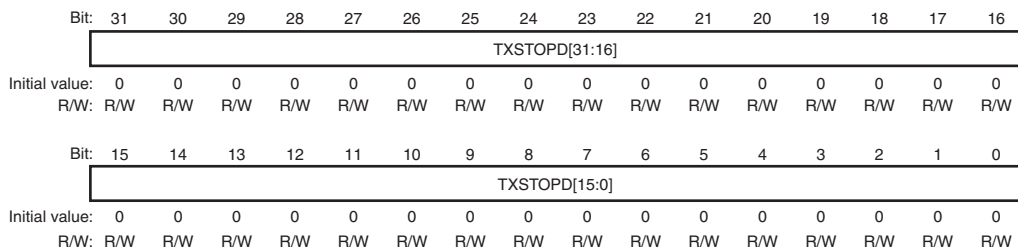
SSISTPBLCNT0 to SSISTPBLCNT5 is a 32-bit readable/writable register that set the number of blocks to be transferred after the TXSTOP0 to TXSTOP5 bits in SSIDMAOR0 and SSIDMAOR1 are set to 1 until the transmit suspension state is entered. This register is decremented after each one block transfer and enters transmit suspension state when it is decremented to 0. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMACH (DMRST bit in SSIDMACOR0 to SSIDMACOR3). To be written to this register, DMEN bit in SSIDMCOR0 to SSIDMCOR5 must be 0.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TXSTOPBL [31:0]	All 0	R/W	Transfer Block Count till Transmit Suspension State  These bits set the number of blocks to be transferred after the TXSTOP0 to TXSTOP5 bits in SSIDMAOR0 and SSIDMAOR1 are set to 1 until the transmit suspension state is entered. Setting value is readable during reading. (enable to read counter value)

### 18.3.8 Transmit Suspension Transfer Data Registers 0 to 5 (SSISTPDR0 to SSISTPDR5)

SSISTPDR0 to SSISTPDR5 is a 32-bit readable/writable register that set data to be transferred to SSI\_CH0 through SSI\_CH5 during transmit suspension state other than the port function. In transmit suspension state, data set in SSISTPDR0 to SSISTPDR5 instead of transmit FIFO buffers is sent to SSI\_CH0 to SSI\_CH5. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAC (DMRST bit in SSIDMACOR0 to SSIDMACOR3). To be written to this register, DMEN bit in SSIDMCOR0 to SSIDMCOR5 must be 0.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TXSTOPD [31:0]	All 0	R/W	Transfer Data during Transmit Suspension State  These bits set data to be transferred to SSI_CH0 through SSI_CH5 during transmit suspension state.

### 18.3.9 Block Count Source Registers 0 to 5 (SSIBLCNTR0 to SSIBLCNTR5)

SSIBLCNTR0 to SSIBLCNTR5 is a 32-bit readable/writable register that set the transfer byte count as SSIBLCNT0 to SSIBLCNT5 increment timing other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAC (DMRST bit in SSIDMACOR0 to SSIDMACOR3). To be written to this register, DMEN bit in SSIDMCOR0 to SSIDMCOR5 must be 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BLCNTR[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
	BLCNTR[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	BLCNTR [31:0]	All 0	R/W	<p>SSIBLCNT0 to SSIBLCNT5 Increment Timing</p> <p>These bits set the transfer byte count as SSIBLCNT0 to SSIBLCNT5 increment timing.</p> <p>The following transfer count should be selected according to the RDMA or WDMA maximum burst size.</p> <p>1 burst: <math>8 \times n</math> [<math>H'08 \times n</math>] (bytes)</p> <p>2 bursts: <math>16 \times n</math> [<math>H'10 \times n</math>] (bytes)</p> <p>4 bursts: <math>32 \times n</math> [<math>H'20 \times n</math>] (bytes)</p>

### 18.3.10 Block Counters 0 to 5 (SSIBLCNT0 to SSIBLCNT5)

SSIBLCNT0 to SSIBLCNT5 is 32-bit readable/writable register that indicates the transfer block count in which the number of bytes specified by SSIBLCNTR0 to SSIBLCNTR5 is handled as one block other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAL (DMRST bit in SSIDMACOR0 to SSIDMACOR3).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BLCNT[31:16]															
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
	BLCNT[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 0	BLCNT [31:0]	All 0	R	Transfer Block Count  These bits indicate the transfer block count in which the number of bytes specified by SSIBLCNTR0 to SSIBLCNTR5 is handled as one block.

### 18.3.11 n-Times Block Transfer Interrupt Count Source Registers 0 to 5 (SSIBLNCNTR0 to SSIBLNCNTR5)

SSIBLNCNTR0 to SSIBLNCNTR5 is a 32-bit readable/writable register that set the transfer byte count as n-times block transfer interrupting generation timing and SSIBLNCNT0 to SSIBLNCNT5 increment timing other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAL (DMRST bit in SSIDMACOR0 to SSIDMACOR3). To be written to this register, DMEN bit in SSIDMCOR0 to SSIDMCOR5 must be 0.



Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BLNCNTR[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
	BLNCNTR[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	BLNCNTR [31:0]	All 0	R/W	n-Times Block Transfer Interrupt Generation Timing These bits set the transfer block count as n-times block transfer interrupt generation timing and SSIBLNCNT0 to SSIBLNCNT5 increment timing.

### 18.3.12 n-Times Block Counters 0 to 5 (SSIBLNCNT0 to SSIBLNCNT5)

SSIBLNCNT0 to SSIBLNCNT5 is a 32-bit readable/writable register that indicates the value incremented for each block count specified by SSIBLNCNTR0 to SSIBLNCNTR5 other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAR (DMRST bit in SSIDMACOR0 to SSIDMACOR3).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BLNCNT[31:16]															
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
	BLNCNT[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 0	BLNCNT [31:0]	All 0	R	n-Times Transfer Block Count These bits indicate the value incremented for each block count specified by SSIBLNCNTR0 to SSIBLNCNTR5.

### 18.3.13 DMA Operation Registers 0 and 1 (SSIDMAOR0 and SSIDMAOR1)

SSIDMAOR0 and SSIDMAOR1 are 32-bit readable/writable registers that set the priority of transmit suspension channels and endian of transmit and receive data other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAC (DMRST bit in SSIDMACOR0 to SSIDMACOR3).

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
	—	—	—	—	—	—	—	—	—	TX STOP2	TX STOP1	TX STOP0	—	—	—	PR[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/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
6	TXSTOP2 (TXSTOP5)	0	R/W	SSI_CH2 (CH5) Transmit Suspension  Temporarily stops the data transfer from SSI_CH2 (CH5) transmit FIFO buffer to SSI_CH2 (CH5) and transfers data set in SSITXSTPDR2 (5) to SSI_CH2 (CH5).  0: SSI_CH2 (CH5) performs normal operation. Data is transferred from SSI_CH2 (CH5) transmit FIFO buffer to SSI_CH2 (CH5).  1: SSI_CH2 (CH5) temporarily stops data transfer. Data is transferred from SSITXSTPDR2 (5) to SSI_CH2 (CH5).

Bit	Bit Name	Initial Value	R/W	Description
5	TXSTOP1 (TXSTOP4)	0	R/W	<p>SSI_CH1 (CH4) Transmit Suspension</p> <p>Temporarily stops the data transfer from SSI_CH1 (CH4) transmit FIFO buffer to SSI_CH1 (CH4) and transfers data set in SSITXSTPDR1 (4) to SSI_CH1 (CH4).</p> <p>0: SSI_CH1 (CH4) performs normal operation. Data is transferred from SSI_CH1 (CH4) transmit FIFO buffer to SSI_CH1 (CH4).</p> <p>1: SSI_CH1 (CH4) temporarily stops data transfer. Data is transferred from SSITXSTPDR1 (4) to SSI_CH1 (CH4).</p>
4	TXSTOP0 (TXSTOP3)	0	R/W	<p>SSI_CH0 (CH3) Transmit Suspension</p> <p>Temporarily stops the data transfer from SSI_CH0 (CH3) transmit FIFO buffer to SSI_CH0 (CH3) and transfers data set in SSITXSTPDR0 (3) to SSI_CH0 (CH3).</p> <p>0: SSI_CH0 (CH3) performs normal operation. Data is transferred from SSI_CH0 (CH3) transmit FIFO buffer to SSI_CH0 (CH3).</p> <p>1: SSI_CH0 (CH3) temporarily stops data transfer. Data is transferred from SSITXSTPDR0 (3) to SSI_CH0 (CH3).</p>
3, 2	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PR[1:0]	All 0	R/W	<p>Priority Modes 1, 0</p> <p>These bits set the priority when multiple transfer requests from SSIs (SSI_CH0 to SSI_CH5) occur simultaneously.</p> <p>00: SSI_CH0 &gt; SSI_CH1 &gt; SSI_CH2 (SSI_CH3 &gt; SSI_CH4 &gt; SSI_CH5)</p> <p>01: SSI_CH0 &gt; SSI_CH2 &gt; SSI_CH1 (SSI_CH3 &gt; SSI_CH5 &gt; SSI_CH4)</p> <p>10: Setting prohibited</p> <p>11: Round robin of SSI_CH0 to SSI_CH2 (Round robin of SSI_CH3 to SSI_CH5)</p>

Note: Descriptions within parenthesis "( )" indicate those for SSIDMAOR1.

### 18.3.14 Interrupt Status Registers 0 and 1 (SSIDMINTSR0 and SSIDMINTSR1)

SSIDMINTSR0 and SSIDMINTSR1 are 32-bit readable/writable registers that indicate the interrupt sources of SSI\_DMAC0/1. Interrupts enabled by the interrupt mask registers (SSIDMINTMR0 and SSIDMINTMR1) will occur other than the port function. Each bit in SSIDMINTSR0 and SSIDMINTSR1 is cleared to 0 by writing 1 to it. Writing 0 to it is ignored. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMAC (DMRST bit in SSIDMACOR0 to SSIDMACOR3).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	BLK END2	BLKN END2	DM END2	TXFIFO FUL2	RXFIFO EMP2
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
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
	—	—	—	BLK END1	BLKN END1	DM END1	TXFIFO FUL1	RXFIFO EMP1	—	—	—	BLK END0	BLKN END0	DM END0	TXFIFO FUL0	RXFIFO EMP0
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
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	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	—	1	R	Reserved This bit is always read as 1. The write value should always be 0.
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BLKEND2 (BLKEND5)	0	R/W	Block Transfer End 2 (5) Indicates that data transfer whose byte count is specified by SSIBLCNTR2 (5) has been completed. 0: Indicates that data transfer whose byte count is specified by SSIBLCNTR2 (5) has not been completed. 1: Indicates that data transfer whose byte count is specified by SSIBLCNTR2 (5) has been completed.
19	BLKNEND2 (BLKNEND5)	0	R/W	n-Times Block Transfer End 2 (5) Indicates that data transfer whose block count is specified by SSIBLNCNTR2 (5) has been completed. 0: Indicates that data transfer whose block count is specified by SSIBLNCNTR2 (5) has not been completed. 1: Indicates that data transfer whose block count is specified by SSIBLNCNTR2 (5) has been completed.
18	DMEND2 (DMEND5)	0	R/W	Transfer End 2 (5) Indicates that data transfer whose word count is specified by SSIRDMCNTR2 (5) or SSIWDMCNTR2 (5) has been completed. 0: Indicates that data transfer whose word count is specified by SSIRDMCNTR2 (5) or SSIWDMCNTR2 (5) has not been completed. 1: Indicates that data transfer whose word count is specified by SSIRDMCNTR2 (5) or SSIWDMCNTR2 (5) has been completed.

Bit	Bit Name	Initial Value	R/W	Description
17	TXFIFOFUL2 (TXFIFOFUL5)	0	R/W	<p>Transmit FIFO Full 2 (5)</p> <p>Indicates that the transmit FIFO buffer for SSI_CH2 (CH5) is full.</p> <p>0: Indicates that the transmit FIFO buffer for SSI_CH2 (CH5) is not full.</p> <p>1: Indicates that the transmit FIFO buffer for SSI_CH2 (CH5) is full.</p>
16	RXFIFOEMP2 (RXFIFOEMP5)	1	R/W	<p>Receive FIFO Empty 2 (5)</p> <p>Indicates that the receive FIFO buffer for SSI_CH2 (CH5) is empty.</p> <p>0: Indicates that the receive FIFO buffer for SSI_CH2 (CH5) is not empty.</p> <p>1: Indicates that the receive FIFO buffer for SSI_CH2 (CH5) is empty.</p>
15 to 13	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
12	BLKEND1 (BLKEND4)	0	R/W	<p>Block Transfer End 1 (4)</p> <p>Indicates that data transfer whose byte count is specified by SSIBLCNTR1 (4) has been completed.</p> <p>0: Indicates that data transfer whose byte count is specified by SSIBLCNTR1 (4) has not been completed.</p> <p>1: Indicates that data transfer whose byte count is specified by SSIBLCNTR1 (4) has been completed.</p>

Bit	Bit Name	Initial Value	R/W	Description
11	BLKNEND1 (BLKNEND4)	0	R/W	<p>n-Times Block Transfer End 1 (4)</p> <p>Indicates that data transfer whose block count is specified by SSIBLNCNTR1 (4) has been completed.</p> <p>0: Indicates that data transfer whose block count is specified by SSIBLNCNTR1 (4) has not been completed.</p> <p>1: Indicates that data transfer whose block count is specified by SSIBLNCNTR1 (4) has been completed.</p>
10	DMEND1 (DMEND4)	0	R/W	<p>Transfer End 1 (4)</p> <p>Indicates that data transfer whose word count is specified by SSIRDMCNTR1 (4) or SSIWDMCNTR1 (4) has been completed.</p> <p>0: Indicates that data transfer whose word count is specified by SSIRDMCNTR1 (4) or SSIWDMCNTR1 (4) has not been completed.</p> <p>1: Indicates that data transfer whose word count is specified by SSIRDMCNTR1 (4) or SSIWDMCNTR1 (4) has been completed.</p>
9	TXFIFOFUL1 (TXFIFOFUL4)	0	R/W	<p>Transmit FIFO Full 1 (4)</p> <p>Indicates that the transmit FIFO buffer for SSI_CH1 (CH4) is full.</p> <p>0: Indicates that the transmit FIFO buffer for SSI_CH1 (CH4) is not full.</p> <p>1: Indicates that the transmit FIFO buffer for SSI_CH1 (CH4) is full.</p>
8	RXFIFOEMP1 (RXFIFOEMP4)	1	R/W	<p>Receive FIFO Empty 1 (4)</p> <p>Indicates that the receive FIFO buffer for SSI_CH1 (CH4) is empty.</p> <p>0: Indicates that the receive FIFO buffer for SSI_CH1 (CH4) is not empty.</p> <p>1: Indicates that the receive FIFO buffer for SSI_CH1 (CH4) is empty.</p>
7 to 5	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
4	BLKEND0 (BLKEND3)	0	R/W	<p>Block Transfer End 0 (3)</p> <p>Indicates that data transfer whose byte count is specified by SSIBLCNTR0 (3) has been completed.</p> <p>0: Indicates that data transfer whose byte count is specified by SSIBLCNTR0 (3) has not been completed.</p> <p>1: Indicates that data transfer whose byte count is specified by SSIBLCNTR0 (3) has been completed.</p>
3	BLKNEND0 (BLKNEND3)	0	R/W	<p>n-Times Block Transfer End 0 (3)</p> <p>Indicates that data transfer whose block count is specified by SSIBLNCNTR0 (3) has been completed.</p> <p>0: Indicates that data transfer whose block count is specified by SSIBLNCNTR0 (3) has not been completed.</p> <p>1: Indicates that data transfer whose block count is specified by SSIBLNCNTR0 (3) has been completed.</p>
2	DMEND0 (DMEND3)	0	R/W	<p>Transfer End 0 (3)</p> <p>Indicates that data transfer whose word count is specified by SSIRDMCNTR0 (3) or SSIWDMCNTR0 (3) has been completed.</p> <p>0: Indicates that data transfer whose word count is specified by SSIRDMCNTR0 (3) or SSIWDMCNTR0 (3) has not been completed.</p> <p>1: Indicates that data transfer whose word count is specified by SSIRDMCNTR0 (3) or SSIWDMCNTR0 (3) has been completed.</p>
1	TXFIFOFUL0 (TXFIFOFUL3)	0	R/W	<p>Transmit FIFO Full 0 (3)</p> <p>Indicates that the transmit FIFO buffer for SSI_CH0 (CH3) is full.</p> <p>0: Indicates that the transmit FIFO buffer for SSI_CH0 (CH3) is not full.</p> <p>1: Indicates that the transmit FIFO buffer for SSI_CH0 (CH3) is full.</p>



Bit	Bit Name	Initial Value	R/W	Description
0	RXFIFOEMP0 (RXFIFOEMP3)	1	R/W	Receive FIFO Empty 0 (3)  Indicates that the receive FIFO buffer for SSI_CH0 (CH3) is empty.  0: Indicates that the receive FIFO buffer for SSI_CH0 (CH3) is not empty.  1: Indicates that the receive FIFO buffer for SSI_CH0 (CH3) is empty.

Note: Descriptions within parenthesis "(" ) indicate those for SSIDMINTSR1.

### 18.3.15 Interrupt Mask Registers 0 and 1 (SSIDMINTMR0 and SSIDMINTMR1)

SSIDMINTMR0 and SSIDMINTMR1 are 32-bit readable/writable registers that mask interrupts sources of SSI\_DMACH0/1 other than the port function. This register value is initialized when either of the conditions is implemented such as hardware reset, software reset or software reset for SSI\_DMACH (DMRST bit in SSIDMACOR0 to SSIDMACOR3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	BLKEND M2	BLKN ENDM2	DMEND M2	TXFIFO FULM2	RXFIFO EMPM2
Initial value:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1
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
	—	—	—	BLKEND M1	BLKN ENDM1	DM ENDM1	TXFIFO FULM1	RXFIFO EMPM1	—	—	—	BLKEND M0	BLKN ENDM0	DMEND M0	TXFIFO FULM0	RXFIFO EMPM0
Initial value:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1
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	R/W	Description
31 to 29	—	All 0	R	Reserved  These bits are always read as 0. The write value should be the same as read value.
28 to 24	—	All 1	R	Reserved  These bits are always read as 1. The write value should be the same as read value.
23 to 21	—	All 0	R	Reserved  These bits are always read as 0. The write value should be the same as read value.

Bit	Bit Name	Initial Value	R/W	Description
20	BLKENDM2 (BLKENDM5)	1	R/W	BLKEND2 (5) Interrupt Source Mask Masks the BLKEND2 (5) interrupt source. 0: The BLKEND2 (5) interrupt source is not masked. 1: The BLKEND2 (5) interrupt source is masked.
19	BLKNENDM2 (BLKNENDM5)	1	R/W	BLKNEND2 (5) Interrupt Source Mask Masks the BLKNEND2 (5) interrupt source. 0: The BLKNEND2 (5) interrupt source is not masked. 1: The BLKNEND2 (5) interrupt source is masked.
18	DMENDM2 (DMENDM5)	1	R/W	DMEND2 (5) Interrupt Source Mask Masks the DMEND2 (5) interrupt source. 0: The DMEND2 (5) interrupt source is not masked. 1: The DMEND2 (5) interrupt source is masked.
17	TXFIFOFULM2 (TXFIFOFULM5)	1	R/W	TXFIFOFUL2 (5) Interrupt Source Mask Masks the TXFIFOFUL2 (5) interrupt source. 0: The TXFIFOFUL2 (5) interrupt source is not masked. 1: The TXFIFOFUL2 (5) interrupt source is masked.
16	RXFIFOEMP2 (RXFIFOEMP5)	1	R/W	RXFIFOEMP2 (5) Interrupt Source Mask Masks the RXFIFOEMP2 (5) interrupt source. 0: The RXFIFOEMP2 (5) interrupt source is not masked. 1: The RXFIFOEMP2 (5) interrupt source is masked.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	BLKENDM1 (BLKENDM4)	1	R/W	BLKEND1 (4) Interrupt Source Mask Masks the BLKEND1 (4) interrupt source. 0: The BLKEND1 (4) interrupt source is not masked. 1: The BLKEND1 (4) interrupt source is masked.
11	BLKNENDM1 (BLKNENDM4)	1	R/W	BLKNEND1 (4) Interrupt Source Mask Masks the BLKNEND1 (4) interrupt source. 0: The BLKNEND1 (4) interrupt source is not masked. 1: The BLKNEND1 (4) interrupt source is masked.

Bit	Bit Name	Initial Value	R/W	Description
10	DMENDM1 (DMENDM4)	1	R/W	DMEND1 (4) Interrupt Source Mask Masks the DMEND1 (4) interrupt source. 0: The DMEND1 (4) interrupt source is not masked. 1: The DMEND1 (4) interrupt source is masked.
9	TXFIFOFULM1 (TXFIFOFULM4)	1	R/W	TXFIFOFUL1 (4) Interrupt Source Mask Masks the TXFIFOFUL1 (4) interrupt source. 0: The TXFIFOFUL1 (4) interrupt source is not masked. 1: The TXFIFOFUL1 (4) interrupt source is masked.
8	RXFIFOEMP1 (RXFIFOEMP4)	1	R/W	RXFIFOEMP1 (4) Interrupt Source Mask Masks the RXFIFOEMP1 (4) interrupt source. 0: The RXFIFOEMP1 (4) interrupt source is not masked. 1: The RXFIFOEMP1 (4) interrupt source is masked.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	BLKENDM0 (BLKENDM3)	1	R/W	BLKEND0 (3) Interrupt Source Mask Masks the BLKEND0 (3) interrupt source. 0: The BLKEND0 (3) interrupt source is not masked. 1: The BLKEND0 (3) interrupt source is masked.
3	BLKNENDM0 (BLKNENDM3)	1	R/W	BLKNEND0 (3) Interrupt Source Mask Masks the BLKNEND0 (3) interrupt source. 0: The BLKNEND0 (3) interrupt source is not masked. 1: The BLKNEND0 (3) interrupt source is masked.
2	DMENDM0 (DMENDM3)	1	R/W	DMEND0 (3) Interrupt Source Mask Masks the DMEND0 (3) interrupt source. 0: The DMEND0 (3) interrupt source is not masked. 1: The DMEND0 (3) interrupt source is masked.

Bit	Bit Name	Initial Value	R/W	Description
1	TXFIFOFULM0 (TXFIFOFULM3)	1	R/W	TXFIFOFUL0 (3) Interrupt Source Mask Masks the TXFIFOFUL0 (3) interrupt source. 0: The TXFIFOFUL0 (3) interrupt source is not masked. 1: The TXFIFOFUL0 (3) interrupt source is masked.
0	RXFIFOEMPM0 (RXFIFOEMPM3)	1	R/W	RXFIFOEMP0 (3) Interrupt Source Mask Masks the RXFIFOEMP0 (3) interrupt source. 0: The RXFIFOEMP0 (3) interrupt source is not masked. 1: The RXFIFOEMP0 (3) interrupt source is masked.

Note: Descriptions within parenthesis "( )" indicate those for SSIDMINTMR1.

### 18.3.16 Control Registers 0 to 5 (SSICR0 to SSICR5)

SSICR0 to SSICR5 control interrupts, select each polarity status, and set operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN	CHNL[1:0]	DWL[2:0]			SWL[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/W	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	—	CKDV[2:0]			MUEN	—	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	R/W	R/W	R/W	R/W	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.
28	DMEN	0	R/W	DMA Enable Enables or disables a DMA request. 0: A DMA request is disabled. 1: A DMA request is enabled.
27	UIEN	0	R/W	Underflow Interrupt Enable Enables or disables an underflow interrupt. 0: An underflow interrupt is disabled. 1: An underflow interrupt is enabled.
26	OIEN	0	R/W	Overflow Interrupt Enable Enables or disables an overflow interrupt. 0: An overflow interrupt is disabled. 1: An overflow interrupt is enabled.
25	IEN	0	R/W	Idle Mode Interrupt Enable Enables or disables an idle mode interrupt. 0: An idle mode interrupt is disabled. 1: An idle mode interrupt is enabled.
24	DIEN	0	R/W	Data Interrupt Enable Enables or disables a data interrupt. 0: A data interrupt is disabled. 1: A data mode interrupt is enabled.
23, 22	CHNL[1:0]	00	R/W	Channel These bits indicate the number of channels in each system word. 00: 1 channel per system word 01: 2 channels per system word 10: 3 channels per system word 11: 4 channels per system word

Bit	Bit Name	Initial Value	R/W	Description
21 to 19	DWL[2:0]	000	R/W	Data Word Length These bits indicate the number of bits in a data word. 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 to 16	SWL[2:0]	000	R/W	System Word Length These bits indicate the number of bits in a system word. 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 clock input, slave mode 1: Serial clock output, master mode Note: Only (SCKD, SWSD) = (0, 0) or (1, 1) is enabled. Other settings are prohibited.
14	SWSD	0	R/W	Serial Word Selection Signal (WS) Direction 0: Serial word select input, slave mode 1: Serial word select output, master mode Note: Only (SCKD, SWSD) = (0, 0) or (1, 1) is enabled. Other settings are prohibited.

Bit	Bit Name	Initial Value	R/W	Description
13	SCKP	0	R/W	Serial Bit Clock Polarity
				0: SSIWS[5:0] and SSIDATA[5:0] change at the falling edge of SSISCK[5:0] (sampled at the rising edge of SSISCK[5:0]).
				1: SSIWS[5:0] and SSIDATA[5:0] change at the rising edge of SSISCK[5:0] (sampled at the falling edge of SSISCK[5:0]).
	</			

Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	<p>Parallel Data Alignment</p> <p>If the data word length is 32, 16 or 8 then this bit has no meaning.</p> <p>This bit is applied to SSIRDR0 to SSIRDR5 in receive mode and to SSITDR0 to SSITDR5 in transmit mode.</p> <p>0: Parallel data (SSITDR0 to SSITDR5 or SSIRDR0 to SSIRDR5) is left aligned.</p> <p>1: Parallel data (SSITDR0 to SSITDR5 or SSIRDR0 to SSIRDR5) is right aligned.</p> <ul style="list-style-type: none"> <li>DWL[2:0] = 000 (data word length: 8 bits), PDTA ignored</li> </ul> <p>All data bits in SSITDR0 to SSITDR5 or SSIRDR0 to SSIRDR5 are used on the audio serial bus. Four data words are transmitted or received in each 32-bit access. The first data word is stored in bits 7 to 0, the second in bits 15 to 8, the third in bits 23 to 16 and the last in bits 31 to 24.</p> <ul style="list-style-type: none"> <li>DWL[2:0] = 001 (data word length: 16 bits), PDTA ignored</li> </ul> <p>All data bits in SSITDR0 to SSITDR5 or SSIRDR0 to SSIRDR5 are used on the audio serial bus. Two data words are transmitted or received in each 32-bit access. The first and second data words are stored in bits 15 to 0 and bits 31 to 16, respectively.</p> <ul style="list-style-type: none"> <li>DWL[2:0] = 010, 011, 100, 101 (data word length: 18, 20, 22 and 24 bits), PDTA = 0 (left aligned)</li> </ul> <p>The data bits which are used in SSITDR0 to SSITDR5 or SSIRDR0 to SSIRDR5 are as follows:</p> <p>Bits 31 to (32 – number of bits having data word length specified by DWL[2:0]).</p> <p>If DWL[2:0] = 011, then data word length is 20 bits and bits 31 to 12 in SSITDR0 to SSITDR5 or SSIRDR0 to SSIRDR5 are used. All other bits are ignored or reserved.</p>



Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	<ul style="list-style-type: none"> <li>DWL[2:0] = 010, 011, 100, 101 (data word length: 18, 20, 22 and 24 bits), PDTA = 1 (right aligned) The data bits which are used in SSITDR0 to SSITDR5 or SSIRDR0 to SSIRDR5 are as follows: Bits (number of bits having data word length specified by DWL - 1) to 0. If DWL[2:0] = 011, then data word length is 20 bits and bits 19 to 0 in SSITDR0 to SSITDR5 or SSIRDR0 to SSIRDR5 are used. All other bits are ignored or reserved.</li> <li>DWL[2:0] = 110 (data word length: 32 bits), PDTA ignored All data bits in SSITDR0 to SSITDR5 or SSIRDR0 to SSIRDR5 are used on the audio serial bus</li> </ul>
8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSIWS[5:0] and SSIDATA[5:0] 1: No delay between SSIWS[5:0] and SSIDATA[5:0]
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	CKDV[2:0]	000	R/W	<p>Serial Oversampling Clock Division Ratio</p> <p>These bits define the division ratio between oversampling clock AUDIO_CKL[5:0] and the serial bit clock.</p> <p>These bits are ignored if SCKD = 0.</p> <p>The serial bit clock is used for the shift register and is provided from the SSISCK[5:0] pin.</p> <p>000: Serial bit clock frequency = oversampling clock frequency/1</p> <p>001: Serial bit clock frequency = oversampling clock frequency/2</p> <p>010: Serial bit clock frequency = oversampling clock frequency/4</p> <p>011: Serial bit clock frequency = oversampling clock frequency/8</p> <p>100: Serial bit clock frequency = oversampling clock frequency/16</p> <p>101: Serial bit clock frequency = oversampling clock frequency/6</p> <p>110: Serial bit clock frequency = oversampling clock frequency/12</p> <p>111: Setting prohibited</p>
3	MUEN	0	R/W	<p>Mute Enable</p> <p>0: SSI_CH0 to SSI_CH5 are not muted.</p> <p>1: SSI_CH0 to SSI_CH5 are muted.</p>
2	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
1	TRMD	0	R/W	<p>Transmit/Receive Mode Selection</p> <p>0: SSI_CH0 to SSI_CH5 are in receive mode.</p> <p>1: SSI_CH0 to SSI_CH5 are in transmit mode.</p>
0	EN	0	R/W	<p>Operation Enable</p> <p>0: SSI_CH0 to SSI_CH5 operation is disabled.</p> <p>1: SSI_CH0 to SSI_CH5 operation is enabled.</p>

### 18.3.17 Status Registers 0 to 5 (SSISR0 to SSISR5)

SSISR0 to SSISR5 are configured by status flags that indicate the operating status of SSI\_CH0 to SSI\_CH5 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	0	1	0	0	0	0	1	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
	—	—	—	—	—	—	—	—	—	—	—	—	CHNO[1:0]	SWNO	IDST	
Initial value:	1	0	1	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	R	R

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	DMRQ	0	R	DMA Request Status Flag This status flag allows the CPU to see the status of the DMA request from SSI_CH0 to SSI_CH5 modules. <ul style="list-style-type: none"> <li>TRMD = 0 (Receive mode) If DMRQ = 1, SSIRDR0 to SSIRDR5 have unread data. If SSIRDR0 to SSIRDR5 are read, then DMRQ = 0 until the SSIRDR0 to SSIRDR5 receive new unread data</li> <li>TRMD = 1 (Transmit mode) If DMRQ = 1, the SSIRDR0 to SSIRDR5 request data to be written to continue the data transmission on the audio serial bus. Once data is written to SSITDR0 to SSITDR5, then DMRQ = 0 until further transmit data is requested.</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
27	UIRQ	0	R/W*	<p>Underflow Error Interrupt Status Flag</p> <p>This status flag indicates that the data has been supplied at a lower rate than the required rate.</p> <p>This bit is set to 1 regardless of the setting of the UIEN bit. In order to clear it to 0, write 0 to it.</p> <p>If UIRQ = 1 and UIEN = 1, then an interrupt will be generated.</p> <ul style="list-style-type: none"><li>• TRMD = 0 (Receive mode) If UIRQ = 1, it indicates that SSIRDR0 to SSIRDR5 were 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 may cause destruction of multi-channel data.</li><li>• TRMD = 1 (Transmit mode) If UIRQ = 1, it indicates that the transmitted data was not written in SSITDR0 to SSITDR5. 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 receive mode.</li></ul> <p>Note: When underflow error occurs, the data in the data buffer will be transmitted until the next data is written in.</p>

---

Bit	Bit Name	Initial Value	R/W	Description
26	OIRQ	0	R/W*	<p>Overflow Error Interrupt Status Flag</p> <p>This status flag indicates that the data has been supplied at a higher rate than the required rate.</p> <p>This bit is set to 1 regardless of the setting of OIEN bit. In order to clear it to 0, write 0 to it.</p> <p>If OIRQ = 1 and OIEN = 1, then an interrupt will be generated.</p> <ul style="list-style-type: none"> <li>TRMD = 0 (Receive mode) <p>If OIRQ = 1, it indicates that the previous unread data had not been read out before new unread data was written in SSIRDR0 to SSIRDR5. This may cause the loss of data, which may cause destruction of multi-channel data.</p> <p>Note: If an overflow error occurs, the data in the data buffer will be overwritten by the next data sent from the SSI interface.</p> </li> <li>TRMD = 1 (Transmit mode) <p>If OIRQ = 1, it indicates that SSITDR0 to SSITDR5 had data written in before the data in SSITDR0 to SSITDR5 were transferred to the shift register. This may cause the loss of data, which can lead to destruction of multi-channel data.</p> </li> </ul>
25	IIRQ	1	R	<p>Idle Mode Interrupt Status Flag</p> <p>This status flag indicates whether the SSI_CH0 to SSI_CH5 are in the idle status. This bit is set to 1 regardless of the setting of ILEN bit, so that polling will be possible.</p> <p>The interrupt can be masked by clearing ILEN bit to 0, but writing 0 in this bit will not clear the interrupt.</p> <p>If IIRQ = 1 and ILEN = 1, then an interrupt will be generated.</p> <p>0: The SSI_CH0 to SSI_CH5 are not in the idle status. 1: The SSI_CH0 to SSI_CH5 are in the idle status.</p>

Bit	Bit Name	Initial Value	R/W	Description
24	DIRQ	0	R	<p>Data Interrupt Status Flag</p> <p>This status flag indicates that the SSI_CH0 to SSI_CH5 request data read or write.</p> <p>This bit is set to 1 regardless of the setting of DIEN bit, so that polling will be possible.</p> <p>The interrupt can be masked by clearing DIEN bit to 0, but writing 0 in this bit will not clear the interrupt.</p> <p>If DIRQ = 1 and DIEN = 1, then an interrupt will be generated.</p> <ul style="list-style-type: none"> <li>• TRMD = 0 (Receive mode)           <p>0: No unread data exists in SSIRDR0 to SSIRDR5.</p> <p>1: Unread data exists in SSIRDR0 to SSIRDR5.</p> </li> <li>• TRMD = 1 (Transmit mode)           <p>0: The transmit buffer is full.</p> <p>1: The transmit buffer is empty, and data write to SSITDR0 to SSITDR5 are requested.</p> </li> </ul>
23 to 4	—	H'10A00	R	<p>Reserved</p> <p>These bits are always read as H'10A00. The write value should always be 0.</p>
3, 2	CHNO[1:0]	00	R	<p>Channel Number</p> <p>These bits indicate the current channel number.</p> <ul style="list-style-type: none"> <li>• TRMD = 0 (Receive mode)           <p>These bits indicate to which channel the current data in SSIRDR0 to SSIRDR5 belongs. When the data in SSIRDR0 to SSIRDR5 is updated by transfer from the shift register, this value will change.</p> </li> <li>• TRMD = 1 (Transmit mode)           <p>These bits indicate the data of which channel should be written in SSITDR0 to SSITDR5. This bit value will change when data is copied to the shift register, regardless whether the data is written in SSITDR0 to SSITDR5.</p> </li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
1	SWNO	1	R	<p>Serial Word Number</p> <p>This bit indicates the current serial word number.</p> <ul style="list-style-type: none"> <li>• TRMD = 0 (Receive mode)</li> </ul> <p>This bit indicates the current serial word number of the current data in SSIRDR0 to SSIRDR5. This bit value will change, when the data in SSIRDR0 to SSIRDR5 is updated by transfer from the shift register, regardless whether the data has been read out from SSIRDR0 to SSIRDR5.</p> <li>• TRMD = 1 (Transmit mode)</li> <p>This bit indicates the current serial word number to be written to SSITDR0 to SSITDR5. This bit value will change when data is copied to the shift register, regardless whether the data is written in SSITDR0 to SSITDR5.</p>

Bit	Bit Name	Initial Value	R/W	Description
0	IDST	1	R	<p>Idle Mode Status Flag</p> <p>This bit indicates that the serial bus operation has been stopped.</p> <p>This bit is cleared if EN = 1 and the serial bus is currently active.</p> <p>This bit can be set to 1 automatically under the following conditions.</p> <ul style="list-style-type: none"> <li>SSI_CH0 to SSI_CH5 = Serial bus master transmitter (SWSD = 1 and TRMD = 1) This bit is set to 1 if the EN bit is cleared to 0 and data written in SSITDR0 to SSITDR5 has been output from serial data I/O pins (SSIDATA0 to SSIDATA5).</li> <li>SSI_CH0 to SSI_CH5 = 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.</li> <li>SSI_CH0 to SSI_CH5 = Slave transmitter/ receiver (SWSD = 0) This bit is set to 1 if the EN bit is cleared and the current system word is completed.</li> </ul> <p>Note: If the external device stops the serial bus clock before the current system word is completed, then this bit will never be set.</p>

Note: \* These bits are readable/writable bits. If writing 0, these bits are initialized, although writing 1 is ignored.



### 18.3.18 Transmit Data Registers 0 to 5 (SSITDR0 to SSITDR5)

SSITDR0 to SSITDR5 store data to be transmitted.

Data written to SSITDR0 to SSITDR5 is transferred to the shift register as it is required for transmission. If the data word length is less than 32 bits, data should be aligned according to the setting of the PDTA control bit in SSICR0 to SSICR5.

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

### 18.3.19 Receive Data Registers 0 to 5 (SSIRDR0 to SSIRDR5)

SSIRDR0 to SSIRDR5 store the received data.

Data in SSIRDR0 to SSIRDR5 is transferred from the shift register as each data word is received. If the data word length is less than 32 bits, data should be aligned according to the setting of the PDTA control bit in SSICR0 to SSICR5.

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

## 18.4 Operation

### 18.4.1 Operation of SSI\_CLKSEL

The SSI\_CLKSEL connects the audio clock and serial-bit clock or serial-word select signal, which are input from the SSI interface for each channel, to SSI\_CH0 through SSI\_CH5 depending on register settings.

- Selecting audio clock

The audio clock for SSI\_CH0 to SSI\_CH5 (AUDIO\_CLK[5:0]) is selected by the SSI\_CLKSEL according to the SCKS[2:0] bits in SSIDMCOR0 to SSIDMCOR5.

In the initial state, the audio clock of the same channel is selected, such as AUDIO\_CLK0 = SSI\_CH0.

- Selecting serial bit clock or serial word select signal

The serial bit clock or serial word select signal for SSI\_CH0 to SSI\_CH5 (SSISCK[5:0]/SSIWS[5:0]) is selected by the SSI\_CLKSEL according to the SCKSIP[2:0] or SCKOP[2:0] bits in SSIDMCOR0 to SSIDMCOR5.

In the initial state, the serial bit clock or serial word select signal of the same channel is selected, such as SSISCK0 = SSI\_CH0.

### 18.4.2 Operation of SSI\_DMAC0 and SSI\_DMAC1

The SSI\_DMAC0 and SSI\_DMAC1 perform data transfer between six SSI channels (SSI\_CH0 to SSI\_CH5) and external memory or on-chip memory.

The SSI\_DMAC0 and SSI\_DMAC1 provide the transmit and receive FIFO buffers (32 bits × 16 stages), which enable high-speed continuous communication effectively.

The SSI\_DMAC0 and SSI\_DMAC1 counts transferred data in any block units. This effectively controls interrupt generation or data transfer including data transfer suspension in block units.

During data transfer suspension, the sound can pause by sending arbitrary data (for example, silent data) continuously to SSI\_CH0 through SSI\_CH5.

- Number of channels  
Six channels corresponding to SSI\_CH0 to SSI\_CH5 transmitters and receivers
- Transferred data size  
8, 16 or 32 bytes
- Maximum transfer byte count  
4,294,967,296 bytes
- Bus mode  
Cycle-steal mode
- Priority order among channels  
Fixed-order or round-robin can be selected for SSI\_CH0 to SSI\_CH2 and SSI\_CH3 to SSI\_CH5.
- Transmit FIFO buffers and receive FIFO buffers  
Transmit and receive FIFO buffers (32 bits x 16 stages) are provided for SSI\_CH0 to SSI\_CH5.
- Interrupt requests  
Block transfer end interrupt  
n-times block transfer end interrupt  
Transfer end interrupt  
Transmit FIFO buffer full interrupt  
Receive FIFO buffer empty interrupt
- Software reset  
A software reset can be executed separately for SSI\_CH0 to SSI\_CH5. Each of transmit or receive FIFO buffer for SSI\_CH0 to SSI\_CH5 can be reset separately.
- Transmit suspension  
Immediate stop or stop after data transfer in block units can be selected. During transmit suspension, arbitrary data (for example, silent data) can be automatically transferred to SSI\_CH0 through SSI\_CH5.

### 18.4.3 Operation of SSI\_CH0 to SSI\_CH5

#### (1) Bus Format

SSI\_CH0 to SSI\_CH5 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 18.8.

**Table 18.8 Bus Formats of SSI Module**

Bus Format	TRMD	SCKD	SWSD	EN	MUEN	DIEN	IEN	OEN	UIEN	DEL	PDTA	SDTA	SPDP	SWSP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-Compressed Slave Receiver	0	0	0	Control bits						Configuration bits								
Non-Compressed Slave Transmitter	1	0	0															
Non-Compressed Master Receiver	0	1	1															
Non-Compressed Master Transmitter	1	1	1															

#### (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.

##### (a) 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.

**(b) 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.

**(c) 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 HAC\_BIT\_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.

**(d) 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 HAC\_BIT\_CLK input clock. The format of these signals is as defined in the configuration bits in the SSI module.

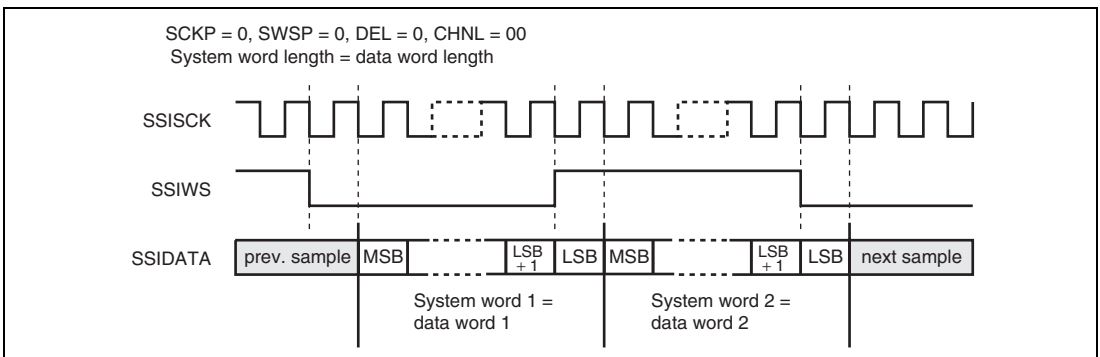
**(e) Configuration Fields - Word Length Related**

All configuration bits relating to the word length of SSICR are valid in non-compressed modes.

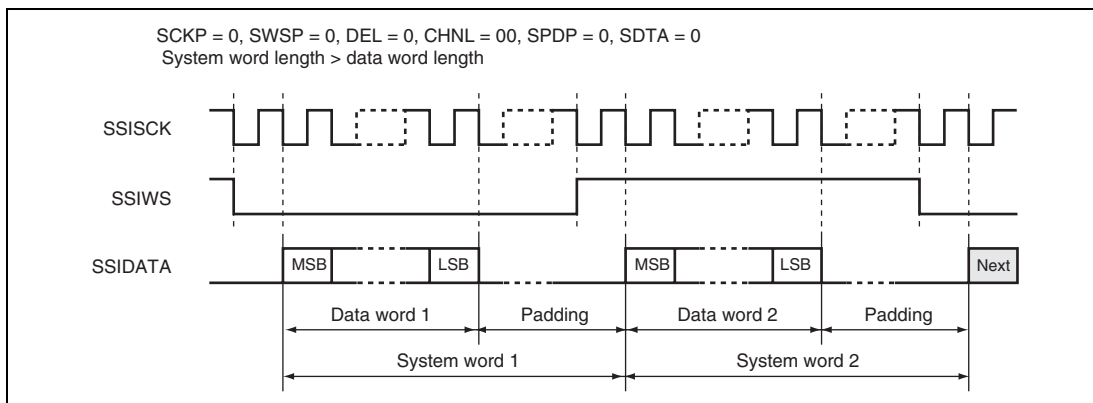
There are many configurations that the SSI module can support and it is not sensible to show all of the Serial Data formats in this document. Some of the combinations are shown below for the popular formats by Philips, Sony, and Matsushita.

- Philips Format**

Figures 18.2 and 18.3 show the supported Philips protocol both with padding and without. Padding occurs when the data word length is smaller than the system word length.



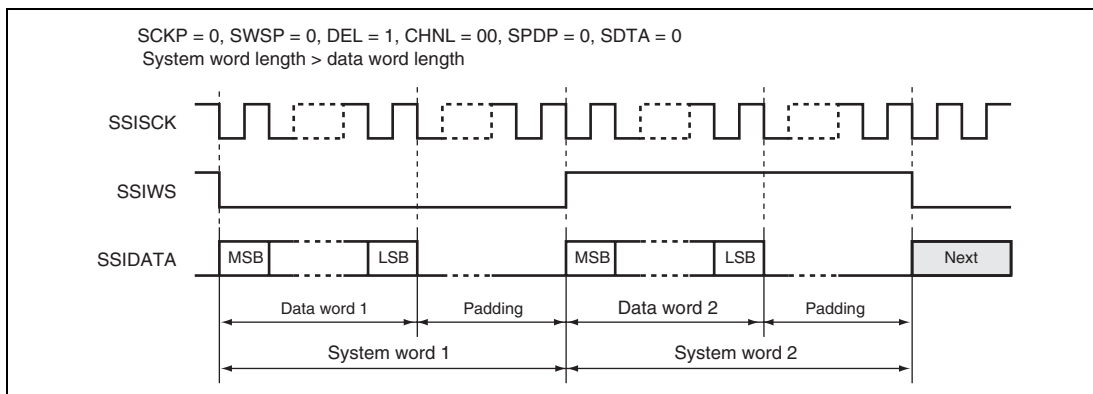
**Figure 18.2 Philips Format (with no Padding)**



**Figure 18.3 Philips Format (with Padding)**

- Sony Format

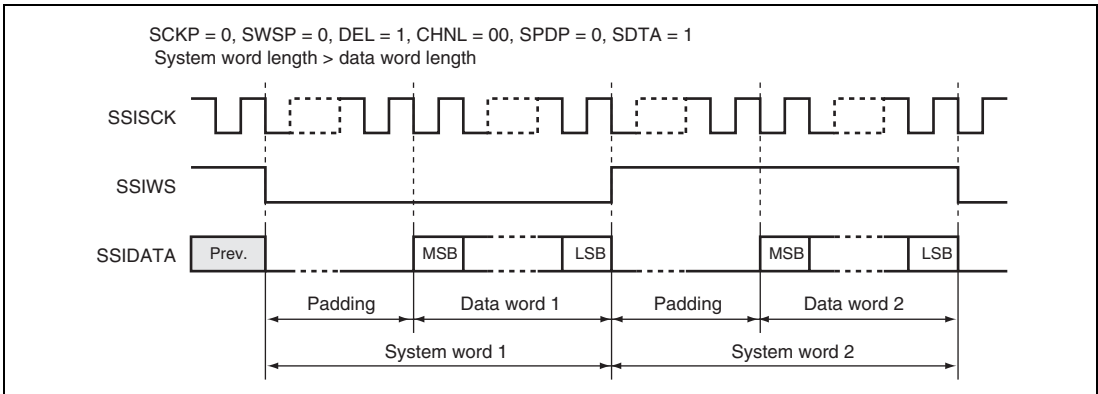
Figure 18.4 shows the format used by Sony. Padding is assumed, but may not be present if the system word length equals the data word length.



**Figure 18.4 Sony Format (with Serial Data First, Followed by Padding Bits)**

- Matsushita Format

Figure 18.5 shows the format used by Matsushita. Padding is assumed, but may not be present if the system word length equals the data word length.



**Figure 18.5 Matsushita Format (with Padding Bits First, Followed by Serial Data)**

**(f) Multi-Channel Formats**

Some devices extend the definition of the specification by Philips and allow more than 2 channels to be transferred within two system words.

SSI\_CH0 to SSI\_CH5 support the transfer of 2, 3 and 4 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 18.9 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 18.9 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

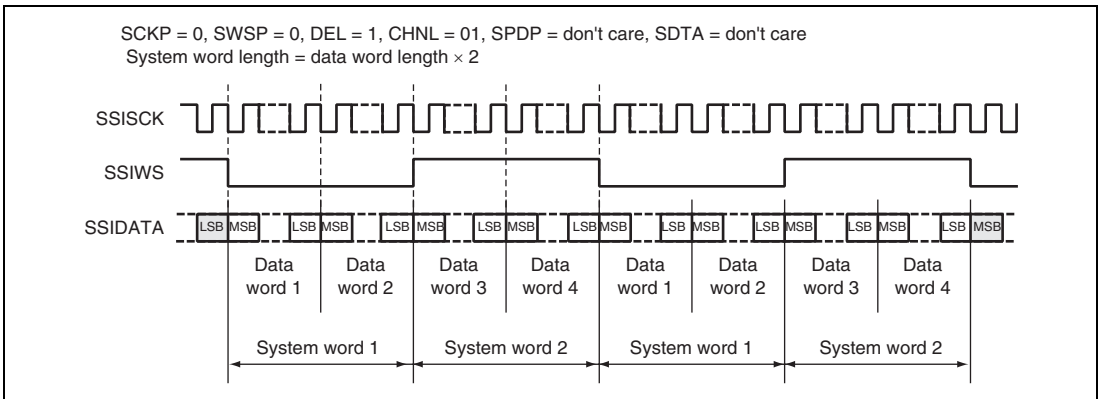


In the case of SSI\_CH0 to SSI\_CH5 configured as a transmitter then each word that is written to SSITDR0 to SSITDR5 is transmitted in order on the serial audio bus.

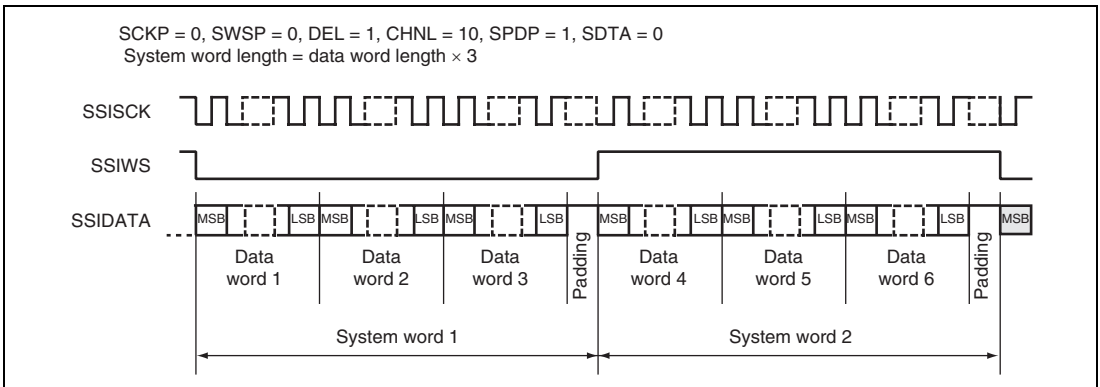
In the case of SSI\_CH0 to SSI\_CH5 configured as a receiver each word received on the Serial Audio Bus is presented for reading in order by SSIRDR0 to SSIRDR5.

Figures 18.6 to 18.8 show how 2, 3 and 4 channels are transferred on the serial audio bus.

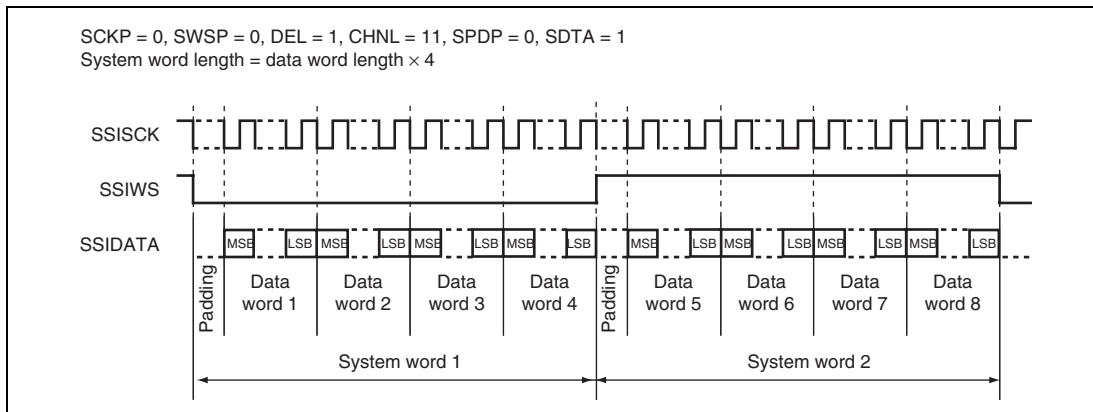
Note that there are no padding bits in the first example, serial data is transmitted/received first and followed by padding bits in the second example, and padding bits are transmitted/received first and followed by serial data in the third example. This selection is purely arbitrary.



**Figure 18.6 Multichannel Format (2 Channels, No Padding)**



**Figure 18.7 Multichannel Format (3 Channels with High Padding)**

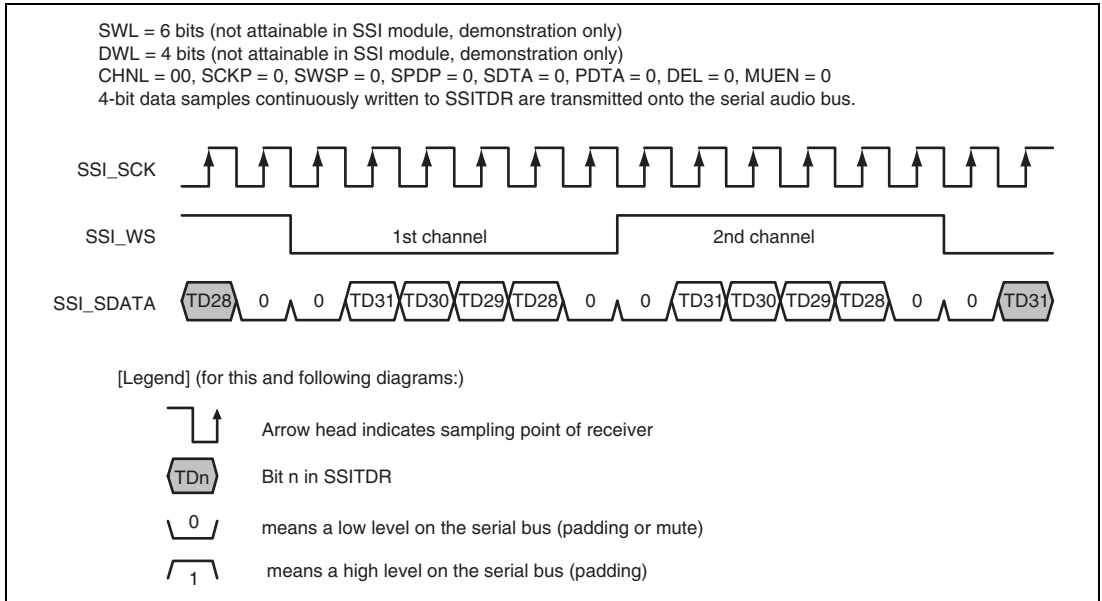


**Figure 18.8 Multichannel Format (4 Channels, with Padding Bits First, Followed by Serial Data, with Padding)**

**(g) 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 bit combinations will probably be prohibited.

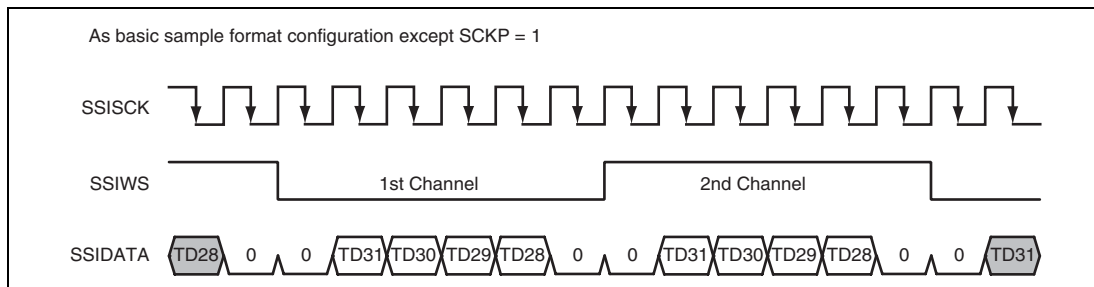
They are demonstrated by referring to the following basic sample format shown in figure 18.9.



**Figure 18.9 Basic Sample Format**  
**(Transmit Mode with Example System/Data Word Length)**

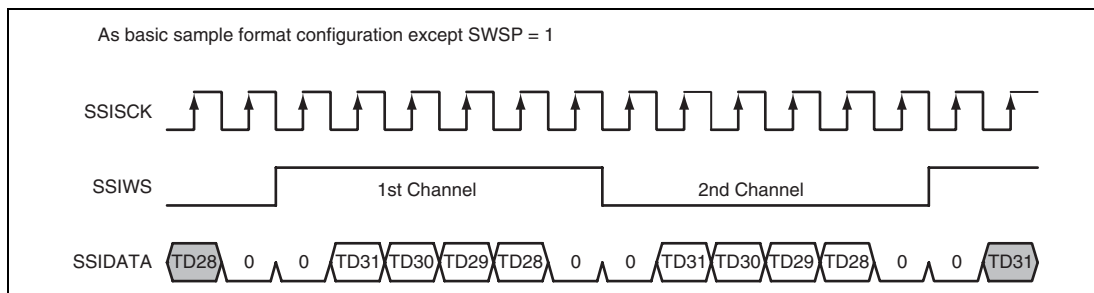
In figure 18.9, system word length of 6 bits and a data word length of 4 bits are used. Neither of these are possible with SSI\_CH0 to SSI\_CH5 but are used only for clarification of the other configuration bits.

- Inverted Clock



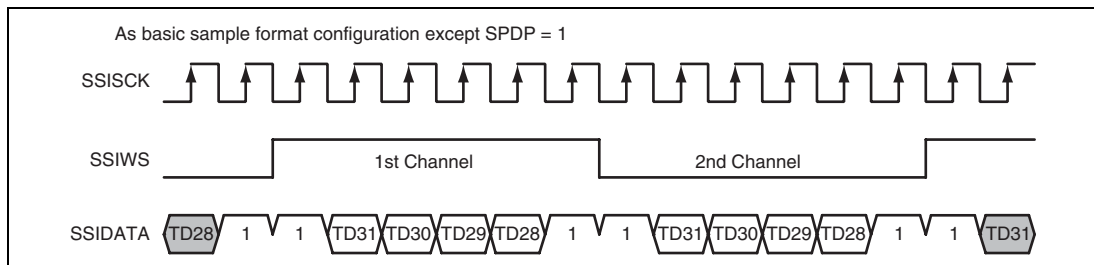
**Figure 18.10 Inverted Clock**

- Inverted Word Select



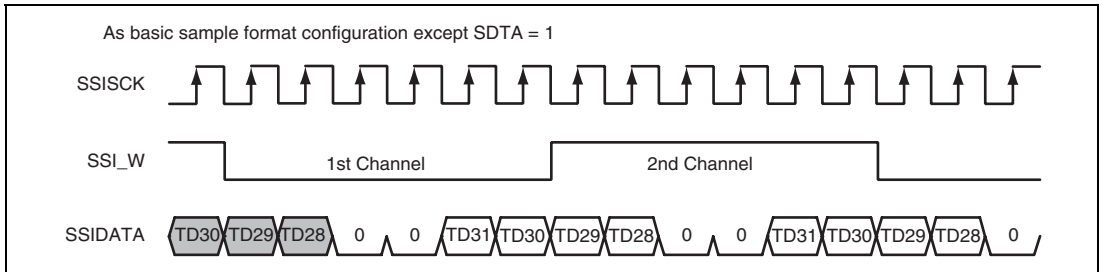
**Figure 18.11 Inverted Word Select**

- Inverted Padding Polarity



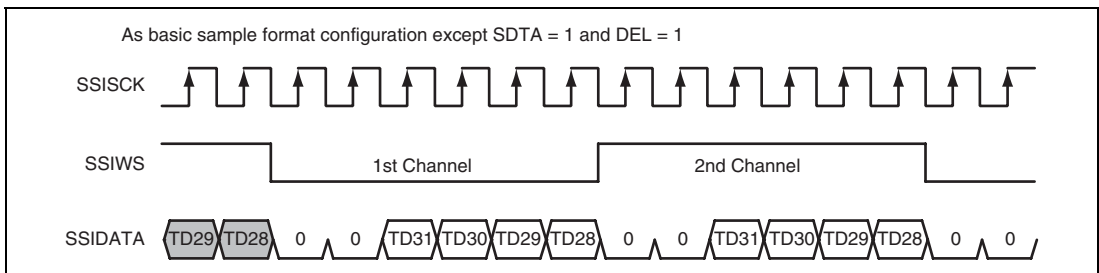
**Figure 18.12 Inverted Padding Polarity**

- Padding Bits First, Followed by Serial Data, with Delay



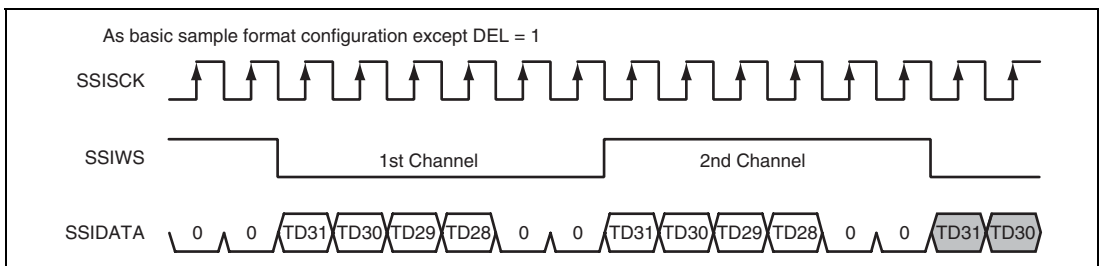
**Figure 18.13 Padding Bits First, Followed by Serial Data, with Delay**

- Padding Bits First, Followed by Serial Data, without Delay



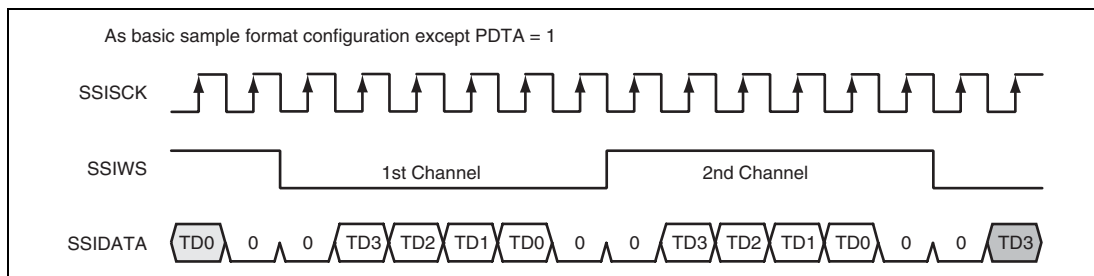
**Figure 18.14 Padding Bits First, Followed by Serial Data, without Delay**

- Serial Data First, Followed by Padding Bits, without Delay



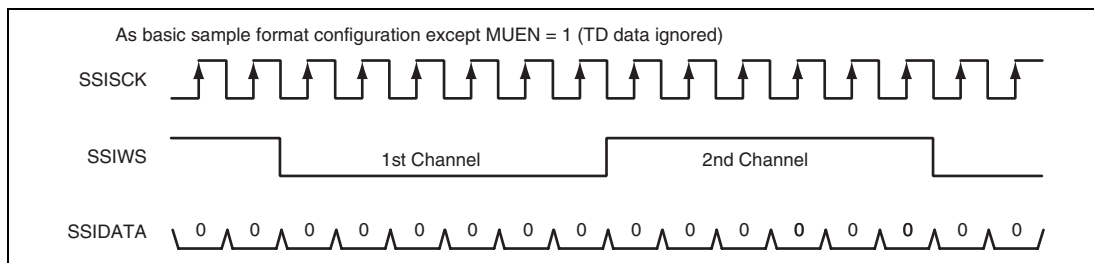
**Figure 18.15 Serial Data First, Followed by Padding Bits, without Delay**

- Parallel Right-Aligned with Delay



**Figure 18.16 Parallel Right-Aligned with Delay**

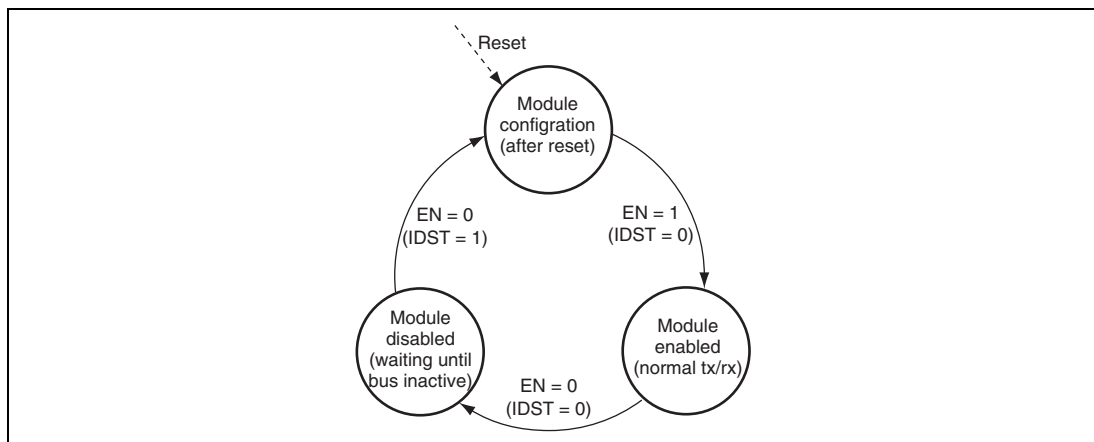
- Mute Enabled



**Figure 18.17 Mute Enabled**

### (3) Operation Modes

There are three modes of operation: configuration, enabled and disabled. Figure 18.18 shows the transition diagram between these operation modes.



**Figure 18.18 Transition Diagram between Operation Modes**

- **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 SSI\_CH0 to SSI\_CH5 are enabled by setting the EN bit.

Setting the EN bit causes SSI\_CH0 to SSI\_CH5 to enter the module enabled mode.

- **Module Enabled Mode:**

Operation in this mode depends on the selected operating mode. For details, see section 18.4.3 (4), Transmit Operation and section 18.4.3 (5), Receive Operation.

#### **(4) Transmit Operation**

Transmission can be controlled in one of two ways: either DMA or an interrupt driven.

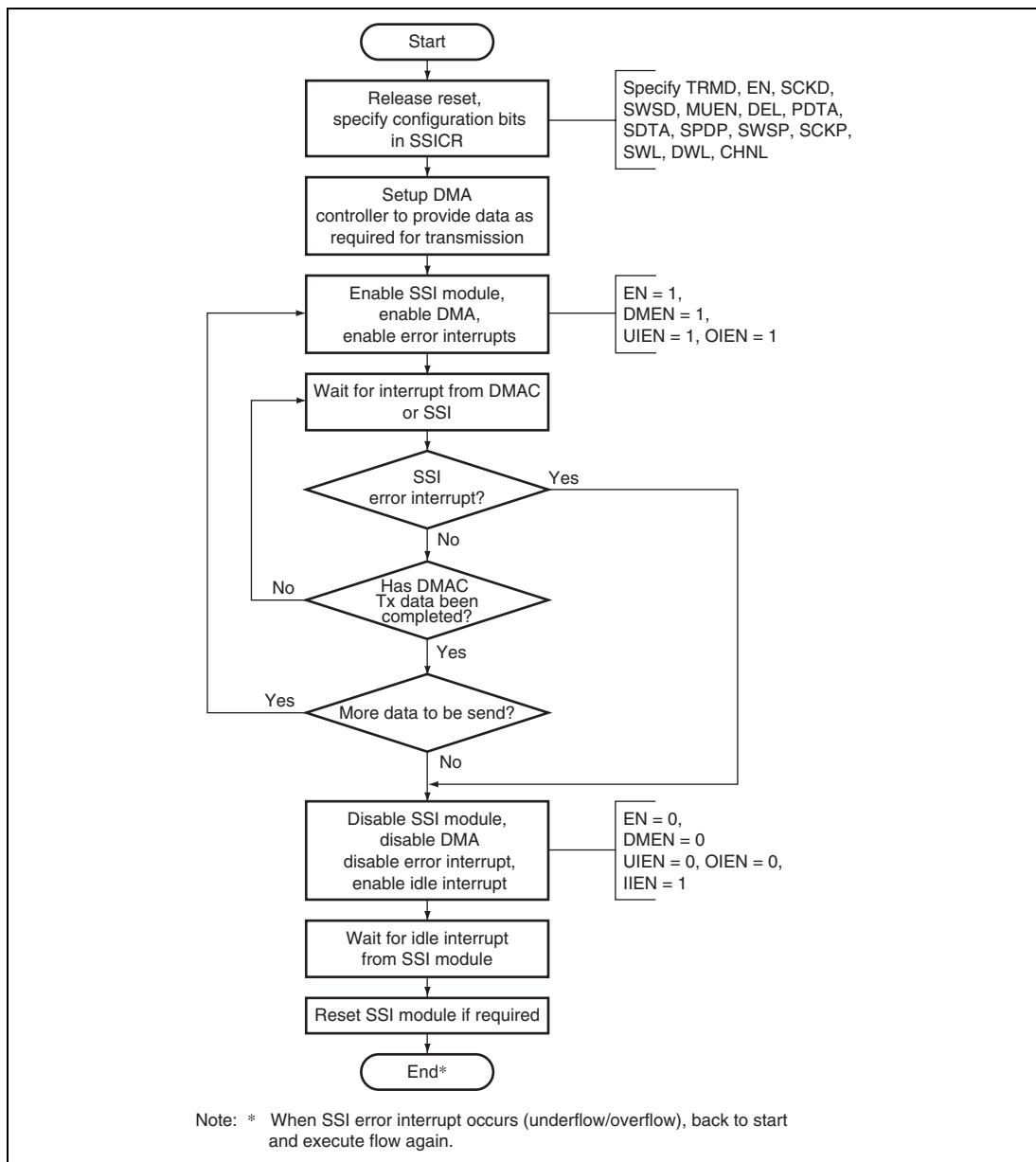
DMA driven is preferred to reduce the CPU load. In DMA control mode, an underflow or overflow of data or DMAC transfer end is notified by using an interrupt.

The alternative is using the interrupts that SSI\_CH0 to SSI\_CH5 generate to supply data as required. This mode has a higher interrupt load as SSI\_CH0 to SSI\_CH5 are only double buffered and will require data to be written at least every system word period.

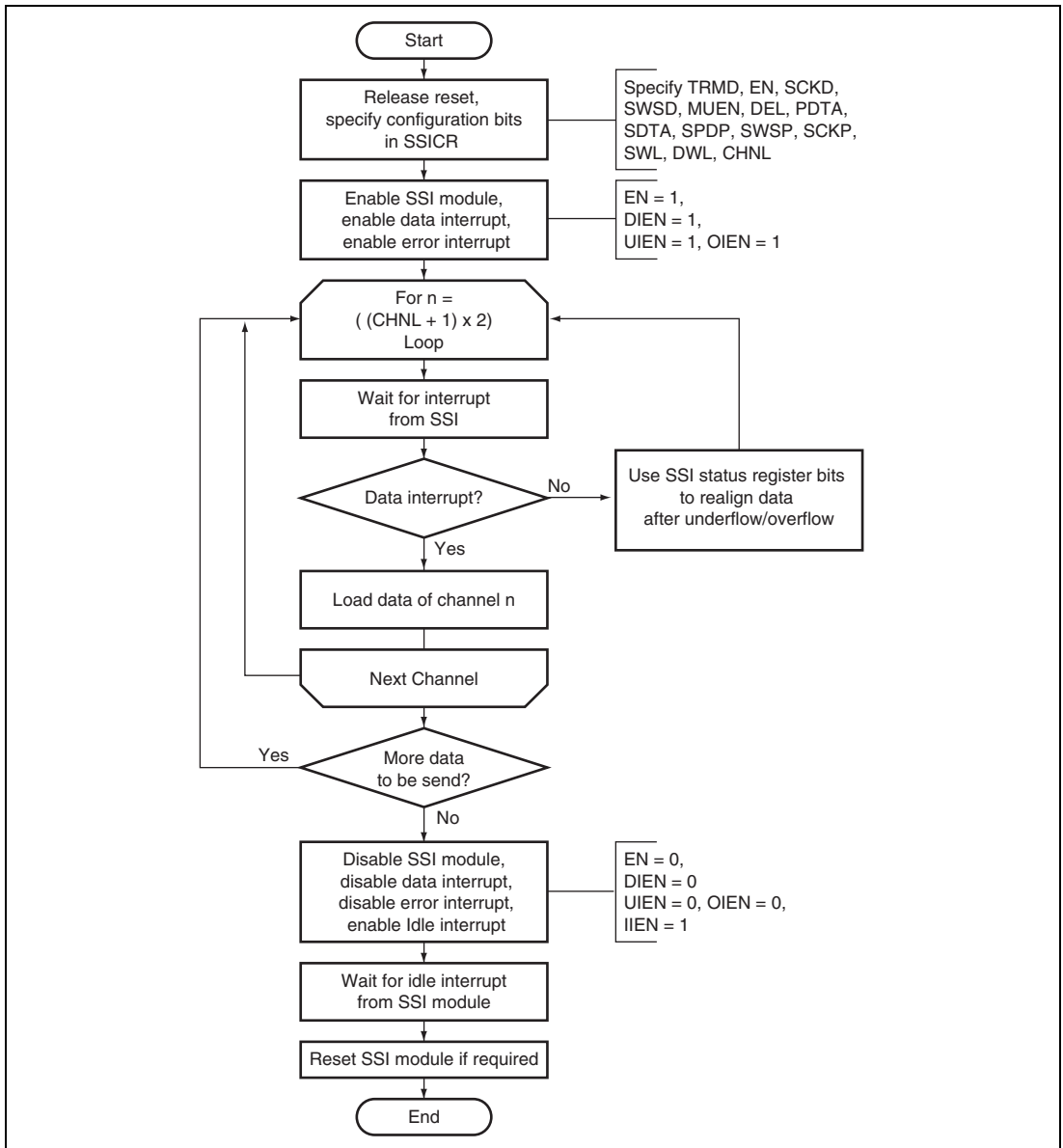
When disabling SSI\_CH0 to SSI\_CH5, the SSI clock\* must be supplied continuously until SSI\_CH0 to SSI\_CH5 enter in the idle state, indicated by the IIRQ bits in SSISR0 to SSISR5.

Figure 18.19 shows the transmit operation in the DMA controller mode. Figure 18.20 shows the transmit operation in the Interrupt controller mode.

Note: \* SCKD = 0: Clock input through the SSISCK[5:0] pins  
SCKD = 1: Clock input through the AUDIO\_CLK[5:0] pins

**(a) Transmission Using SSI\_DMAC0 and SSI\_DMAC1****Figure 18.19 Transmission Using SSI\_DMAC0 and SSI\_DMAC1**



**(b) Transmission using Interrupt Data Flow Control****Figure 18.20 Transmission Using Interrupt Data Flow Control**

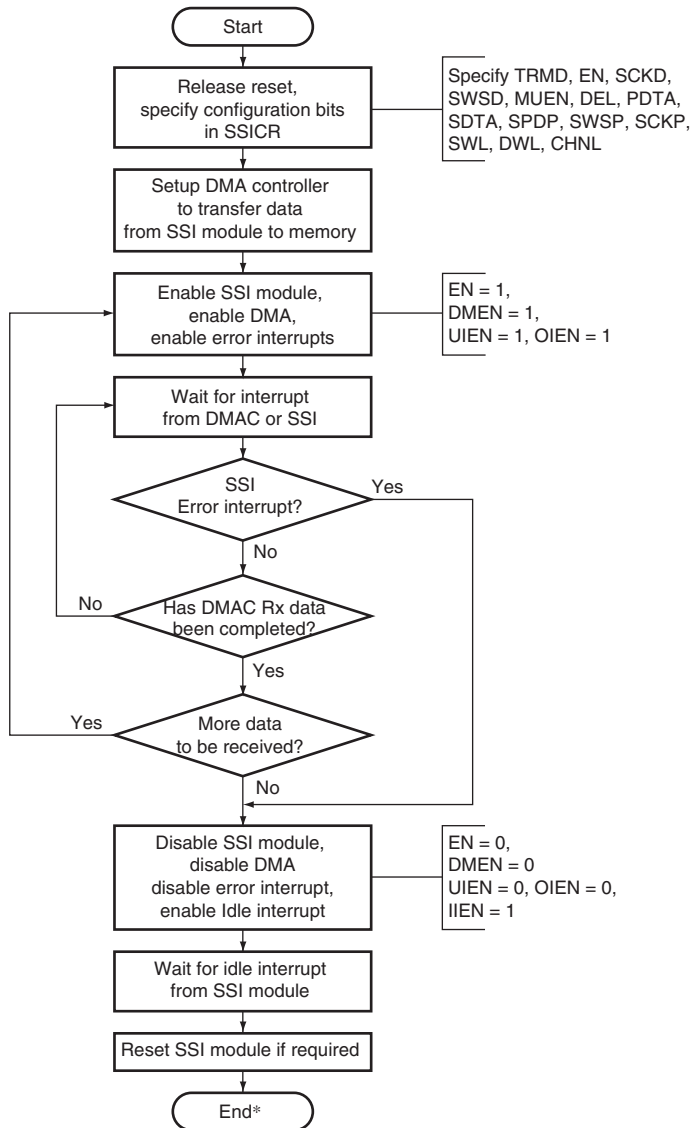
## (5) Receive Operation

As with transmission the reception can be controlled in one of two ways: either DMA or an interrupt driven.

Figures 18.21 and 18.22 show the flow of operation.

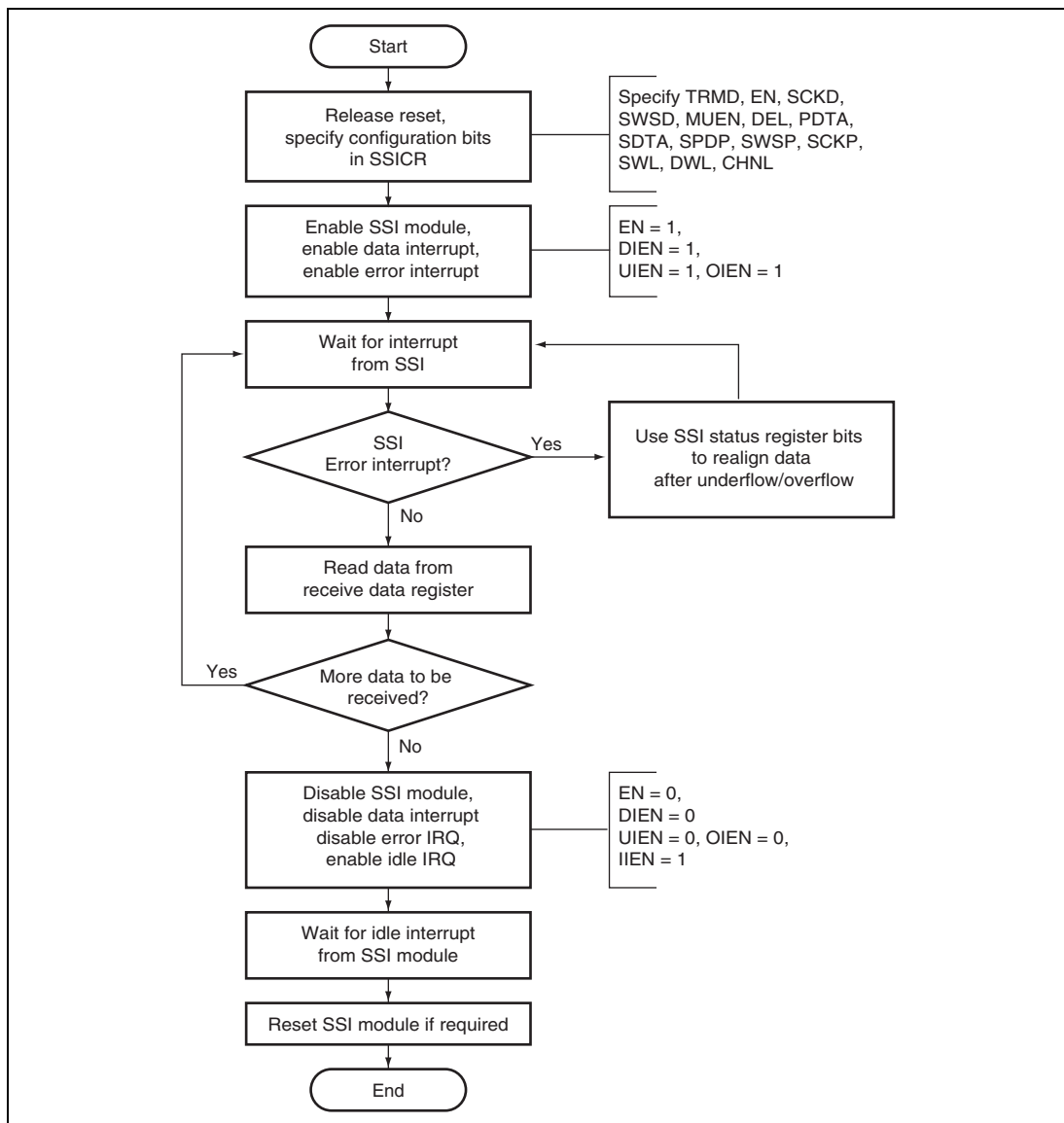
When disabling SSI\_CH0 to SSI\_CH5, the SSI clock must be supplied continuously until SSI\_CH0 to SSI\_CH5 enter in the idle state, which is indicated by the IIRQ bits in SSISR0 to SSISR5.

Note: \* SCKD = 0: Clock input through the SSISCK[5:0] pins  
SCKD = 1: Clock input through the AUDIO\_CLK[5:0] pins

**(a) Reception Using SSI\_DMAC0 and SSI\_DMAC1**

Note: \* When SSI error interrupt occurs (underflow/overflow), back to start and execute flow again.

**Figure 18.21 Reception Using SSI\_DMAC0 and SSI\_DMAC1**

**(b) Reception using Interrupt Data Flow Control****Figure 18.22 Reception Using Interrupt Data Flow Control**

When an underflow or overflow error condition is met, the CHNO[1:0] and SWNO bits in SSISR0 to SSISR5 can be used to recover SSI\_CH0 to SSI\_CH5 to a known status. When an underflow or overflow occurs, the host CPU can read the number of channels and the number of system words to determine what point the serial audio stream has reached. In the transmitter case, the host CPU can skip forward through the data it wants to transmit until it finds the sample data that matches what SSI\_CH0 to SSI\_CH5 are expecting to transmit next, and so resynchronize with the audio data stream. In the receiver case, the host CPU can skip forward storing null sample data until it is ready to store the sample data that SSI\_CH0 to SSI\_CH5 are indicating that it will receive next to ensure consistency of the number of received data, and so resynchronize with the audio data stream.

## **(6) Serial Clock Control**

This function is used to control and select which clock is used for the serial bus interface.

If the serial clock direction is set to input (SCKD in SSICR = 0), SSI\_CH0 to SSI\_CH5 are in clock slave mode, then the bit clock that is used in the shift register is derived from the SSISCK[5:0] pins.

If the serial clock direction is set to output (SCKD in SSICR = 1), SSI\_CH0 to SSI\_CH5 are in clock master mode, and the shift register uses the bit clock derived from the AUDIO\_SCK[5:0] input pins or its clock divided. This input clock is then divided by the ratio in the serial oversampling clock division ratio (CKDV) bit in SSICR0 to SSICR5 and used as the bit clock in the shift register.

In either case, the SSISCK[5:0] pin outputs are the same as the bit clock.

## 18.5 Usage Note

### 18.5.1 Restrictions when an Overflow Occurs during Receive DMA Operation

If an overflow occurs during receive DMA operation, the corresponding module of SSI\_CH0 to SSI\_CH5 must be reactivated.

The receive buffers of SSI\_CH0 to SSI\_CH5 consist of 32-bit registers common to both left and right channels. If an overflow occurs under the condition of control registers 0 to 5 (SSICR0 to SSICR5) data-word length (DWL2 to DWL0) is 32-bit and system-word length (SWL2 to SWL0) is 32-bit, the SSI has received the data at right channel that should be received at left channel.

If an overflow occurs through an overflow error interrupt or overflow error status flag (the OIRQ bit in SSISR0 to SSISR5), disable the DMA transfer of the SSI\_CH0 to SSI\_CH5 to halt its operation by writing 0 to the EN bit and DMEN bit in SSICR0 to SSICR5 (then terminate the SSI\_DMACH0 and SSI\_DMACH1 settings). And clear the overflow status flag by writing 0 to the OIRQ bit, set the DMA again and transfer restart.

### 18.5.2 Restrictions during Operation in Slave Mode

In slave mode, data transfer should be terminated (EN in SSICR0 to SSICR5 = 0) before the serial word select signal (SSIWS[5:0]) input is stopped.

In slave mode, data transfer is terminated by detecting a falling edge of the serial word select signal (SSIWS[5:0]) after the EN bit is cleared (transfer stop).

If the serial word select signal input stops, the falling edge of it cannot be detected. In this case, data transfer cannot be terminated normally.

### 18.5.3 Restrictions when Specify Each Register

- Consider the FIFO buffer size (64 bytes: 32 bits × 16 stages) to specify the RDMA maximum burst size (RDMBSZ) and the WDMA maximum burst size (WDMBSZ) in the SSIDMMR, which is the DMA mode register.
- In the RDMA data transfer count register (SSIRDMCNTR), specify the number of byte that must be a multiple of the maximum burst size number of the RDMA (RDMBSZ). For example, in the case of the maximum burst size such as; RDMBSZ = 4 burst (32 bytes), the number of 32,64,96,128 and so are able to be selected. Also same to the WDMA data transfer count register (SSIWDMCNTR), specify the number of byte that must be a multiple of the maximum burst size number of the WDMA (WDMBSZ). The RDMA data transfer count

register (SSIRDMCNTR) and the WDMA data transfer count register (SSIWDMCNTR) should be read only the setting value.

- In the block count source register (SSIBLCNTR), specify the number of byte that must be a multiple of the maximum burst size number of the RDMA or WDMA
- The transmit suspension block counter (SSISTPBLCNT) and the transmit suspension transfer data register (SSIWDMCNTR) are unable to be written in the conditions of the DMA is transferring data.
- The RDMA data transfer count register (SSIRDMCNTR) and the WDMA data transfer count register (SSIWDMCNTR) should be read only the setting value.
- Refer to the block counter register (SSIBLCNT) and the n-times block counter (SSIBLNCNT) for the number of data transferring.
- The block counter register (SSIBLCNT) and the n-times block counter register (SSIBLNCNT) are only readable and cleared by the software reset (DMRST). The transferred data count timing for the conditions of implementing the transmit operations is when data transfer from the transmit FIFO in the SSI\_DMAC to the data buffers in the SSI and for the conditions of implementing the receiving operation is when data transfer from the data buffers in the SSI to the receive FIFO in the SSI\_DMAC.





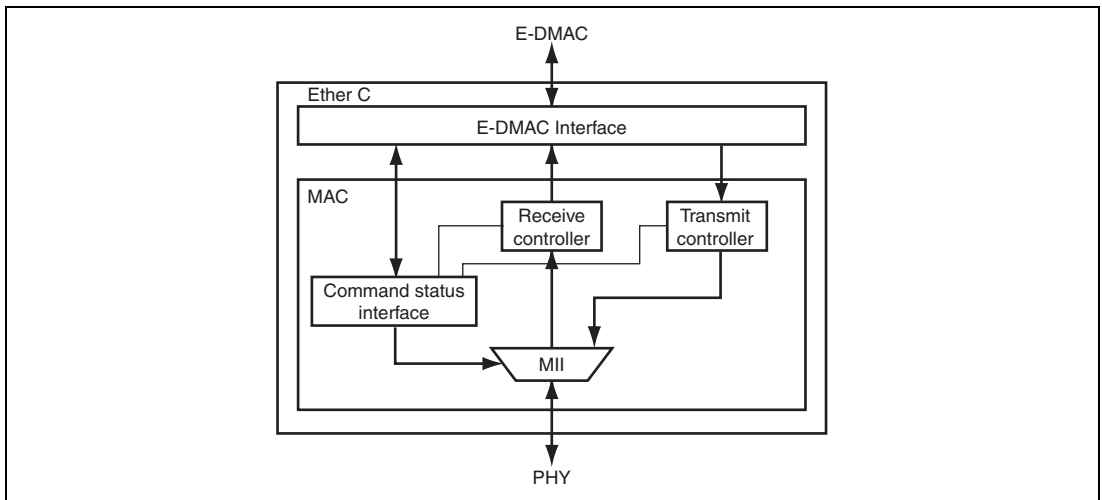
## Section 19 Ethernet Controller (EtherC)

This LSI has an on-chip Ethernet controller (EtherC) conforming to the Ethernet or the IEEE802.3 MAC (Media Access Control) layer standard. Connecting a physical-layer LSI (PHY-LSI) complying with this standard enables the Ethernet controller (EtherC) to perform transmission and reception of Ethernet/IEEE802.3 frames. The LSI has one MAC layer interface port. The Ethernet controller is connected to the Ethernet Direct Memory Access Controller (E-DMAC) for Ethernet controller inside the LSI, and carries out high-speed data transfer to and from the memory.

Figure 19.1 shows a configuration of the EtherC.

### 19.1 Features

- Transmission and reception of Ethernet/IEEE802.3 frames
- Supports 10/100 Mbps receive/transfer
- Supports full-duplex and half-duplex modes
- Conforms to IEEE802.3u standard MII (Media Independent Interface)
- Magic Packet detection and Wake-On-LAN (WOL) signal output
- Flow control conforming to IEEE802.3x



**Figure 19.1 Configuration of EtherC**

## 19.2 Input/Output Pins

Table 19.1 lists the pin configuration of the EtherC.

**Table 19.1 Pin Configuration**

Name	Abbreviation	I/O	Function
Transmit clock*	TX-CLK	I	TX-EN, MII_TXD3 to MII_TXD0, TX-ER timing reference signal
Receive clock*	RX-CLK	I	RX-DV, MII_RXD3 to MII_RXD0, RX-ER timing reference signal
Transmit enable*	TX-EN	O	Indicates that transmit data is ready on MII_TXD3 to MII_TXD0
Transmit data*	MII_TXD3 to MII_TXD0	O	4-bit transmit data
Transmit error*	TX-ER	O	Notifies PHY_LSI of error during transmission
Receive data valid*	RX-DV	I	Indicates that valid receive data is on MII_RXD3 to MII_RXD0
Receive data*	MII_RXD3 to MII_RXD0	I	4-bit receive data
Receive error*	RX-ER	I	Identifies error state occurred during data reception
Carrier detection*	CRS	I	Carrier detection signal
Collision detection*	COL	I	Collision detection signal
Management data clock*	MDC	O	Reference clock signal for information transfer via MDIO
Management data I/O*	MDIO	I/O	Bidirectional signal for exchange of management information between this LSI and PHY
Link status	LNKSTA	I	Inputs link status from PHY
General-purpose external output	EXOUT	O	External output pin
Wake-On-LAN	WOL	O	Signal indicating reception of Magic Packet

Notes: \* MII signal conforming to IEEE802.3u

## 19.3 Register Descriptions

Table 19.2 shows the configuration of registers of EtherC. Table 19.3 shows the state of registers in each processing mode.

**Table 19.2 Register Configuration**

Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
EtherC mode register	ECMR	R/W	H'FEF0 0100	H'1EF0 0100	32
EtherC status register	ECSR	R/W	H'FEF0 0110	H'1EF0 0110	32
EtherC interrupt permission register	ECSIPR	R/W	H'FEF0 0118	H'1EF0 0118	32
Receive frame length register	RFLR	R/W	H'FEF0 0108	H'1EF0 0108	32
PHY interface register	PIR	R/W	H'FEF0 0120	H'1EF0 0120	32
MAC address high register	MAHR	R/W	H'FEF0 01C0	H'1EF0 01C0	32
MAC address low register	MALR	R/W	H'FEF0 01C8	H'1EF0 01C8	32
PHY status register	PSR	R	H'FEF0 0128	H'1EF0 0128	32
Transmit retry over counter register	TROCR	R/W	H'FEF0 01D0	H'1EF0 01D0	32
Delayed collision detect counter register	CDCR	R/W	H'FEF0 01D4	H'1EF0 01D4	32
Lost carrier counter register	LCCR	R/W	H'FEF0 01D8	H'1EF0 01D8	32
Carrier not detect counter register	CNDCR	R/W	H'FEF0 01DC	H'1EF0 01DC	32
CRC error frame receive counter register	CEFCR	R/W	H'FEF0 01E4	H'1EF0 01E4	32
Frame receive error counter register	FRECR	R/W	H'FEF0 01E8	H'1EF0 01E8	32
Too-short frame receive counter register	TSFRCR	R/W	H'FEF0 01EC	H'1EF0 01EC	32
Too-long frame receive counter register	TLFRCR	R/W	H'FEF0 01F0	H'1EF0 01F0	32
Residual-bit frame receive counter register	RFCR	R/W	H'FEF0 01F4	H'1EF0 01F4	32
Multicast address frame receive counter register	MAFCR	R/W	H'FEF0 01F8	H'1EF0 01F8	32
IPG register	IPGR	R/W	H'FEF0 0150	H'1EF0 0150	32
Automatic PAUSE frame register	APR	R/W	H'FEF0 0154	H'1EF0 0154	32
Manual PAUSE frame register	MPR	R/W	H'FEF0 0158	H'1EF0 0158	32
Automatic PAUSE frame retransmit count register	TPAUSER	R/W	H'FEF0 0164	H'1EF0 0164	32

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
Random number generation counter upper limit setting register	RDMLR	R/W	H'FEF0 0140	H'1EF0 0140	32
PAUSE Frame Receive Counter Register	RFCF	R	H'FEF0 0160	H'1EF0 0160	32
PAUSE frame retransmit counter register	TPAUSECR	R	H'FEF0 0168	H'1EF0 0168	32
Broadcast frame receive count setting register	BCFRR	R/W	H'FEF0 016C	H'1EF0 016C	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 19.3 Register States in Each Operation Mode**

Name	Abbreviation	Software Reset
EtherC mode register	ECMR	Initialised
EtherC status register	ECSR	Initialised
EtherC interrupt permission register	ECSIPR	Initialised
Receive frame length register	RFLR	Initialised
PHY interface register	PIR	Initialised
MAC address high register	MAHR	Initialised
MAC address low register	MALR	Initialised
PHY status register	PSR	Initialised
Transmit retry over counter register	TROCR	Initialised
Delayed collision detect counter register	CDCR	Initialised
Lost carrier counter register	LCCR	Initialised
Carrier not detect counter register	CNDCR	Initialised
CRC error frame receive counter register	CEFCR	Initialised
Frame receive error counter register	FRECR	Initialised
Too-short frame receive counter register	TSFRCR	Initialised
Too-long frame receive counter register	TLFRCR	Initialised
Residual-bit frame receive counter register	RFCR	Initialised
Multicast address frame receive counter register	MAFCR	Initialised
IPG register	IPGR	Initialised
Automatic PAUSE frame register	APR	Initialised
Manual PAUSE frame register	MPR	Initialised

Name	Abbreviation	Software Reset
Automatic PAUSE frame retransmit count register	TPAUSER	Initialised
Random number generation counter upper limit setting register	RDMLR	Initialised
PAUSE Frame Receive Counter Register	RFCF	Initialised
PAUSE frame retransmit counter register	TPAUSECR	Initialised
Broadcast frame receive count setting register	BCFRR	Initialised

### 19.3.1 EtherC Mode Register (ECMR)

ECMR is a 32-bit readable/writable register that specifies the operating mode of the EtherC. The settings in this register are normally made in the initialization process following a reset.

The operating mode setting must not be changed while the transmitting and receiving functions are enabled. To switch the operating mode, return the EtherC and E-DMAC to their initial states by means of the SWR bit in EDMR before making settings again.

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
	—	—	—	PRCEF	—	—	MPDE	—	—	RE	TE	—	ILB	—	DM	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	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	TPC	0	R/W	PAUSE Frame Transmission 0: PAUSE frame is not transmitted in a PAUSE period 1: PAUSE frame is transmitted even in a PAUSE period

Bit	Bit Name	Initial Value	R/W	Description
19	ZPF	0	R/W	<p>PAUSE Frame Usage with TIME = 0 Enable</p> <p>0: Control of a PAUSE frame whose TIME parameter value is 0 is disabled. The next frame is not transmitted until the time specified by the Timer value has elapsed. If a PAUSE frame whose time specified by the Timer value is 0 is received, that PAUSE frame is discarded.</p> <p>1: Control of a PAUSE frame whose TIME parameter value is 0 is enabled. When the data size in the receive FIFO becomes smaller than the FCFTR setting before the time specified by the Timer value elapses, an automatic PAUSE frame with a Timer value of 0 is transmitted. On receiving a PAUSE frame with a Timer value of 0, the transmission wait state is canceled.</p>
18	PFR	0	R/W	<p>PAUSE Frame Receive Mode</p> <p>0: PAUSE frame is not transferred to E-DMAC</p> <p>1: PAUSE frame is transferred to E-DMAC</p>
17	RXF	0	R/W	<p>Operating Mode for Receiving Port Flow Control</p> <p>0: PAUSE frame detection is disabled</p> <p>1: Flow control for the receiving port is enabled</p>
16	TXF	0	R/W	<p>Operating Mode for Transmitting Port Flow Control</p> <p>0: Flow control for the transmitting port is disabled (Automatic PAUSE frame is not transmitted)</p> <p>1: Flow control for the transmitting port is enabled (Automatic PAUSE frame is transmitted as required)</p>
15 to 13	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
12	PRCEF	0	R/W	<p>CRC Error Frame Reception Enable</p> <p>0: A frame with a CRC error is received as a frame with an error</p> <p>1: A frame with a CRC error is received as a frame without an error</p>

Bit	Bit Name	Initial Value	R/W	Description
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	MPDE	0	R/W	Magic Packet Detection Enable Enables or disables Magic Packet detection by hardware to allow activation from the Ethernet. 0: Magic Packet detection is not enabled 1: Magic Packet detection is enabled
8, 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	RE	0	R/W	Reception Enable If a switch is made from receiving function enabled (RE = 1) to disabled (RE = 0) while a frame is being received, the receiving function will be enabled until reception of the corresponding frame is completed. 0: Receiving function is disabled 1: Receiving function is enabled
5	TE	0	R/W	Transmission Enable If a switch is made from transmitting function enabled (TE = 1) to disabled (TE = 0) while a frame is being transmitted, the transmitting function will be enabled until transmission of the corresponding frame is completed. 0: Transmitting function is disabled 1: Transmitting function is enabled
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	ILB	0	R/W	Internal Loop Back Mode Specifies loopback mode in the EtherC. 0: Normal data transmission/reception is performed 1: Data loopback is performed inside the MAC in the EtherC when DM = 1

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	DM	0	R/W	Duplex Mode Specifies the EtherC transfer method. 0: Half-duplex transfer is specified 1: Full-duplex transfer is specified
0	PRM	0	R/W	Promiscuous Mode Setting this bit enables all Ethernet frames to be received. All Ethernet frames means all receivable frames, irrespective of differences or enabled/disabled status (destination address, broadcast address, multicast bit, etc.). 0: EtherC performs normal operation 1: EtherC performs promiscuous mode operation



### 19.3.2 EtherC Status Register (ECSR)

ECSR is a 32-bit readable/writable register that indicates the status in the EtherC. This status can be notified to the CPU by interrupts. When 1 is written to the PFROI, LCHNG, MPD, and ICD bits, the corresponding flags can be cleared. Writing 0 does not affect the flag. For bits that generate interrupts, the interrupt can be enabled or disabled by the corresponding bit in ECSIPR.

The interrupts generated due to this status register are indicated in ECI bit in EESR of the E-DMAC.

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
	—	—	—	—	—	—	—	—	—	—	BFR	PFROI	—	LCHNG	MPD	ICD
Initial value:	0	0	0	0	0	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	R/W	Description
31 to 6	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
5	BFR	0	R/W	Continuous Broadcast Frame Reception Interrupt (Interrupt Source)  Indicates that Broadcast frames have been received continuously.
4	PFROI	0	R/W	PAUSE Frame Retransmit Retry Over  Indicates whether the retransmit count for retransmitting a PAUSE frame when flow control is enabled has exceeded the retransmit upper-limit set in the automatic PAUSE frame retransmit count register (TPAUSER).  0: PAUSE frame retransmit count has not exceeded the upper limit 1: PAUSE frame retransmit count has exceeded the upper limit

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	LCHNG	0	R/W	Link Signal Change Indicates that the LNKSTA signal input from the PHY-LSI has changed from high to low or low to high. To check the current Link state, refer to the LMON bit in the PHY status register (PSR). 0: Change in the LNKSTA signal has not been detected 1: Change in the LNKSTA signal has been detected (high to low or low to high)
1	MPD	0	R/W	Magic Packet Detection Indicates that a Magic Packet has been detected on the line. 0: Magic Packet has not been detected 1: Magic Packet has been detected
0	ICD	0	R/W	Illegal Carrier Detection Indicates that the PHY-LSI has detected an illegal carrier on the line. More specifically, this bit is set when the signals transmitted from the PHY-LSI to this LSI through the RX-DV, RX-ER and MII-RXD3 to 0 pins are 0,1, and 1110, respectively (see figure 19.4 (6)). If a change in the signal input from the PHY-LSI occurs in a period shorter than the software recognition period, the correct information may not be obtained. Refer to the timing specification for the PHY-LSI used. 0: PHY-LSI has not detected an illegal carrier on the line 1: PHY-LSI has detected an illegal carrier on the line

### 19.3.3 EtherC Interrupt Permission Register (ECSIPR)

ECSIPR is a 32-bit readable/writable register that enables or disables the interrupt sources indicated by ECSR. Each bit can disable or enable interrupts corresponding to the bits in ECSR.

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
	—	—	—	—	—	—	—	—	—	—	BFSIPR	PFROIP	—	LCHNGIP	MPDIP	ICDIP
Initial value:	0	0	0	0	0	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	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	BFSIPR	0	R/W	Continuous Broadcast Frame Reception Interrupt Enable 0: Enables an interrupt requested by the BFR bit in ECSR 1: Disables an interrupt requested by the BFR bit in ECSR
4	PFROIP	0	R/W	PAUSE Frame Retransmit Interrupt Enable 0: Interrupt notification by the PFROI bit is disabled 1: Interrupt notification by the PFROI bit is enabled
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	LCHNGIP	0	R/W	LINK Signal Change Interrupt Enable 0: Interrupt notification by the LCHNG bit is disabled 1: Interrupt notification by the LCHNG bit is enabled

Bit	Bit Name	Initial Value	R/W	Description
1	MPDIP	0	R/W	Magic Packet Detect Interrupt Enable 0: Interrupt notification by the MPD bit is disabled 1: Interrupt notification by the MPD bit is enabled
0	ICDIP	0	R/W	Illegal Carrier Detect Interrupt Enable 0: Interrupt notification by the ICD bit is disabled 1: Interrupt notification by the ICD bit is enabled

### 19.3.4 PHY Interface Register (PIR)

PIR is a 32-bit readable/writable register that provides a means of accessing the PHY-LSI internal registers via the MII.

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	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	MDI	Undefined	R	MII Management Data-In Indicates the level of the MDIO pin.
2	MDO	0	R/W	MII Management Data-Out Outputs the value set in this bit from the MDIO pin when the MMD bit is 1.

Bit	Bit Name	Initial Value	R/W	Description
1	MMD	0	R/W	MII Management Mode Specifies the data read/write direction with respect to the MII. 0: Read direction is specified 1: Write direction is specified
0	MDC	0	R/W	MII Management Data Clock Outputs the value set in this bit from the MDC pin and supplies the MII with the management data clock. For the method of accessing the MII registers, see section 19.4.4, Accessing MII Registers.

### 19.3.5 MAC Address High Register (MAHR)

MAHR is a 32-bit readable/writable register that specifies the upper 32 bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Return the EtherC and E-DMAC to their initial states by means of the SWR bit in EDMR before making settings again.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MA[47:32]															
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
	MA[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	Bit Name	Initial Value	R/W	Description
31 to 0	MA[47:16]	All 0	R/W	MAC Address Bits 47 to 16 These bits are used to set the upper 32 bits of the MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), set H'01234567 in this register.

### 19.3.6 MAC Address Low Register (MALR)

MALR is a 32-bit readable/writable register that specifies the lower 16 bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Return the EtherC and E-DMAC to their initial states by means of the SWR bit in EDMR before making settings again.

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
	MA[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 These bits are always read as 0. The write value should always be 0.
15 to 0	MA[15:0]	All 0	R/W	MAC Address Bits 15 to 0 These bits are used to set the lower 16 bits of the MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), set H'89AB in this register.

### 19.3.7 Receive Frame Length Register (RFLR)

RFLR is a 32-bit readable/writable register that specifies the maximum frame length (in bytes) that can be received by this LSI. The settings in this register must not be changed while the receiving function is enabled.

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
	—	—	—	—	RFL[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 12	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
11 to 0	RFL[11:0]	All 0	R/W	Receive Frame Length  The frame data described here refers to all fields from the destination address up to the CRC data. Frame contents from the destination address up to the data are actually transferred to memory. CRC data is not included in the transfer. When data that exceeds the specified value is received, the part of data that exceeds the specified value is discarded.  H'000 to H'5EE: 1,518 bytes H'5EF: 1,519 bytes H'5F0: 1,520 bytes : : H'7FF: 2,047 bytes H'800 to H'FFF: 2048 bytes

### 19.3.8 PHY Status Register (PSR)

PSR is a read-only register that can read interface signals from the PHY-LSI.

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LMON
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	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	LMON	Undefined	R	LNKSTA Pin Status The Link status can be read by connecting the Link signal output from the PHY-LSI to the LNKSTA pin. For the polarity, refer to the specifications of the PHY-LSI to be connected.



### 19.3.9 Transmit Retry Over Counter Register (TROCR)

TROCR is a 32-bit counter that indicates the number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer. When 16 transmission attempts have failed, this register is incremented by 1. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TROCR[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

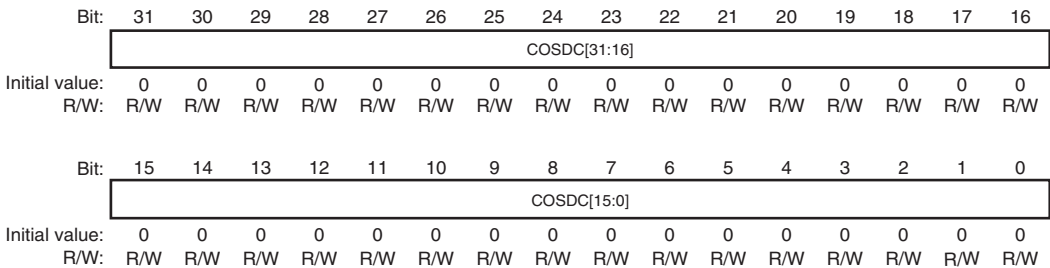
  

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TROCR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TROCR[31:0]	All 0	R/W	Transmit Retry Over Count  These bits indicate the number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer.

19.3.10 Delayed Collision Detect Counter Register (CDCR)

CDCR is a 32-bit counter that indicates the number of all delayed collisions that occurred on the line after the start of data transmission. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any value.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	COSDC[31:0]	All 0	R/W	Delayed Collision Detect Count
				These bits indicate the number of all delayed collisions after the start of data transmission.

### 19.3.11 Lost Carrier Counter Register (LCCR)

LCCR is a 32-bit counter that indicates the number of times the carrier was lost during data transmission. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LCC[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

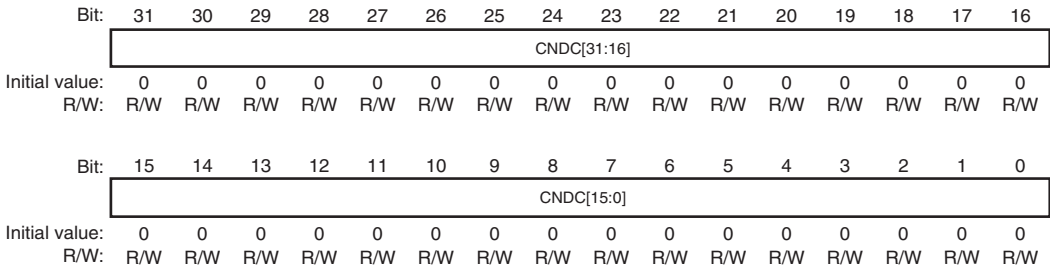
  

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	LCC[31:0]	All 0	R/W	Lost Carrier Count These bits indicate the number of times the carrier was lost during data transmission.

19.3.12 Carrier Not Detect Counter Register (CNDCR)

CNDCR is a 32-bit counter that indicates the number of times carrier could not be detected while the preamble is being sent. When the value in this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by a write to this register with any value.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CNDC[31:0]	All 0	R/W	Carrier Not Detect Count These bits indicate the number of times carrier was not detected.

### 19.3.13 CRC Error Frame Receive Counter Register (CEFCR)

CEFCR is a 32-bit counter that indicates the number of times a frame with a CRC error was received. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CEFC[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

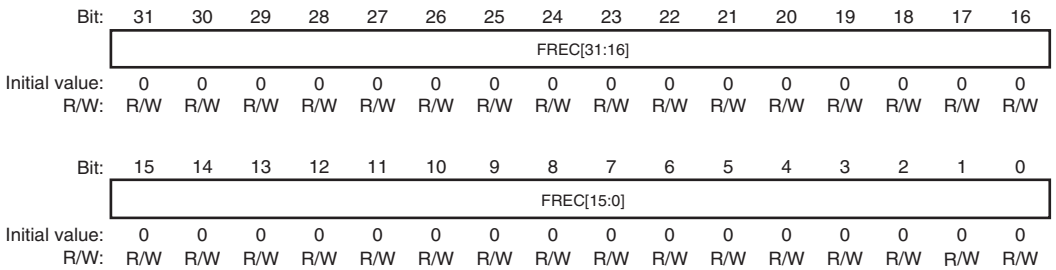
  

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CEFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CEFC[31:0]	All 0	R/W	CRC Error Frame Count These bits indicate the number of CRC error frames received.

19.3.14 Frame Receive Error Counter Register (FRECR)

FRECR is a 32-bit counter that indicates the number of frames for which a receive error was generated by the RX-ER pin input from the PHY-LSI. FRECR is incremented each time the RX-ER pin becomes active. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any value.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FRECR[31:0]	All 0	R/W	Frame Receive Error Count
				These bits indicate the number of errors during frame reception.

### 19.3.15 Too-Short Frame Receive Counter Register (TSFRCCR)

TSFRCCR is a 32-bit counter that indicates the number of frames received with a length fewer than 64 bytes. When the value in this register reaches H'FFFFFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSFC[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

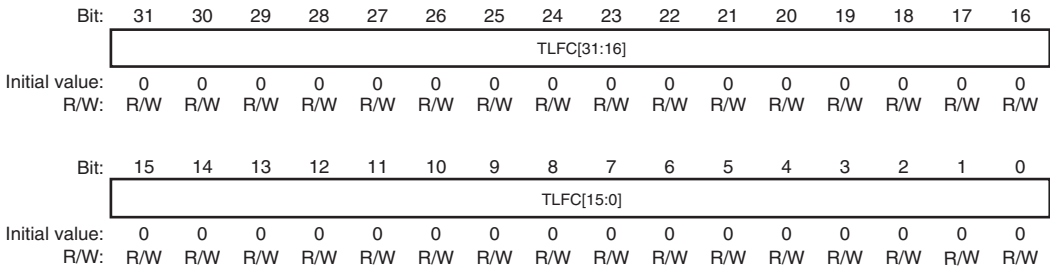
  

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TSFC[31:0]	All 0	R/W	Too-Short Frame Receive Count These bits indicate the number of frames received with a length of less than 64 bytes.

19.3.16 Too-Long Frame Receive Counter Register (TLFRCR)

TLFRCR is a 32-bit counter that indicates the number of frames received with a length exceeding the value specified by the receive frame length register (RFLR). When the value in this register reaches H'FFFFFFFF, count-up is halted. This register is not incremented when a frame containing residual bits is received. In this case, the reception of the frame is indicated in the residual-bit frame receive counter register (RFCR). The counter value is cleared to 0 by a write to this register with any value.

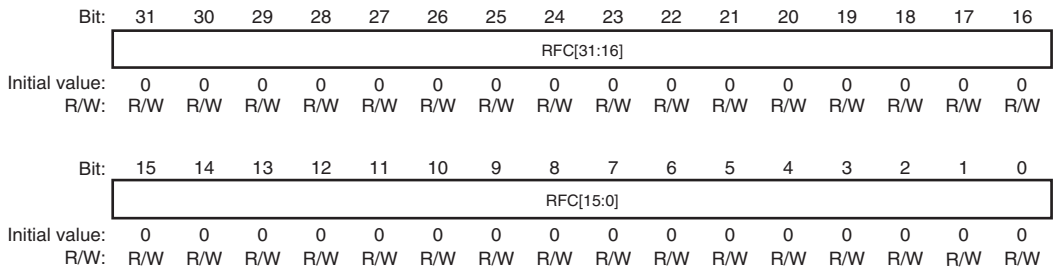


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TLFC[31:0]	All 0	R/W	Too-Long Frame Receive Count
				These bits indicate the number of frames received with a length exceeding the value in RFLR.



### 19.3.17 Residual-Bit Frame Receive Counter Register (RFCR)

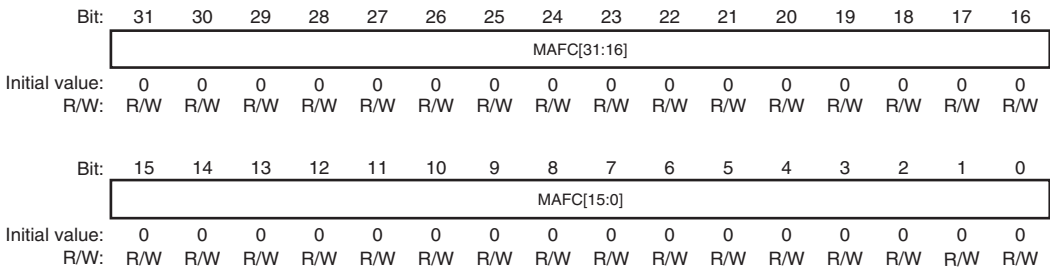
RFCR is a 32-bit counter that indicates the number of frames received containing residual bits (less than an 8-bit unit). When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any value.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RFC[31:0]	All 0	R/W	Residual-Bit Frame Receive Count
				These bits indicate the number of frames received containing residual bits.

19.3.18 Multicast Address Frame Receive Counter Register (MAFCR)

MAFCR is a 32-bit counter that indicates the number of frames received with a specified multicast address. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any value.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MAFC[31:0]	All 0	R/W	Multicast Address Frame Count These bits indicate the number of multicast frames received.

### 19.3.19 IPG Register (IPGR)

IPGR sets the IPG (Inter Packet Gap). This register must not be changed while the transmitting and receiving functions of the EtherC mode register (ECMR) are enabled. (For details, refer to section 19.4.6, Operation by IPG Setting.)

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	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	IPG[4:0]	H'14	R/W	Inter Packet Gap Sets the IPG value every 4-bit time. H'00: 16-bit time H'01: 20-bit time : : H'14: 96-bit time (Default) : : H'1F: 140-bit time

### 19.3.20 Automatic PAUSE Frame Register (APR)

APR 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
	AP[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 These bits are always read as 0. The write value should always be 0.
15 to 0	AP[15:0]	All 0	R/W	Automatic PAUSE These bits set the TIME parameter value of an automatic PAUSE frame. One bit is equivalent to 512 bit-time.

### 19.3.21 Manual PAUSE Frame Register (MPR)

MPR 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:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MP[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 These bits are always read as 0. The write value should always be 0.
15 to 0	MP[15:0]	All 0	R/W	Manual PAUSE These bits set the TIME parameter value of a manual PAUSE frame. One bit is equivalent to 512 bit-time. Read value is undefined.

19.3.22 Automatic PAUSE Frame Retransmit Count Register (TPAUSER)

TPAUSER is used to set the upper limit for the number of times to retransmit an automatic PAUSE frame. The settings in this register must not be changed while the transmitting function is enabled.

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
	TPAUSE[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  These bits are always read as 0. The write value should always be 0.
15 to 0	TPAUSE[15:0]	All 0	R/W	Upper Limit for Automatic PAUSE Frame Retransmission  H'0000: Retransmit count is unlimited H'0001: Retransmit count is 1 : : H'FFFF: Retransmit count is 65,535

### 19.3.23 Random Number Generation Counter Upper Limit Setting Register (RDMLR)

RDMLR is used to set the upper limit for the counter used in the random number generation block.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RMD[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
	RMD[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 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 0	RMD[19:0]	All 0	R/W	Upper Limit for Counter Used in Random Number Generation Block H'00000: Set value in normal operation H'00001 to H'FFFFE: Upper limit for the counter

**Note:** The operation of the random number generation block in the feLic depends on the setting in this register. Accordingly, special attention should be paid when setting a value other than 0.

19.3.24 PAUSE Frame Receive Counter Register (RFCF)

RFCF is an 8-bit counter that indicates the number of times a PAUSE frame is received.

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	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	TXP[7:0]	All 0	R	PAUSE Frame Receive Count



### 19.3.25 PAUSE Frame Retransmit Counter Register (TPAUSECR)

PFTCR is a 16-bit counter that indicates the number of times a PAUSE frame is retransmitted.

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[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	TXP[7:0]	All 0	R	PAUSE Frame Retransmit Count

### 19.3.26 Broadcast Frame Receive Count Setting Register (BCFRR)

BCFRR is used to set the number of Broadcast frames that can be received continuously.

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
	BCF[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 These bits are always read as 0. The write value should always be 0.
15 to 0	BCF[15:0]	All 0	R/W	Receive Count for Continuous Broadcast Frames The DA can receive a Broadcast address frame up to the number of times set in these bits. If the reception is performed for more times than the set value, the excess Broadcast frames are discarded. H'0000: No limitation for receive count H'0001: 1 frame can be received : : H'FFFF: 65,535 continuous frames can be received

## 19.4 Operation

The following outlines the operations of the Ethernet controller (EtherC).

The Ethernet controller (EtherC) supports flow control functions conforming to IEEE802.3x, and transmission/reception of PAUSE frames used for the control is possible.

### 19.4.1 Transmission

The EtherC transmitter assembles the transmit data on the frame and outputs to MII when there is a transmit request from the E-DMAC. The data transmitted via the MII is transmitted to the lines by PHY-LSI. Figure 19.2 shows the status change of the EtherC transmitter.

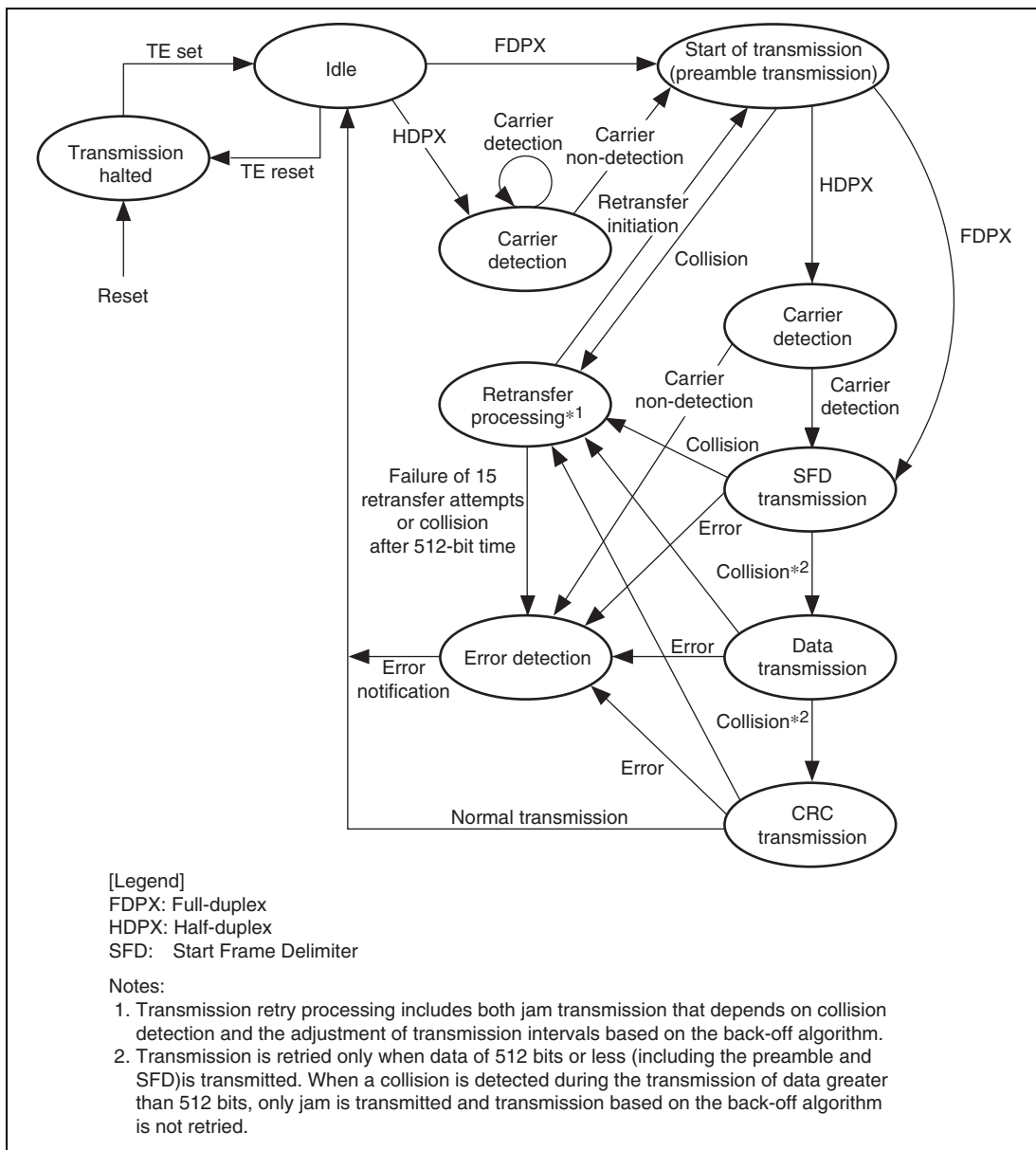
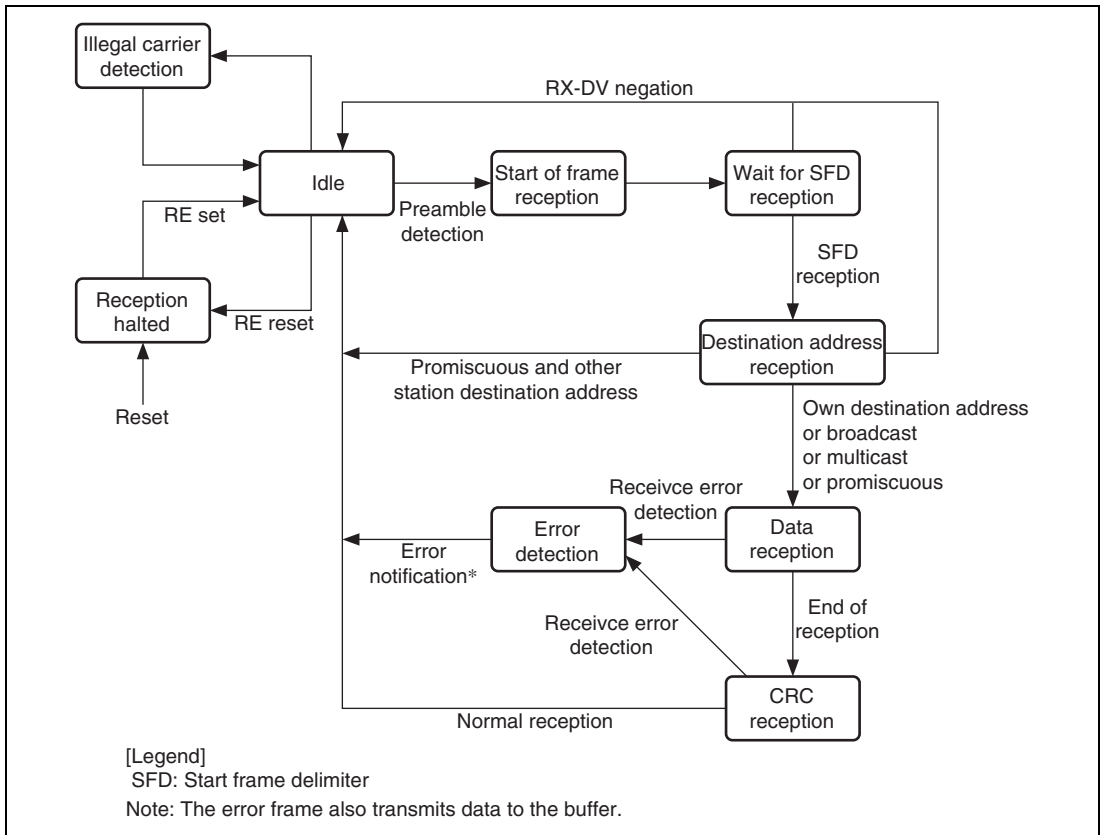


Figure 19.2 EtherC Transmitter State Transitions

1. When the transmit enable (TE) bit is set, the transmitter enters the transmit idle state.
2. When a transmit request is issued by the transmit E-DMAC, the EtherC sends the preamble after a transmission delay equivalent to the frame interval time. If full-duplex transfer is selected, which does not require carrier detection, the preamble is sent as soon as a transmit request is issued by the E-DMAC.
3. The transmitter sends the SFD, data, and CRC sequentially. At the end of transmission, the transmit E-DMAC generates a transmission complete interrupt (TC). If a collision or the carrier-not-detected state occurs during data transmission, these are reported as interrupt sources.
4. After waiting for the frame interval time, the transmitter enters the idle state, and if there is more transmit data, continues transmitting.

### **19.4.2 Reception**

The EtherC receiver separates the frame from the MII into preamble, SFD, data and CRC, and the fields from DA (destination address) to the CRC data are transferred to the receive E-DMAC. Figure 19.3 shows the state transitions of the EtherC receiver.

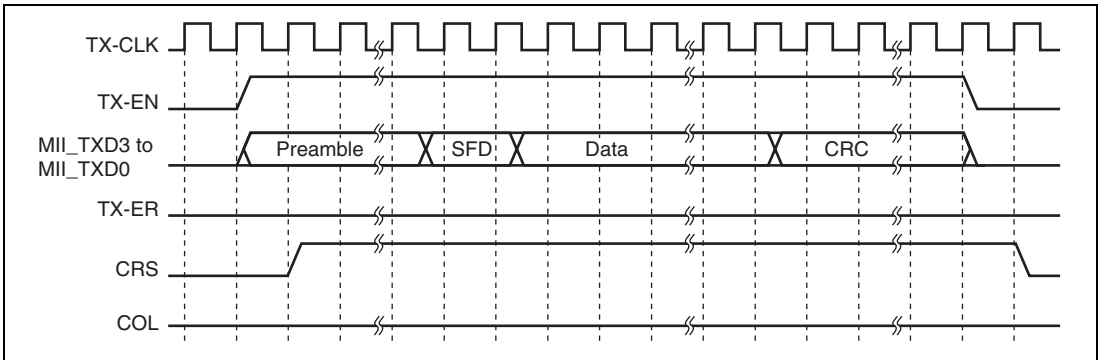


**Figure 19.3 EtherC Receiver State Transmissions**

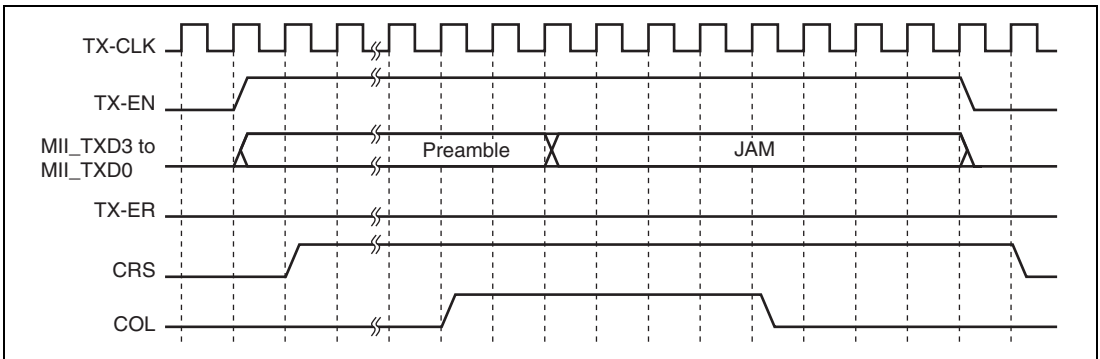
1. When the receive enable (RE) bit is set, the receiver enters the receive idle state.
2. When an SFD (start frame delimiter) is detected after a receive packet preamble, the receiver starts receive processing. Discards a frame with an invalid pattern.
3. In normal mode, if the destination address matches the receiver's own address, or if broadcast or multicast transmission or promiscuous mode is specified, the receiver starts data reception.
4. Following data reception from the MII, the receiver carries out a CRC check. The result is indicated as a status bit in the descriptor after the frame data has been written to memory. Reports an error status in the case of an abnormality.
5. After one frame has been received, if the receive enable bit is set (RE = 1) in the EtherC mode register, the receiver prepares to receive the next frame.

### 19.4.3 MII Frame Timing

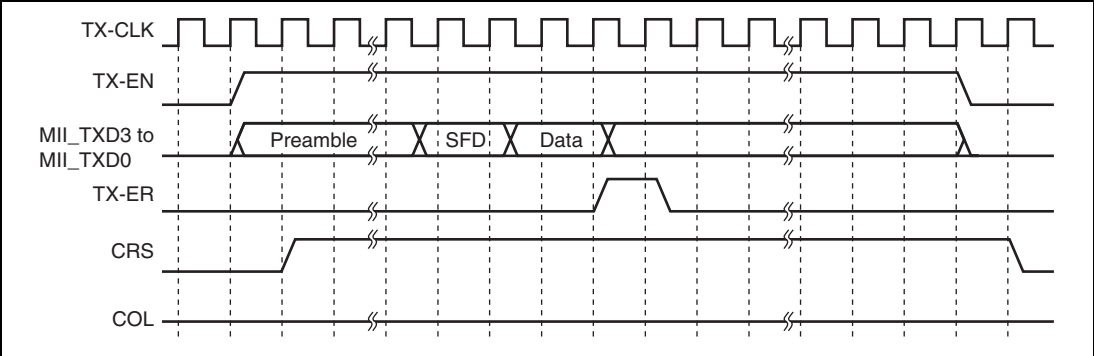
Each MII Frame timing is shown in figure 19.4.



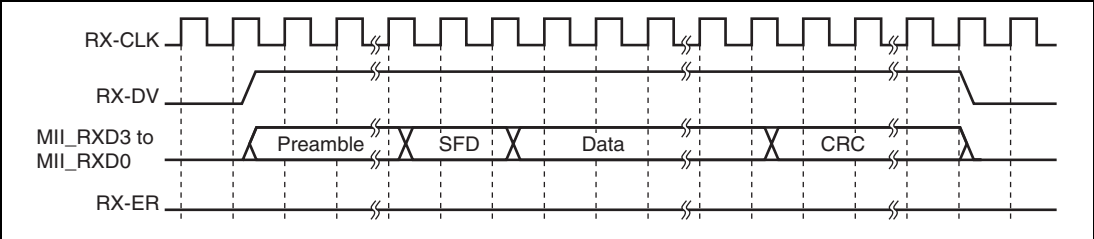
**Figure 19.4 (1) MII Frame Transmit Timing (Normal Transmission)**



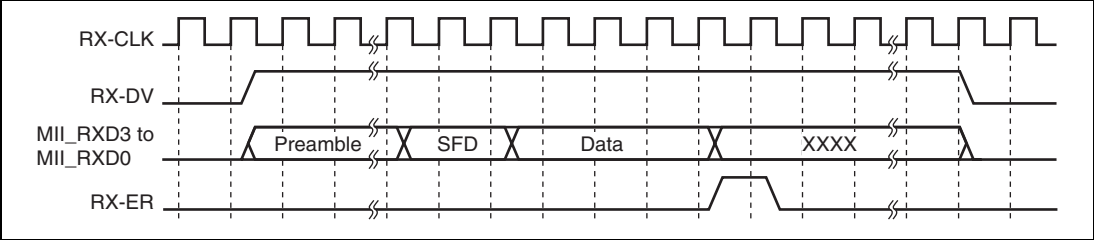
**Figure 19.4 (2) MII Frame Transmit Timing (Collision)**



**Figure 19.4 (3) MII Frame Transmit Timing (Transmit Error)**

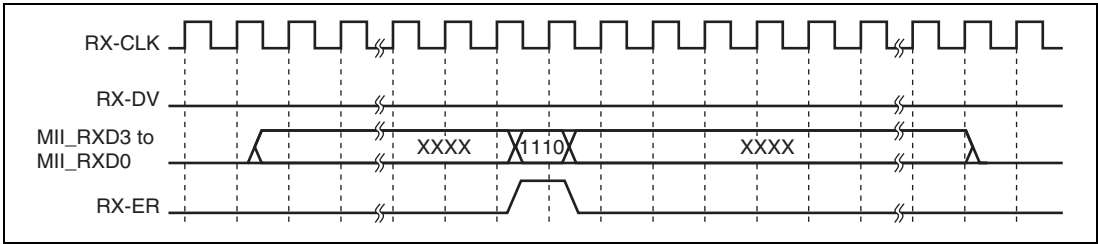


**Figure 19.4 (4) MII Frame Receive Timing (Normal Reception)**



**Figure 19.4 (5) MII Frame Receive Timing (Reception Error (1): Receive Error Notification)**





**Figure 19.4 (6) MII Frame Receive Timing (Reception Error (2): Carrier Error Notification)**

#### 19.4.4 Accessing MII Registers

MII registers in the PHY-LSI are accessed via this LSI's PHY interface register (PIR). Connection is made as a serial interface in accordance with the MII frame format specified in IEEE802.3u.

**MII Management Frame Format:** The format of an MII management frame is shown in figure 19.5. To access an MII register, a management frame is implemented by the program in accordance with the procedures shown in MII Register Access 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	00001	RRRRR	Z0	D..D	
Write	1..1	01	01	00001	RRRRR	10	D..D	X

[Legend]

PRE: 32 consecutive 1s

ST: Write of 01 indicating start of frame

OP: Write of code indicating access type

PHYAD: Write of 0001 if the PHY-LSI address is 1 (sequential write starting with the MSB).

This bit changes depending on the PHY-LSI address.

REGAD: Write of 0001 if the register address is 1 (sequential write starting with the MSB).

This bit changes depending on the PHY-LSI register address.

TA: Time for switching data transmission source on MII interface

(a) Write: 10 written

(b) Read: Bus release (notation: Z0) performed

DATA: 16-bit data. Sequential write or read from MSB

(a) Write: 16-bit data write

(b) Read: 16-bit data read

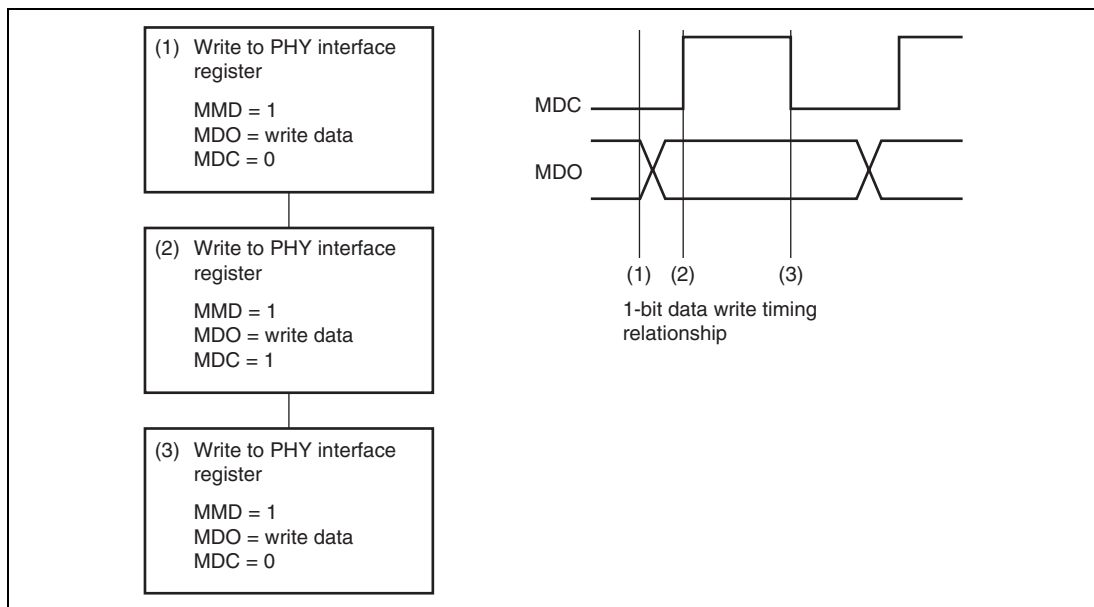
IDLE: Wait time until next MII management format input

(a) Write: Independent bus release (notation: X) performed

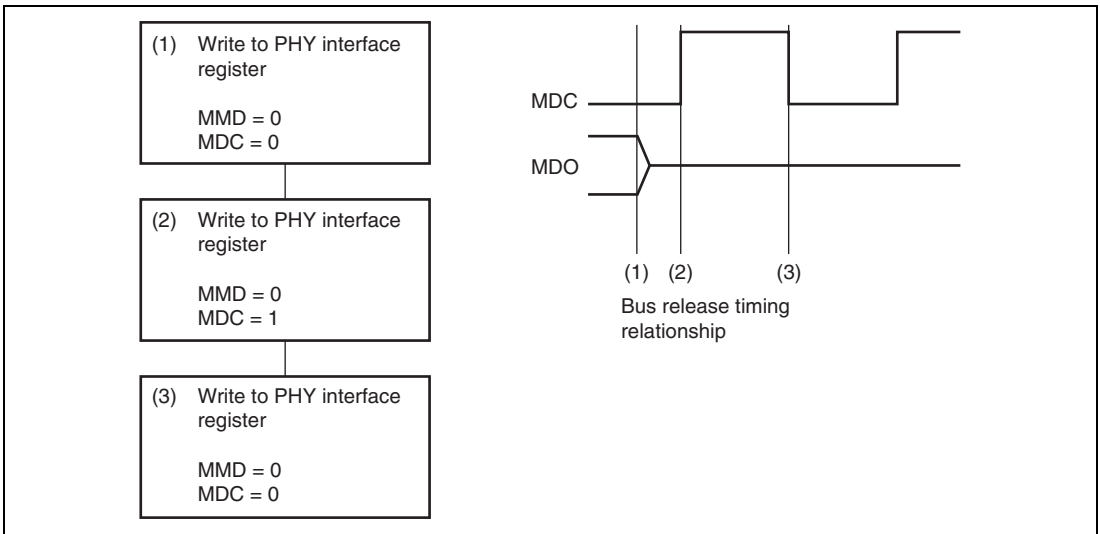
(b) Read: Bus already released in TA; control unnecessary

**Figure 19.5 MII Management Frame Format**

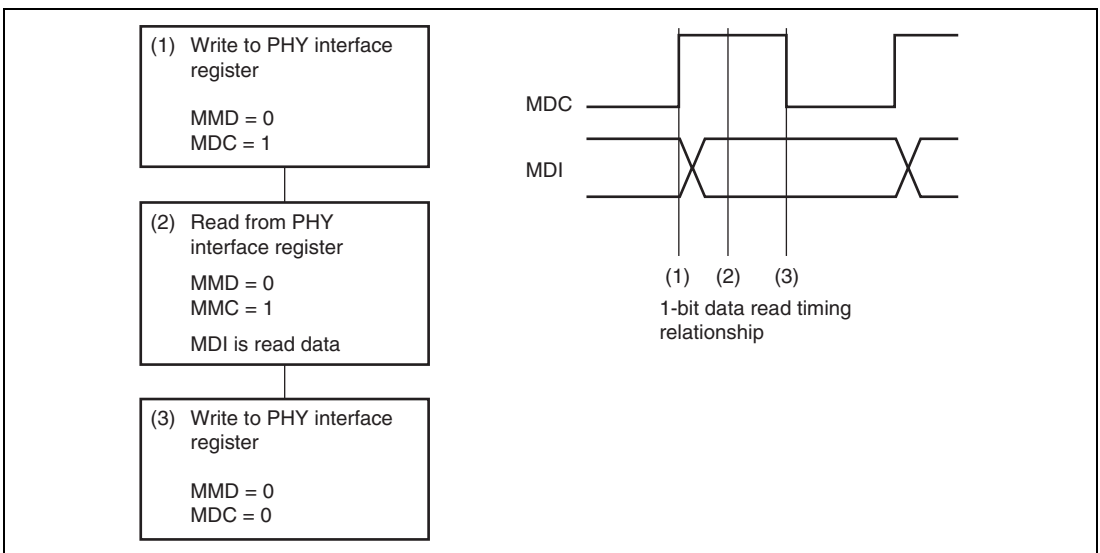
**MII Register Access Procedure:** The program accesses MII registers via the PHY interface register (PIR). Access is implemented by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. Figure 19.6 shows the MII register access timing. The timing will differ depending on the PHY-LSI type.



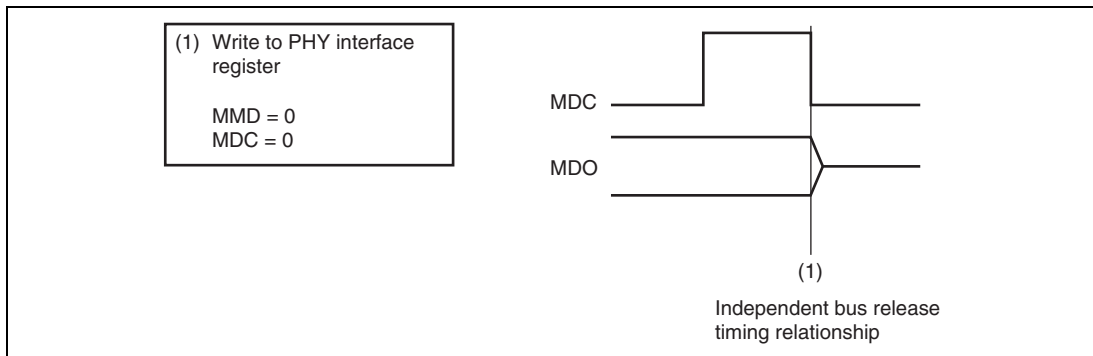
**Figure 19.6 (1) 1-Bit Data Write Flowchart**



**Figure 19.6 (2) Bus Release Flowchart (TA in Read in Figure 19.5)**



**Figure 19.6 (3) 1-Bit Data Read Flowchart**



**Figure 19.6 (4) Independent Bus Release Flowchart (IDLE in Write in Figure 19.5)**

### 19.4.5 Magic Packet Detection

The EtherC has a Magic Packet detection function. This function provides a Wake-On-LAN (WOL) facility that activates various peripheral devices connected to a LAN from the host device or other source. This makes it possible to construct a system in which a peripheral device receives a Magic Packet sent from the host device or other source, and activates itself. When the Magic Packet is detected, data is stored in the FIFO of the E-DMAC by the broadcast packet that has received data previously and the EtherC is notified of the receiving status. To return to normal operation from the interrupt processing, initialize the EtherC and E-DMAC by using SWR bit in the E-DMAC mode register (EDMR).

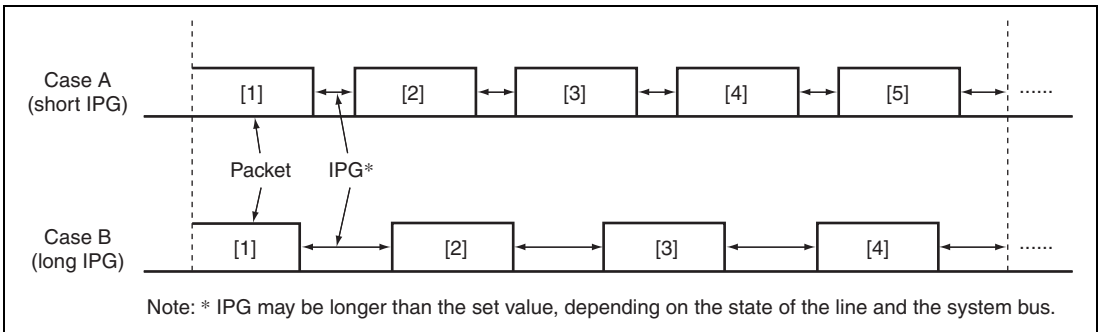
With a Magic Packet, reception is performed regardless of the destination address. As a result, this function is valid, and the WOL pin enabled, only in the case of a match with the destination address specified by the format in the Magic Packet. Further information on Magic Packets can be found in the technical documentation published by AMD Corporation.

The procedure for using the WOL function with this LSI is as follows.

1. Disable interrupt source output by means of the various interrupt enable/mask registers.
2. Set the Magic Packet detection enable bit (MPDE) in the EtherC mode register (ECMR).
3. Set the Magic Packet detection interrupt enable bit (MPDIP) in the EtherC interrupt enable register (ECSIPR) to the enable setting.
4. If necessary, set the CPU operating mode to sleep mode or set peripheral modules to module standby mode.
5. When a Magic Packet is detected, an interrupt is sent to the CPU. The WOL pin notifies peripheral LSIs that the Magic Packet has been detected.

### 19.4.6 Operation by IPG Setting

The EtherC has a function to change the non-transmission period IPG (Inter Packet Gap) between transmit frames. By changing the set values of the IPG setting register (IPGR), the transmission efficiency can be raised and lowered from the standard value. IPG settings are prescribed in IEEE802.3 standards. When changing settings, adequately check that the respective devices can operate smoothly on the same network.



**Figure 19.7 Changing IPG and Transmission Efficiency**

### 19.4.7 Flow Control

The EtherC supports flow control functions conforming to IEEE802.3x for full-duplex operation. The flow control can be applied to both receive and transmit operations. When transmitting PAUSE frames, flow control can be performed by the following two procedures :

#### (1) Automatic PAUSE Frame Transmission

For receive frames, PAUSE frames are automatically transmitted when the number of data written to the receive FIFO reaches the value set in FCFTR. The TIME parameter included in the PAUSE frame is set by APR. The automatic PAUSE frame transmission is repeated until the number of data in the receive FIFO becomes less than the value set in FCFTR as the receive data is read from the FIFO. Using TPAUSER, the upper limit of retransmission counts of the PAUSE frames can also be set. In this case, PAUSE frame transmission is repeated until the number of receive FIFO data becomes less than the FCFTR value, or the number of transmits reaches the value set by TPAUSER.

The automatic PAUSE frame transmission is enabled when the TXF bit in ECMR is 1.

## **(2) Manual PAUSE Frame Transmission**

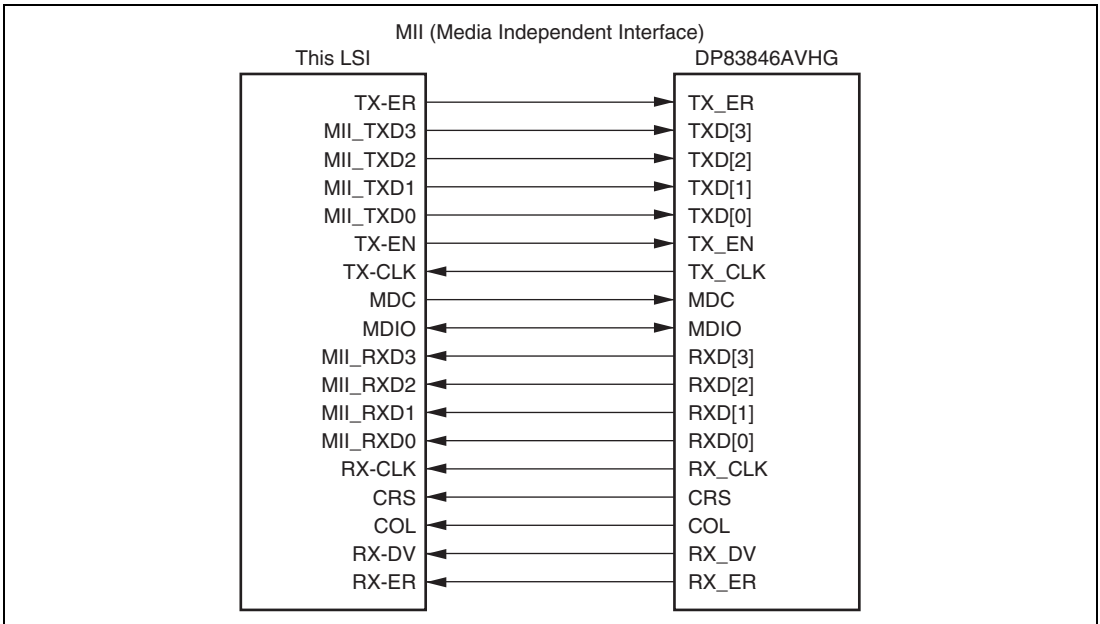
PAUSE frames are transmitted by directives from the software. When writing the Timer value to MPR, manual PAUSE frame transmission is started. With this method, PAUSE frame transmission is carried out only once.

## **(3) PAUSE Frame Reception**

The next frame is not transmitted until the time indicated by the Timer value elapses after receiving a PAUSE frame. However, the transmission of the current frame is continued. A received PAUSE frame is valid only when the RXF bit in ECMR is set to 1. The number of times of PAUSE frame receptions is counted.

## 19.5 Connection to LSI

Figure 19.8 shows the example of connection to a DP83846AVHG (National Semiconductor Corporation).



**Figure 19.8 Example of Connection to DP83846AVHG**

## 19.6 Usage Notes

Attention should be paid to the following when the EtherC is used.

### (1) Conditions for setting the LCHNG bit

The LCHNG bit in the ECSR register may be set even when the input level on the LNKSTA pin has not changed. It may be set when the LNKSTA pin is selected by the PSEL bit in the GPIO or when a high level is applied to the LNKSTA pin while the EtherC/E-DMAC is released from the software reset state by the SWR bit in the EDMR register.

This is because the LNKSTA signal is internally fixed low regardless of the external pin level when the LNKSTA pin is not selected by the GPIO or while the EtherC/E-DMAC is in the software reset state,.

In order not to request the LINK signal change interrupt accidentally, clear the LCHNG bit before setting the LCHNGIP bit in the ECSIPR register.



## Section 20 Ethernet Controller Direct Memory Access Controller (E-DMAC)

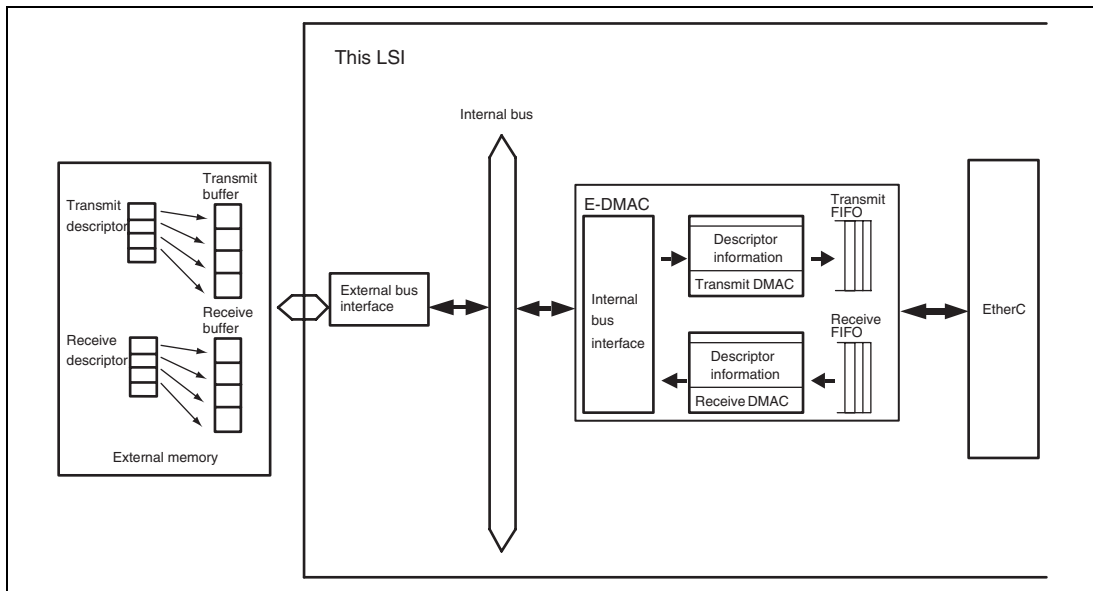
This LSI has an on-chip direct memory access controller (E-DMAC) directly connected to the Ethernet controller (EtherC). The E-DMAC controls the most part of the buffer management by using descriptors. This reduces the load on the CPU, thus enabling efficient data transmission and reception.

Figure 20.1 shows the configuration of the E-DMAC, and the descriptors and transmit/receive buffers in memory.

### 20.1 Features

The E-DMAC has the following features:

- The load on the CPU is reduced by means of a descriptor management system
- Transmit/receive frame status information is indicated in descriptors
- Achieves efficient system bus utilization through the use of DMA block transfer (32-byte units)
- Supports single-frame/multi-buffer operation
- Improves software performance by padding insertion in receive data.



**Figure 20.1 Configuration of E-DMAC, and Descriptors and Buffers**

## 20.2 Register Descriptions

Table 20.1 shows the configuration of registers of the E-DMAC. Table 20.2 shows the state of registers in each processing mode.

**Table 20.1 Register Configuration**

Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size
E-DMAC mode register	EDMR	R/W	H'FEF0 0000	H'1EF0 0000	32
E-DMAC transmit request register	EDTRR	R/W	H'FEF0 0008	H'1EF0 0008	32
E-DMAC receive request register	EDRRR	R/W	H'FEF0 0010	H'1EF0 0010	32
Transmit descriptor list start address register	TDLAR	R/W	H'FEF0 0018	H'1EF0 0018	32
Receive descriptor list start address register	RDLAR	R/W	H'FEF0 0020	H'1EF0 0020	32
EtherC/E-DMAC status register	EESR	R/W	H'FEF0 0028	H'1EF0 0028	32
EtherC/E-DMAC status interrupt permission register	EESIPR	R/W	H'FEF0 0030	H'1EF0 0030	32
Transmit/receive status copy enable register	TRSCER	R/W	H'FEF0 0038	H'1EF0 0038	32
Receive missed-frame counter register	RMFCR	R	H'FEF0 0040	H'1EF0 0040	32
Transmit FIFO threshold register	TFTR	R/W	H'FEF0 0048	H'1EF0 0048	32
FIFO depth register	FDR	R/W	H'FEF0 0050	H'1EF0 0050	32
Receiving method control register	RMCR	R/W	H'FEF0 0058	H'1EF0 0058	32
Transmit FIFO Underrun Counter	TFUCR	R/W	H'FEF0 0064	H'1EF0 0064	32
Receive FIFO Overflow Counter	RFOCR	R/W	H'FEF0 0068	H'1EF0 0068	32
Receive buffer write address register	RBWAR	R	H'FEF0 00C8	H'1EF0 00C8	32
Receive descriptor fetch address register	RDFAR	R	H'FEF0 00CC	H'1EF0 00CC	32
Transmit buffer read address register	TBRAR	R	H'FEF0 00D4	H'1EF0 00D4	32
Transmit descriptor fetch address register	TDFAR	R	H'FEF0 00D8	H'1EF0 00D8	32
Flow Control Start FIFO Threshold Setting Register	FCFTR	R/W	H'FEF0 0070	H'1EF0 0070	32
Receive Data Padding Insert Register	RPADIR	R/W	H'FEF0 0078	H'1EF0 0078	32

Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size
Transmit Interrupt Setting Register	TRIMD	R/W	H'FEF0 007C	H'1EF0 007C	32
Independent Output Signal Setting Register	IOSR	R/W	H'FEF0 006C	H'1EF0 006C	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 20.2 Register States in Each Operating Mode**

Name	Abbreviation	Software Reset
E-DMAC mode register	EDMR	Initialized
E-DMAC transmit request register	EDTRR	Initialized
E-DMAC receive request register	EDRRR	Initialized
Transmit descriptor list start address register	TDLAR	Retained
Receive descriptor list start address register	RDLAR	Retained
EtherC/E-DMAC status register	EESR	Initialized
EtherC/E-DMAC status interrupt permission register	EESIPR	Initialized
Transmit/receive status copy enable register	TRSCER	Initialized
Receive missed-frame counter register	RMFCR	Retained
Transmit FIFO threshold register	TFTR	Initialized
FIFO depth register	FDR	Initialized
Receiving method control register	RMCR	Initialized
Transmit FIFO Underrun Counter	TFUCR	Retained
Receive FIFO Overflow Counter	RFOCR	Retained
Receive buffer write address register	RBWAR	Initialized
Receive descriptor fetch address register	RDFAR	Initialized
Transmit buffer read address register	TBRAR	Initialized
Transmit descriptor fetch address register	TDFAR	Initialized
Flow Control Start FIFO Threshold Setting Register	FCFTR	Initialized
Receive Data Padding Insert Register	RPADIR	Initialized
Transmit Interrupt Setting Register	TRIMD	Initialized
Independent Output Signal Setting Register	IOSR	Initialized

## 20.2.1 E-DMAC Mode Register (EDMR)

EDMR is a 32-bit readable/writable register that specifies E-DMAC operating mode. This register should usually be set at initialization after a reset. If the EtherC and E-DMAC are initialized with this register during data transmission, abnormal data may be transmitted on the line. It is prohibited to modify the operating mode while transmission or reception function is enabled. Before modifying the operating mode, the EtherC and E-DMAC should be initialized by setting the software reset bit (SWR). Note that it takes 64 cycles of internal bus clock B $\phi$  for the EtherC and E-DMAC to be completely initialized. Therefore, the registers in the EtherC or E-DMAC should be accessed after that.

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	DL[1:0]	—	—	—	—	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	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	DE	0	R/W	Big/Little Endian Mode 0: Big endian (longword access) 1: Little endian (longword access) The setting applies to transmit and receive data. The setting does not apply to transmit/receive descriptors or registers (only big endian mode is available).
5, 4	DL[1:0]	00	R/W	Transmit/Receive Descriptor Length 00: 16 bytes (Initial value) 01: 32 bytes 10: 64 bytes 11: 16 bytes

Bit	Bit Name	Initial Value	R/W	Description
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SWR	0	R/W	Software Reset [Writing] 0: Disabled 1: Internal hardware is reset. For the registers that are reset, see tables 19.3 and 20.2.

### 20.2.2 E-DMAC Transmit Request Register (EDTRR)

EDTRR is a 32-bit readable/writable register that issues transmit directives to the E-DMAC. After having transmitted one frame, the E-DMAC reads the next descriptor. If the transmit descriptor valid bit in this descriptor is set (valid), the E-DMAC continues transmission. Otherwise, the E-DMAC clears the TR bit and stops the transmit DMAC operation.

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TR
Initial value:	0	0	0	0	0	0	0	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	TR	0	R/W	Transmit Request 0: Transmission-halted state. Writing 0 does not stop transmission. Termination of transmission is controlled by the TACT bit of the transmit descriptor. 1: Transmit DMA operation being performed by the E-DMAC. After writing 1 to this bit, the E-DMAC starts reading a transmit descriptor.

### 20.2.3 E-DMAC Receive Request Register (EDRRR)

EDRRR is a 32-bit readable/writable register that issues receive directives to the E-DMAC. After writing 1 to the RR bit in this register, the E-DMAC reads the receive descriptor. If the RACT bit of this receive descriptor is set to 1 (valid), the E-DMAC starts receive DMA transfer. When DMA transfer based on the first receive descriptor is completed, the E-DMAC reads the next receive descriptor. If the RACT bit of that receive descriptor is set to 1 (valid), the E-DMAC continues receive DMA operation. If the RACT bit of the receive descriptor is cleared to 0 (invalid), the E-DMAC clears the RR bit and stops receive DMA operation.

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

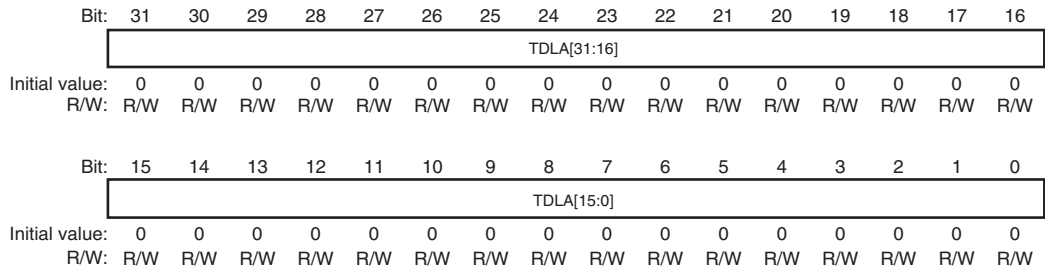
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

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	RR	0	R/W	Receive Request 0: Receiving function is disabled* 1: Receive descriptor is read, and the E-DMAC is ready to receive

Note: \* If the receiving function is disabled during frame reception, write-back is not performed successfully to the receive descriptor. Following pointers to read a receive descriptor become abnormal and the E-DMAC cannot operate successfully. In this case, to make E-DMAC reception enabled again, execute a software reset by the SWR bit in EDMR. To disable the E-DMAC receiving function without executing a software reset, specify the RE bit in EDCMR. Next, after the E-DMAC has completed the reception and write-back to the receive descriptor has been confirmed, disable the receiving function using this register.

20.2.4 Transmit Descriptor List Start Address Register (TDLAR)

TDLAR is a 32-bit readable/writable register that specifies the start address of the transmit descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bits in EDMR. This register must not be modified during transmission. Modifications to this register should only be made in the transmission-halted state specified by bits TR[1:0] (= 00) in the E-DMAC transmit request register (EDTRR).



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDLA[31:0]	All 0	R/W	Transmit Descriptor Start Address The lower bits are set according to the specified descriptor length. 16-byte boundary: TDLA[3:0] = 0000 32-byte boundary: TDLA[4:0] = 00000 64-byte boundary: TDLA[5:0] = 000000



## 20.2.5 Receive Descriptor List Start Address Register (RDLAR)

RDLAR is a 32-bit readable/writable register that specifies the start address of the receive descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bits in EDMR. This register must not be modified during reception. Modifications to this register should only be made while reception is disabled by the RR bit (= 0) in the E-DMAC receive request register (EDRRR).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDLA[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
	RDLA[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	RDLA[31:0]	All 0	R/W	Receive Descriptor Start Address  The lower bits are set according to the specified descriptor length.  16-byte boundary: RDLA[3:0] = 0000 32-byte boundary: RDLA[4:0] = 00000 64-byte boundary: RDLA[5:0] = 000000

### 20.2.6 E-MAC/E-DMAC Status Register (EESR)

EESR is a 32-bit readable/writable register that shows communications status information on the E-DMAC in combination with the E-MAC. The information in this register is reported in the form of interrupt sources. Individual bits are cleared by writing 1 (however, bit 22 (ECI) is a read-only bit that is not cleared by writing 1) and are not affected by writing 0. Each interrupt source can also be masked by means of the corresponding bit in the E-MAC/E-DMAC status interrupt permission register (EESIPR).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TWB	—	—	—	TABT	RABT	RFCOF	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
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/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CND	DLC	CD	TRO	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
Initial 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	R	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	TWB	0	R/W	Write-Back Complete  Indicates that write-back from the E-DMAC to the corresponding descriptor after frame transmission has completed. This operation is enabled only when the TIS bit in TRIMD is set to 1.  0: Write-back has not completed, or no transmission directive 1: Write-back has completed
29 to 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	TABT	0	R/W	<p>Transmit Abort Detect</p> <p>Indicates that the EtherC aborts transmitting a frame because of failures during frame transmission.</p> <p>0: Frame transmission has not been aborted or no transmission directive</p> <p>1: Frame transmission has been aborted</p>
25	RABT	0	R/W	<p>Receive Abort Detect</p> <p>Indicates that the EtherC aborts receiving a frame because of failures during frame reception.</p> <p>0: Frame reception has not been aborted or no reception directive</p> <p>1: Frame reception has been aborted</p>
24	RFCOF	0	R/W	<p>Receive Frame Counter Overflow</p> <p>Indicates that the frame counter in the receive FIFO has overflowed.</p> <p>0: Receive frame counter has not overflowed</p> <p>1: Receive frame counter has overflowed</p>
23	ADE	0	R/W	<p>Address Error</p> <p>Indicates that the memory address that the E-DMAC tried to transfer is found illegal.</p> <p>0: Illegal memory address not detected (normal operation)</p> <p>1: Illegal memory address detected</p> <p>Note: When an address error is detected, the E-DMAC halts transmitting/receiving. To resume the operation, execute a software reset with the SWR bit in EDMR.</p>
22	ECI	0	R	<p>EtherC Status Register Source</p> <p>This bit is a read-only bit. When the source of an ECSR interrupt is cleared, this bit is also cleared.</p> <p>0: EtherC status interrupt source has not been detected</p> <p>1: EtherC status interrupt source has been detected</p>

Bit	Bit Name	Initial Value	R/W	Description
21	TC	0	R/W	<p>Frame Transmit Complete</p> <p>Indicates that all the data specified by the transmit descriptor has been transmitted from the EtherC. This bit is set to 1, assuming the completion of transmission, when transmission of one frame is completed in single-frame/single-buffer operation or when the last data of a frame has been transmitted and the transmit descriptor valid bit (TACT) of the next descriptor is not set in for the processing of multi-buffer frame. After frame transmission, the E-DMAC writes the transmission status back to the relevant descriptor.</p> <p>0: Transfer not complete, or no transfer directive 1: Transfer complete</p>
20	TDE	0	R/W	<p>Transmit Descriptor Empty</p> <p>Indicates that the transmit descriptor valid bit (TACT) of a transmit descriptor read by the E-DMAC is not set if the previous descriptor does not represent the end of a frame in multi-buffer frame processing based on single-frame/multi-descriptor operation. As a result, an incomplete frame may be sent.</p> <p>0: Transmit descriptor active bit TACT = 1 detected 1: Transmit descriptor active bit TACT = 0 detected</p> <p>When transmit descriptor empty (TDE = 1) occurs, execute a software reset and initiate transmission. In this case, transmission starts from the address that is stored in the transmit descriptor list start address register (TDLAR).</p>
19	TFUF	0	R/W	<p>Transmit FIFO Underflow</p> <p>Indicates that an underflow has occurred in the transmit FIFO during frame transmission. Incomplete data is sent onto the line.</p> <p>0: Underflow has not occurred 1: Underflow has occurred</p>

Bit	Bit Name	Initial Value	R/W	Description
18	FR	0	R/W	<p>Frame Reception</p> <p>Indicates that a frame has been received and the receive descriptor has been updated. This bit is set to 1 each time a frame is received.</p> <p>0: Frame has not been received</p> <p>1: Frame has been received</p>
17	RDE	0	R/W	<p>Receive Descriptor Empty</p> <p>When receive descriptor empty (RDE = 1) occurs, reception can be resumed by setting the RACT bit (cleared to 0) of the receive descriptor to 1 and then restarting the receive operation.</p> <p>0: Receive descriptor active bit RACT = 1 detected</p> <p>1: Receive descriptor active bit RACT = 0 detected</p>
16	RFOF	0	R/W	<p>Receive FIFO Overflow</p> <p>Indicates that the receive FIFO has overflowed during frame reception.</p> <p>0: Overflow has not occurred</p> <p>1: Overflow has occurred</p>
15 to 12	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
11	CND	0	R/W	<p>Carrier Not Detect</p> <p>Indicates the carrier detection status during preamble transmission.</p> <p>0: A carrier is detected when transmission starts</p> <p>1: A carrier is not detected</p>
10	DLC	0	R/W	<p>Detect Loss of Carrier</p> <p>Indicates that loss of the carrier has been detected during frame transmission.</p> <p>0: Loss of carrier has not been detected</p> <p>1: Loss of carrier has been detected</p>

Bit	Bit Name	Initial Value	R/W	Description
9	CD	0	R/W	Delayed Collision Detect Indicates that a delayed collision has been detected during frame transmission. 0: Delayed collision has not been detected 1: Delayed collision has been detected
8	TRO	0	R/W	Transmit Retry Over Indicates that a retry-over condition has occurred during frame transmission. Total 16 transmission retries including 15 retries based on the back-off algorithm have failed after the E-MAC transmission starts. 0: Transmit retry-over condition not detected 1: Transmit retry-over condition detected
7	RMAF	0	R/W	Receive Multicast Address Frame 0: Multicast address frame has not been received 1: Multicast address frame has been received
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRF	0	R/W	Receive Residual-Bit Frame 0: Residual-bit frame has not been received 1: Residual-bit frame has been received
3	RTLF	0	R/W	Receive Too-Long Frame Indicates that a frame whose byte size exceeds the upper limit for the receive frame length set by RFLR in EtherC has been received. 0: Too-long frame has not been received 1: Too-long frame has been received
2	RTSF	0	R/W	Receive Too-Short Frame Indicates that a frame of fewer than 64 bytes has been received. 0: Too-short frame has not been received 1: Too-short frame has been received

Bit	Bit Name	Initial Value	R/W	Description
1	PRE	0	R/W	PHY-LSI Receive Error 0: PHY-LSI receive error has not been detected 1: PHY-LSI receive error has been detected
0	CERF	0	R/W	CRC Error on Received Frame 0: CRC error has not been detected 1: CRC error has been detected

## 20.2.7 E-MAC/E-DMAC Status Interrupt Permission Register (EESIPR)

EESIPR is a 32-bit readable/writable register that enables interrupts corresponding to individual bits in the E-MAC/E-DMAC status register (EESR). An interrupt is enabled by writing 1 to the corresponding bit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TWB IP	—	—	—	TABT IP	RABT IP	RFCOF IP	ADE IP	ECI IP	TC IP	TDE IP	TFUF IP	FR IP	RDE IP	RFOF IP
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/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
	—	—	—	—	CND IP	DLC IP	CD IP	TRO IP	RMAF IP	—	—	RRF IP	RTLF IP	RTSF IP	PRE IP	CERF IP
Initial 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	R	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	TWBIP	0	R/W	Write-Back Complete Interrupt Enable 0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled
29 to 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	TABTIP	0	R/W	Transmit Abort Detect Interrupt Enable 0: Transmit abort detect interrupt is disabled 1: Transmit abort detect interrupt is enabled
25	RABTIP	0	R/W	Receive Abort Detect Interrupt Enable 0: Receive abort detect interrupt is disabled 1: Receive abort detect interrupt is enabled
24	RFCOFIP	0	R/W	Receive Frame Counter Overflow Interrupt Enable 0: Receive frame counter overflow interrupt is disabled 1: Receive frame counter overflow interrupt is enabled
23	ADEIP	0	R/W	Address Error Interrupt Enable 0: Address error interrupt is disabled 1: Address error interrupt is enabled
22	ECIIP	0	R/W	EtherC Status Register Source Interrupt Enable 0: EtherC status interrupt is disabled 1: EtherC status interrupt is enabled
21	TCOIP	0	R/W	Frame Transmission Complete Interrupt Enable 0: Frame transmission complete interrupt is disabled 1: Frame transmission complete interrupt is enabled
20	TDEIP	0	R/W	Transmit Descriptor Empty Interrupt Enable 0: Transmit descriptor empty interrupt is disabled 1: Transmit descriptor empty interrupt is enabled
19	TFUFIP	0	R/W	Transmit FIFO Underflow Interrupt Enable 0: Underflow interrupt is disabled 1: Underflow interrupt is enabled
18	FRIP	0	R/W	Frame Reception Interrupt Enable 0: Frame reception interrupt is disabled 1: Frame reception interrupt is enabled
17	RDEIP	0	R/W	Receive Descriptor Empty Interrupt Enable 0: Receive descriptor empty interrupt is disabled 1: Receive descriptor empty interrupt is enabled



Bit	Bit Name	Initial Value	R/W	Description
16	RFOFIP	0	R/W	Receive FIFO Overflow Interrupt Enable 0: Overflow interrupt is disabled 1: Overflow interrupt is enabled
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	CNDIP	0	R/W	Carrier Not Detect Interrupt Enable 0: Carrier not detect interrupt is disabled 1: Carrier not detect interrupt is enabled
10	DLCIP	0	R/W	Detect Loss of Carrier Interrupt Enable 0: Detect loss of carrier interrupt is disabled 1: Detect loss of carrier interrupt is enabled
9	CDIP	0	R/W	Delayed Collision Detect Interrupt Enable 0: Delayed collision detect interrupt is disabled 1: Delayed collision detect interrupt is enabled
8	TROIP	0	R/W	Transmit Retry Over Interrupt Enable 0: Transmit retry over interrupt is disabled 1: Transmit retry over interrupt is enabled
7	RMAFIP	0	R/W	Receive Multicast Address Frame Interrupt Enable 0: Receive multicast address frame interrupt is disabled 1: Receive multicast address frame interrupt is enabled
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRFIP	0	R/W	Receive Residual-Bit Frame Interrupt Enable 0: Receive residual-bit frame interrupt is disabled 1: Receive residual-bit frame interrupt is enabled
3	RTLFIIP	0	R/W	Receive Too-Long Frame Interrupt Enable 0: Receive too-long frame interrupt is disabled 1: Receive too-long frame interrupt is enabled

Bit	Bit Name	Initial Value	R/W	Description
2	RTSFIP	0	R/W	Receive Too-Short Frame Interrupt Enable 0: Receive too-short frame interrupt is disabled 1: Receive too-short frame interrupt is enabled
1	PREIP	0	R/W	PHY-LSI Receive Error Interrupt Enable 0: PHY-LSI receive error interrupt is disabled 1: PHY-LSI receive error interrupt is enabled
0	CERFIP	0	R/W	CRC Error on Received Frame Interrupt Enable 0: CRC error interrupt is disabled 1: CRC error interrupt is enabled

### 20.2.8 Transmit/Receive Status Copy Enable Register (TRSCER)

TRSCER specifies whether the information for the transmit and receive state reported by bits in the E-MAC/E-DMAC status register (EESR) is to be reflected in the TFS25 to TFS0 or RFS26 to RFS0 bits of the corresponding descriptor. The bits in this register correspond to bits 11 to 0 in EESR. When a bit is cleared to 0, the transmit status (bits 11 to 8 in EESR) is reflected in the TFS3 to TFS0 bits of the transmit descriptor, and the receive status (bits 7 to 0 in EESR) is reflected in the RFS7 to RFS0 bits of the receive descriptor. When a bit is set to 1, the occurrence of the corresponding source is not reflected in the descriptor. After this LSI is reset, all bits are 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
	—	—	—	—	CND CE	DLC CE	CD CE	TRO CE	RMAF CE	—	—	RRF CE	RTLF CE	RTSF CE	PRE CE	CERF CE
Initial 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	R	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	CNDCE	0	R/W	CND Bit Copy Directive 0: Reflects the CND bit status in the TFS bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFS bit of the transmit descriptor
10	DLCCE	0	R/W	DLC Bit Copy Directive 0: Reflects the DLC bit status in the TFE bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFE bit of the transmit descriptor
9	CDCE	0	R/W	CD Bit Copy Directive 0: Reflects the CD bit status in the TFE bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFE bit of the transmit descriptor
8	TROCE	0	R/W	TRO Bit Copy Directive 0: Reflects the TRO bit status in the TFE bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFE bit of the transmit descriptor
7	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Reflects the RMAF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRFCE	0	R/W	RRF Bit Copy Directive 0: Reflects the RRF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor

Bit	Bit Name	Initial Value	R/W	Description
3	RTLFCE	0	R/W	RTLF Bit Copy Directive 0: Reflects the RTLF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
2	RTSFCE	0	R/W	RTSF Bit Copy Directive 0: Reflects the RTSF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
1	PRECE	0	R/W	PRE Bit Copy Directive 0: Reflects the PRE bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
0	CERFCE	0	R/W	CERF Bit Copy Directive 0: Reflects the CERF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor

## 20.2.9 Receive Missed-Frame Counter Register (RMFCR)

RMFCR is a 16-bit counter that indicates the number of frames that could not be saved in the receive buffer and so were discarded during reception. When the receive FIFO overflows, the receive frames in the FIFO are discarded. The number of frames discarded at this time is counted. When the value in this register reaches H'FFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any 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
	MFC[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 16	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
15 to 0	MFC[15:0]	All 0	R	Missed-Frame Counter  These bits indicate the number of frames that are discarded and not transferred to the receive buffer during reception.

### 20.2.10 Transmit FIFO Threshold Register (TFTR)

TFTR is a 32-bit readable/writable register that specifies the transmit FIFO threshold at which the first transmission is started. The actual threshold is 4 times the set value. The EtherC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes specified by this register, when the transmit FIFO is full, or when one frame of data write is performed.

When setting this register, do so in the transmission-halt state.

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
	—	—	—	—	—	TFT[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	R/W	Description
31 to 11	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 0	TFT[10:0]	All 0	R/W	<p>Transmit FIFO Threshold</p> <p>A value and smaller than the FIFO size specified by FDR must be set as the transmit FIFO threshold.</p> <p>H'000: Store and forward modes</p> <p>H'001 to H'00C: Setting prohibited</p> <p>H'00D: 52 bytes</p> <p>H'00E: 56 bytes</p> <p>: :</p> <p>H'01F: 124 bytes</p> <p>H'020: 128 bytes</p> <p>: :</p> <p>H'03F: 252 bytes</p> <p>H'040: 256 bytes</p> <p>: :</p> <p>H'07F: 508 bytes</p> <p>H'080: 512 bytes</p> <p>: :</p> <p>H'0FF: 1,020 bytes</p> <p>H'100: 1,024 bytes</p> <p>: :</p> <p>H'1FF: 2,044 bytes</p> <p>H'200: 2,048 bytes</p> <p>H'201 to H'7FF: Setting prohibited</p>

- Notes:
1. When starting transmission before one frame of data write has completed, take care no underflow occurs.
  2. Operation cannot be guaranteed when the value set in this register is greater than the transmit FIFO size.
  3. To prevent a transmit underflow, setting the initial value (store and forward modes) is recommended.

### 20.2.11 FIFO Depth Register (FDR)

FDR is a 32-bit readable/writable register that specifies the sizes 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
	—	—	—	TFD[4:0]					—	—	—	RFD[4:0]				
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
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	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 to 8	TFD[4:0]	B'00111	R/W	Transmit FIFO Size Specifies the size of the transmit FIFO. The setting must not be changed after transmission/reception has started. 00000: 256 bytes 00001: 512 bytes 00010: 768 bytes 00011: 1024 bytes 00100: 1280 bytes 00101: 1536 bytes 00110: 1792 bytes 00111: 2048 bytes Other than above: Setting prohibited
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 to 0	RFD[4:0]	B'00111	R/W	<p>Receive FIFO Size</p> <p>Specifies the size of the receive FIFO. The setting must not be changed after transmission/reception has started.</p> <p>00000: 256 bytes</p> <p>00001: 512 bytes</p> <p>00010: 768 bytes</p> <p>00011: 1024 bytes</p> <p>00100: 1280 bytes</p> <p>00101: 1536 bytes</p> <p>00110: 1792 bytes</p> <p>00111: 2048 bytes</p> <p>Other than above: Setting prohibited</p>

Note: Operation cannot be guaranteed when the value set in this register is greater than the transmit FIFO size.

## 20.2.12 Receiving Method Control Register (RMCR)

RMCR is a 32-bit readable/writable register that specifies the control method for the RE bit in ECMR while a frame is received. This register must be set during the receiving-halted state.

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RNC	RNR
Initial value:	0	0	0	0	0	0	0	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	RNC	0	R/W	Receive Start Bit Non-Reset Mode 0: nop 1: Allows the software to reset the receive start bit (RR) in EDRRR. In this case, even when the RACT bit in the fetched descriptor is 0 (receive descriptor empty), the receive start bit (RR) in EDRRR is not automatically reset and the receive descriptor is continuously fetched to continue DMA transfers of the receive frames.
0	RNR	0	R/W	Receive Start Bit Reset 0: Allows the hardware to reset the receive start bit (RR) in EDRRR automatically upon completion of reception of one frame. Control is possible for each frame. To receive the subsequent receive frame, the receive start bit in EDRRR needs to be set again. 1: Allows the higher-level software to control the receive start bit (RR) in EDRRR. Once the receive start bit (RR) in EDRRR is set to 1, the hardware continues to fetch the receive descriptor and receive frames automatically until the RR bit in EDRRR is cleared to 0. In other words, continuous reception of multiple frames are possible. It is recommended to set this bit to 1 when continuous reception is used. However, when a receive descriptor empty is detected, the hardware clears the RR bit in EDRRR automatically.

### 20.2.13 Transmit FIFO Underrun Counter (TFUCR)

TFUCR is a register that indicates the count of underruns having occurred in the transmit FIFO. The count value is cleared to 0 by writing any value to 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
	UNDER[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 These bits are always read as 0. The write value should always be 0.
15 to 0	UNDER [15:0]	All 0	R/W	Transmit FIFO Underflow Count Indicates the count of underflows having occurred in the transmit FIFO. The counter stops when the count value reaches H'FFFF.

**20.2.14 Receive FIFO Overflow Counter (RFOCR)**

RFOCR is a register that indicates the count of overflows having occurred in the receive FIFO. The count value is cleared to 0 by writing any value to 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
	OVER[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 These bits are always read as 0. The write value should always be 0.
15 to 0	OVER[15:0]	All 0	R/W	Receive FIFO Overflow Count Indicates the count of overflows having occurred in the receive FIFO. The counter stops when the count value reaches H'FFFF.

### 20.2.15 Receive Buffer Write Address Register (RBWAR)

RBWAR stores the address of data to be written in the receiving buffer when the E-DMAC writes data to the receiving buffer. Which addresses in the receiving buffer are processed by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address that the E-DMAC is actually processing may be different from the value read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RBWA[31:16]															
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
	RBWA[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 0	RBWA[31:0]	All 0	R	Receiving-Buffer Write Address
				These bits can only be read. Writing is prohibited.

### 20.2.16 Receive Descriptor Fetch Address Register (RDFAR)

RDFAR stores the descriptor start address that is required when the E-DMAC fetches descriptor information from the receive descriptor. Which receive descriptor information is used for processing by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address from which the E-DMAC is actually fetching a descriptor may be different from the value read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDFA[31:16]															
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
	RDFA[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 0	RDFA[31:0]	All 0	R	Receive Descriptor Fetch Address
				Writing to these bits during the reception is prohibited.

### 20.2.17 Transmit Buffer Read Address Register (TBRAR)

TBRAR stores the address of the transmission buffer when the E-DMAC reads data from the transmission buffer. Which addresses in the transmission buffer are processed by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address from which the E-DMAC is actually reading in the buffer may be different from the value read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TBRA[31:16]															
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
	TBRA[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 0	TBRA[31:0]	All 0	R	Transmission-Buffer Read Address
				These bits can only be read. Writing is prohibited.

## 20.2.18 Transmit Descriptor Fetch Address Register (TDFAR)

TDFAR stores the descriptor start address that is required when the E-DMAC fetches descriptor information from the transmit descriptor. Which transmit descriptor information is used for processing by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address from which the E-DMAC is actually fetching a descriptor may be different from the value read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDFAR[31:16]															
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
	TDFAR[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 0	TDFAR[31:0]	All 0	R	Transmit Descriptor Fetch Address
				Writing to these bits during transmission is prohibited.



### 20.2.19 Flow Control Start FIFO Threshold Setting Register (FCFTR)

FCFTR is a 32-bit readable/writable register that sets the flow control of the EtherC (sets the threshold of automatic PAUSE output). The threshold can be set in terms of the data size in the receive FIFO (RFDO[2:0]) and the number of receive frames (RFFO[2:0]). Flow control is turned on when either of the data size in the receive FIFO or the number of receive frames is determined as the threshold value.

If the same receive FIFO size as set by the FIFO depth register (FDR) is set when flow control is to be turned on according to the RFDO setting condition, flow control is turned on with (FIFO data size – 64) bytes. For instance, when the RFD bits in FDR = 1 and the RFDO bits in this register = 1, flow control is turned on when (2,048 – 64) bytes of data is stored in the receive FIFO. The value set in the RFDO bits in this register should be equal to or less than the value set in the RFD bits in FDR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFO[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
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFDO[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	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 to 16	RFFO[2:0]	111	R/W	Receive Frame Count Overflow BSY Output Threshold 000: When two receive frames have been stored in the receive FIFO. 001: When four receive frames have been stored in the receive FIFO. 010: When six receive frames have been stored in the receive FIFO. : 110: When 14 receive frames have been stored in the receive FIFO. 111: When 16 receive frames have been stored in the receive FIFO.
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	RFDO[2:0]	111	R/W	Receive FIFO Overflow BSY Output Threshold 000: When (256 – 32) bytes of data is stored in the receive FIFO. 001: When (512 – 32) bytes of data is stored in the receive FIFO. : 110: When (1792 – 32) bytes of data is stored in the receive FIFO. 111: When (2048 – 32) bytes of data is stored in the receive FIFO.

### 20.2.20 Receive Data Padding Insert Register (RPADIR)

RPADIR is a 32-bit readable/writable register that sets padding insertion in receive data. Before modifying the settings of this register, execute a software reset by means of the SWR bit in the E-DMAC mode register (EDMR).

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	PADS[1:0]	All 0	R/W	Padding Size 00: No padding insertion 01: 1-byte insertion 10: 2-byte insertion 11: 3-byte insertion
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	PADR[5:0]	All 0	R/W	Padding Slot H'00: Inserts the specified padding size immediately before the first byte of the receive data. H'01: Inserts the specified padding size immediately before the second byte of the receive data. : H'3E: Inserts the specified padding size immediately before the 63rd byte of the receive data. H'3F: Inserts the specified padding size immediately before the 64th byte of the receive data.

### 20.2.21 Transmit Interrupt Setting Register (TRIMD)

TRIMD is a 32-bit readable/writable register that specifies whether to notify write-back completion of each frame during transmission by means of the TWB bit in EESR or the interrupt

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	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	TIM	0	R/W	Transmit Interrupt Mode 0: Per-transmit-frame mode An interrupt is notified upon write-back completion of each frame. 1: Interrupt mode An interrupt is notified upon write-back completion of the transmit descriptor with the TWBI bit set to 1.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TIS	0	R/W	Transmit Interrupt Setting 0: An interrupt is not notified in the mode selected by the TIM bit. When this bit is 0, the TIM bit setting is invalid. 1: An interrupt is notified by setting the TWB bit in EESR to 1 in the mode selected by the TIM bit.

### 20.2.22 Independent Output Signal Setting Register (IOSR)

The value set in the ELB bit in this register is directly output via the general external output pin (EXOUT) of this LSI. The EXOUT pin can be used to specify loopback mode for the PHY-LSI. To use the loopback function of the PHY-LSI through this register, the PHY-LSI needs to be provided with the pin to be connected to the EXOUT pin.

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ELB
Initial value:	0	0	0	0	0	0	0	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	ELB	0	R/W	External Loopback Mode 0: The EXOUT pin outputs a low level signal. 1: The EXOUT pin outputs a high level signal.

## 20.3 Operation

The E-DMAC, connected to the EtherC, allows efficient transfer of transmit/receive data between the EtherC and memory (buffers) without CPU intervention. The E-DMAC automatically reads the control information referred to as descriptors. The descriptors corresponding to each buffer hold buffer pointers and other information. The E-DMAC reads transmit data from the transmit buffer and writes receive data to the receive buffer according to the control information. By arranging such multiple descriptors continuously (i.e., making a descriptor list), continuous transmission or reception is possible.

### 20.3.1 Descriptor Lists and Data Buffers

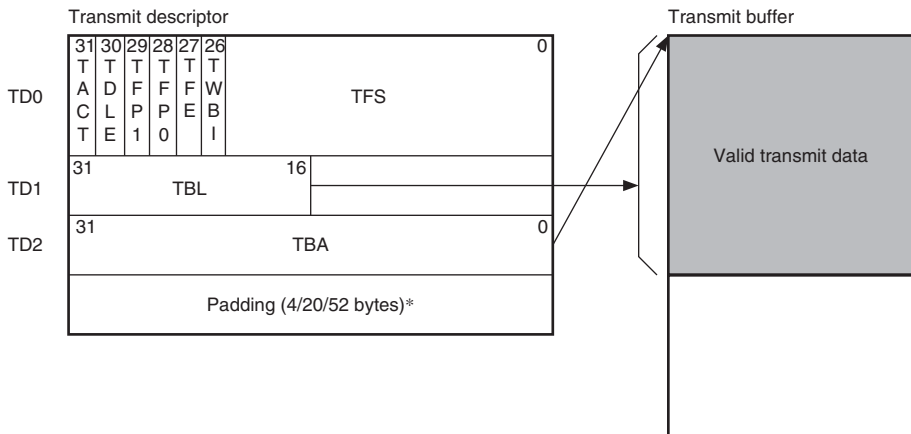
By the communication program, a transmit descriptor list and a receive descriptor list should be created in memory space prior to transmission and reception. The start addresses of these lists should be set to the transmit descriptor list start address register and receive descriptor list start address register.

The start addresses of the descriptor lists should be placed on the address boundaries in accordance with the descriptor length specified by the E-DMAC mode register (EDMR). Here, the start address of the transmit buffer can be placed on a longword, word, or byte boundary.

#### (1) Transmit Descriptor

Figure 20.2 shows the relationship between a transmit descriptor and a transmit buffer. The descriptor can relate one transmit frame to one transmit buffer (single-frame/single-buffer operation) or multiple transmit buffers (single-frame/multi-buffer operation).

When the transmit buffer length (TBL) is to be set to 1 to 16 bytes, the buffer address needs to be placed on a 32-byte boundary. When the transmit buffer length (TBL) is set to 0 byte, operation cannot be guaranteed.



Note: \* According to the descriptor length (16/32/64 bytes), the padding size is determined.

**Figure 20.2 Relationship between Transmit Descriptor and Transmit Buffer**

**(a) Transmit Descriptor 0 (TD0)**

TD0 indicates the transmit frame status informing frame transmission status.

(The underlined bits are subject to write-back in the table below.)

Bit	Bit Name	Initial Value	R/W	Description
<u>31</u>	<u>TACT</u>	0	R/W	Transmit Descriptor Valid Indicates that the corresponding descriptor is valid. This bit is set to 1 by software. This bit is cleared to 0 by hardware when a transmit frame has been completely transferred or when transmission has been aborted due to some cause.

Bit	Bit Name	Initial Value	R/W	Description
30	TDLE	0	R/W	<p>Transmit Descriptor Ring End</p> <p>When set to 1, this bit indicates that the corresponding descriptor is the last one of the descriptor ring.</p>
29	TFP1	0	R/W	Transmit Frame Positions 1 and 0
28	TFP0	0	R/W	<p>These bits relate the transmit buffer to the transmit frame. The settings of these bits and the TBL bits should be logically correct in the consecutive descriptors.</p> <p>00: Transmission of the frame of the transmit buffer specified by this descriptor is continued. (The frame is incomplete.)</p> <p>01: The transmit buffer specified by this descriptor contains the end of the frame (The frame is complete.)</p> <p>10: The transmit buffer specified by this descriptor is the start of the frame (The frame is incomplete.)</p> <p>11: The contents in the transmit buffer specified by this descriptor correspond to one frame (single-frame/single-buffer).</p>



Bit	Bit Name	Initial Value	R/W	Description
<u>27</u>	<u>TFE</u>	0	R/W	<p>Transmit Frame Error</p> <p>When set to 1, this bit indicates that an error is indicated by any of the TFS bits. (By so setting TRSCER, it is possible to prevent this bit from being set by an event indicated by TFS7 to TFS0. It is impossible, however, if an event indicated by TFS7 to TFS0 also causes TFS8 to be set.)</p> <p>1: Frame transmission has been aborted.</p>
26	TWBI	0	R/W	<p>Write-Back Completion Interrupt Notification</p> <p>(This bit is valid when TRIMD is set so.)</p> <p>0: nop</p> <p>1: An interrupt is generated upon completion of write-back to this descriptor.</p>
25 to 0	TFS	All 0	R/W	<p>Transmit Frame Status</p> <p>TFS25 to TFS9 [Reserved (The write value should always be 0.)]:</p> <p>TFS8 [Detect Transmit Abort];</p> <p>When set to 1, this bit indicates that the abort signal is set to 1 during frame transmission. (causing TFE to be set)</p> <p>TFS7 to TFS4 [Reserved (The write value should always be 0.)];</p> <p>TFS3 [Detect of No Carrier (corresponding to the CND bit in EESR)];</p> <p>TFS2 [Detect Loss of Carrier (corresponding to the DLC bit in EESR)];</p> <p>TFS1 [Detect of Delayed Collision during Transmission (corresponding to the CD bit in EESR)];</p> <p>TFS0 [Transmit Retry Over (corresponding to the TRO bit in EESR)];</p> <p>When set to 1, these bits indicate that TFS8 to TFS1 have been set to 1 during frame transmission. (Although TFE is normally set when these bits are set to 1, it can be prevented from being set by so setting TRSCER.)</p>

**(b) Transmit Descriptor 1 (TD1)**

TD1 indicates the length of the transmit buffer.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	TBL	All 0	R/W	Transmit Buffer Length Indicates the length of the relevant transmit buffer in terms of valid bytes.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

**(c) Transmit Descriptor 2 (TD2)**

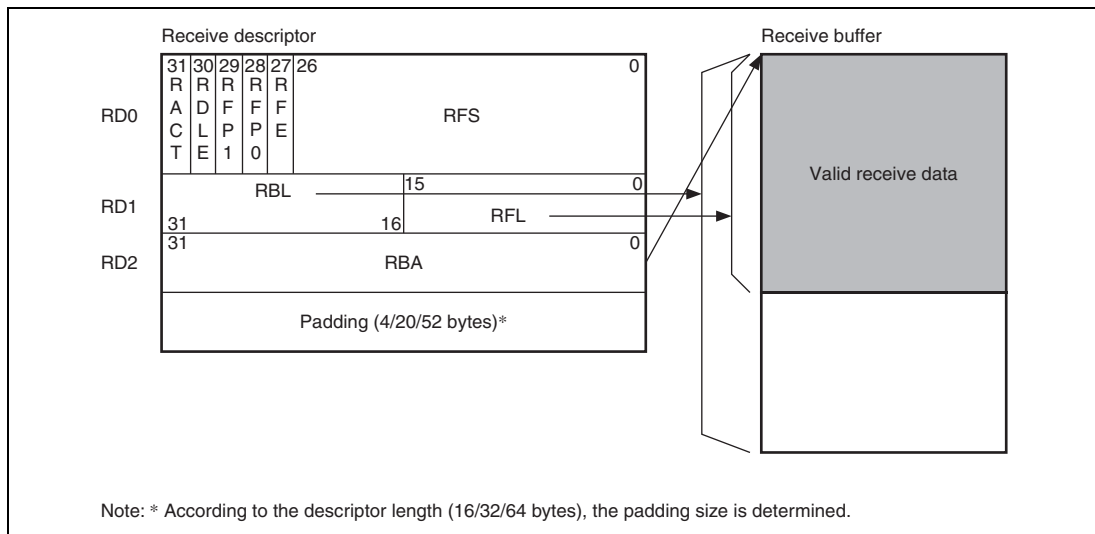
TD2 indicates the start address of the relevant transmit buffer.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TBA	All 0	R/W	Transmit Buffer Address Indicates the start address of the transmit buffer.

## (2) Receive Descriptor

Figure 20.3 shows the relationship between a receive descriptor and a receive buffer. The receive buffer address is to be placed on a 32-byte boundary.

When the receive buffer length (RBL) is set to 0 byte, operation specified by the descriptor cannot be guaranteed.



**Figure 20.3 Relationship between Receive Descriptor and Receive Buffer**

**(a) Receive Descriptor 0 (RD0)**

RD0 indicates the receive frame status informing frame reception status.

(The underlined bits are subject to write-back in the table below.)

Bit	Bit Name	Initial Value	R/W	Description
<u>31</u>	<u>RACT</u>	0	R/W	<p>Receive Descriptor Valid</p> <p>Indicates that the corresponding descriptor is valid. This bit is set to 1 by software. This bit is cleared to 0 by hardware when an entire receive frame has been completely transferred to the buffer address specified by RD2 or when the receive buffer becomes full.</p>
30	RDLE	0	R/W	<p>Receive Descriptor Ring End</p> <p>When set to 1, this bit indicates that the corresponding descriptor is the last one of the descriptor ring.</p>
<u>29, 28</u>	<u>RFP[1:0]</u>	00	R/W	<p>Receive Frame Positions 1 and 0</p> <p>These bits relate the receive buffer to the receive frame. The settings of these bits and the TBL bits should be logically correct in the consecutive descriptors.</p> <p>00: Reception of the frame of the receive buffer specified by this descriptor is continued. (The frame is incomplete.)</p> <p>01: The receive buffer specified by this descriptor contains the end of the frame (The frame is complete.)</p> <p>10: The receive buffer specified by this descriptor is the start of the frame (The frame is incomplete.)</p> <p>11: The contents in the receive buffer specified by this descriptor correspond to one frame (single-frame/single-buffer).</p>
<u>27</u>	<u>RFE</u>	0	R/W	<p>Receive Frame Error</p> <p>When set to 1, this bit indicates that an error is indicated by any of the RFS bits. (By so setting TRSCER, it is possible to prevent this bit from being set by an event indicated by RFS7 to RFS0. It is impossible, however, if an event indicated by RFS7 to RFS0 also causes RFS8 to be set.)</p>

Bit	Bit Name	Initial Value	R/W	Description
<u>26 to 0</u>	<u>RFS</u>	All 0	R/W	<p>Receive Frame Status</p> <p>RF26 to RF10 [Reserved (The write value should always be 0.)];</p> <p>RFS9 [Receive FIFO Overflow (corresponding to the RFOF bit in EESR)]:</p> <p>When set to 1, this bit indicates that a receive FIFO overflow has occurred terminating the frame halfway and that the frame has been written back. (causing RFE to be set)</p> <p>TFS8 [Detect Receive Abort];</p> <p>When set to 1, this bit indicates that the abort signal is set to 1 during frame transmission. (causing RFE to be set)</p> <p>RFS7 [Multicast address frame received (corresponding to the RMAF bit in EESR)];</p> <p>RFS6 and RFS5 [Reserved (The write value should always be 0.)];</p> <p>RFS4 [Residual-bit frame receive error (corresponding to the RRF bit in EESR)];</p> <p>RFS3 [Long frame receive error (corresponding to the RTLf bit in EESR)];</p> <p>RFS2 [Short frame receive error (corresponding to the RTSF bit in EESR)];</p> <p>RFS1 [PHY-LSI receive error (corresponding to the PRE bit in EESR)];</p> <p>RFS0 [CRC error detected in receive frame (corresponding to the CERF bit in EESR)]:</p> <p>When set to 1, these bits indicate that RFS8 to RFS1 have been set to 1 during frame reception. (Although RFE is normally set when these bits are set to 1, it can be prevented from being set by so setting TRSCER.)</p>

**(b) Receive Descriptor 1 (RD1)**

RD1 indicates the length of the receive buffer.

(The underlined bits are subject to write-back in the table below.)

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	RBL	All 0	R/W	Receive Buffer Length Indicates the length of the relevant receive buffer in terms of bytes. The buffer length should be set as $n$ in $32 \times n$ , which represents the buffer size.
<u>15 to 0</u>	<u>RFL</u>	All 0	R	Receive Data Length Indicates the length of (number of bytes in) a receive frame stored in the buffer The number of bytes for padding insertion specified by RPADIR are excluded. These bits are written back to the descriptor containing the end of a frame.

**(c) Receive Descriptor 2 (RD2)**

RD2 indicates the start address of the relevant receive buffer.

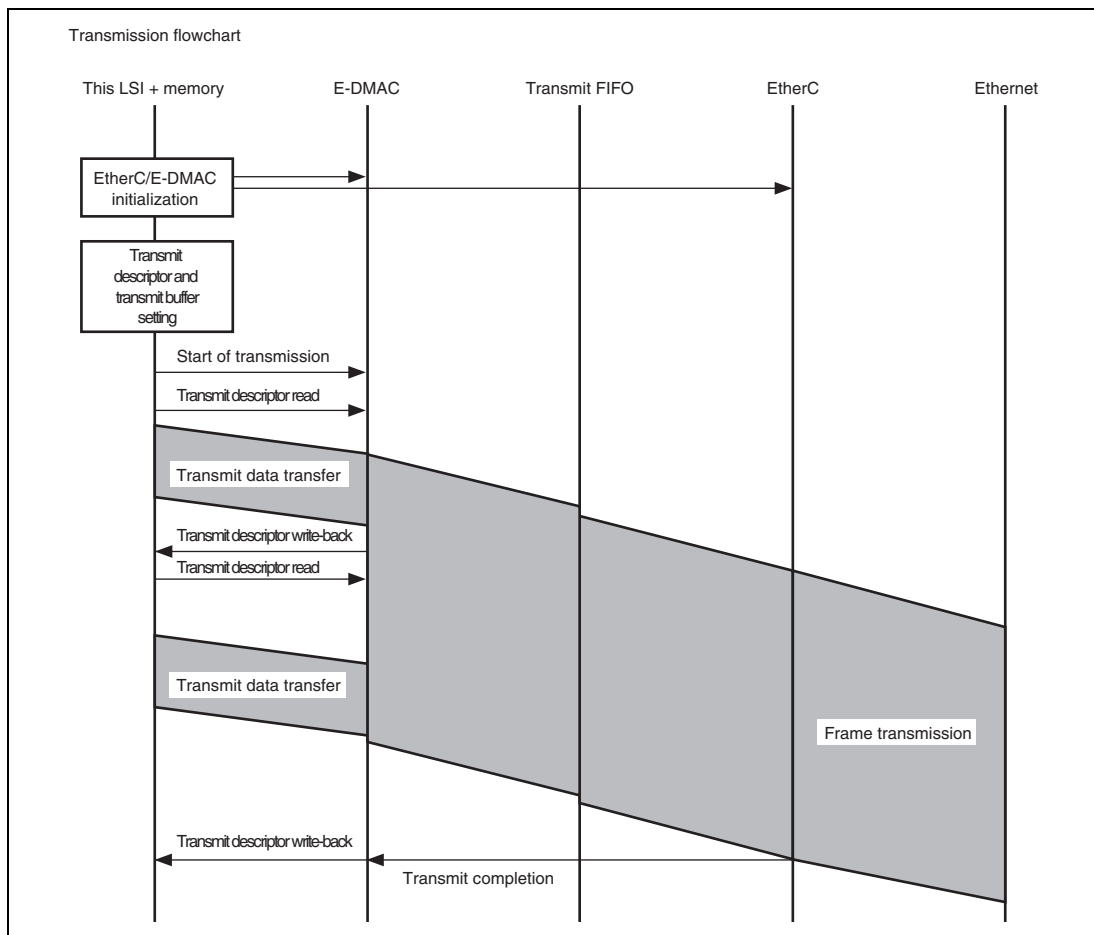
Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RBA	All 0	R/W	Receive Buffer Address Indicates the start address of the receive buffer. The buffer address should be set on a 32-byte boundary.

### 20.3.2 Transmission

When the transmit request bit (TR) in the E-DMAC transmit request register (EDTRR) is set while the transmission function is enabled, the E-DMAC reads the descriptor following the previously used descriptor from the transmit descriptor list (or the descriptor indicated by the transmit descriptor start address register (TDLAR) at the initial start time). If the TACT bit of the read descriptor is set to 1 (valid), the E-DMAC sequentially reads transmit frame data from the transmit buffer start address specified by TD2 for transfer to the EtherC. The EtherC creates a transmit frame and starts transmission to the MII. After DMA transfer of data equivalent to the buffer length specified in the descriptor, the following processing is carried out according to the TFP value.

1. TFP = 00 or 10 (frame continuation):  
Descriptor write-back (writing 0 to the TACT bit) is performed after DMA transfer.
2. TFP = 01 or 11 (frame end):  
Descriptor write-back (writing 0 to the TACT bit and writing status) is performed after completion of frame transmission.

As long as the TACT bit of a read descriptor is set to 1 (valid), the reading of E-DMAC descriptors and the transmission of frames continue. When a descriptor with the TACT bit cleared to 0 (invalid) is read, the E-DMAC clears the TR bit in EDTRR to 0 and completes transmit processing.



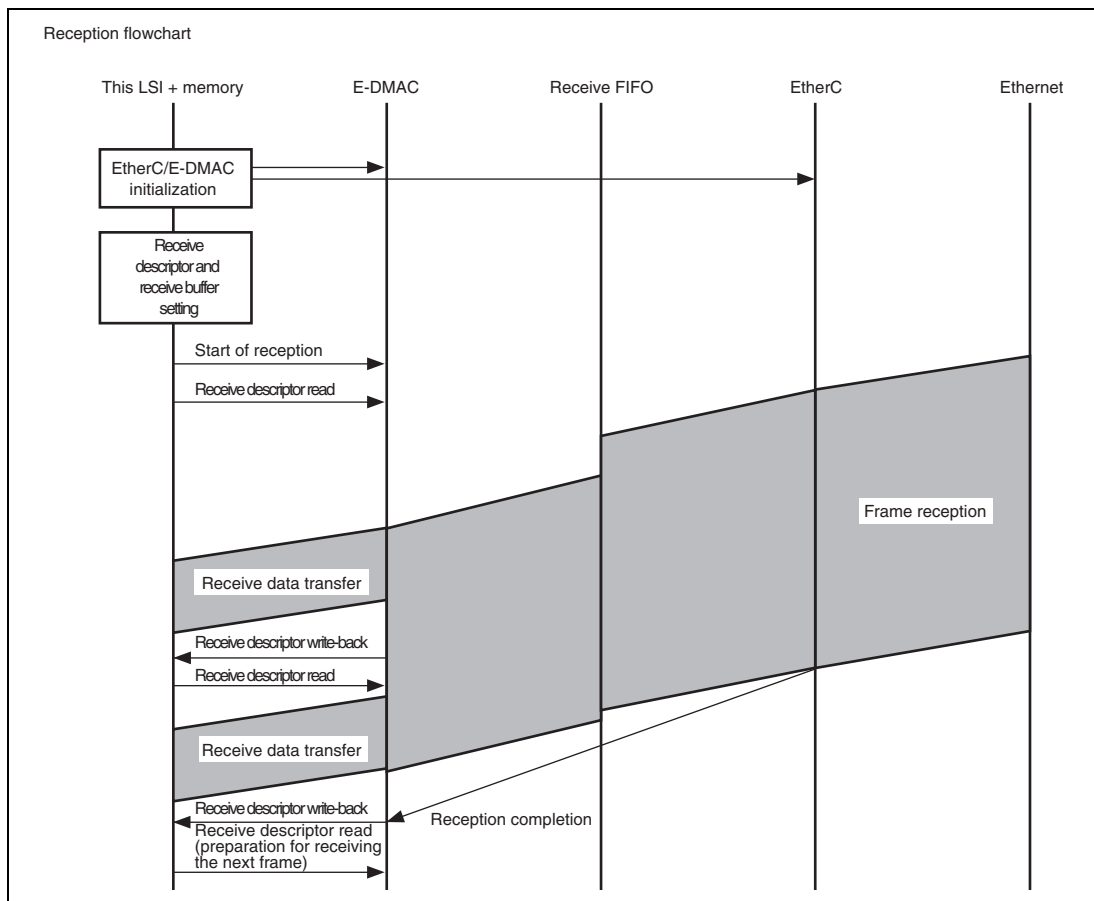
**Figure 20.4 Sample Transmission Flowchart (Single-Frame/Two-Descriptor)**



### 20.3.3 Reception

When the CPU sets the receive request bit (RR) in the E-DMAC receive request register (EDRRR) while the receive function is enabled, the E-DMAC reads the descriptor following the previously used descriptor from the receive descriptor list (or the descriptor indicated by the receive descriptor start address register (RDLAR) at the initial start time) then enters the receive standby state. When the EtherC receives a frame for this LSI (with an address enabled for reception by this LSI), the EtherC stores the receive data in the receive FIFO. Upon receiving the frame for own station while the RACT bit is set to 1 (valid), the E-DMAC transfers the frame to the receive buffer specified by RD2. If the data length of a received frame is longer than the buffer length specified by RD1, the E-DMAC performs a write-back operation to the descriptor (with RFP set to 10 or 00) when the buffer becomes full, then reads the next descriptor. The E-DMAC then continues to transfer data to the receive buffer specified by the new RD2. When frame reception is completed, or if frame reception is suspended because of a certain kind of error, the E-DMAC performs write-back to the relevant descriptor (with RFP set to 11 or 01), and then ends the receive processing. The E-DMAC then reads the next descriptor and enters the receive standby state again.

To receive frames continuously, the receive enable control bit (RNC) must be set to 1 in the receive method control register (RMCR). The initial value is 0.



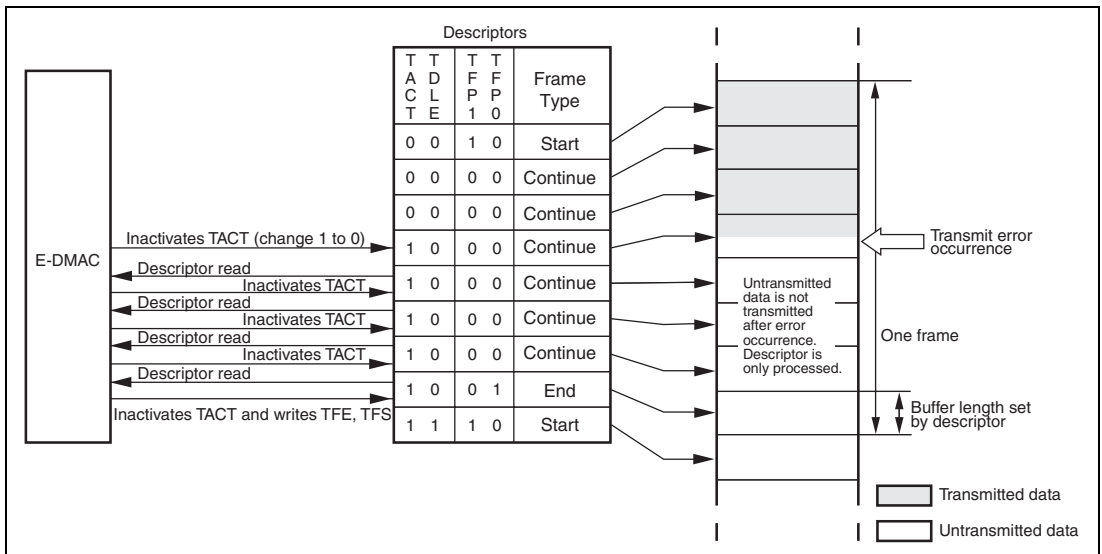
**Figure 20.5 Sample Reception Flowchart (Single-Frame/Two-Descriptor)**

### 20.3.4 Transmit/Receive Processing of Multi-Buffer Frame (Single-Frame/Multi-Descriptor)

#### (1) Multi-Buffer Frame Transmit Processing

If an error occurs during multi-buffer frame transmission, the processing shown in figure 20.6 is carried out by the E-DMAC.

In the figure where the transmit descriptor is shown as inactive (TACT bit = 0), buffer data has already been transmitted normally, and where the transmit descriptor is shown as active (TACT bit = 1), buffer data has not been transmitted. If a frame transmit error occurs in the first descriptor part where the transmit descriptor is active (TACT bit = 1), transmission is halted, and the TACT bit cleared to 0, immediately. The next descriptor is then read, and the position within the transmit frame is determined on the basis of bits TFP1 and TFP0 (continuing [B'00] or end [B'01]). In the case of a continuing descriptor, the TACT bit is cleared to 0, only, and the next descriptor is read immediately. If the descriptor is the final descriptor, not only is the TACT bit cleared to 0, but write-back is also performed to the TFE and TFS bits at the same time. Data in the buffer is not transmitted between the occurrence of an error and write-back to the final descriptor. If error interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EESIPR), an interrupt is generated immediately after the final descriptor write-back.



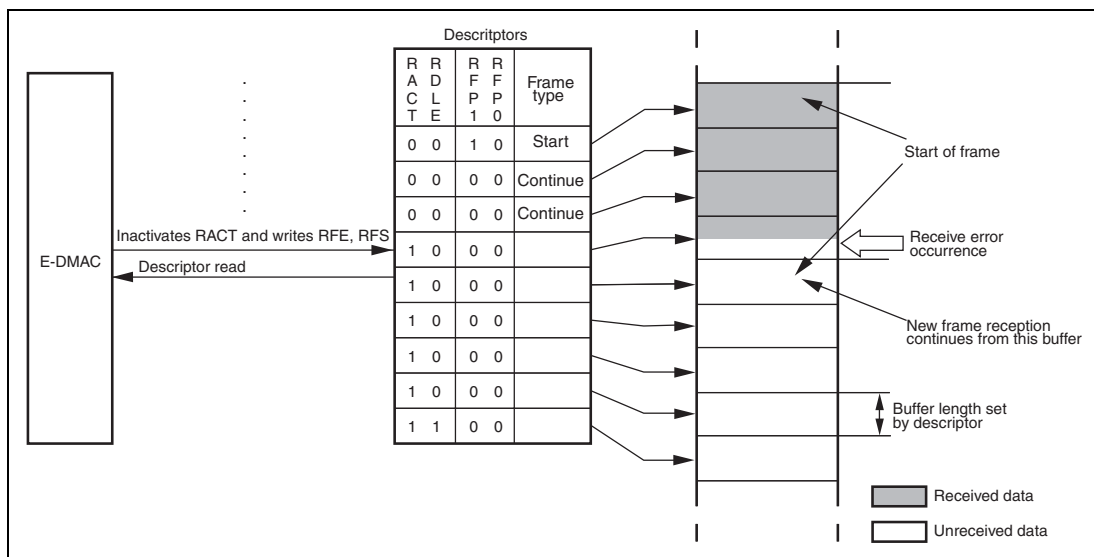
**Figure 20.6 E-DMAC Operation after Transmit Error**

## (2) Receive Processing in Case of Multi-Buffer Frame

If an error occurs during reception in the case of a multi-buffer frame where a receive frame is divided for storage in multiple buffers, the E-DMAC performs the processing shown in Figure 20.7.

In the figure, the invalid receive descriptors (with the RACT bit cleared to 0) represent the normal reception of data to be stored in buffers, and the valid receive descriptors (with the RACT bit set to 1) represent unreceived buffers. If a frame receive error occurs with a descriptor shown in the figure, the status is written back to the corresponding descriptor.

If error interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EESIPR), an interrupt is generated immediately after the write-back. If there is a new frame receive request, reception is continued from the buffer after that in which the error occurred.



**Figure 20.7 E-DMAC Operation after Receive Error**

## Section 21 USB 2.0 Host/Function Module (USB)

The USB 2.0 host/function module (USB) is a USB controller which provides capabilities as a USB host controller and USB function controller function. This module supports high-speed transfer defined by USB (universal serial bus) Specification 2.0 and full-speed transfer when used as the host controller, and supports high-speed transfer and full-speed transfer when used as the function controller. This module has a USB transceiver and supports all of the transfer types defined by the USB specification.

This module has a 5-Kbyte buffer memory for data transfer, providing a maximum of ten pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE9, based on the peripheral devices or user system for communication.

### 21.1 Features

#### (1) Host Controller and Function Controller Supporting USB High-Speed Operation

- The USB host controller and USB function controller are incorporated.
- The USB host controller and USB function controller can be switched by register settings.
- USB transceiver is incorporated.

#### (2) Reduced Number of External Pins and Space-Saving Installation

- The VBUS signal can be directly connected to the input pin of this module.
- On-chip D+ pull-up resistor (during USB function operation)
- On-chip D+ and D- pull-down resistor (during USB host operation)
- On-chip D+ and D- terminal resistor (during high-speed operation)
- On-chip D+ and D- output impedance (during full-speed operation)

#### (3) All Types of USB Transfers Supported

- Control transfer
- Bulk transfer
- Interrupt transfer (high bandwidth transfers not supported)
- Isochronous transfer (high bandwidth transfers not supported)

#### (4) Internal Bus Interfaces

- Two DMA interface channels are incorporated.

## (5) Pipe Configuration

- Up to 5 Kbytes of buffer memory for USB communications are supported
- 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 (default control pipe: DCP), 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

## (6) Features of the USB Host Controller

- High-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) are supported.
- Communications with multiple peripheral devices connected via a single HUB
- Automatic response to the reset handshake
- Automatic scheduling for SOF and packet transmissions
- Programmable intervals for isochronous and interrupt transfers

## (7) Features of the USB Function Controller

- Both high-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) are supported.
- 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

**(8) Other Features**

- Transfer ending function using transaction count
- BRDY interrupt event notification timing change function (BFRE)
- Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0 or 1) port has been read (DCLRM)
- NAK setting function for response PID generated by end of transfer (SHTNAK)

## 21.2 Input / Output Pins

Table 21.1 shows the pin configuration of the USB.

**Table 21.1 USB Pin Configuration**

Pin Name	I/O	Function	Description
XIN	Input	Crystal resonator /External clock input for USB	Connect to a crystal resonator or input an external clock for USB operation
XOUT	Output	Crystal resonator connection pin for USB	Connect to a crystal resonator for USB operation
MODE7	Input	USB clock mode control	0: Input an external clock from XIN. XOUT should be open.  1: Connect XIN and XOUT to a 48 MHz crystal resonator.
DP	I/O	D+	D+ data for USB bus
DM	I/O	D-	D- data for USB bus
VBUS	Input	Vbus	Vbus for USB bus
REFRIN	Input	Reference input	Connect to the analog ground through a 5.6 k $\Omega$ $\pm$ 1% resistor.
VDD_USB	—	USB PHY digital power supply	Connect to a voltage of 1.2 V
VSS_USB	—	USB PHY digital ground	Connect to a voltage of 0 V
VDDQ_USB	—	USB PHY digital power supply	Connect to a voltage of 3.3 V
VSSQ_USB	—	USB PHY digital ground	Connect to a voltage of 0 V
VDDA_USB	—	USB PHY analog power supply	Connect to a voltage of 1.2 V
VSSA_USB	—	USB PHY analog ground	Connect to a voltage of 0 V
VDDQA_USB	—	USB PHY analog power supply	Connect to a voltage of 3.3 V
VSSQA_USB	—	USB PHY analog ground	Connect to a voltage of 0 V
UV12	—	USB 480 MHz power supply	Connect to a voltage of 1.2 V
UG12	—	USB 480 MHz ground	Connect to a voltage of 0 V



## 21.3 Register Description

Table 21.2 shows the register configuration of the USB. Table 21.3 shows the register state in each processing mode.

**Table 21.2 Register Configuration**

Register Name	Abbreviation	R/W	Address	Access Size
System configuration control register	SYSCFG	R/W	H'FE40 0000	16
CPU bus wait setting register	BUSWAIT	R/W	H'FE40 0002	16
System configuration status register	SYSSTS	R	H'FE40 0004	16
Device state control register	DVSTCTR	R/W	H'FE40 0008	16
Test mode register	TESTMODE	R/W	H'FE40 000C	16
DMA0-FIFO bus configuration register	D0FBCFG	R/W	H'FE40 0010	16
DMA1-FIFO bus configuration register	D1FBCFG	R/W	H'FE40 0012	16
CFIFO port register	CFIFO	R/W	H'FE40 0014	8/16/32
D0FIFO port register	D0FIFO	R/W	H'FE40 0018 H'FE40 0180	8/16/32
D1FIFO port register	D1FIFO	R/W	H'FE40 001C H'FE40 01C0	8/16/32
CFIFO port select register	CFIFOSEL	R/W	H'FE40 0020	16
CFIFO port control register	CFIFOCTR	R/W	H'FE40 0022	16
D0FIFO port select register	D0FIFOSEL	R/W	H'FE40 0028	16
D0FIFO port control register	D0FIFOCTR	R/W	H'FE40 002A	16
D1FIFO port select register	D1FIFOSEL	R/W	H'FE40 002C	16
D1FIFO port control register	D1FIFOCTR	R/W	H'FE40 002E	16
Interrupt enable register 0	INTENB0	R/W	H'FE40 0030	16
Interrupt enable register 1	INTENB1	R/W	H'FE40 0032	16
BRDY interrupt enable register	BRDYENB	R/W	H'FE40 0036	16
NRDY interrupt enable register	NRDYENB	R/W	H'FE40 0038	16
BEMP interrupt enable register	BEMPENB	R/W	H'FE40 003A	16
SOF output configuration register	SOFCFG	R/W	H'FE40 003C	16

Register Name	Abbreviation	R/W	Address	Access Size
Interrupt status register 0	INTSTS0	R/W	H'FE40 0040	16
Interrupt status register 1	INTSTS1	R/W	H'FE40 0042	16
BRDY interrupt status register	BRDYSTS	R/W	H'FE40 0046	16
NRDY interrupt status register	NRDYSTS	R/W	H'FE40 0048	16
BEMP interrupt status register	BEMPSTS	R/W	H'FE40 004A	16
Frame number register	FRMNUM	R/W	H'FE40 004C	16
μFrame number register	UFRMNUM	R/W	H'FE40 004E	16
USB address register	USBADDR	R	H'FE40 0050	16
USB request type register	USBREQ	R	H'FE40 0054	16
USB request value register	USBVAL	R	H'FE40 0056	16
USB request index register	USBINDX	R	H'FE40 0058	16
USB request length register	USBLENG	R	H'FE40 005A	16
DCP configuration register	DCPCFG	R/W	H'FE40 005C	16
DCP maximum packet size register	DCPMAXP	R/W	H'FE40 005E	16
DCP control register	DCPCTR	R/W	H'FE40 0060	16
Pipe window select register	PIPESEL	R/W	H'FE40 0064	16
Pipe configuration register	PIPECFG	R/W	H'FE40 0068	16
Pipe buffer setting register	PIPEBUF	R/W	H'FE40 006A	16
Pipe maximum packet size register	PIPEMAXP	R/W	H'FE40 006C	16
Pipe cycle control register	PIPEPERI	R/W	H'FE40 006E	16
Pipe 1 control register	PIPE1CTR	R/W	H'FE40 0070	16
Pipe 2 control register	PIPE2CTR	R/W	H'FE40 0072	16
Pipe 3 control register	PIPE3CTR	R/W	H'FE40 0074	16
Pipe 4 control register	PIPE4CTR	R/W	H'FE40 0076	16
Pipe 5 control register	PIPE5CTR	R/W	H'FE40 0078	16
Pipe 6 control register	PIPE6CTR	R/W	H'FE40 007A	16
Pipe 7 control register	PIPE7CTR	R/W	H'FE40 007C	16
Pipe 8 control register	PIPE8CTR	R/W	H'FE40 007E	16
Pipe 9 control register	PIPE9CTR	R/W	H'FE40 0080	16
Pipe 1 transaction counter enable register	PIPE1TRE	R/W	H'FE40 0090	16
Pipe 1 transaction counter register	PIPE1TRN	R/W	H'FE40 0092	16

Register Name	Abbreviation	R/W	Address	Access Size
Pipe 2 transaction counter enable register	PIPE2TRE	R/W	H'FE40 0094	16
Pipe 2 transaction counter register	PIPE2TRN	R/W	H'FE40 0096	16
Pipe 3 transaction counter enable register	PIPE3TRE	R/W	H'FE40 0098	16
Pipe 3 transaction counter register	PIPE3TRN	R/W	H'FE40 009A	16
Pipe 4 transaction counter enable register	PIPE4TRE	R/W	H'FE40 009C	16
Pipe 4 transaction counter register	PIPE4TRN	R/W	H'FE40 009E	16
Pipe 5 transaction counter enable register	PIPE5TRE	R/W	H'FE40 00A0	16
Pipe 5 transaction counter register	PIPE5TRN	R/W	H'FE40 00A2	16
Device address 0 configuration register	DEVADD0	R/W	H'FE40 00D0	16
Device address 1 configuration register	DEVADD1	R/W	H'FE40 00D2	16
Device address 2 configuration register	DEVADD2	R/W	H'FE40 00D4	16
Device address 3 configuration register	DEVADD3	R/W	H'FE40 00D6	16
Device address 4 configuration register	DEVADD4	R/W	H'FE40 00D8	16
Device address 5 configuration register	DEVADD5	R/W	H'FE40 00DA	16
Device address 6 configuration register	DEVADD6	R/W	H'FE40 00DC	16
Device address 7 configuration register	DEVADD7	R/W	H'FE40 00DE	16
Device address 8 configuration register	DEVADD8	R/W	H'FE40 00E0	16
Device address 9 configuration register	DEVADD9	R/W	H'FE40 00E2	16
Device address A configuration register	DEVADDA	R/W	H'FE40 00E4	16

**Table 21.3 Register State in Each Processing Mode**

<b>Register Abbreviation</b>	<b>Power-On Reset</b>	<b>Software Standby</b>	<b>Module Standby</b>	<b>Sleep</b>
SYSCFG	Initialized	Retained	Retained	Retained
BUSWAIT	Initialized	Retained	Retained	Retained
SYSSTS	Initialized	Retained	Retained	Retained
DVSTCTR	Initialized	Retained	Retained	Retained
TESTMODE	Initialized	Retained	Retained	Retained
D0FBCFG	Initialized	Retained	Retained	Retained
D1FBCFG	Initialized	Retained	Retained	Retained
CFIFO	Initialized	Retained	Retained	Retained
D0FIFO	Initialized	Retained	Retained	Retained
D1FIFO	Initialized	Retained	Retained	Retained
CFIFOSEL	Initialized	Retained	Retained	Retained
CFIFOCTR	Initialized	Retained	Retained	Retained
D0FIFOSEL	Initialized	Retained	Retained	Retained
D0FIFOCTR	Initialized	Retained	Retained	Retained
D1FIFOSEL	Initialized	Retained	Retained	Retained
D1FIFOCTR	Initialized	Retained	Retained	Retained
INTENB0	Initialized	Retained	Retained	Retained
INTENB1	Initialized	Retained	Retained	Retained
BRDYENB	Initialized	Retained	Retained	Retained
NRDYENB	Initialized	Retained	Retained	Retained
BEMPENB	Initialized	Retained	Retained	Retained
SOFCFG	Initialized	Retained	Retained	Retained
INTSTS0	Initialized	Retained	Retained	Retained
INTSTS1	Initialized	Retained	Retained	Retained
BRDYSTS	Initialized	Retained	Retained	Retained
NRDYSTS	Initialized	Retained	Retained	Retained
BEMPSTS	Initialized	Retained	Retained	Retained
FRMNUM	Initialized	Retained	Retained	Retained
UFRMNUM	Initialized	Retained	Retained	Retained

<b>Register Abbreviation</b>	<b>Power-On Reset</b>	<b>Software Standby</b>	<b>Module Standby</b>	<b>Sleep</b>
USBADDR	Initialized	Retained	Retained	Retained
USBREQ	Initialized	Retained	Retained	Retained
USBVAL	Initialized	Retained	Retained	Retained
USBINDX	Initialized	Retained	Retained	Retained
USBLENG	Initialized	Retained	Retained	Retained
DCPCFG	Initialized	Retained	Retained	Retained
DCPMAXP	Initialized	Retained	Retained	Retained
DCPCTR	Initialized	Retained	Retained	Retained
PIPESEL	Initialized	Retained	Retained	Retained
PIPECFG	Initialized	Retained	Retained	Retained
PIPEBUF	Initialized	Retained	Retained	Retained
PIPEMAXP	Initialized	Retained	Retained	Retained
PIPEPERI	Initialized	Retained	Retained	Retained
PIPE1CTR	Initialized	Retained	Retained	Retained
PIPE2CTR	Initialized	Retained	Retained	Retained
PIPE3CTR	Initialized	Retained	Retained	Retained
PIPE4CTR	Initialized	Retained	Retained	Retained
PIPE5CTR	Initialized	Retained	Retained	Retained
PIPE6CTR	Initialized	Retained	Retained	Retained
PIPE7CTR	Initialized	Retained	Retained	Retained
PIPE8CTR	Initialized	Retained	Retained	Retained
PIPE9CTR	Initialized	Retained	Retained	Retained
PIPE1TRE	Initialized	Retained	Retained	Retained
PIPE1TRN	Initialized	Retained	Retained	Retained
PIPE2TRE	Initialized	Retained	Retained	Retained
PIPE2TRN	Initialized	Retained	Retained	Retained
PIPE3TRE	Initialized	Retained	Retained	Retained
PIPE3TRN	Initialized	Retained	Retained	Retained
PIPE4TRE	Initialized	Retained	Retained	Retained
PIPE4TRN	Initialized	Retained	Retained	Retained

<b>Register Abbreviation</b>	<b>Power-On Reset</b>	<b>Software Standby</b>	<b>Module Standby</b>	<b>Sleep</b>
PIPE5TRE	Initialized	Retained	Retained	Retained
PIPE5TRN	Initialized	Retained	Retained	Retained
DEVADD0	Initialized	Retained	Retained	Retained
DEVADD1	Initialized	Retained	Retained	Retained
DEVADD2	Initialized	Retained	Retained	Retained
DEVADD3	Initialized	Retained	Retained	Retained
DEVADD4	Initialized	Retained	Retained	Retained
DEVADD5	Initialized	Retained	Retained	Retained
DEVADD6	Initialized	Retained	Retained	Retained
DEVADD7	Initialized	Retained	Retained	Retained
DEVADD8	Initialized	Retained	Retained	Retained
DEVADD9	Initialized	Retained	Retained	Retained
DEVADDA	Initialized	Retained	Retained	Retained

### 21.3.1 System Configuration Control Register (SYSCFG)

SYSCFG is a register that enables high-speed operation, selects the host controller function or function controller function, controls the DP and DM pins, and enables operation of this module.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SCKE	—	—	HSE	DCFM	DRPD	DPRPU	—	—	—	USBE
Initial value:	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	R	R/W	R/W	R/W	R/W	R	R	R	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	SCKE	0	R/W	USB Module Clock Enable Stops or enables supplying 48-MHz clock signal to this module. 0: Stops supplying the clock signal to the USB module. 1: Enables supplying the clock signal to the USB module. When this bit is 0, only this register and the BUSWAIT register allow both writing and reading; the other registers in the USB module allows reading only.
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	HSE	0	R/W	<p>High-Speed Operation Enable</p> <p>0: High-speed operation is disabled</p> <p>When the function controller function is selected: Only full-speed operation is enabled.</p> <p>When the host controller function is selected: Only full-speed operation is enabled.</p> <p>1: High-speed operation is enabled (detected by this module)</p> <p>(1) When the host controller function is selected</p> <p>When HSE = 0, the USB port performs full-speed operation.</p> <p>When HSE = 1, this module executes the reset handshake protocol, and automatically allows the USB port to perform high-speed or full-speed operation according to the protocol execution result.</p> <p>This bit should be modified after detecting device connection (after detecting the ATTCH interrupt) and before executing a USB bus reset (before setting USBRESET to 1).</p> <p>(2) When the function controller function is selected</p> <p>When HSE = 0, this module performs full-speed operation.</p> <p>When HSE = 1, this module executes the reset handshake protocol, and automatically performs high-speed or full-speed operation according to the protocol execution result.</p> <p>This bit should be modified while DPRPU is 0.</p>
6	DCFM	0	R/W	<p>Controller Function Select</p> <p>Selects the host controller function or function controller function.</p> <p>0: Function controller function is selected.</p> <p>1: Host controller function is selected.</p> <p>This bit should be modified while DPRPU and DRPD are 0.</p>



Bit	Bit Name	Initial Value	R/W	Description
5	DRPD	0	R/W	<p>D+/D− Line Resistor Control</p> <p>Enables or disables pulling down D+ and D- lines when the host controller function is selected.</p> <p>0: Pulling down the lines is disabled.</p> <p>1: Pulling down the lines is enabled.</p> <p>This bit should be set to 1 if the host controller function is selected, and should be set to 0 if the function controller function is selected.</p>
4	DPRPU	0	R/W	<p>D+ Line Resistor Control</p> <p>Enables or disables pulling up D+ line when the function controller function is selected.</p> <p>0: Pulling up the line is disabled.</p> <p>1: Pulling up the line is enabled.</p> <p>Setting this bit to 1 when the function controller function is selected allows this module to pull up the D+ line to 3.3 V, thus notifying the USB host of connection. Modifying this bit from 1 to 0 allows this module to cancel pulling up the D+ line, thus notifying the USB host of disconnection.</p> <p>This bit should be set to 1 if the function controller function is selected, and should be set to 0 if the host controller function is selected.</p>
3 to 1	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
0	USBE	0	R/W	<p>USB Module Operation Enable</p> <p>Enables or disables operation of this module.</p> <p>0: USB module operation is disabled.</p> <p>1: USB module operation is enabled.</p> <p>Modifying this bit from 1 to 0 initializes some register bits as listed in tables 21.4 and 21.5.</p> <p>This bit should be modified while SCKE is 1.</p> <p>When the host controller function is selected, this bit should be set to 1 after setting DPRD to 1, eliminating LNST bit chattering, and checking that the USB bus has been settled.</p>

**Table 21.4 Register Bits Initialized by Writing USBE = 0 (when Function Controller Function is Selected)**

Register Name	Bit Name	Remarks
SYSSTS	LNST	The value is retained when the host controller function is selected.
DVSTCTR	RHST	
INTSTS0	DVSQ	The value is retained when the host controller function is selected.
USBADDR	USBADDR	The value is retained when the host controller function is selected.
USEREQ	BRequest, bmRequestType	The values are retained when the host controller function is selected.
USBVAL	wValue	The value is retained when the host controller function is selected.
USBINDX	wIndex	The value is retained when the host controller function is selected.
USBLENG	wLength	The value is retained when the host controller function is selected.

**Table 21.5 Register Bits Initialized by Writing USBE = 0 (when Host Controller Function is Selected)**

Register Name	Bit Name	Remarks
DVSTCTR	RHST	
FRMNUM	FRNM	The value is retained when the function controller function is selected.
UFRMNUM	UFRNM	The value is retained when the function controller function is selected.

### 21.3.2 CPU Bus Wait Setting Register (BUSWAIT)

BUSWAIT is a register that specifies the number of wait cycles to be inserted during an access from the CPU to this module.

This register can be modified even when the SCKE bit in SYSCFG is 0.

This register is initialized by a power-on reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	BWAIT[3:0]	1111	R/W	CPU Bus Wait Specifies the number of wait cycles to be inserted during an access to this module. 0000: Setting prohibited 0100: 4 wait cycle (6 access cycles) 0110: 6 wait cycle (8 access cycles) 1111: 15 wait cycles (17 access cycles) (initial value)

In this controller, there are restrictions for a cycle accessing to the register address after H'04 as follows.

Restrictions: In this controller, a cycle accessing to the register continually must be above 80ns. To satisfy the restrictions, a wait cycle needs to be controlled by frequency of SHwy clock to be input. Since the initial value has to be maximum (17 clock cycle), select a suitable value. Indicates a formula to figure out the BWAIT of SHwy clock as follows.

Formula :

$$80\text{ ns} \div \left\{ (1 \div \text{SHwy clock frequency}) \div 10^{-9} \right\} - 1$$

BWAIT setting value corresponding to SHwy clock:

Shwy clock:	BWAIT setting value:
80 MHz	0100
108 MHz	0110

21.3.3 System Configuration Status Register (SYSSTS)

SYSSTS is a register that monitors the line status (D + and D – lines) of the USB data bus.

This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LNST[1:0]	
Initial value:	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	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
9 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	LNST[1:0]	Undefined*	R	USB Data Line Status Monitor Indicates the status of the USB data bus lines (D+ and D-) as shown in table 21.6. These bits should be read after setting DPRPU to 1 to notify connection when the function controller function is selected; whereas after setting DRPD to 1 to enable pulling down the lines when the host controller function is selected.

Note: \* Depends on the DP and DM pin status.

**Table 21.6 USB Data Bus Line Status**

LNST[1]	LNST[0]	During Full-Speed Operation	During High-Speed Operation	During 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

**[Legend]**

Chirp:	The reset handshake protocol is being executed in high-speed operation enabled state (the HSE bit in SYSCFG is set to 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

### 21.3.4 Device State Control Register (DVSTCTR)

DVSTCTR is a register that controls and confirms the state of the USB data bus.

This register is initialized by a power-on reset. After a USB bus reset, WKUP is initialized but RESUME is undefined.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W*	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
8	WKUP	0	R/W	<p>Wakeup Output</p> <p>Enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller function is selected.</p> <p>0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.</p> <p>The module controls the output time of a remote wakeup signal. When this bit is set to 1, this module clears this bit to 0 after outputting the 10-ms K state.</p> <p>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 this module writes 1 to this bit right after detection of suspended state, the K state will be output after 2 ms.</p> <p>Note: Do not write 1 to this bit, unless the device state is in the suspended state (the DVSQ bit in the INTSTS0 register is set to 1xx) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while SCKE is 1).</p> <p>This bit should be set to 0 if the host controller function is selected.</p>

Bit	Bit Name	Initial Value	R/W	Description
7	RWUPE	0	R/W	<p>Wakeup Detection Enable</p> <p>Enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller function is selected.</p> <p>0: Downstream port wakeup is disabled.</p> <p>1: Downstream port wakeup is enabled.</p> <p>With this bit set to 1, on detecting the remote wakeup signal, this module detects the resume signal (K-state for 2.5 <math>\mu</math>s) from the downstream port device and performs the resume process (drives the port to the K-state).</p> <p>With this bit set to 0, this module ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the downstream port.</p> <p>While this bit is 1, the internal clock should not be stopped even in the suspended state (SCKE should be set to 1). Also note that the USB bus should not be reset from the suspended state (USBRST should not be set to 1); it is prohibited by USB Specification 2.0.</p> <p>This bit should be set to 0 if the function controller function is selected.</p>

Bit	Bit Name	Initial Value	R/W	Description
6	USBRST	0	R/W	<p><b>Bus Reset Output</b></p> <p>Controls the USB bus reset signal output when the host controller function is selected.</p> <p>0: USB bus reset signal is not output.</p> <p>1: USB bus reset signal is output.</p> <p>When the host controller function is selected, setting this bit to 1 allows this module to drive the USB port to SE0 to reset the USB bus. Here, this module performs the reset handshake protocol if the HSE bit is 1.</p> <p>This module continues outputting SE0 while USBRST is 1 (until software sets USBRST to 0). USBRST should be 1 (= USB bus reset period) for the time defined by USB Specification 2.0.</p> <p>Writing 1 to this bit during communication (UACT = 1) or during the resume process (RESUME = 1) prevents this module from starting the USB bus reset process until both UACT and RESUME become 0.</p> <p>Write 1 to the UACT bit simultaneously with the end of the USB bus reset process (writing 0 to USBRST).</p> <p>This bit should be set to 0 if the function controller function is selected.</p>
5	RESUME	0	R/W	<p><b>Resume Output</b></p> <p>Controls the resume signal output when the host controller function is selected.</p> <p>0: Resume signal is not output.</p> <p>1: Resume signal is output.</p> <p>Setting this bit to 1 allows this module to drive the port to the K-state and output the resume signal.</p> <p>This module continues outputting K-state while RESUME is 1 (until software sets RESUME to 0). RESUME should be 1 (= resume period) for the time defined by USB Specification 2.0.</p> <p>This bit should be set to 1 in the suspended state.</p> <p>Write 1 to the UACT bit simultaneously with the end of the resume process (writing 0 to RESUME).</p> <p>This bit should be set to 0 if the function controller function is selected.</p>



Bit	Bit Name	Initial Value	R/W	Description
4	UACT	0	R/W	<p><b>USB Bus Enable</b></p> <p>Enables operation of the USB bus (controls the SOF or <math>\mu</math>SOF packet transmission to the USB bus) when the host controller function is selected.</p> <p>0: Downstream port is disabled (SOF/<math>\mu</math>SOF transmission is disabled).</p> <p>1: Downstream port is enabled (SOF/<math>\mu</math>SOF transmission is enabled).</p> <p>With this bit set to 1, this module puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception.</p> <p>This module starts outputting SOF/<math>\mu</math>SOF within 1 (<math>\mu</math>) frame after software has written 1 to UACT.</p> <p>With this bit set to 0, this module enters the idle state after outputting SOF/<math>\mu</math>SOF.</p> <p>This module sets this bit to 0 on any of the following conditions.</p> <ul style="list-style-type: none"> <li>• A DTCH interrupt is detected during communication (while UACT = 1).</li> <li>• An EOFERR interrupt is detected during communication (while UACT = 1).</li> </ul> <p>Writing 1 to this bit should be done at the end of the USB reset process (writing 0 to USBRST) or at the end of the resume process from the suspended state (writing 0 to RESUME).</p> <p>This bit should be set to 0 if the function controller function is selected.</p>
3	—	0	R	<p><b>Reserved</b></p> <p>This bit is always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	RHST[2:0]	000	R	<p>Reset Handshake</p> <p>Indicates the status of the reset handshake.</p> <p>(1) When the host controller function is selected</p> <p>000: Communication speed not determined (powered state or no connection)</p> <p>1xx: Reset handshake in progress</p> <p>010: Full-speed connection</p> <p>011: High-speed connection</p> <p>These bits indicate 100 after software has written 1 to USBRST.</p> <p>If HSE has been set to 1, these bits indicate 111 as soon as this module detects Chirp-K from the peripheral device.</p> <p>This module fixes the value of the RHST bits when software writes 0 to USBRST and this module completes SE0 driving.</p> <p>(2) When the function controller function is selected</p> <p>000: Communication speed not determined</p> <p>100: Reset handshake in progress</p> <p>010: Full-speed connection</p> <p>011: High-speed connection</p> <p>If HSE has been set to 1, these bits indicate 100 as soon as this module detects the USB bus reset.</p> <p>Then, these bits indicate 011 as soon as this module outputs Chirp-K and detects Chirp-JK from the USB host three times. If the connection speed is not fixed to high speed within 2.5 ms after Chirp-K output, these bits indicate 010.</p> <p>If HSE has been set to 0, these bits indicate 010 as soon as this module detects the USB bus reset.</p> <p>A DVST interrupt is generated as soon as this module detects the USB bus reset and then the value of the RHST bits is fixed to 010 or 011.</p>

Note: \* Only 1 can be written.

### 21.3.5 Test Mode Register (TESTMODE)

TESTMODE is a register that controls the USB test signal output during high-speed operation.

This register is initialized by a power-on reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	UTST[3:0]	0000	R/W	Test Mode This module outputs the USB test signals during the high-speed operation, when these bits are written appropriate value. Table 21.7 shows test mode operation of this module.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	UTST[3:0]	0000	R/W	<p>(1) When the host controller function is selected</p> <p>These bits can be set after writing 1 to DRPD. This module outputs waveforms to the USB port for which both DPRD and UACT have been set to 1. This module also performs high-speed termination for the USB port.</p> <ul style="list-style-type: none"> <li>• Procedure for setting the UTST bits <ol style="list-style-type: none"> <li>1. Power-on reset.</li> <li>2. Start the clock supply (Set SCKE to 1 after the crystal oscillation and the PLL for USB are settled).</li> <li>3. Set DCFM and DPRD to 1 (setting HSE to 1 is not required).</li> <li>4. Set USBE to 1.</li> <li>5. Set the UTST bits to the appropriate value according to the test specifications.</li> <li>6. Set the UACT bit to 1.</li> </ol> </li> <li>• Procedure for modifying the UTST bits <ol style="list-style-type: none"> <li>1. (In the state after executing step 6 above) Set UACT and USBE to 0.</li> <li>2. Set USBE to 1.</li> <li>3. Set the UTST bits to the appropriate value according to the test specifications.</li> <li>4. Set the UACT bit to 1.</li> </ol> </li> </ul> <p>When these bits are set to Test_SE0_NAK (1011), this module does not output the SOF packet to the port even when 1 has been set to UACT for the port.</p> <p>When these bits are set to Test_Force_Enable (1101), this module outputs the SOF packet to the port for which 1 has been set to UACT. In this test mode, this module does not perform hardware control consequent to detection of high-speed disconnection (detection of the DTCH interrupt).</p> <p>When setting the UTST bits, the PID bits for all the pipes should be set to NAK.</p> <p>To return to normal USB communication after a test mode has been set and executed, a power-on reset should be applied.</p>

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	UTST[3:0]	0000	R/W	<p>(2) When the function controller function is selected</p> <p>The appropriate value should be set to these bits according to the SetFeature request from the USB host during high-speed communication.</p> <p>This module does not make a transition to the suspended state while these bits are 0001 to 0100.</p>

**Table 21.7 Test Mode Operation**

Test Mode	UTST Bit Setting	
	When Function Controller Function is Selected	When Host Controller Function is Selected
Normal operation	0000	0000
Test_J	0001	1001
Test_K	0010	1010
Test_SE0_NAK	0011	1011
Test_Packet	0100	1100
Test_Force_Enable	—	1101
Reserved	0101 to 0111	1110 to 1111

### 21.3.6 DMA-FIFO Bus Configuration Registers (D0FBCFG, D1FBCFG)

D0FBCFG is a register that controls DMA0-FIFO bus accesses. D1FBCFG is a register that controls DMA1-FIFO bus accesses.

These registers are initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	DFACC		—	—	—	—	—	—	—	—	—	—	—	—
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
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	DFACC	00	R/W	DMA <sub>n</sub> -FIFO Buffer Access Mode (n = 0, 1) Specifies DMA0-FIFO or DMA1-FIFO port access mode. 00: Cycle steal mode (initial value) 01: 16-byte continuous access mode 10: 32-byte continuous access mode 11: Invalid
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

### 21.3.7 FIFO Port Registers (CFIFO, D0FIFO, D1FIFO)

CFIFO, D0FIFO and D1FIFO are port registers that are used to read data from the FIFO buffer memory and writing data to the FIFO buffer memory.

There are three FIFO ports: the CFIFO, D0FIFO and D1FIFO ports. Each FIFO port is configured of a port register (CFIFO, D0FIFO, D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a select register (CFIFOSEL, D0FIFOSEL, D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a control register (CFIFOCTR, D0FIFOCTR, D1FIFOCTR).

Each FIFO port has the following features.

- The DCP FIFO buffer should be accessed through the CFIFO port.
- Accessing the FIFO buffer using DMA transfer should be performed through the D0FIFO or D1FIFO port.
- The D1FIFO and D0FIFO ports can be accessed also by the CPU.
- When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed (when the DMA transfer function is used, etc.).
- Registers configuring a FIFO port do not affect other FIFO ports.
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.
- These addresses, H'FE40 0180 of D0FIFO port register and H'FE40 01C0 of D1FIFO port register shown in table 21.2 are exclusive for 16- or 32- byte consecutive accesses. When accesses the FIFO buffer with 16- or 32-byte consecutively, the bit width has to be 32bit. Select to be used either of endian. If D0FIFO or D1FIFO is accessed in cycle steal mode or by the CPU, address H'FE400018 for D0FIFO or H'FE40001C for D1FIFO should be accessed. However, in the case of access to CFIFO, D0FIFO or D1FIFO as little-endian data with any width other than 21 bits, the address is changed. See tables 21.8 to 21.10 for details.

These registers are initialized by a power-on reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIFOPORT[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
	FIFOPORT[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	FIFOPORT [31:0]	All 0	R/W	<p>FIFO Port</p> <p>Accessing these bits allow reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.</p> <p>These bits can be accessed only while the FRDY bit in each control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.</p> <p>The valid bits in this register depend on the settings of the MBW bits (access bit width setting) and BIGEND bit (endian setting) as shown in tables 21.8 to 21.10.</p>

Table 21.8 Endian Operation in 32-Bit Access (when MBW = 10)

BIGEND Bit	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0	Access Address		
					CFIFO	D0FIFO	D1FIFO
0	N + 3 address	N + 2 address	N + 1 address	N + 0 address	H'FE40 0014	H'FE40 0018	H'FE40 001C
1	N + 0 address	N + 1 address	N + 2 address	N + 3 address			



**Table 21.9 Endian Operation in 16-Bit Access (when MBW = 01)**

BIGEND Bit	Bits	Bits	Bits	Bits	Access Address		
	31 to 24	23 to 16	15 to 8	7 to 0	CFIFO	D0FIFO	D1FIFO
0	Write: invalid Read: prohibited*		N + 1 address	N + 0 address	H'FE40 0016	H'FE40 001A	H'FE40 001E
1	N + 0 address	N + 1 address	Write: invalid Read: prohibited*		H'FE40 0014	H'FE40 0018	H'FE40 001C

Note: \* Reading data from the invalid bits in a word or byte unit is prohibited.

**Table 21.10 Endian Operation in 8-Bit Access (when MBW = 00)**

BIGEND Bit	Bits	Bits	Bits	Bits	Access Address		
	31 to 24	23 to 16	15 to 8	7 to 0	CFIFO	D0FIFO	D1FIFO
0	Write: invalid Read: prohibited*			N + 0 address	H'FE40 0017	H'FE40 001B	H'FE40 001F
1	N + 0 address	Write: invalid Read: prohibited*			H'FE40 0014	H'FE40 0018	H'FE40 001C

Note: \* Reading data from the invalid bits in a word or byte unit is prohibited.

### 21.3.8 FIFO Port Select Registers (CFIFOSEL, D0FIFOSEL, D1FIFOSEL)

CFIFOSEL, D0FIFOSEL and D1FIFOSEL are registers that assign the pipe to the FIFO port, and control access to the corresponding port.

The same pipe should not be specified by the CURPIPE bits in CFIFOSEL, D0FIFOSEL and D1FIFOSEL. When the CURPIPE bits in D0FIFOSEL and D1FIFOSEL are cleared to B'000, no pipe is selected.

The pipe number should not be changed while the DMA transfer is enabled.

These registers are initialized by a power-on reset.

#### (1) CFIFOSEL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	—	—	MBW[1:0]		—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W*	R	R	R/W	R/W	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RCNT	0	R/W	<p>Read Count Mode</p> <p>Specifies the read mode for the value in the DTLN bits in CFIFOCTR.</p> <p>0: The DTLN bit is cleared when all of the receive data has been read from the CFIFO.</p> <p>(In double buffer mode, the DTLN bit value is cleared when all the data has been read from a single plane.)</p> <p>1: The DTLN bit is decremented when the receive data is read from the CFIFO.</p>

Bit	Bit Name	Initial Value	R/W	Description
14	REW	0	R/W*	<p>Buffer Pointer Rewind</p> <p>Specifies whether or not to rewind the buffer pointer.</p> <p>0: The buffer pointer is not rewind.</p> <p>1: The buffer pointer is rewind.</p> <p>When the selected pipe is in the receiving direction, setting this bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).</p> <p>Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, be sure to check that FRDY is 1.</p> <p>To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.</p>
13, 12	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
11, 10	MBW[1:0]	00	R/W	<p>CFIFO Port Access Bit Width</p> <p>Specifies the bit width for accessing the CFIFO port.</p> <p>00: 8-bit width</p> <p>01: 16-bit width</p> <p>10: 32-bit width</p> <p>11: Setting prohibited</p> <p>When the selected pipe is in the receiving direction, once reading data is started after setting these bits, these bits should not be modified until all the data has been read.</p> <p>When the selected pipe is in the receiving direction, set the CURPIPE and MBW bits simultaneously.</p> <p>When the selected pipe is in the transmitting direction, the bit width cannot be changed from the 8-bit width to the 16-/32-bit width or from the 16-bit width to the 32-bit width while data is being written to the buffer memory.</p> <p>The odd number of bytes can also be written through byte-access control even when 8- or 16-bit width is selected.</p>

Bit	Bit Name	Initial Value	R/W	Description
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	BIGEND	0	R/W	CFIFO Port Endian Control Specifies the byte endian for the CFIFO port. 0: Little endian 1: Big endian
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	ISEL	0	R/W	CFIFO Port Access Direction When DCP is Selected 0: Reading from the buffer memory is selected 1: Writing to the buffer memory is selected After writing to this bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. Even if an attempt is made to modify the setting of this bit during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective thus enabling continuous access. Set this bit and the CURPIPE bits simultaneously.
4	—	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
3 to 0	CURPIPE[3:0]	0000	R/W	<p>CFIFO Port Access Pipe Specification</p> <p>Specifies the pipe number using which data is read or written through the CFIFO port.</p> <p>0000: DCP</p> <p>0001: Pipe 1</p> <p>0010: Pipe 2</p> <p>0011: Pipe 3</p> <p>0100: Pipe 4</p> <p>0101: Pipe 5</p> <p>0110: Pipe 6</p> <p>0111: Pipe 7</p> <p>1000: Pipe 8</p> <p>1001: Pipe 9</p> <p>Other than above: Setting prohibited</p> <p>After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.</p> <p>Do not set the same pipe number to the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.</p> <p>Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective thus enabling continuous access.</p>

Note: \* Only 0 can be read.

## (2) D0FIFOSEL, D1FIFOSEL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	DCLRM	DREQE	MBW[1:0]	—	BIG END	—	—	—	—	—	CURPIPE[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W*	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RCNT	0	R/W	<p>Read Count Mode</p> <p>Specifies the read mode for the value in the DTLN bits in DnFIFOCTR.</p> <p>0: The DTLN bit is cleared when all of the receive data has been read from the DnFIFO.</p> <p>(In double buffer mode, the DTLN bit value is cleared when all the data has been read from a single plane.)</p> <p>1: The DTLN bit is decremented when the receive data is read from the DnFIFO.</p> <p>When accessing DnFIFO with the BFRE bit set to 1, set this bit to 0.</p>
14	REW	0	R/W*	<p>Buffer Pointer Rewind</p> <p>Specifies whether or not to rewind the buffer pointer.</p> <p>0: The buffer pointer is not rewound.</p> <p>1: The buffer pointer is rewound.</p> <p>When the selected pipe is in the receiving direction, setting this bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).</p> <p>Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, be sure to check that FRDY is 1.</p> <p>When accessing DnFIFO with the BFRE bit set to 1, do not set this bit to 1 in the state in which the short packet data has been read out.</p> <p>To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.</p>

Bit	Bit Name	Initial Value	R/W	Description
13	DCLRM	0	R/W	<p>Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read</p> <p>Enables or disables the buffer memory to be cleared automatically after data has been read out using the selected pipe.</p> <p>0: Auto buffer clear mode is disabled.</p> <p>1: Auto buffer clear mode is enabled.</p> <p>With this bit set to 1, this module sets BCLR to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while BFRE is 1.</p> <p>When using this module with the BRDYM bit set to 1, set this bit to 0.</p>
12	DREQE	0	R/W	<p>DMA Transfer Request Enable</p> <p>Enables or disables the DMA transfer request to be issued.</p> <p>0: Request disabled</p> <p>1: Request enabled</p> <p>Before setting this bit to 1 to enable the DMA transfer request to be issued, set the CURPIPE bits.</p> <p>Before modifying the CURPIPE bit setting, set this bit to 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
11, 10	MBW[1:0]	All 0	R/W	<p>FIFO Port Access Bit Width</p> <p>Specifies the bit width for accessing the DnFIFO port.</p> <p>00: 8-bit width</p> <p>01: 16-bit width</p> <p>10: 32-bit width</p> <p>11: Setting prohibited</p> <p>When the selected pipe is in the receiving direction, once reading data is started after setting these bits, these bits should not be modified until all the data has been read.</p> <p>When the selected pipe is in the receiving direction, set the CURPIPE and MBW bits simultaneously.</p> <p>When the selected pipe is in the transmitting direction, the bit width cannot be changed from the 8-bit width to the 16-/32-bit width or from the 16-bit width to the 32-bit width while data is being written to the buffer memory.</p> <p>The odd number of bytes can be written through byte-access control even when 8- or 16-bit width is selected.</p>
9	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
8	BIGEND	0	R/W	<p>FIFO Port Endian Control</p> <p>Specifies the byte endian for the DnFIFO port.</p> <p>0: Little endian</p> <p>1: Big endian</p>
7 to 4	—	All 0	R/W	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>



Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CURPIPE[3:0]	0000	R/W	<p>FIFO Port Access Pipe Specification</p> <p>Specifies the pipe number using which data is read or written through the D0FIFO/D1FIFO port.</p> <p>0000: No pipe specified</p> <p>0001: Pipe 1</p> <p>0010: Pipe 2</p> <p>0011: Pipe 3</p> <p>0100: Pipe 4</p> <p>0101: Pipe 5</p> <p>0110: Pipe 6</p> <p>0111: Pipe 7</p> <p>1000: Pipe 8</p> <p>1001: Pipe 9</p> <p>Other than above: Setting prohibited</p> <p>After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.</p> <p>Do not set the same pipe number to the CURPIPE bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.</p> <p>Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective thus enabling continuous access.</p>

Note: \* Only 0 can be read.

### 21.3.9 FIFO Port Control Registers (CFIFOCTR, D0FIFOCTR, D1FIFOCTR)

CFIFOCTR, D0FIFOCTR and D1FIFOCTR are registers that determine whether or not writing to the buffer memory has been finished, the buffer accessed from the CPU has been cleared, and the FIFO port is accessible. CFIFOCTR, D0FIFOCTR, and D1FIFOCTR are used for the corresponding FIFO ports.

These registers are initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BVAL	BCLR	FRDY	—	DTLN[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*2	R/W*1	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	BVAL	0	R/W*2	<p>Buffer Memory Valid Flag</p> <p>This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE bits (selected pipe).</p> <p>0: Invalid</p> <p>1: Writing ended</p> <p>When the selected pipe is in the transmitting direction, set this bit to 1 in the following cases. Then, this module switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.</p> <ul style="list-style-type: none"> <li>To transmit a short packet, set this bit to 1 after data has been written.</li> <li>To transmit a zero-length packet, set this bit to 1 before data is written to the FIFO buffer.</li> <li>Set this bit to 1 after the number of data bytes has been written for the pipe in continuous transfer mode, where the number is a natural integer multiple of the maximum packet size and less than the buffer size.</li> </ul> <p>When the data of the maximum packet size has been written for the pipe in continuous transfer mode, this module sets this bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.</p> <p>Writing 1 to this bit should be done while FRDY indicates 1 (set by this module).</p> <p>When the selected pipe is in the receiving direction, do not set this bit to 1.</p>

Bit	Bit Name	Initial Value	R/W	Description
14	BCLR	0	R/W* <sup>1</sup>	<p><b>CPU Buffer Clear</b></p> <p>This bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.</p> <p>0: Invalid</p> <p>1: Clears the buffer memory on the CPU side.</p> <p>When double buffer mode is set for the FIFO buffer assigned to the selected pipe, this module clears only one plane of the FIFO buffer even when both planes are read-enabled.</p> <p>When the selected pipe is the DCP, setting BCLR to 1 allows this module to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the PID bits for the DCP to NAK before setting BCLR to 1.</p> <p>When the selected pipe is in the transmitting direction, if 1 is written to BVAL and BCLR bits simultaneously, this module clears the data that has been written before it, enabling transmission of a zero-length packet.</p> <p>When the selected pipe is not the DCP, writing 1 to this bit should be done while FRDY indicates 1 (set by this module).</p>
13	FRDY	0	R	<p><b>FIFO Port Ready</b></p> <p>Indicates whether the FIFO port can be accessed by the CPU (DMAC).</p> <p>0: FIFO port access is disabled.</p> <p>1: FIFO port access is enabled.</p> <p>In the following cases, this module sets FRDY to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.</p> <ul style="list-style-type: none"> <li>• A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.</li> <li>• A short packet is received and the data is completely read while BFRE is 1.</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 to 0	DTLN[11:0]	H'000	R	Receive Data Length Indicates the length of the receive data. While the FIFO buffer is being read, these bits indicate the different values depending on the RCNT bit value as described below. <ul style="list-style-type: none"> <li>RCNT = 0: This module sets these bits to indicate the length of the receive data until the CPU (DMAC) has read all the received data from a single FIFO buffer plane. While BFRE is 1, these bits retain the length of the receive data until BCLR is set to 1 even after all the data has been read.</li> <li>RCNT = 1: This module decrements the value indicated by these bits each time data is read from the FIFO buffer. (The value is decremented by one when MBW is 0, and by two when MBW is 1.) This module sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, this module sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane. When RCNT is 1, reading these bits while the FIFO buffer is being read returns the latest value within 150 ns after the FIFO port read cycle.</li> </ul>

Notes: 1. Only 0 can be read and 1 can be written to.  
2. Only 1 can be written to.

### 21.3.10 Interrupts Enable Register 0 (INTENB0)

INTENB0 is a register that specifies the various interrupt masks. On detecting the interrupt corresponding to the bit in this register to which software has set 1, this module generates the USB interrupt.

This module sets 1 to each status bit in INTSTS0 when a detection condition of the corresponding interrupt source has been satisfied regardless of the set value in INTENB0 (regardless of whether the interrupt output is enabled or disabled).

While the status bit in INTSTS0 corresponding to the interrupt source indicates 1, this module generates the USB interrupt when software modifies the corresponding interrupt enable bit in INTENB0 from 0 to 1.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
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	Bit Name	Initial Value	R/W	Description
15	VBSE	0	R/W	VBUS Interrupts Enable Enables or disables the USB interrupt output when the VBINT interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
14	RSME	0	R/W	Resume Interrupts Enable Enables or disables the USB interrupt output when the RESM interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
13	SOFE	0	R/W	<p>Frame Number Update Interrupts Enable</p> <p>Enables or disables the USB interrupt output when the SOFR interrupt is detected.</p> <p>0: Interrupt output disabled</p> <p>1: Interrupt output enabled</p>
12	DVSE	0	R/W	<p>Device State Transition Interrupts Enable*</p> <p>Enables or disables the USB interrupt output when the DVST interrupt is detected.</p> <p>0: Interrupt output disabled</p> <p>1: Interrupt output enabled</p>
11	CTRE	0	R/W	<p>Control Transfer Stage Transition Interrupts Enable*</p> <p>Enables or disables the USB interrupt output when the CTRT interrupt is detected.</p> <p>0: Interrupt output disabled</p> <p>1: Interrupt output enabled</p>
10	BEMPE	0	R/W	<p>Buffer Empty Interrupts Enable</p> <p>Enables or disables the USB interrupt output when the BEMP interrupt is detected.</p> <p>0: Interrupt output disabled</p> <p>1: Interrupt output enabled</p>
9	NRDYE	0	R/W	<p>Buffer Not Ready Response Interrupts Enable</p> <p>Enables or disables the USB interrupt output when the NRDY interrupt is detected.</p> <p>0: Interrupt output disabled</p> <p>1: Interrupt output enabled</p>

Bit	Bit Name	Initial Value	R/W	Description
8	BRDYE	0	R/W	Buffer Ready Interrupts Enable Enables or disables the USB interrupt output when the BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: \* The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller function is selected; do not set these bits to 1 to enable the corresponding interrupt output when the host controller function is selected.

### 21.3.11 Interrupt Enable Register 1 (INTENB1)

INTENB1 is a register that specifies the various interrupt masks when the host controller function is selected. On detecting the interrupt corresponding to the bit in this register to which software has set 1, this module generates the USB interrupt.

This module sets 1 to each status bit in INTSTS1 when a detection condition of the corresponding interrupt source has been satisfied regardless of the set value in INTENB1 (regardless of whether the interrupt output is enabled or disabled).

While the status bit in INTSTS1 corresponding to the interrupt source indicates 1, this module generates the USB interrupt when software modifies the corresponding interrupt enable bit in INTENB1 from 0 to 1.

When the function controller function is selected, the interrupts should not be enabled. This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHGE	—	DTCHE	ATT CHE	—	—	—	—	EOF ERRE	SIGNE	SACKE	—	—	—	—
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/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	BCHGE	0	R/W	USB Bus Change Interrupt Enable Enables or disables the USB interrupt output when the BCHG interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	DTCHE	0	R/W	Disconnection Detection Interrupt Enable Enables or disables the USB interrupt output when the DTCH interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
11	ATTCH	0	R/W	Connection Detection Interrupt Enable Enables or disables the USB interrupt output when the ATTCH interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
10 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	EOFERRE	0	R/W	EOF Error Detection Interrupt Enable Enables or disables the USB interrupt output when the EOFERR interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled
5	SIGNE	0	R/W	Setup Transaction Error Interrupt Enable Enables or disables the USB interrupt output when the SIGN interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled



Bit	Bit Name	Initial Value	R/W	Description
4	SACKE	0	R/W	Setup Transaction Normal Response Interrupt Enable  Enables or disables the USB interrupt output when the SACK interrupt is detected.  0: Interrupt output disabled 1: Interrupt output enabled
3 to 0	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.

Note: The INTENB1 register bits can be set to 1 only when the host controller function is selected; do not set these bits to 1 to enable the corresponding interrupt output when the function controller function is selected.

### 21.3.12 BRDY Interrupt Enable Register (BRDYENB)

BRDYENB is a register that enables or disables the BRDY bit in INTSTS0 to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in this register to which software has set 1, this module sets 1 to the corresponding PIPEBRDY bit in BRDYSTS and the BRDY bit in INTSTS0, and generates the BRDY interrupt.

While at least one PIPEBRDY bit in BRDYSTS indicates 1, this module generates the BRDY interrupt when software modifies the corresponding interrupt enable bit in BRDYENB from 0 to 1.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BRDYE	PIPE8 BRDYE	PIPE7 BRDYE	PIPE6 BRDYE	PIPE5 BRDYE	PIPE4 BRDYE	PIPE3 BRDYE	PIPE2 BRDYE	PIPE1 BRDYE	PIPE0 BRDYE
Initial value:	0	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	PIPE9BRDYE	0	R/W	BRDY interrupt Enable for PIPE9 0: Interrupt output disabled 1: Interrupt output enabled
8	PIPE8BRDYE	0	R/W	BRDY interrupt Enable for PIPE8 0: Interrupt output disabled 1: Interrupt output enabled
7	PIPE7BRDYE	0	R/W	BRDY interrupt Enable for PIPE7 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6BRDYE	0	R/W	BRDY interrupt Enable for PIPE6 0: Interrupt output disabled 1: Interrupt output enabled
5	PIPE5BRDYE	0	R/W	BRDY interrupt Enable for PIPE5 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4BRDYE	0	R/W	BRDY interrupt Enable for PIPE4 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3BRDYE	0	R/W	BRDY interrupt Enable for PIPE3 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2BRDYE	0	R/W	BRDY interrupt Enable for PIPE2 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
1	PIPE1BRDYE	0	R/W	BRDY interrupt Enable for PIPE1 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0BRDYE	0	R/W	BRDY interrupt Enable for PIPE0 0: Interrupt output disabled 1: Interrupt output enabled

### 21.3.13 NRDY Interrupt Enable Register (NRDYENB)

NRDYENB is a register that enables or disables the NRDY bit in INTSTS0 to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in this register to which software has set 1, this module sets 1 to the corresponding PIPENRDY bit in NRDYSTS and the NRDY bit in INTSTS0, and generates the NRDY interrupt.

While at least one PIPENRDY bit in NRDYSTS indicates 1, this module generates the NRDY interrupt when software modifies the corresponding interrupt enable bit in NRDYENB from 0 to 1.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 NRDYE	PIPE8 NRDYE	PIPE7 NRDYE	PIPE6 NRDYE	PIPE5 NRDYE	PIPE4 NRDYE	PIPE3 NRDYE	PIPE2 NRDYE	PIPE1 NRDYE	PIPE0 NRDYE
Initial value:	0	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	PIPE9NRDYE	0	R/W	NRDY Interrupt Enable for PIPE9 0: Interrupt output disabled 1: Interrupt output enabled
8	PIPE8NRDYE	0	R/W	NRDY Interrupt Enable for PIPE8 0: Interrupt output disabled 1: Interrupt output enabled
7	PIPE7NRDYE	0	R/W	NRDY Interrupt Enable for PIPE7 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6NRDYE	0	R/W	NRDY Interrupt Enable for PIPE6 0: Interrupt output disabled 1: Interrupt output enabled
5	PIPE5NRDYE	0	R/W	NRDY Interrupt Enable for PIPE5 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4NRDYE	0	R/W	NRDY Interrupt Enable for PIPE4 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3NRDYE	0	R/W	NRDY Interrupt Enable for PIPE3 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2NRDYE	0	R/W	NRDY Interrupt Enable for PIPE2 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
1	PIPE1NRDYE	0	R/W	NRDY Interrupt Enable for PIPE1 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0NRDYE	0	R/W	NRDY Interrupt Enable for PIPE0 0: Interrupt output disabled 1: Interrupt output enabled

### 21.3.14 BEMP Interrupt Enable Register (BEMPENB)

BEMPENB is a register that enables or disables the BEMP bit in INTSTS0 to be set to 1 when the BEMP interrupt is detected for each pipe.

On detecting the BEMP interrupt for the pipe corresponding to the bit in this register to which software has set 1, this module sets 1 to the corresponding PIPEBEMP bit in BEMPSTS and the BEMP bit in INTSTS0, and generates the BEMP interrupt.

While at least one PIPEBEMP bit in BEMPSTS indicates 1, this module generates the BEMP interrupt when software modifies the corresponding interrupt enable bit in BEMPENB from 0 to 1.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BEMPE	PIPE8 BEMPE	PIPE7 BEMPE	PIPE6 BEMPE	PIPE5 BEMPE	PIPE4 BEMPE	PIPE3 BEMPE	PIPE2 BEMPE	PIPE1 BEMPE	PIPE0 BEMPE
Initial value:	0	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	PIPE9BEMPE	0	R/W	BEMP Interrupt Enable for PIPE9 0: Interrupt output disabled 1: Interrupt output enabled
8	PIPE8BEMPE	0	R/W	BEMP Interrupt Enable for PIPE8 0: Interrupt output disabled 1: Interrupt output enabled
7	PIPE7BEMPE	0	R/W	BEMP Interrupt Enable for PIPE7 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6BEMPE	0	R/W	BEMP Interrupt Enable for PIPE6 0: Interrupt output disabled 1: Interrupt output enabled
5	PIPE5BEMPE	0	R/W	BEMP Interrupt Enable for PIPE5 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4BEMPE	0	R/W	BEMP Interrupt Enable for PIPE4 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3BEMPE	0	R/W	BEMP Interrupt Enable for PIPE3 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2BEMPE	0	R/W	BEMP Interrupt Enable for PIPE2 0: Interrupt output disabled 1: Interrupt output enabled
1	PIPE1BEMPE	0	R/W	BEMP Interrupt Enable for PIPE1 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0BEMPE	0	R/W	BEMP Interrupt Enable for PIPE0 0: Interrupt output disabled 1: Interrupt output enabled

### 21.3.15 SOF Control Register (SOFCFG)

SOFCFG is a register that specifies the transaction-enabled time and BRDY interrupt status clear timing.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	BRDYM	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0*	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R	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	BRDYM	0	R/W	BRDY Interrupt Status Clear Timing for each Pipe Specifies the timing for clearing the BRDY interrupt status for each pipe. 0: Software clears the status. 1: This module clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.
5	—	0*	R	Reserved This bit is reserved. The previously read value should be written to this bit. Note: Although this bit is initialized to 0 by a power-on reset, be sure to set this bit to 1 using the initialization routine of this module.
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: \* Although this bit is initialized to 0 by a power-on reset, be sure to set this bit to 1 using the initialization routine of this module.

### 21.3.16 Interrupt Status Register 0 (INTSTS0)

INTSTS0 is a register that indicates the status of the various interrupts detected.

This register is initialized by a power-on reset. By a USB bus reset, the DVSQ2 to DVSQ0 bits are initialized.

Bit:	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[2:0]			VALID	CTSQ[2:0]		
Initial value:	0	0	0	0	0	0	0	0	*3	*2	*2	*2	0	0	0	0
R/W:	R/W*7	R/W*7	R/W*7	R/W*7	R/W*7	R	R	R	R	R	R	R	R/W*7	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	VBINT	0	R/W*7	VBUS Interrupt Status*4*5 0: VBUS interrupts not generated 1: VBUS interrupts generated This module sets this bit to 1 on detecting a level change (high to low or low to high) in the VBUS pin input value. This module sets the VBSTS bit to indicate the VBUS pin input value. When the VBUS interrupt is generated, use software to repeat reading the VBSTS bit until the same value is read three or more times, and eliminate chattering.
14	RESM	0	R/W*7	Resume Interrupt Status*4*5*6 0: Resume interrupts not generated 1: Resume interrupts generated When the function controller function is selected, this module sets this bit to 1 on detecting the falling edge of the signal on the DP pin in the suspended state (DVSQ = 1XX). When the host controller function is selected, the read value is invalid.



Bit	Bit Name	Initial Value	R/W	Description
13	SOFR	0	R/W* <sup>7</sup>	<p>Frame Number Refresh Interrupt Status*<sup>4</sup></p> <p>0: SOF interrupts not generated 1: SOF interrupts generated</p> <p>(1) When the host controller function is selected This module sets this bit to 1 on updating the frame number when software has set the UACT bit to 1. (This interrupt is detected every 1 ms.)</p> <p>(2) When the function controller function is selected This module sets this bit to 1 on updating the frame number. (This interrupt is detected every 1 ms.)</p> <p>This module can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.</p>
12	DVST	0/1* <sup>1</sup>	R/W* <sup>7</sup>	<p>Device State Transition Interrupt Status*<sup>4</sup>*<sup>6</sup></p> <p>0: Device state transition interrupts not generated 1: Device state transition interrupts generated</p> <p>When the function controller function is selected, this module updates the DVSQ value and sets this bit to 1 on detecting a change in the device state.</p> <p>When this interrupt is generated, clear the status before this module detects the next device state transition.</p> <p>When the host controller function is selected, the read value is invalid.</p>

Bit	Bit Name	Initial Value	R/W	Description
11	CTRT	0	R/W* <sup>7</sup>	<p>Control Transfer Stage Transition Interrupt Status*<sup>4</sup>*<sup>6</sup></p> <p>0: Control transfer stage transition interrupts not generated</p> <p>1: Control transfer stage transition interrupts generated</p> <p>When the function controller function is selected, this module updates the CTSQ value and sets this bit to 1 on detecting a change in the control transfer stage.</p> <p>When this interrupt is generated, clear the status before this module detects the next control transfer stage transition.</p> <p>When the host controller function is selected, the read value is invalid.</p>
10	BEMP	0	R	<p>Buffer Empty Interrupt Status</p> <p>0: BEMP interrupts not generated</p> <p>1: BEMP interrupts generated</p> <p>This module sets this bit to 1 when at least one PIPEBEMP bit in BEMPSTS is set to 1 among the PIPEBEMP bits corresponding to the PIPEBEMPE bits in BEMPENB to which 1 has been set (when this module detects the BEMP interrupt status in at least one pipe among the pipes for which software enables the BEMP interrupt output).</p> <p>For the conditions for PIPEBEMP status assertion, refer to (3) BEMP Interrupts under section 21.4.2, Interrupt Functions.</p> <p>This module clears this bit to 0 when software writes 0 to all the PIPEBEMP bits corresponding to the PIPEBEMPE bits to which 1 has been set.</p> <p>This bit cannot be cleared to 0 even if software writes 0 to this bit.</p>

Bit	Bit Name	Initial Value	R/W	Description
9	NRDY	0	R	<p>Buffer Not Ready Interrupt Status</p> <p>0: NRDY interrupts not generated 1: NRDY interrupts generated</p> <p>This module sets this bit to 1 when at least one PIPENRDY bit in NRDYSTS is set to 1 among the PIPENRDY bits corresponding to the PIPENRDYE bits in NRDYENB to which 1 has been set (when this module detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).</p> <p>For the conditions for PIPENRDY status assertion, refer to (2) NRDY Interrupts under section 21.4.2, Interrupt Functions.</p> <p>This module clears this bit to 0 when software writes 0 to all the PIPENRDY bits corresponding to the PIPENRDYE bits to which 1 has been set.</p> <p>This bit cannot be cleared to 0 even if software writes 0 to this bit.</p>
8	BRDY	0	R	<p>Buffer Ready Interrupt Status</p> <p>Indicates the BRDY interrupt status.</p> <p>0: BRDY interrupts not generated 1: BRDY interrupts generated</p> <p>This module sets this bit to 1 when at least one PIPEBRDY bit in BRDYSTS is set to 1 among the PIPEBRDY bits corresponding to the PIPEBRDYE bits in BRDYENB to which 1 has been set (when this module detects the BRDY interrupt status in at least one pipe among the pipes for which software enables the BRDY interrupt output).</p> <p>For the conditions for PIPEBRDY status assertion, refer to (1) BRDY Interrupts under section 21.4.2, Interrupt Functions.</p> <p>This module clears this bit to 0 when software writes 0 to all the PIPEBRDY bits corresponding to the PIPEBRDYE bits to which 1 has been set.</p> <p>This bit cannot be cleared to 0 even if software writes 0 to this bit.</p>

Bit	Bit Name	Initial Value	R/W	Description
7	VBSTS	0/1* <sup>3</sup>	R	VBUS Input Status 0: The VBUS pin is low level. 1: The VBUS pin is high level.
6 to 4	DVSQ[2:0]	000/001* <sup>2</sup>	R	Device State 000: Powered state 001: Default state 010: Address state 011: Configured state 1xx: Suspended state When the host controller function is selected, the read value is invalid.
3	VALID	0	R/W* <sup>7</sup>	USB Request Reception 0: Not detected 1: Setup packet reception When the host controller function is selected, the read value is invalid.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	CTSQ[2:0]	000	R	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: Setting prohibited When the host controller function is selected, the read value is invalid.

- Notes:
1. This bit is initialized to B'0 by a power-on reset and B'1 by a USB bus reset.
  2. These bits are initialized to B'000 by a power-on reset and B'001 by a USB bus reset.
  3. This bit is initialized to 0 when the level of the VBUS pin input is high and 1 when low.
  4. To clear the VBINT, RESM, SOFR, DVST, or CTRT bit, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.
  5. A change in the status indicated by the VBINT and RESM bits can be detected even while the clock supply is stopped (while SCKE is 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.
  6. A change in the status of the RESM, DVST, and CTRT bits occur only when the function controller function is selected; disable the corresponding interrupt enable bits (set to 0) when the function controller function is selected.
  7. Only 0 can be written to.

### 21.3.17 Interrupt Status Register 1 (INTSTS1)

INTSTS1 is a register that is used to confirm interrupt status.

Interrupt generation can be confirmed simply by referencing one of the registers: INTSTS0 when the function controller function is selected and INTSTS1 when the host controller function is selected.

The various interrupts indicated by the bits in this register should be enabled only when the host controller function is selected.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHG	—	DTCH	ATTCH	—	—	—	—	EOF ERR	SIGN	SACK	—	—	—	—
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	R/W*1	R/W*1	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved
				This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14	BCHG	0	R/W* <sup>1</sup>	<p>USB Bus Change Interrupt Status</p> <p>Indicates the status of the USB bus change interrupt.</p> <p>0: BCHG interrupts not generated</p> <p>1: BCHG interrupts generated</p> <p>This module detects the BCHG interrupt when a change in the full-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets this bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, this module generates the interrupt.</p> <p>This module sets the LNST bits in SYSSTS0 to indicate the current input state of the USB port.</p> <p>When the BCHG interrupt is generated, use software to repeat reading the LNST bits until the same value is read three or more times, and eliminate chattering.</p> <p>A change in the USB bus state can be detected even while the internal clock supply is stopped.</p> <p>When the function controller function is selected, the read value is invalid.</p>
13	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
12	DTCH	0	R/W* <sup>1</sup>	<p>USB Disconnection Detection Interrupt Status</p> <p>Indicates the status of the USB disconnection detection interrupt when the host controller function is selected.</p> <p>0: DTCH interrupts not generated 1: DTCH interrupts generated</p> <p>This module detects the DTCH interrupt on detecting USB bus disconnection, and sets this bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, this module generates the interrupt. This module detects bus disconnection based on USB Specification 2.0.</p> <p>After detecting the DTCH interrupt, this module controls hardware as described below (irrespective of the set value of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried out for the USB port and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation).</p> <ul style="list-style-type: none"> <li>• Modifies the UACT bit for the port in which a DTCH interrupt has been detected to 0.</li> <li>• Puts the port in which a DTCH interrupt has been generated into the idle state.</li> </ul> <p>When the function controller function is selected, the read value is invalid.</p>



Bit	Bit Name	Initial Value	R/W	Description
11	ATTCH	0	R/W* <sup>1</sup>	<p><b>ATTCH Interrupt Status</b></p> <p>Indicates the status of the ATTCH interrupt when the host controller function is selected.</p> <p>0: ATTCH interrupts not generated</p> <p>1: ATTCH interrupts generated</p> <p>This module detects the ATTCH interrupt on detecting J-state or K-state of the full-speed level signal for 2.5 <math>\mu</math>s, and sets this bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, this module generates the interrupt.</p> <p>Specifically, this module detects the ATTCH interrupt on any of the following conditions.</p> <ul style="list-style-type: none"> <li>• K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 <math>\mu</math>s.</li> <li>• J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 <math>\mu</math>s.</li> </ul> <p>When the function controller function is selected, the read value is invalid.</p>
10 to 7	—	All 0	R	<p><b>Reserved</b></p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
6	EOFERR	0	R/W* <sup>1</sup>	<p><b>EOF Error Detection Interrupt Status</b></p> <p>Indicates the status of the EOFERR interrupt when the host controller function is selected.</p> <p>0: EOFERR interrupt not generated 1: EOFERR interrupt generated</p> <p>This module detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0, and sets this bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, this module generates the EOFERR interrupt.</p> <p>After detecting the EOFERR interrupt, this module controls hardware as described below (irrespective of the set value of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried for the USB port and perform re-enumeration of the USB port.</p> <ul style="list-style-type: none"><li>• Modifies the UACT bit for the port in which an EOFERR interrupt has been detected to 0.</li><li>• Puts the port in which an EOFERR interrupt has been generated into the idle state.</li></ul> <p>When the function controller function is selected, the read value is invalid.</p>

Bit	Bit Name	Initial Value	R/W	Description
5	SIGN	0	R/W* <sup>1</sup>	<p>Setup Transaction Error Interrupt Status</p> <p>Indicates the status of the setup transaction error interrupt when the host controller function is selected.</p> <p>0: SIGN interrupts not generated</p> <p>1: SIGN interrupts generated</p> <p>This module detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets this bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, this module generates the SIGN interrupt.</p> <p>Specifically, this module detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.</p> <ul style="list-style-type: none"> <li>• Timeout is detected when the peripheral device has returned no response.</li> <li>• A damaged ACK packet is received.</li> <li>• A handshake other than ACK (NAK, NYET, or STALL) is received.</li> </ul> <p>When the function controller function is selected, the read value is invalid.</p>

Bit	Bit Name	Initial Value	R/W	Description
4	SACK	0	R/W* <sup>1</sup>	<p>Setup Transaction Normal Response Interrupt Status</p> <p>Indicates the status of the setup transaction normal response interrupt when the host controller function is selected.</p> <p>0: SACK interrupts not generated 1: SACK interrupts generated</p> <p>This module detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by this module, and sets this bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, this module generates the SACK interrupt.</p>
3 to 0	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Notes: 1. To clear the status indicated by the bits in this register, write 0 only to the bits to be cleared; write 1 to the other bits.

2. A change in the status indicated by the BCHG bit can be detected even while the clock supply is stopped (while SCKE is 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after enabling the clock supply.

No interrupts other than BCHG can be detected while the clock supply is stopped (while SCKE is 0).

### 21.3.18 BRDY Interrupt Status Register (BRDYSTS)

BRDYSTS is a register that indicates the BRDY interrupt status for each pipe.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BRDY	PIPE8 BRDY	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	PIPE3 BRDY	PIPE2 BRDY	PIPE1 BRDY	PIPE0 BRDY
Initial value:	0	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* <sup>1</sup>	R/W* <sup>1</sup>	R/W* <sup>1</sup>	R/W* <sup>1</sup>	R/W* <sup>1</sup>	R/W* <sup>1</sup>	R/W* <sup>1</sup>	R/W* <sup>1</sup>	R/W* <sup>1</sup>	R/W* <sup>1</sup>

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	PIPE9BRDY	0	R/W* <sup>1</sup>	BRDY Interrupt Status for PIPE9* <sup>2</sup> 0: Interrupts not generated 1: Interrupts generated
8	PIPE8BRDY	0	R/W* <sup>1</sup>	BRDY Interrupt Status for PIPE8* <sup>2</sup> 0: Interrupts not generated 1: Interrupts generated
7	PIPE7BRDY	0	R/W* <sup>1</sup>	BRDY Interrupt Status for PIPE7* <sup>2</sup> 0: Interrupts not generated 1: Interrupts generated
6	PIPE6BRDY	0	R/W* <sup>1</sup>	BRDY Interrupt Status for PIPE6* <sup>2</sup> 0: Interrupts not generated 1: Interrupts generated
5	PIPE5BRDY	0	R/W* <sup>1</sup>	BRDY Interrupt Status for PIPE5* <sup>2</sup> 0: Interrupts not generated 1: Interrupts generated
4	PIPE4BRDY	0	R/W* <sup>1</sup>	BRDY Interrupt Status for PIPE4* <sup>2</sup> 0: Interrupts not generated 1: Interrupts generated
3	PIPE3BRDY	0	R/W* <sup>1</sup>	BRDY Interrupt Status for PIPE3* <sup>2</sup> 0: Interrupts not generated 1: Interrupts generated
2	PIPE2BRDY	0	R/W* <sup>1</sup>	BRDY Interrupt Status for PIPE2* <sup>2</sup> 0: Interrupts not generated 1: Interrupts generated

Bit	Bit Name	Initial Value	R/W	Description
1	PIPE1BRDY	0	R/W* <sup>1</sup>	BRDY Interrupt Status for PIPE1* <sup>2</sup> 0: Interrupts not generated 1: Interrupts generated
0	PIPE0BRDY	0	R/W* <sup>1</sup>	BRDY Interrupt Status for PIPE0* <sup>2</sup> 0: Interrupts not generated 1: Interrupts generated

Notes: 1. When BRDYM is 0, to clear the status indicated by the bits in this register, write 0 only to the bits to be cleared; write 1 to the other bits.  
2. When BRDYM is 0, clearing this bit should be done before accessing the FIFO.

### 21.3.19 NRDY Interrupt Status Register (NRDYSTS)

NRDYSTS is a register that indicates the NRDY interrupt status for each pipe.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 NRDY	PIPE8 NRDY	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	PIPE3 NRDY	PIPE2 NRDY	PIPE1 NRDY	PIPE0 NRDY
Initial value:	0	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	PIPE9NRDY	0	R/W*	NRDY Interrupt Status for PIPE9 0: Interrupts not generated 1: Interrupts generated
8	PIPE8NRDY	0	R/W*	NRDY Interrupt Status for PIPE8 0: Interrupts not generated 1: Interrupts generated

Bit	Bit Name	Initial Value	R/W	Description
7	PIPE7NRDY	0	R/W*	NRDY Interrupt Status for PIPE7 0: Interrupts not generated 1: Interrupts generated
6	PIPE6NRDY	0	R/W*	NRDY Interrupt Status for PIPE6 0: Interrupts not generated 1: Interrupts generated
5	PIPE5NRDY	0	R/W*	NRDY Interrupt Status for PIPE5 0: Interrupts not generated 1: Interrupts generated
4	PIPE4NRDY	0	R/W*	NRDY Interrupt Status for PIPE4 0: Interrupts not generated 1: Interrupts generated
3	PIPE3NRDY	0	R/W*	NRDY Interrupt Status for PIPE3 0: Interrupts not generated 1: Interrupts generated
2	PIPE2NRDY	0	R/W*	NRDY Interrupt Status for PIPE2 0: Interrupts not generated 1: Interrupts generated
1	PIPE1NRDY	0	R/W*	NRDY Interrupt Status for PIPE1 0: Interrupts not generated 1: Interrupts generated
0	PIPE0NRDY	0	R/W*	NRDY Interrupt Status for PIPE0 0: Interrupts not generated 1: Interrupts generated

Note: \* To clear the status indicated by the bits in this register, write 0 only to the bits to be cleared; write 1 to the other bits.

### 21.3.20 BEMP Interrupt Status Register (BEMPSTS)

BEMPSTS is a register that indicates the BEMP interrupt status for each pipe.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BEMP	PIPE8 BEMP	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	PIPE3 BEMP	PIPE2 BEMP	PIPE1 BEMP	PIPE0 BEMP
Initial value:	0	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	PIPE9BEMP	0	R/W*	BEMP Interrupts for PIPE9 0: Interrupts not generated 1: Interrupts generated
8	PIPE8BEMP	0	R/W*	BEMP Interrupts for PIPE8 0: Interrupts not generated 1: Interrupts generated
7	PIPE7BEMP	0	R/W*	BEMP Interrupts for PIPE7 0: Interrupts not generated 1: Interrupts generated
6	PIPE6BEMP	0	R/W*	BEMP Interrupts for PIPE6 0: Interrupts not generated 1: Interrupts generated



Bit	Bit Name	Initial Value	R/W	Description
5	PIPE5BEMP	0	R/W*	BEMP Interrupts for PIPE5 0: Interrupts not generated 1: Interrupts generated
4	PIPE4BEMP	0	R/W*	BEMP Interrupts for PIPE4 0: Interrupts not generated 1: Interrupts generated
3	PIPE3BEMP	0	R/W*	BEMP Interrupts for PIPE3 0: Interrupts not generated 1: Interrupts generated
2	PIPE2BEMP	0	R/W*	BEMP Interrupts for PIPE2 0: Interrupts not generated 1: Interrupts generated
1	PIPE1BEMP	0	R/W*	BEMP Interrupts for PIPE1 0: Interrupts not generated 1: Interrupts generated
0	PIPE0BEMP	0	R/W*	BEMP Interrupts for PIPE0 0: Interrupts not generated 1: Interrupts generated

Note: \* To clear the status indicated by the bits in this register, write 0 only to the bits to be cleared; write 1 to the other bits.

### 21.3.21 Frame Number Register (FRMNUM)

FRMNUM is a register that determines the source of isochronous error notification and indicates the frame number.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	OVRN	0	R/W*	<p>Overrun/Underrun Detection Status</p> <p>Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer.</p> <p>0: No error</p> <p>1: An error occurred</p> <p>Software can clear this bit to 0 by writing 0 to the bit. Here, 1 should be written to the other bits in this register.</p> <p>(1) When the host controller function is selected</p> <p>This module sets this bit to 1 on any of the following conditions.</p> <ul style="list-style-type: none"> <li>For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.</li> <li>For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty.</li> </ul> <p>(2) When the function controller function is selected</p> <p>This module sets this bit to 1 on any of the following conditions.</p> <ul style="list-style-type: none"> <li>For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.</li> <li>For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
14	CRCE	0	R/W*	<p>Receive Data Error</p> <p>Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.</p> <p>0: No error</p> <p>1: An error occurred</p> <p>Software can clear this bit to 0 by writing 0 to the bit. Here, 1 should be written to the other bits in this register.</p> <p>(1) When the host controller function is selected</p> <p>On detecting a CRC error, this module generates the internal NRDY interrupt request.</p> <p>(2) When the function controller function is selected</p> <p>On detecting a CRC error, this module does not generate the internal NRDY interrupt request.</p>
13 to 11	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
10 to 0	FRNM[10:0]	H'000	R	<p>Frame Number</p> <p>This module sets these bits to indicate the latest frame number, which is updated every time an SOF packet is issued or received (every 1 ms)</p> <p>Repeat reading these bits until the same value is read twice.</p>

Note: \* Only 0 can be written to

### 21.3.22 $\mu$ Frame Number Register (UFRMNUM)

UFRMNUM is a register that indicates the  $\mu$ frame number.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	UFRNM[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	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
2 to 0	UFRNM[2:0]	000	R	$\mu$ Frame  The $\mu$ frame number can be confirmed.  This module sets these bits to indicate the $\mu$ frame number during high-speed operation.  During operation other than high-speed operation, this module sets these bits to B'000.  Repeat reading these bits until the same value is read twice.

### 21.3.23 USB Address Register (USBADDR)

USBADDR is a register that indicates the USB address. This register is valid only when the function controller function is selected. When the host controller function is selected, peripheral device addresses should be set using the DEVSEL bits in PIPEMAXP.

This register is initialized by a power-on reset or a USB bus reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
6 to 0	USBADDR [6:0]	H'00	R	USB Address  When the function controller function is selected, these bits indicate the USB address assigned by the host when the SET_ADDRESS request is successfully processed.  On detecting the USB reset, this module sets these bits to H'00.  When the host controller function is selected, these bits are invalid.

21.3.24 USB Request Type Register (USBREQ)

USBREQ is a register that stores setup requests for control transfers. When the function controller function is selected, the values of bRequest and bmRequestType that have been received are stored. When the host controller function is selected, the values of bRequest and bmRequestType to be transmitted are set.

This register is initialized by a power-on reset or a USB bus reset.



Bit	Bit Name	Initial Value	R/W	Description
15 to 8	BREQUEST [7:0]	H'00	R/W*	<div>Request</div> <div>These bits store the USB request bRequest value.</div> <div>(1) When the host controller function is selected</div> <div>The USB request data value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1.</div> <div>(2) When the function controller function is selected</div> <div>Indicates the USB request data value received during the setup transaction. Writing to these bits is invalid.</div>

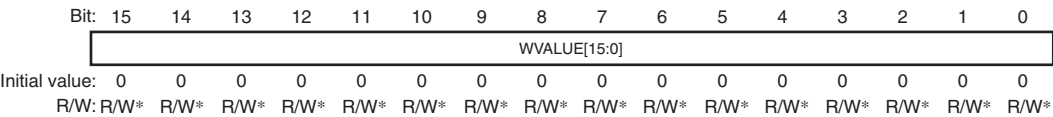
Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BMREQUEST- TYPE[7:0]	H'00	R/W*	<p>Request Type</p> <p>These bits store the USB request bmRequestType value.</p> <p>(1) When the host controller function is selected</p> <p>The USB request type value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1.</p> <p>(2) When the function controller function is selected</p> <p>Indicates the USB request type value received during the setup transaction. Writing to these bits is invalid.</p>

Note: \* When the function controller function is selected, these bits can only be read, and writing to these bits is invalid. When the host controller function is selected, these bits can be read and written to.

21.3.25 USB Request Value Register (USBVAL)

USBVAL is a register that stores setup requests for control transfers. When the function controller function is selected, the value of wValue that has been received is stored. When the host controller function is selected, the value of wValue to be transmitted is set.

This register is initialized by a power-on reset or a USB bus reset.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WValue[15:0]	H'0000	R/W*	<div>Value</div> <div>These bits store the USB request wValue value.</div> <div>(1) When the host controller function is selected</div> <div>The USB request wValue value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1.</div> <div>(2) When the function controller function is selected</div> <div>Indicates the USB request wValue value received during the setup transaction. Writing to these bits is invalid.</div>

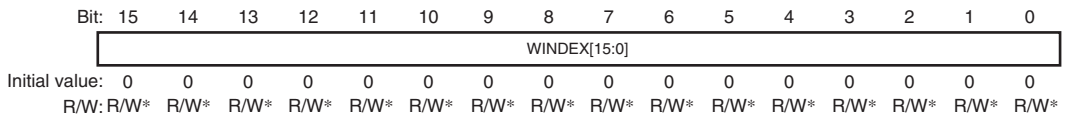
Note: \* When the function controller function is selected, these bits can only be read, and writing to these bits is invalid. When the host controller function is selected, these bits can be read and written to.



### 21.3.26 USB Request Index Register (USBINDEX)

USBINDEX is a register that stores setup requests for control transfers. When the function controller function is selected, the value of wIndex that has been received is stored. When the host controller function is selected, the value of wIndex to be transmitted is set.

This register is initialized by a power-on reset or a USB bus reset.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WINDEX[15:0]	H'0000	R/W*	<p>Index</p> <p>These bits store the USB request wIndex value.</p> <p>(1) When the host controller function is selected</p> <p>The USB request wIndex value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1.</p> <p>(2) When the function controller function is selected</p> <p>Indicates the USB request wIndex value received during the setup transaction. Writing to these bits is invalid.</p>

Note: \* When the function controller function is selected, these bits can only be read, and writing to these bits is invalid. When the host controller function is selected, these bits can be read and written to.

21.3.27 USB Request Length Register (USBLENG)

USBLENG is a register that stores setup requests for control transfers. When the function controller function is selected, the value of wLength that has been received is stored. When the host controller function is selected, the value of wLength to be transmitted is set.

This register is initialized by a power-on reset or a USB bus reset.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WLENGTH [15:0]	H'0000	R/W*	<div>Length</div> <div>These bits store the USB request wLength value.</div> <div>(1) When the host controller function is selected</div> <div>The USB request wLength value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1.</div> <div>(2) When the function controller function is selected</div> <div>Indicates the USB request wLength value received during the setup transaction. Writing to these bits is invalid.</div>

Note: \* When the function controller function is selected, these bits can only be read, and writing to these bits is invalid. When the host controller function is selected, these bits can be read and written to.

### 21.3.28 DCP Configuration Register (DCPCFG)

DCPCFG is a register that specifies the data transfer direction for the default control pipe (DCP).

This register is initialized by a power-on reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	DIR	0	R/W	Transfer Direction When the host controller function is selected, this bit sets the transfer direction of data stage. 0: Data receiving direction 1: Data transmitting direction When the function controller function is selected, this bit should be cleared to 0.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

### 21.3.29 DCP Maximum Packet Size Register (DCPMAXP)

DCPMAXP is a register that specifies the maximum packet size for the DCP.

This register is initialized by a power-on reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	DEVSEL[3:0]	0000	R/W	<p>Device Select</p> <p>When the host controller function is selected, these bits specify the communication target peripheral device address.</p> <p>0000: Address 0000</p> <p>0001: Address 0001</p> <p>: :</p> <p>1001: Address 1001</p> <p>1010: Address 1010</p> <p>Other than above: Setting prohibited</p> <p>These bits should be set after setting the address to the DEVADDn register corresponding to the value to be set in these bits.</p> <p>For example, before setting DEVSEL to 0010, the address should be set to the DEVADD2 register.</p> <p>These bits should be set while CSSTS is 0, PID is NAK, and SUREQ is 0.</p> <p>Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p> <p>When the function controller function is selected, these bits should be set to B'0000.</p>

Bit	Bit Name	Initial Value	R/W	Description
11 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	MXPS[6:0]	H'40	R/W	Maximum Packet Size Specifies the maximum data payload (maximum packet size) for the DCP. These bits are initialized to H'40 (64 bytes). These bits should be set to the value based on the USB Specification. These bits should be set while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary. While MXPS is 0, do not write to the FIFO buffer or do not set PID to BUF.

### 21.3.30 DCP Control Register (DCPCTR)

DCPCTR is a register that is used to confirm the buffer memory status, change and confirm the data PID sequence bit, and set the response PID for the DCP.

This register is initialized by a power-on reset. The CCPL and PID[1:0] bits are initialized by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	SUREQ	CSCLR	CSCTS	SUREQ CLR	—	—	SQCLR	SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R	R/W*2	R/W*1	R	R/W*1	R	R	R/W*1	R/W*1	R	R	R/W	R	R/W*1	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	<p>Buffer Status</p> <p>Indicates whether DCP FIFO buffer access is enabled or disabled.</p> <p>0: Buffer access is disabled.</p> <p>1: Buffer access is enabled.</p> <p>The meaning of the BSTS bit depends on the ISEL bit setting as follows.</p> <ul style="list-style-type: none"> <li>When ISEL = 0, BSTS indicates whether the received data can be read from the buffer.</li> <li>When ISEL = 1, BSTS indicates whether the data to be transmitted can be written to the buffer.</li> </ul>
14	SUREQ	0	R/W* <sup>2</sup>	<p>SETUP Token Transmission</p> <p>Transmits the setup packet by setting this bit to 1 when the host controller function is selected.</p> <p>0: Invalid</p> <p>1: Transmits the setup packet.</p> <p>After completing the setup transaction process, this module generates either the SACK or SIGN interrupt and clears this bit to 0.</p> <p>This module also clears this bit to 0 when software sets the SUREQCLR bit to 1.</p> <p>Before setting this bit to 1, set the DEVSEL bits, USBREQ register, USBVAL register, USBINDX register, and USBLENG register appropriately to transmit the desired USB request in the setup transaction.</p> <p>Before setting this bit to 1, check that the PID bits for the DCP are set to NAK. After setting this bit to 1, do not modify the DEVSEL bits, USBREQ register, USBVAL register, USBINDX register, or USBLENG register until the setup transaction is completed (SUREQ = 1).</p> <p>Write 1 to this bit only when transmitting the setup token; for the other purposes, write 0.</p> <p>When the function controller function is selected, be sure to write 0 to this bit.</p>

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R/W* <sup>1</sup>	<p><b>C-SPLIT Status Clear for Split Transaction</b></p> <p>When the host controller function is selected, setting this bit to 1 clears the CSSTS bit to 0 for the transfer using the split transaction. In this case, the next DCP transfer restarts with the S-SPLIT.</p> <p>0: Invalid</p> <p>1: Clears the CSSTS bit to 0.</p> <p>When software sets this bit to 1, this module clears the CSSTS bit to 0.</p> <p>For the transfer using the split transaction, to restart the next transfer with the S-SPLIT forcibly, set this bit to 1 through software. However, for the normal split transaction, this module automatically clears the CSSTS bit to 0 upon completion of the C-SPLIT; therefore, clearing the CSSTS bit through software is not necessary.</p> <p>Controlling the CSSTS bit through this bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.</p> <p>Setting this bit to 1 while CSSTS is 0 has no effect.</p> <p>When the function controller function is selected, be sure to write 0 to this bit.</p>
12	CSSTS	0	R	<p><b>COMPLETE SPLIT (C-SPLIT) Status of Split Transaction</b></p> <p>Indicates the C-SPLIT status of the split transaction when the host controller function is selected.</p> <p>0: START-SPLIT (S-SPLIT) transaction being processed or the device not using the split transaction being processed</p> <p>1: C-SPLIT transaction being processed</p> <p>This module sets this bit to 1 upon start of the C-SPLIT and clears this bit to 0 upon detection of C-SPLIT completion.</p> <p>When the function controller function is selected, the read value is invalid.</p>

Bit	Bit Name	Initial Value	R/W	Description
11	SUREQCLR	0	R/W* <sup>1</sup>	<p>SUREQ Bit Clear</p> <p>When the host controller function is selected, setting this bit to 1 clears the SUREQ bit to 0.</p> <p>0: Invalid</p> <p>1: Clears the SUREQ bit to 0.</p> <p>This bit always indicates 0.</p> <p>Set this bit to 1 through software when communication has stopped with SUREQ being 1 during the setup transaction. However, for normal setup transactions, this module automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary.</p> <p>Controlling the SUREQ bit through this bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.</p> <p>When the function controller function is selected, be sure to write 0 to this bit.</p>
10, 9	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>



Bit	Bit Name	Initial Value	R/W	Description
8	SQCLR	0	R/W* <sup>1</sup>	<p>Toggle Bit Clear</p> <p>Specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.</p> <p>0: Invalid</p> <p>1: Specifies DATA0.</p> <p>This bit always indicates 0.</p> <p>Do not set the SQCLR and SQSET bits to 1 simultaneously.</p> <p>Set this bit to 1 while CSCTS is 0, PID is NAK, and CURPIPE bits are not yet set.</p> <p>Before setting this bit to 1 after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0.</p> <p>However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>
7	SQSET	0	R/W* <sup>1</sup>	<p>Toggle Bit Set</p> <p>Specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.</p> <p>0: Invalid</p> <p>1: Specifies DATA1.</p> <p>Do not set the SQCLR and SQSET bits to 1 simultaneously.</p> <p>Set this bit to 1 while CSCTS is 0, PID is NAK, and CURPIPE bits are not yet set.</p> <p>Before setting this bit to 1 after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0.</p> <p>However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

Bit	Bit Name	Initial Value	R/W	Description
6	SQMON	1	R	<p>Sequence Toggle Bit Monitor</p> <p>Indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer.</p> <p>0: DATA0 1: DATA1</p> <p>This module allows this bit to toggle upon normal completion of the transaction. However, this bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.</p> <p>When the function controller function is selected, this module sets this bit to 1 (specifies DATA1 as the expected value) upon normal reception of the setup packet.</p> <p>When the function controller function is selected, this module does not reference to this bit during the IN/OUT transaction of the status stage, and does not allow this bit to toggle upon normal completion.</p>
5	PBUSY	0	R	<p>Pipe Busy</p> <p>This bit indicates whether DCP is used or not for the transaction when USB changes the PID bits from BUF to NAK.</p> <p>0: DCP is not used for the transaction. 1: DCP is used for the transaction.</p> <p>This module modifies this bit from 0 to 1 upon start of the USB transaction for the pertinent pipe, and modifies the bit from 1 to 0 upon completion of one transaction.</p> <p>Reading this bit after software has set PID to NAK allows checking that modification of the pipe settings is possible.</p> <p>For details, refer to (1) Pipe Control Register Switching Procedures under section 21.4.3, Pipe Control.</p>

Bit	Bit Name	Initial Value	R/W	Description
4	PINGE	0	R/W	<p>PING Token Issue Enable</p> <p>When the host controller function is selected, setting this bit to 1 allows this module to issue the PING token during transfers in the transmitting direction and start a transfer in the transmitting direction with the PING transaction.</p> <p>0: Disables issuing PING token. 1: Enables normal PING operation.</p> <p>When having detected the ACK handshake during PING transactions, this module performs the OUT transaction as the next transaction.</p> <p>When having detected the NAK handshake during OUT transactions, this module performs the PING transaction as the next transaction.</p> <p>When the host controller function is selected, setting this bit to 0 through software prevents this module from issuing the PING token during transfers in the transmitting direction and only allows this module to perform OUT transactions for the transfers in the transmitting direction.</p> <p>These bits should be modified while CSSTS is 0 and PID is NAK.</p> <p>Before setting this bit to 1 after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p> <p>When the function controller function is selected, be sure to write 0 to this bit.</p>
3	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
2	CCPL	0	R/W* <sup>1</sup>	<p>Control Transfer End Enable</p> <p>When the function controller function is selected, setting this bit to 1 enables the status stage of the control transfer to be completed.</p> <p>0: Invalid</p> <p>1: Completion of control transfer is enabled.</p> <p>When software sets this bit to 1 while the corresponding PID bits are set to BUF, this module completes the control transfer stage.</p> <p>Specifically, during control read transfer, this module transmits the ACK handshake in response to the OUT transaction from the USB host, and outputs the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, this module operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of this bit.</p> <p>This module modifies this bit from 1 to 0 on receiving the new setup packet.</p> <p>Software cannot write 1 to this bit while VALID is 1.</p> <p>When the host controller function is selected, be sure to write 0 to this bit.</p>

Bit	Bit Name	Initial Value	R/W	Description
1,0	PID[1:0]	00	R/W	<p>Response PID</p> <p>Controls the response type of this module during control transfer.</p> <p>00: NAK response</p> <p>01: BUF response (depending on the buffer state)</p> <p>10: STALL response</p> <p>11: STALL response</p> <p>(1) When the host controller function is selected</p> <p>Modify the setting of these bits from NAK to BUF using the following procedure.</p> <ul style="list-style-type: none"> <li>When the transmitting direction is set</li> </ul> <p>Write all the transmit data to the FIFO buffer while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, this module executes the OUT transaction (or PING transaction).</p> <ul style="list-style-type: none"> <li>When the receiving direction is set</li> </ul> <p>Check that the FIFO buffer is empty (or empty the buffer) while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, this module executes the IN transaction.</p> <p>This module modifies the setting of these bits as follows.</p> <ul style="list-style-type: none"> <li>This module sets PID to STALL (11) on receiving the data of the size exceeding the maximum packet size when software has set PID to BUF.</li> <li>This module sets PID to NAK on detecting a receive error such as a CRC error three consecutive times.</li> <li>This module also sets PID to STALL (11) on receiving the STALL handshake.</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
1,0	PID[1:0]	00	R/W	<p>Even if software modifies the PID bits to NAK after this module has issued S-SPLIT of the split transaction for the selected pipe (while CSSTS indicates 1), this module continues the transaction until C-SPLIT completes. On completion of C-SPLIT, this module sets PID to NAK.</p> <p>(2) When the function controller function is selected</p> <p>This module modifies the setting of these bits as follows.</p> <ul style="list-style-type: none"> <li>• This module modifies PID to NAK on receiving the setup packet. Here, this module sets VALID to 1. Software cannot modify the setting of PID until software sets VALID to 0.</li> <li>• This module sets PID to STALL (11) on receiving the data of the size exceeding the maximum packet size when software has set PID to BUF.</li> <li>• This module sets PID to STALL (1x) on detecting the control transfer sequence error.</li> <li>• This module sets PID to NAK on detecting the USB bus reset.</li> </ul> <p>This module does not reference to the setting of the PID bits while the SET_ADDRESS request is processed (auto processing).</p>

Notes: 1. This bit is always read as 0. Only 1 can be written to.

2. Only 1 can be written to.

### 21.3.31 Pipe Window Select Register (PIPESEL)

PIPE1 to PIPE 9 should be set using PIPESEL, PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN. After selecting the pipe using PIPESEL, functions of the pipe should be set using PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in PIPESEL.

For a power-on reset and a USB bus reset, the corresponding bits for not only the selected pipe but all of the pipes are initialized.

This register is initialized by a power-on reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	PIPESEL[3:0]	0000	R/W	<p>Pipe Window Select</p> <p>Selects the pipe number corresponding to the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers which data is written to or read from.</p> <p>0000: No pipe selected</p> <p>0001: PIPE1</p> <p>0010: PIPE2</p> <p>0011: PIPE3</p> <p>0100: PIPE4</p> <p>0101: PIPE5</p> <p>0110: PIPE6</p> <p>0111: PIPE7</p> <p>1000: PIPE8</p> <p>1001: PIPE9</p> <p>Other than above: Setting prohibited</p> <p>Selecting a pipe number through these bits allows writing to and reading from the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers that correspond to the selected pipe number.</p> <p>When PIPESEL = 0000, 0 is read from all of the bits in PIPECFG, PIPEBUF, PIPEMAXP, PIPEERI and PIPEnCTR. Writing to these bits is invalid.</p>

### 21.3.32 Pipe Configuration Register (PIPECFG)

PIPECFG is a register that specifies the transfer type, buffer memory access direction, and endpoint numbers for PIPE1 to PIPE9. It also selects continuous or non-continuous transfer mode, single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

This register is initialized by a power-on reset. Only the TYPE[1:0] bits are initialized by a USB bus reset.



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHT NAK	—	—	DIR	EPNUM[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	TYPE[1:0]	00	R/W	<p>Transfer Type</p> <p>Selects the transfer type for the pipe selected by the PIPESEL bits (selected pipe)</p> <ul style="list-style-type: none"> <li>PIPE1 and PIPE2</li> <li>00: Pipe not used</li> <li>01: Bulk transfer</li> <li>10: Setting prohibited</li> <li>11: Isochronous transfer</li> <li>PIPE3 to PIPE5</li> <li>00: Pipe not used</li> <li>01: Bulk transfer</li> <li>10: Setting prohibited</li> <li>11: Setting prohibited</li> <li>PIPE6 and PIPE7</li> <li>00: Pipe not used</li> <li>01: Setting prohibited</li> <li>10: Interrupt transfer</li> <li>11: Setting prohibited</li> </ul> <p>Before setting PID to BUF for the selected pipe (before starting USB communication using the selected pipe), be sure to set these bits to the value other than 00.</p> <p>Modify these bits while the PID bits for the selected pipe are set to NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

Bit	Bit Name	Initial Value	R/W	Description
13 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	BFRE	0	R/W	<p><b>BRDY Interrupt Operation Specification</b></p> <p>Specifies the BRDY interrupt generation timing from this module to the CPU with respect to the selected pipe.</p> <p>0: BRDY interrupt upon transmitting or receiving of data</p> <p>1: BRDY interrupt upon completion of reading of data</p> <p>When software has set this bit to 1 and the selected pipe is in the receiving direction, this module detects the transfer completion and generates the BRDY interrupt on having read the pertinent packet.</p> <p>When the BRDY interrupt is generated with the above conditions, software needs to write 1 to BCLR. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to BCLR.</p> <p>When software has set this bit to 1 and the selected pipe is in the transmitting direction, this module does not generate the BRDY interrupt.</p> <p>For details, refer to (1) BRDY Interrupt under section 21.4.2, Interrupt Functions.</p> <p>Modify these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits.</p> <p>To modify these bits after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously through software to clear the FIFO buffer assigned to the selected pipe while the CSSTS, PID, and CURPIPE bits are in the above-described state.</p> <p>Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

Bit	Bit Name	Initial Value	R/W	Description
9	DBLB	0	R/W	<p>Double Buffer Mode</p> <p>Selects either single or double buffer mode for the FIFO buffer used by the selected pipe.</p> <p>0: Single buffer 1: Double buffer</p> <p>This bit is valid when PIPE1 to PIPE5 are selected.</p> <p>When software has set this bit to 1, this module assigns two planes of the FIFO buffer size specified by the BUFSIZE bits in PIPEBUF to the selected pipe.</p> <p>Specifically, the following expression determines the FIFO buffer size assigned to the selected pipe by this module.</p> $(\text{BUFSIZE} + 1) * 64 * (\text{DBLB} + 1) \text{ [bytes]}$ <p>When software has set this bit to 1 and the selected pipe is in the transmitting direction, this module does not generate the BRDY interrupt.</p> <p>For details, refer to (1) BRDY Interrupt under section 21.4.2, Interrupt Functions.</p> <p>Modify these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits.</p> <p>To modify these bits after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously through software to clear the FIFO buffer assigned to the selected pipe while the CSSTS, PID, and CURPIPE bits are in the above-described state.</p> <p>Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

Bit	Bit Name	Initial Value	R/W	Description
8	CNTMD	0	R/W	<p>Continuous Transfer Mode</p> <p>Specifies whether to use the selected pipe in continuous transfer mode.</p> <p>0: Non-continuous transfer mode</p> <p>1: Continuous transfer mode</p> <p>This bit is valid when PIPE1 to PIPE5 are selected by the PIPESEL bits and bulk transfer is selected (TYPE = 01).</p> <p>Modify these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits.</p> <p>To modify these bits after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously through software to clear the FIFO buffer assigned to the selected pipe while the CSSTS, PID, and CURPIPE bits are in the above-described state.</p> <p>Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

Bit	Bit Name	Initial Value	R/W	Description
7	SHTNAK	0	R/W	<p>Pipe Disabled at End of Transfer</p> <p>Specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction.</p> <p>0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer</p> <p>This bit is valid when the selected pipe is PIPE1 to PIPE5 in the receiving direction.</p> <p>When software has set this bit to 1 for the selected pipe in the receiving direction, this module modifies the PID bits corresponding to the selected pipe to NAK on determining the end of the transfer. This module determines that the transfer has ended on any of the following conditions.</p> <ul style="list-style-type: none"> <li>• A short packet (including a zero-length packet) is successfully received.</li> <li>• The transaction counter is used and the number of packets specified by the counter are successfully received.</li> </ul> <p>Modify these bits while CSSTS is 0 and PID is NAK.</p> <p>Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p> <p>This bit should be cleared to 0 for the pipe in the transmitting direction.</p>
6, 5	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
4	DIR	0	R/W	<p>Transfer Direction</p> <p>Specifies the transfer direction for the selected pipe.</p> <p>0: Receiving direction</p> <p>1: Sending direction</p> <p>When software has set this bit to 0, this module uses the selected pipe in the receiving direction, and when software has set this bit to 1, this module uses the selected pipe in the transmitting direction.</p> <p>Modify these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits.</p> <p>To modify these bits after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously through software to clear the FIFO buffer assigned to the selected pipe while the CSSTS, PID, and CURPIPE bits are in the above-described state.</p> <p>Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>
3 to 0	EPNUM[3:0]	0000	R/W	<p>Endpoint Number</p> <p>These bits specify the endpoint number for the selected pipe.</p> <p>Setting 0000 means unused pipe.</p> <p>Modify these bits while CSSTS is 0 and PID is NAK.</p> <p>Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p> <p>Do not make the settings such that the combination of the set values in the DIR and EPNUM bits should be the same for two or more pipes (EPNUM = 0000 can be set for all the pipes).</p>

### 21.3.33 Pipe Buffer Setting Register (PIPEBUF)

PIPEBUF is a register that specifies the buffer size and buffer number for PIPE1 to PIPE9.

This register is initialized by a power-on reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15	—	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
14 to 10	BUFSIZE[4:0]	H'00	R/W	<p><b>Buffer Size</b></p> <p>Specifies the size of the buffer for the pipe selected by the PIPESEL bits (selected pipe) in terms of blocks, where one block comprises 64 bytes.</p> <p>00000 (H'00): 64 bytes  00001 (H'01): 128 bytes  : :  11111 (H'1F): 2 Kbytes</p> <p>When software has set the DBLB bit to 1, this module assigns two planes of the FIFO buffer size specified by the BUFSIZE bits to the selected pipe.</p> <p>Specifically, the following expression determines the FIFO buffer size assigned to the selected pipe by this module.</p> $(BUFSIZE + 1) * 64 * (DBLB + 1) \text{ [bytes]}$ <p>The valid value for these bits depends on the selected pipe.</p> <ul style="list-style-type: none"> <li>PIPE1 to PIPE5: Any value from H'00 to H'1F is valid.</li> <li>PIPE6 to PIPE9: H'00 should be set.</li> </ul> <p>When used with CNTMD = 1, set an integer multiple of the maximum packet size to the BUFSIZE bits.</p> <p>Modify these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits.</p> <p>Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>
9, 8	—	All 0	R	<p><b>Reserved</b></p> <p>These bits are always read as 0. The write value should always be 0.</p>



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BUFNMB[7:0]	H'00	R/W	<p>Buffer Number</p> <p>These bits specify the FIFO buffer number for the selected pipe (from H'04 to H'4F).</p> <p>When the selected pipe is one of PIPE1 to PIPE5, any value can be set to these bits according to the user system.</p> <p>BUFNMB = H'00 to H'03 are used exclusively for DCP.</p> <p>BUFNMB = H'04 is used exclusively for PIPE6.</p> <p>When PIPE6 is not used, H'04 can be used for other pipes.</p> <p>When PIPE6 is selected, writing to these bits is invalid and H'04 is automatically assigned by this module.</p> <p>BUFNMB = H'05 is used exclusively for PIPE7.</p> <p>When PIPE7 is not used, H'05 can be used for other pipes.</p> <p>When PIPE7 is selected, writing to these bits is invalid and H'05 is automatically assigned by this module.</p> <p>BUFNMB = H'06 is used exclusively for PIPE8.</p> <p>When PIPE8 is not used, H'06 can be used for other pipes.</p> <p>When PIPE8 is selected, writing to these bits is invalid and H'06 is automatically assigned by this module.</p> <p>BUFNMB = H'07 is used exclusively for PIPE9.</p> <p>When PIPE9 is not used, H'07 can be used for other pipes.</p> <p>When PIPE9 is selected, writing to these bits is invalid and H'07 is automatically assigned by this module.</p> <p>Modify these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits.</p> <p>Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

### 21.3.34 Pipe Maximum Packet Size Register (PIPEMAXP)

PIPEMAXP is a register that specifies the maximum packet size for PIPE1 to PIPE9.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEVSEL[3:0]				—	MXPS[10:0]										
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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	DEVSEL[3:0]	00	R/W	<p>Device Select</p> <p>When the host controller function is selected, these bits specify the USB address of the communication target peripheral device.</p> <p>0000: Address 0000 0001: Address 0001 0010: Address 0010 : 1010: Address 1010</p> <p>Other than above: Setting prohibited</p> <p>These bits should be set after setting the address to the DEVADDn (n = 0 to A) register corresponding to the value to be set in these bits.</p> <p>For example, before setting DEVSEL to 0010, the address should be set to the DEVADD2 register.</p> <p>Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p> <p>When the function controller function is selected, these bits should be set to B'0000.</p>
11	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
10 to 0	MXPS[10:0]	*	R/W	<p>Maximum Packet Size</p> <p>Specifies the maximum data payload (maximum packet size) for the selected pipe. The valid value for these bits depends on the pipe as follows.</p> <p>PIPE1, PIPE2: 1 byte (H'001) to 1,024 bytes (H'400)</p> <p>PIPE3 to PIPE5: 8 bytes (H'008), 16 bytes (H'010), 32 bytes (H'020), 64 bytes (H'040), and 512 bytes (H'200) (Bits 2 to 0 are not provided.)</p> <p>PIPE6 to PIPE9: 1 byte (H'001) to 64 bytes (H'040)</p> <p>These bits should be set to the appropriate value for each transfer type based on the USB Specification.</p> <p>For split transactions using the isochronous pipe, these bits should be set to 188 bytes or less.</p> <p>Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p> <p>While MXPS is 0, do not write to the FIFO buffer or set PID to BUF.</p>

Note: \* The initial value of MXPS is H'000 when no pipe is selected with the PIPESEL bits in PIPESEL and H'040 when a pipe is selected with the PIPESEL bit in PIPESEL.

### 21.3.35 Pipe Timing Control Register (PIPEPERI)

PIPEPERI is a register that selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for PIPE1 to PIPE9.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[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/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	IFIS	0	R/W	<p><b>Isochronous IN Buffer Flush</b></p> <p>Specifies whether to flush the buffer when the pipe selected by the PIPESEL bits (selected pipe) is used for isochronous IN transfers.</p> <p>0: The buffer is not flushed. 1: The buffer is flushed.</p> <p>When the function controller function is selected and the selected pipe is for isochronous IN transfers, this module automatically clears the FIFO buffer when this module fails to receive the IN token from the USB host within the interval set by the IITV bits in terms of (<math>\mu</math>) frames.</p> <p>In double buffer mode (DBLB = 1), this module only clears the data in the plane used earlier.</p> <p>This module clears the FIFO buffer on receiving the SOF packet immediately after the (<math>\mu</math>) frame in which this module has expected to receive the IN token. Even if the SOF packet is corrupted, this module also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation.</p> <p>When the host controller function is selected, set this bit to 0.</p> <p>When the selected pipe is not for the isochronous transfer, set this bit to 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
11 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	IITV[2:0]	000	R/W	Interval Error Detection Interval Specifies the interval error detection timing for the selected pipe in terms of frames, which is expressed as n-th power of 2 (n is the value to be set). As described later, the detailed functions are different in host controller mode and in function controller mode. Modify these bits while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary. Before modifying these bits after USB communication has been completed with these bits set to a certain value, set PID to NAK and then set ACLRM to 1 to initialize the interval timer. The IITV bits are invalid for PIPE3 to PIPE5; set these bits to 000 for these pipes.

### 21.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9)

PIPEnCTR is a register that is used to confirm the buffer memory status for the corresponding pipe, change and confirm the data PID sequence bit, determine whether auto response mode is set, determine whether auto buffer clear mode is set, and set a response PID for PIPE1 to PIPE9. This register can be set regardless of the pipe selection in PIPESEL.

This register is initialized by a power-on reset. PID[1:0] are initialized by a USB bus reset.

#### (1) PIPEnCTR (n = 1 to 5)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	INBUFM	CSCLR	CSSTS	—	AT REPM	ACLARM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[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/W*2	R	R	R/W	R/W	R/W*1	R/W*1	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	<p>Buffer Status</p> <p>Indicates the FIFO buffer status for the pertinent pipe.</p> <p>0: Buffer access from CPU is disabled.</p> <p>1: Buffer access from CPU is enabled.</p> <p>The meaning of this bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in table 21.11.</p>

Bit	Bit Name	Initial Value	R/W	Description
14	INBUFM	0	R	<p><b>IN Buffer Monitor</b></p> <p>Indicates the pertinent FIFO buffer status when the pertinent pipe is in the transmitting direction.</p> <p>0: There is no data to be transmitted in the buffer memory.</p> <p>1: There is data to be transmitted in the buffer memory.</p> <p>When the pertinent pipe is in the transmitting direction (DIR = 1), this module sets this bit to 1 when software (or DMAC) completes writing data to at least one FIFO buffer plane.</p> <p>This module sets this bit to 0 when this module completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (DBLB = 1), this module sets this bit to 0 when this module completes transmitting the data from the two FIFO buffer planes before software (or DMAC) completes writing data to one FIFO buffer plane.</p> <p>This bits indicates the same value as the BSTS bit when the pertinent pipe is in the receiving direction (DIR = 0).</p>

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R/W* <sup>2</sup>	<p><b>C-SPLIT Status Clear Bit</b></p> <p>When the host controller function is selected, setting this bit to 1 through software allows this module to clear the CSSTS bit to 0.</p> <p>0: Writing invalid</p> <p>1: Clears the CSSTS bit to 0.</p> <p>For the transfer using the split transaction, to restart the next transfer with the S-SPLIT forcibly, set this bit to 1 through software. However, for the normal split transaction, this module automatically clears the CSSTS bit to 0 upon completion of the C-SPLIT; therefore, clearing the CSSTS bit through software is not necessary.</p> <p>Controlling the CSSTS bit through this bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.</p> <p>Setting this bit to 1 while CSSTS is 0 has no effect.</p> <p>When the function controller function is selected, be sure to write 0 to this bit.</p>
12	CSSTS	0	R	<p><b>CSSTS Status Bit</b></p> <p>Indicates the C-SPLIT status of the split transaction when the host controller function is selected.</p> <p>0: START-SPLIT (S-SPLIT) transaction being processed or the transfer not using the split transaction in progress</p> <p>1: C-SPLIT transaction being processed</p> <p>This module sets this bit to 1 upon start of the C-SPLIT and clears this bit to 0 upon detection of C-SPLIT completion.</p> <p>Indicates the valid value only when the host controller function is selected.</p>
11	—	0	R	<p><b>Reserved</b></p> <p>This bit is always read as 0. The write value should always be 0.</p>



Bit	Bit Name	Initial Value	R/W	Description
10	ATREPM	0	R/W	<p><b>Auto Response Mode</b></p> <p>Enables or disables auto response mode for the pertinent pipe.</p> <p>0: Auto response disabled</p> <p>1: Auto response enabled</p> <p>When the function controller function is selected and the pertinent pipe is for bulk transfer, this bit can be set to 1.</p> <p>When this bit is set to 1, this module responds to the token from the USB host as described below.</p> <p>(1) When the pertinent pipe is for bulk IN transfer (TYPE = 01 and DIR = 1)</p> <p>When ATREPM = 1 and PID = BUF, this module transmits a zero-length packet in response to the IN token.</p> <p>This module updates (allows toggling of) the sequence toggle bit (DATA-PID) each time this module receives the ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.).</p> <p>In this case, this module does not generate the BRDY or BEMP interrupt.</p> <p>(2) When the pertinent pipe is for bulk OUT transfer (TYPE = 01 and DIR = 0)</p> <p>When ATREPM = 1 and PID = BUF, this module returns NAK in response to the OUT (or PING) token and generates the NRDY interrupt.</p> <p>Modify this bit while CSSTS is 0 and PID is NAK. Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

Bit	Bit Name	Initial Value	R/W	Description
10	ATREPM	0	R/W	<p>For USB communication in auto response mode, set this bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.</p> <p>When the pertinent pipe is for isochronous transfer, be sure to set this bit to 0.</p> <p>When the host controller function is selected, set this bit to 0.</p>
9	ACLRM	0	R/W	<p>Auto Buffer Clear Mode</p> <p>Enables or disables automatic buffer clear mode for the pertinent pipe.</p> <p>0: Disabled</p> <p>1: Enabled (all buffers are initialized)</p> <p>To delete the information in the FIFO buffer assigned to the pertinent pipe completely, write 1 and then 0 to this bit continuously.</p> <p>Table 21.12 shows the information cleared by writing 1 and 0 to this bit continuously and the cases in which clearing the information is necessary.</p> <p>Modify this bit while CSSTS is 0 and PID is NAK.</p> <p>Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

Bit	Bit Name	Initial Value	R/W	Description
8	SQCLR	0	R/W* <sup>1</sup>	<p><b>Toggle Bit Clear</b></p> <p>This bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the pertinent pipe.</p> <p>0: Invalid</p> <p>1: Specifies DATA0.</p> <p>Setting this bit to 1 through software allows this module to set DATA0 as the expected value of the sequence toggle bit of the pertinent pipe. This module always sets this bit to 0.</p> <p>When the host controller function is selected, setting this bit to 1 for the pipe for bulk OUT transfer, this module starts the next transfer of the pertinent pipe with the PING token.</p> <p>Set the SQCLR bit to 1 while CSCTS is 0 and PID is NAK.</p> <p>Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

Bit	Bit Name	Initial Value	R/W	Description
7	SQSET	0	R/W* <sup>1</sup>	<p><b>Toggle Bit Set</b></p> <p>This bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the pertinent pipe.</p> <p>0: Invalid</p> <p>1: Specifies DATA1.</p> <p>Setting this bit to 1 through software allows this module to set DATA1 as the expected value of the sequence toggle bit of the pertinent pipe. This module always sets this bit to 0.</p> <p>Set the SQSET bit to 1 while CSSTS is 0 and PID is NAK.</p> <p>Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>
6	SQMON	0	R	<p><b>Toggle Bit Confirmation</b></p> <p>Indicates the expected value of the sequence toggle bit for the next transaction of the pertinent pipe.</p> <p>0: DATA0</p> <p>1: DATA1</p> <p>When the pertinent pipe is not for the isochronous transfer, this module allows this bit to toggle upon normal completion of the transaction. However, this bit is not allowed to toggle when a DATA-PID disagreement occurs during the receiving transfer.</p>

Bit	Bit Name	Initial Value	R/W	Description
5	PBUSY	0	R	<p>Pipe Busy</p> <p>This bit indicates whether the relevant pipe is used or not for the transaction.</p> <p>0: The relevant pipe is not used for the transaction.</p> <p>1: The relevant pipe is used for the transaction.</p> <p>This module modifies this bit from 0 to 1 upon start of the USB transaction for the pertinent pipe, and modifies the bit from 1 to 0 upon completion of one transaction.</p> <p>Reading this bit after software has set PID to NAK allows checking that modification of the pipe settings is possible.</p> <p>For details, refer to (1) Pipe Control Register Switching Procedures under section 21.4.3, Pipe Control.</p>
4 to 2	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	<p>Response PID</p> <p>Specifies the response type for the next transaction of the pertinent pipe.</p> <p>00: NAK response</p> <p>01: BUF response (depending on the buffer state)</p> <p>10: STALL response</p> <p>11: STALL response</p> <p>The default setting of these bits is NAK. Modify the setting to BUF to use the pertinent pipe for USB transfer. Tables 21.13 and 21.14 show the basic operation (operation when there are no errors in the transmitted and received packets) of this module depending on the PID bit setting.</p> <p>After modifying the setting of these bits through software from BUF to NAK during USB communication using the pertinent pipe, check that PBUSY is 1 to see if USB communication using the pertinent pipe has actually entered the NAK state. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p> <p>This module modifies the setting of these bits as follows.</p> <ul style="list-style-type: none"> <li>• This module sets PID to NAK on recognizing the completion of the transfer when the pertinent pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1.</li> <li>• This module sets PID to STALL (11) on receiving the data packet with the payload exceeding the maximum packet size of the pertinent pipe.</li> <li>• This module sets PID to NAK on detecting a USB bus reset when the function controller function is selected.</li> <li>• This module sets PID to NAK on detecting a receive error such as a CRC error three consecutive times when the host controller function is selected.</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	<ul style="list-style-type: none"> <li>This module sets PID to STALL (11) on receiving the STALL handshake when the host controller function is selected.</li> </ul> <p>To specify each response type, set these bits as follows.</p> <ul style="list-style-type: none"> <li>To make a transition from NAK (00) to STALL, set 10.</li> <li>To make a transition from BUF (01) to STALL, set 11.</li> <li>To make a transition from STALL (11) to NAK, set 10 and then 00.</li> <li>To make a transition from STALL to BUF, set 00 (NAK) and then 01 (BUF).</li> </ul>

Notes: 1. Only 0 can be read and 1 can be written to.

2. Only 1 can be written to.

**Table 21.11 Meaning of BSTS Bit**

DIR Bit	BFRE Bit	DCLRM Bit	Meaning of BSTS Bit
0	0	0	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	1: The received data can be read from the FIFO buffer. 0: Software has set BCLR to 1 after the received data has been completely read from the FIFO buffer.
		1	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
	1	0	1: The transmit data can be written to the FIFO buffer. 0: The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
1	1	0	Setting prohibited
		1	Setting prohibited

**Table 21.12 Information Cleared by this Module by Setting ACLRM = 1**

<b>No.</b>	<b>Information Cleared by ACLRM Bit Manipulation</b>	<b>Cases in which Clearing the Information is Necessary</b>
1	All the information in the FIFO buffer assigned to the pertinent pipe (all the information in two FIFO buffer planes in double buffer mode)	
2	The interval count value when the pertinent pipe is for isochronous transfer	When the interval count value is to be reset
3	Values of the internal flags related to the BFRE bit	When the BFRE setting is modified
4	FIFO buffer toggle control	When the DBLB setting is modified
5	Values of the internal flags related to the transaction count	When the transaction count function is forcibly terminated

**Table 21.13 Operation of This Module depending on PID Setting (when Host Controller Function is Selected)**

<b>PID</b>	<b>Transfer Type</b>	<b>Transfer Direction (DIR Bit)</b>	<b>Operation of This Module</b>
00 (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01 (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while UACT is 1 and the FIFO buffer corresponding to the pertinent pipe is ready for transmission and reception.  Does not issue tokens while UACT is 0 or the FIFO buffer corresponding to the pertinent pipe is not ready for transmission or reception.
	Isochronous	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the pertinent pipe.
10 (STALL) or 11 (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.



**Table 21.14 Operation of This Module depending on PID Setting (when Function Controller Function is Selected)**

<b>PID</b>	<b>Transfer Type</b>	<b>Transfer Direction (DIR Bit)</b>	<b>Operation of This Module</b>
00 (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host.  For the operation when ATREPM is 1, refer to the description of the ATREPM bit.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01 (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Returns NAK if not ready.  Returns ACK in response to the PING token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Returns NYET if not ready.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Returns NAK if not ready.
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the pertinent pipe is ready for reception. Discards data if not ready.

PID	Transfer Type	Transfer Direction (DIR Bit)	Operation of This Module
01 (BUF)	Isochronous	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.
10 (STALL) or 11 (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting	Returns nothing in response to the token from the USB host.

**(2) PIPEnCTR (n = 6 to 9)**

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[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/W*	R	R	R	R/W	R/W*	R/W*	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	<p>Buffer Status</p> <p>Indicates the FIFO buffer status for the pertinent pipe.</p> <p>0: Buffer access is disabled.</p> <p>1: Buffer access is enabled.</p> <p>The meaning of this bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in table 21.11.</p>
14	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
13	CSCLR	0	R/W* <sup>1</sup>	<p><b>C-SPLIT Status Clear Bit</b></p> <p>Setting this bit to 1 allows this module to clear the CSSTS bit of the pertinent pipe to 0.</p> <p>0: Writing invalid</p> <p>1: Clears the CSSTS bit to 0.</p> <p>For the transfer using the split transaction, to restart the next transfer with the S-SPLIT forcibly, set this bit to 1 through software. However, for the normal split transaction, this module automatically clears the CSSTS bit to 0 upon completion of the C-SPLIT; therefore, clearing the CSSTS bit through software is not necessary.</p> <p>Controlling the CSSTS bit through this bit must be done while UACT is 0 thus communication is halted or while no transfer is being performed with bus disconnection detected.</p> <p>Setting this bit to 1 while CSSTS is 0 has no effect.</p> <p>When the function controller function is selected, be sure to write 0 to this bit.</p>
12	CSSTS	0	R	<p><b>CSSTS Status Bit</b></p> <p>Indicates the C-SPLIT status of the split transaction when the host controller function is selected.</p> <p>0: START-SPLIT (S-SPLIT) transaction being processed or the transfer not using the split transaction in progress</p> <p>1: C-SPLIT transaction being processed</p> <p>This module sets this bit to 1 upon start of the C-SPLIT and clears this bit to 0 upon detection of C-SPLIT completion.</p> <p>Indicates the valid value only when the host controller function is selected.</p>
11, 10	—	All 0	R	<p><b>Reserved</b></p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
9	ACLRM	0	R/W	<p>Auto Buffer Clear Mode*3*4</p> <p>Enables or disables automatic buffer clear mode for the pertinent pipe.</p> <p>0: Disabled</p> <p>1: Enabled (all buffers are initialized)</p> <p>To delete the information in the FIFO buffer assigned to the pertinent pipe completely, write 1 and then 0 to this bit continuously.</p> <p>Table 21.15 shows the information cleared by writing 1 and 0 to this bit continuously and the cases in which clearing the information is necessary.</p> <p>Modify this bit while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits.</p> <p>Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

Bit	Bit Name	Initial Value	R/W	Description
8	SQCLR	0	R/W* <sup>1</sup>	<p>Toggle Bit Clear*<sup>3</sup>*<sup>4</sup></p> <p>This bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the pertinent pipe.</p> <p>0: Invalid</p> <p>1: Specifies DATA0.</p> <p>Setting this bit to 1 through software allows this module to set DATA0 as the expected value of the sequence toggle bit of the pertinent pipe. This module always sets this bit to 0.</p> <p>When the host controller function is selected, setting this bit to 1 for the pipe for bulk OUT transfer, this module starts the next transfer of the pertinent pipe with the PING token.</p> <p>Set the SQCLR bit to 1 while CSCTS is 0 and PID is NAK.</p> <p>Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>

Bit	Bit Name	Initial Value	R/W	Description
7	SQSET	0	R/W* <sup>1</sup>	<p>Toggle Bit Set*<sup>3</sup>*<sup>4</sup></p> <p>This bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the pertinent pipe.</p> <p>0: Invalid</p> <p>1: Specifies DATA1.</p> <p>Setting this bit to 1 through software allows this module to set DATA1 as the expected value of the sequence toggle bit of the pertinent pipe. This module always sets this bit to 0.</p> <p>Set the SQSET bit to 1 while CSSTS is 0 and PID is NAK.</p> <p>Before modifying this bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p>
6	SQMON	0	R	<p>Toggle Bit Confirmation</p> <p>Indicates the expected value of the sequence toggle bit for the next transaction of the pertinent pipe.</p> <p>0: DATA0</p> <p>1: DATA1</p> <p>When the pertinent pipe is not for the isochronous transfer, this module allows this bit to toggle upon normal completion of the transaction. However, this bit is not allowed to toggle when a DATA-PID disagreement occurs during the receiving transfer.</p>

Bit	Bit Name	Initial Value	R/W	Description
5	PBUSY	0	R	<p>Pipe Busy</p> <p>This bit indicates whether the relevant pipe is used or not for the transaction.</p> <p>0: The relevant pipe is not used for the transaction.</p> <p>1: The relevant pipe is used for the transaction.</p> <p>This module modifies this bit from 0 to 1 upon start of the USB transaction for the pertinent pipe, and modifies the bit from 1 to 0 upon completion of one transaction.</p> <p>Reading this bit after software has set PID to NAK allows checking that modification of the pipe settings is possible.</p> <p>For details, refer to (1) Pipe Control Register Switching Procedures under section 21.4.3, Pipe Control.</p>
4 to 2	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	<p>Response PID</p> <p>Specifies the response type for the next transaction of the pertinent pipe.</p> <p>00: NAK response</p> <p>01: BUF response (depending on the buffer state)</p> <p>10: STALL response</p> <p>11: STALL response</p> <p>The default setting of these bits is NAK. Modify the setting to BUF to use the pertinent pipe for USB transfer. Tables 21.13 and 21.14 show the basic operation (operation when there are no errors in the transmitted and received packets) of this module depending on the PID bit setting.</p> <p>After modifying the setting of these bits through software from BUF to NAK during USB communication using the pertinent pipe, check that PBUSY is 1 to see if USB communication using the pertinent pipe has actually entered the NAK state. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p> <p>This module modifies the setting of these bits as follows.</p> <ul style="list-style-type: none"> <li>• This module sets PID to NAK on recognizing the completion of the transfer when the pertinent pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1.</li> <li>• This module sets PID to STALL (11) on receiving the data packet with the payload exceeding the maximum packet size of the pertinent pipe.</li> <li>• This module sets PID to NAK on detecting a USB bus reset when the function controller function is selected.</li> </ul>



Bit	Bit Name	Initial Value	R/W	Description
1, 0	PID[1:0]	00	R/W	<ul style="list-style-type: none"> <li>This module sets PID to NAK on detecting a receive error such as a CRC error three consecutive times when the host controller function is selected.</li> <li>This module sets PID to STALL (11) on receiving the STALL handshake when the host controller function is selected.</li> </ul> <p>To specify each response type, set these bits as follows.</p> <ul style="list-style-type: none"> <li>To make a transition from NAK (00) to STALL, set 10.</li> <li>To make a transition from BUF (01) to STALL, set 11.</li> <li>To make a transition from STALL (11) to NAK, set 10 and then 00.</li> <li>To make a transition from STALL to BUF, set 00 (NAK) and then 01 (BUF).</li> </ul>

- Notes:
1. Only 0 can be read and 1 can be written to.
  2. Only 1 can be written to.
  3. The ACLRM, SQCLR, or SQSET bits should be set while CSSTS is 0 and PID is NAK and before the pipe is selected by the CURPIPE bits.
  4. Before modifying ACLRM, SQCLR, or SQSET bits after modifying the PID bits from BUF to NAK, it should be checked that CSSTS and PBUSY for the selected pipe are 0. However, if the PID bits have been modified to NAK through hardware control, checking PBUSY is not necessary.

**Table 21.15 Information Cleared by this Module by Setting ACLRM = 1**

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing the Information is Necessary
1	All the information in the FIFO buffer assigned to the pertinent pipe	
2	When the host controller function is selected, the interval count value when the pertinent pipe is for isochronous transfer	When the interval count value is to be reset
3	Values of the internal flags related to the BFRE bit	When the BFRE setting is modified
4	Values of the internal flags related to the transaction count	When the transaction count function is forcibly terminated

**21.3.37 PIPE<sub>n</sub> Transaction Counter Enable Registers (PIPE<sub>n</sub>TRE) (n = 1 to 5)**

PIPE<sub>n</sub>TRE is a register that enables or disables the transaction counter corresponding to PIPE1 to PIPE5, and clears the transaction counter.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Initial value:	0	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
15 to 10	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	TRENB	0	R/W	<p>Transaction Counter Enable</p> <p>Enables or disables the transaction counter.</p> <p>0: The transaction counter is disabled.</p> <p>1: The transaction counter is enabled.</p> <p>For the pipe in the receiving direction, setting this bit to 1 after setting the total number of the packets to be received in the TRNCNT bits through software allows this module to control hardware as described below on having received the number of packets equal to the set value in the TRNCNT bits.</p> <ul style="list-style-type: none"> <li>• In continuous transmission/reception mode (CNTMD = 1), this module switches the FIFO buffer to the CPU side even if the FIFO buffer is not full on completion of reception.</li> <li>• While SHTNAK is 1, this module modifies the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the set value in the TRNCNT bits.</li> <li>• While BFRE is 1, this module asserts the BRDY interrupt on having received the number of packets equal to the set value in the TRNCNT bits and then reading out the last received data.</li> </ul> <p>For the pipe in the transmitting direction, set this bit to 0.</p> <p>When the transaction counter is not used, set this bit to 0.</p> <p>When the transaction counter is used, set the TRNCNT bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.</p>
8	TRCLR	0	R/W	<p>Transaction Counter Clear</p> <p>Clears the current value of the transaction counter corresponding to the pertinent pipe and then sets this bit to 0.</p> <p>0: Invalid</p> <p>1: The current counter value is cleared.</p>

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

**Note:** Modify each bit in this register while CSSTS is 0 and PID is NAK. Before modifying each bit after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.

### 21.3.38 PIPE<sub>n</sub> Transaction Counter Registers (PIPE<sub>n</sub>TRN) (n = 1 to 5)

PIPE<sub>n</sub>TRN is a transaction counter corresponding to PIPE1 to PIPE5.

This register is initialized by a power-on reset, but retains the set value by a USB bus reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TRNCNT[15:0]	All 0	R/W	Transaction Counter When written to: Specifies the number of transactions to be transferred through DMA. When read from: Indicates the specified number of transactions if TREN <sub>B</sub> is 0. Indicates the number of currently counted transaction if TREN <sub>B</sub> is 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TRNCNT[15:0]	All 0	R/W	<p>This module increments the value of these bits by one when all of the following conditions are satisfied on receiving the packet.</p> <ul style="list-style-type: none"> <li>TRENB is 1.</li> <li>(TRNCNT set value <math>\neq</math> current counter value + 1) on receiving the packet.</li> <li>The payload of the received packet agrees with the set value in the MXPS bits.</li> </ul> <p>This module clears the value of these bits to 0 when any of the following conditions are satisfied.</p> <ul style="list-style-type: none"> <li>All the following conditions are satisfied. TRENB is 1. (TRNCNT set value = current counter value + 1) on receiving the packet. The payload of the received packet agrees with the set value in the MXPS bits.</li> <li>All the following conditions are satisfied. TRENB is 1. This module has received a short packet.</li> <li>All the following conditions are satisfied. TRENB is 1. Software has set the TRCLR bit to 1.</li> </ul> <p>For the pipe in the transmitting direction, set these bits to 0.</p> <p>When the transaction counter is not used, set these bits to 0.</p> <p>Modify these bits while CSSTS is 0, PID is NAK, and TRENB is 0.</p> <p>Before modifying these bits after modifying the PID bits for the corresponding pipe from BUF to NAK, check that CSSTS and PBUSY are 0. However, if the PID bits have been modified to NAK by this module, checking PBUSY through software is not necessary.</p> <p>To modify the value of these bits, set TRNCNT to 1 before setting TRENB to 1.</p>

21.3.39 Device Address n Configuration Registers (DEVADDn) (n = 0 to A)

DEVADDn is a register that specifies the address and port number of the hub to which the communication target peripheral device is connected and that also specifies the transfer speed of the peripheral device for PIPE0 to PIPEA.

When the host controller function is selected, this register should be set before starting communication using each pipe.

The bits in this register should be modified while no valid pipes are using the settings of this register. Valid pipes refer to the ones satisfying both of condition 1 and 2 below.

- 1. This register is selected by the DEVSEL bits as the communication target.
- 2. The PID bits are set to BUF for the selected pipe or the selected pipe is the DCP with SUREQ being 1.

This register is initialized by a power-on reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved  This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 11	UPPHUB[3:0]	0000	R/W	<p>Address of Hub to which Communication Target is Connected</p> <p>Specifies the USB address of the hub to which the communication target peripheral device is connected.</p> <p>0000: The peripheral device is directly connected to the port of this LSI.</p> <p>0001 to 1010: USB address of the hub</p> <p>1011 to 1111: Setting prohibited</p> <p>When the host controller function is selected, this module refers to the setting of these bits to generate packets for split transactions.</p> <p>When the function controller function is selected, set these bits to 0000.</p>
10 to 8	HUBPORT[2:0]	000	R/W	<p>Port Number of Hub to which Communication Target is Connected</p> <p>Specifies the port number of the hub to which the communication target peripheral device is connected.</p> <p>000: The peripheral device is directly connected to the port of this LSI.</p> <p>001 to 111: Port number of the hub</p> <p>When the host controller function is selected, this module refers to the setting of these bits to generate packets for split transactions.</p> <p>When the function controller function is selected, set these bits to 000.</p>

Bit	Bit Name	Initial Value	R/W	Description
7, 6	USBSPD[1:0]	00	R/W	<p>Transfer Speed of the Communication Target Device</p> <p>Specifies the USB transfer speed of the communication target peripheral device.</p> <p>00: DEVADDn is not used.</p> <p>01: Setting prohibited</p> <p>10: Full speed</p> <p>11: High speed</p> <p>When the host controller function is selected, this module refers to the setting of these bits to generate packets.</p> <p>When the function controller function is selected, set these bits to 00.</p>
5 to 0	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>



## 21.4 Operation

### 21.4.1 System Control and Oscillation Control

This section describes the register operations that are necessary to the initial settings of this module, and the registers necessary for power consumption control.

#### (1) Resets

Table 21.16 lists the types of controller resets. For the initialized states of the registers following the reset operations, see section 21.3, Register Description.

**Table 21.16 Types of Reset**

Name	Operation
Power-on reset	Low level input from the PRESET pin
USB bus reset	Automatically detected by this module from the D+ and D– lines when the function controller function is selected

#### (2) Controller Function Selection

This module can select the host controller function or function controller function using the DCFM bit in SYSCFG. Changing the DCFM bit should be done in the initial settings immediately after a power-on reset or in the D+ pull-up disabled (DPRPU = 0) and D + /D – pull-down disabled (DRPD = 0) state.

#### (3) Enabling High-Speed Operation

This module can select a USB communication speed (communication bit rate) using software. When the host controller function is selected, either of the high-speed operation or full-speed operation can be selected. In order to enable the high-speed operation for this module, the HSE bit in SYSCFG should be set to 1. If high-speed mode has been enabled, this module executes the reset handshake protocol, and the USB communication speed is set automatically. The results of the reset handshake can be confirmed using the RHST bit in DVSTCTR.

If high-speed operation has been disabled, this module operates at full-speed. If the function controller function is also selected, this module operates at full-speed.

Changing the HSE bit should be done between the ATTCH interrupt detection and bus reset execution when the host controller function is selected, or with the D+ line pull-up disabled (DPRPU = 0) when the function controller function is selected.

#### **(4) USB Data Bus Resistor Control**

Figure 21.1 shows a diagram of the connections between this module and the USB connectors.

This module incorporates 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 SYSCFG.

This module controls the terminal resistor for the D+ and D- signals during high-speed operation and the output resistor for the signals during full-speed operation. This module automatically switches the resistor after connection with the host controller or peripheral device by means of reset handshake, suspended state and resume detection.

When the function controller function is selected and the DPRPU bit in SYSCFG 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 USB host of the device disconnection.

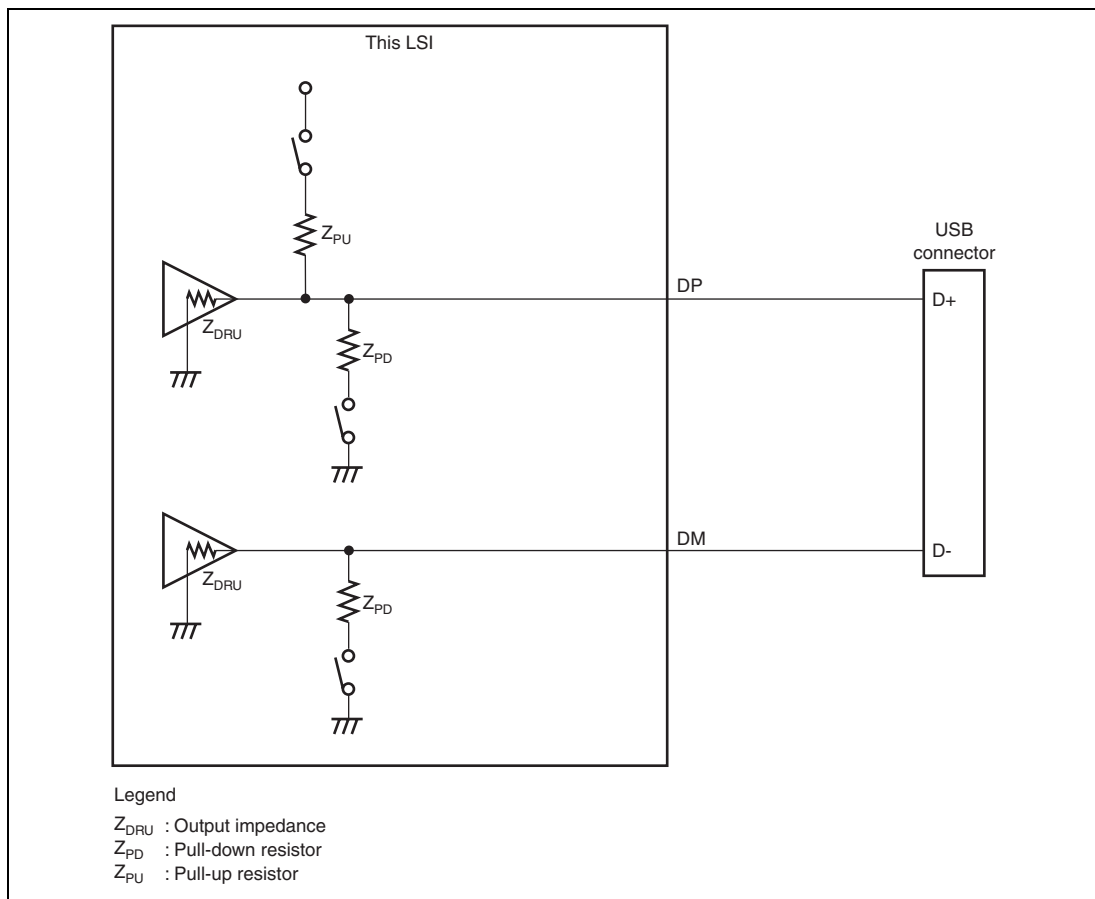


Figure 21.1 UBS Connector Connection

### 21.4.2 Interrupt Functions

Table 21.17 lists the interrupt generation conditions for this module.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, this module issues a USB interrupt request to the INTC.

**Table 21.17 Interrupt Generation Conditions**

<b>Bit</b>	<b>Interrupt Name</b>	<b>Cause of Interrupt</b>	<b>Function That Generates the Interrupt</b>	<b>Related Status</b>
VBINT	VBUS interrupt	When a change in the state of the VBUS input pin has been detected (low to high or high to low)	Host, function	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)	Function	—
SOFR	Frame number update interrupt	When the host controller function is selected: <ul style="list-style-type: none"> <li>When an SOF packet with a different frame number has been transmitted</li> </ul> When the function controller function is selected: <ul style="list-style-type: none"> <li>SOFRM = 0: When an SOF packet with a different frame number is received</li> <li>SOFRM = 1: When the SOF with the <math>\mu</math>frame number 0 cannot be received due to a corruption of a packet</li> </ul>	Host, function	—
DVST	Device state transition interrupt	When a device state transition is detected <ul style="list-style-type: none"> <li>A USB bus reset detected</li> <li>The suspend state detected</li> <li>SET_ADDRESS request received</li> <li>SET_CONFIGURATION request received</li> </ul>	Function	DVSQ

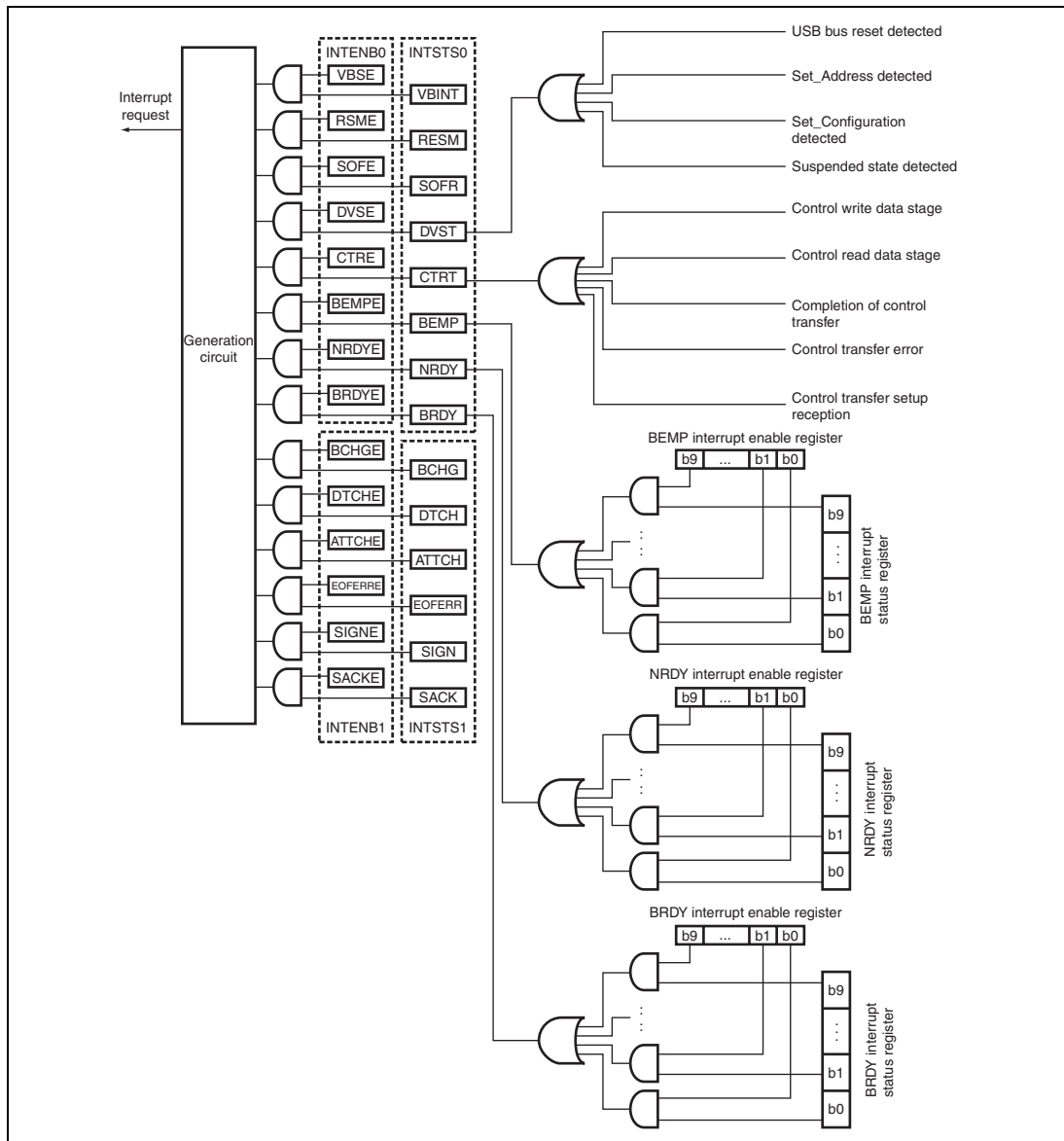
Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
CTRT	Control transfer stage transition interrupt	When a stage transition is detected in control transfer <ul style="list-style-type: none"> <li>• Setup stage completed</li> <li>• Control write transfer status stage transition</li> <li>• Control read transfer status stage transition</li> <li>• Control transfer completed</li> <li>• A control transfer sequence error occurred</li> </ul>	Function	CTSQ
BEMP	Buffer empty interrupt	<ul style="list-style-type: none"> <li>• When transmission of all of the data in the buffer memory has been completed</li> <li>• When an excessive maximum packet size error has been detected</li> </ul>	Host, Function	BEMPSTS. PIPEBEMP

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
NRDY	Buffer not ready interrupt	<p>When the host controller function is selected:</p> <ul style="list-style-type: none"> <li>When STALL is received from the peripheral side for the issued token</li> <li>When a response cannot be received correctly from the peripheral side for the issued token (No response is returned three consecutive times or a packet reception error occurred three consecutive times.)</li> <li>When an overrun/underrun occurred during isochronous transfer</li> </ul> <p>When the function controller function is selected:</p> <ul style="list-style-type: none"> <li>When NAK is returned for an IN/OUT/PING token.</li> <li>When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer</li> <li>When an overrun/underrun occurred during data reception in isochronous transfer</li> </ul>	Host, function	NRDYSTS. PIPENRDY
BRDY	Buffer ready interrupt	When the buffer is ready (reading or writing is enabled)	Host, function	BRDYSYS PIPEBRDY
BCHG	Bus change interrupt	When a change of USB bus state is detected	Host, function	—
DTCH	Disconnection detection during full-speed operation	When disconnection of a peripheral device during full-speed operation is detected	Host	—
ATTCH	Device connection detection	<p>When J-state or K-state is detected on the USB port for 2.5 <math>\mu</math>s.</p> <p>Used for checking whether a peripheral device is connected.</p>	Host	—

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
EOFERR	EOF error detection	When EOF error of a peripheral device is detected	Host	—
SACK	Normal setup operation	When the normal response (ACK) for the setup transaction is received	Host	—
SIGN	Setup error	When a setup transaction error (no response or ACK packet corruption) is detected three consecutive times.	Host	—

Note: All the bits without register name indication are in INTSTS0.

Figure 21.2 shows a diagram relating to interrupts of this module.



**Figure 21.2 Items Relating to Interrupts**



## (1) BRDY Interrupt

The BRDY interrupt is generated when either of the host controller function or function controller function is selected. The following shows the conditions under which this module sets 1 to a corresponding bit in BRDYSTS. Under this condition, this module generates BRDY interrupt, if software sets the PIPEBRDYE bit in BRDYENB that corresponds to the pipe to 1 and the BRDYE bit in INTENB0 to 1.

The conditions for generating and clearing the BRDY interrupt depend on the settings of the BRDYM bit and BFRE bit for the pertinent pipe as described below.

### (a) When the BRDYM bit is 0 and BFRE bit is 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, this module generates the internal BRDY interrupt request trigger and sets 1 to the PIPEBRDY bit corresponding to the pertinent pipe.

- (i) For the pipe in the transmitting direction:
  - When software changes the DIR bit from 0 to 1.
  - When packet transmission is completed using the pertinent pipe when write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).  
In continuous transmission/reception mode, the request trigger is generated on completion of transmitting data of one plane of the FIFO buffer.
  - When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.  
The request trigger is not generated until completion of writing data to the currently-written FIFO buffer plane even if transmission to the other FIFO buffer is completed.
  - When the hardware flushes the buffer of the pipe for isochronous transfers.
  - When 1 is written to the ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

The request trigger is not generated for the DCP (that is, during data transmission for control transfers).

## (ii) For the pipe in the receiving direction:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read when read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).

The request trigger is not generated for the transaction in which DATA-PID disagreement occurs.

In continuous transmission/reception mode, the request trigger is not generated when the data is of the specified maximum packet size and the buffer has available space.

When a short packet is received, the request trigger is generated even if the FIFO buffer has available space.

When the transaction counter is used, the request trigger is generated on receiving the specified number of packets. In this case, the request trigger is generated even if the FIFO buffer has available space.

- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.

The request trigger is not generated until completion of reading data from the currently-read FIFO buffer plane even if reception by the other FIFO buffer is completed.

When the function controller function is selected, the BRDY interrupt is not generated in the status stage of control transfers.

The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY interrupt status bit in the BRDYSTS register through software. In this case, 1s should be written to the PIPEBRDY interrupt status bits for the other pipes.

Be sure to clear the BRDY status before accessing the FIFO buffer.

**(b) When the BRDYM bit is 0 and the BFRE bit is 1**

With these settings, this module generates the BRDY interrupt on completion of reading all the data for a single transfer using the pipe in the receiving direction, and sets 1 to the PIPEBRDY bit corresponding to the pertinent pipe.

On any of the following conditions, this module determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the transaction counter register (TRNCNT bits) is used and the number of packets specified by the TRNCNT bits are completely received.

When the pertinent data is completely read out after any of the above determination conditions has been satisfied, this module determines that all the data for a single transfer has been completely read out.

When a zero-length packet is received when the FIFO buffer is empty, this module determines that all the data for a single transfer has been completely read out upon passing the zero-length packet data to the CPU. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFOCTR register through software.

With these settings, this module does not detect the BRDY interrupt for the pipe in the transmitting direction.

The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY interrupt status bit through software. In this case, 1s should be written to the PIPEBRDY interrupt status bits for the other pipes.

In this mode, the BFRE bit setting should not be modified until all the data for a single transfer has been processed. When it is necessary to modify the BFRE bit before completion of processing, all the FIFO buffers for the pertinent pipe should be cleared using the ACLRM bit.

**(c) When the BRDYM bit is 1 and the BFRE bit is 0**

With these settings, the PIPEBRDY values are linked to the BSTS bit settings for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by this module depending on the FIFO buffer status.

**(i) For the pipe in the transmitting direction:**

The BRDY interrupt status bits are set to 1 when the FIFO buffer is write-enabled and are set to 0 when write-disabled.

However, the BRDY interrupt is not generated if the DCP in the transmitting direction is write-enabled.

**(ii) For the pipe in the receiving direction:**

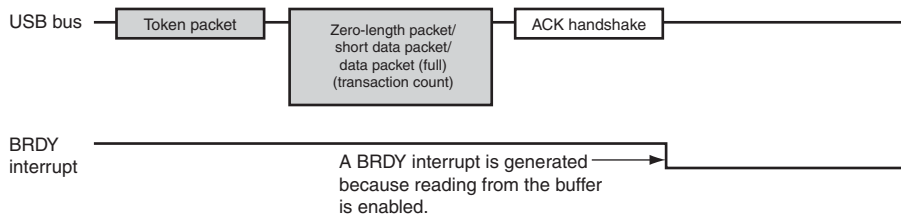
The BRDY interrupt status bits are set to 1 when the FIFO buffer is read-enabled and are set to 0 when all the data have been read (read-disabled).

When a zero-length packet is received when the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.

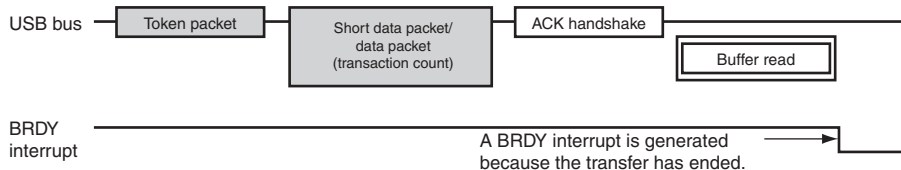
With this setting, the PIPEBRDY bit cannot be cleared to 0 through software. When BRDYM is set to 1, all of the BFRE bits (for all pipes) should be cleared to 0.

Figure 21.3 shows the timing at which the BRDY interrupt is generated.

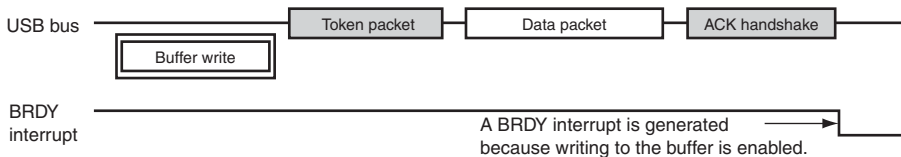
- (1) Zero-length packet reception or data packet reception when BFRE = 0  
(short packet reception/transaction counter completion/buffer full)



- (2) Data packet reception when BFRE = 1 (short packet reception/transaction counter completion)



- (3) Packet transmission



**Figure 21.3 Timing at which a BRDY Interrupt is Generated**

## (2) NRDY Interrupt

On generating the internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, this module sets the corresponding PIPENRDY bit in NRDYSTS to 1. If the corresponding bit in NRDYENB is set to 1 by software, this module sets the NRDY bit in INTSTS0 to 1, allowing the USB interrupt to be generated.

The following describes the conditions on which this module generates the internal NRDY interrupt request for a given pipe.

However, the internal NRDY interrupt request is not generated during setup transaction execution when the host controller function is selected. During setup transactions when the host controller function is selected, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller function is selected.

### (a) When the host controller function is selected and when the connection is used in which no split transactions occur

#### (i) For the pipe in the transmitting direction:

On any of the following conditions, this module detects the NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes in a state in which there is no data to be transmitted in the FIFO buffer.  
In this case, this module transmits a zero-length packet following the OUT token, setting the corresponding PIPENRDY bit and the OVRN bit to 1.
- During communications other than setup transactions using the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.  
In this case, this module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK.
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device (including the STALL handshake in response to PING in addition to the STALL handshake in response to OUT).  
In this case, this module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to STALL (11).

## (ii) For the pipe in the receiving direction

- For the pipe for isochronous transfers, when the time to issue an IN token comes in a state in which there is no space available in the FIFO buffer.  
In this case, this module discards the received data for the IN token, setting the PIPENRDY bit of the corresponding pipe and the OVRN bit to 1.  
When a packet error is detected in the received data for the IN token, this module also sets the CRCE bit to 1.
- For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by this module (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.  
In this case, this module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK.
- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device.  
In this case, this module sets the corresponding PIPENRDY bit to 1. (The setting of the PID bits of the corresponding pipe to NAK is not modified.)
- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.  
In this case, this module sets the corresponding PIPENRDY bit and CRCE bit to 1.
- When the STALL handshake is received.  
In this case, this module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to STALL.

**(b) When the host controller function is selected and when the connection is used in which split transactions occur****(i) For the pipe in the transmitting direction:**

- For the pipe for isochronous transfers, when the time to issue an OUT token comes in a state in which there is no data to be transmitted in the FIFO buffer.  
In this case, this module transmits a zero-length packet following the OUT token, setting the corresponding PIPENRDY bit and the OVRN bit to 1 at the issuance of the start-split transaction (S-SPLIT).
- For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the HUB for the S-SPLIT or complete-split transaction (C-SPLIT) (when timeout is detected before detection of the handshake packet from the HUB) and 2) an error is detected in the packet from the HUB.  
In this case, this module sets the PIPENRDY bit of the corresponding pipe to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK.  
If the NRDY interrupt is detected when the C-SPLIT is issued, this module clears the CSSTS bit to 0.
- When the STALL handshake is received in response to the C-SPLIT.  
In this case, this module sets the corresponding PIPENRDY bit to 1, modifies the setting of the PID bits of the corresponding pipe to STALL (11) and clears the CSSTS bit to 0.  
This interrupt is not detected for SETUP transactions.
- When the NYET is received in response to the C-SPLIT and the microframe number = 4.  
In this case, this module sets the corresponding PIPENRDY bit to 1 and clears the CSSTS bit to 0 (does not modify the setting of the PID bits).

**(ii) For the pipe in the receiving direction:**

- For the pipe for isochronous transfers, when the time to issue an IN token comes in a state in which there is no space available in the FIFO buffer.  
In this case, this module discards the received data for the IN token, setting the corresponding PIPENRDY bit and the OVRN bit to 1 at the issuance of the S-SPLIT.
- During bulk-pipe transfers or the transfers other than SETUP transactions with the DCP, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the HUB for the IN token issued by this module at the issuance of S-SPLIT or C-SPLIT (when timeout is detected before detection of the DATA packet from the HUB) and 2) an error is detected in the packet from the HUB.



In this case, this module sets the corresponding PIPENRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK. When the condition is generated during the C-SPLIT transaction, this module clears the CSSTS bit to 0.

- During the C-SPLIT transaction for the pipe for isochronous transfers or interrupt transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the HUB for the IN token issued by this module (when timeout is detected before detection of the DATA packet from the HUB) and 2) an error is detected in the packet from the HUB.

On generating this condition for the pipe for interrupt transfers, this module sets the corresponding PIPENRDY bit to 1, modifies the setting of the PID bits of the corresponding pipe to NAK and clears the CSSTS bit to 0.

On generating this condition for the pipe for isochronous transfers, this module sets the corresponding PIPENRDY bit to 1 and CRCE bit to 1, and clears the CSSTS bit to 0 (does not modify the setting of the PID bits).

- During the C-SPLIT transaction, when the STALL handshake is received for the pipe for the transfers other than isochronous transfers.

In this case, this module sets the corresponding PIPENRDY bit to 1, modifies the setting of the PID bits of the corresponding pipe to STALL (11) and clears the CSSTS bit to 0.

- During the C-SPLIT transaction, when the NYET handshake is received for the pipe for the isochronous transfers or interrupt transfers and the microframe number = 4.

In this case, this module sets the corresponding PIPENRDY bit to 1 and CRCE bit to 1, and clears the CSSTS bit to 0 (does not modify the setting of the PID bits).

### (c) When the function controller function is selected

#### (i) For the pipe in the transmitting direction:

- On receiving an IN token when there is no data to be transmitted in the FIFO buffer.

In this case, this module generates a NRDY interrupt request at the reception of the IN token, setting the PIPENRDY bit to 1. For the pipe for the isochronous transfers in which an interrupt is generated, this module transmits a zero-length packet, setting the OVRN bit to 1.

#### (ii) For the pipe in the receiving direction:

- On receiving an OUT token when there is no space available in the FIFO buffer.

For the pipe for the isochronous transfers in which an interrupt is generated, this module generates a NRDY interrupt request, setting the PIPENRDY bit to 1 and OVRN bit to 1.

For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, this module generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token was received, setting the PIPENRDY bit to 1.

However, during re-transmission (due to DATA-PID disagreement), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.

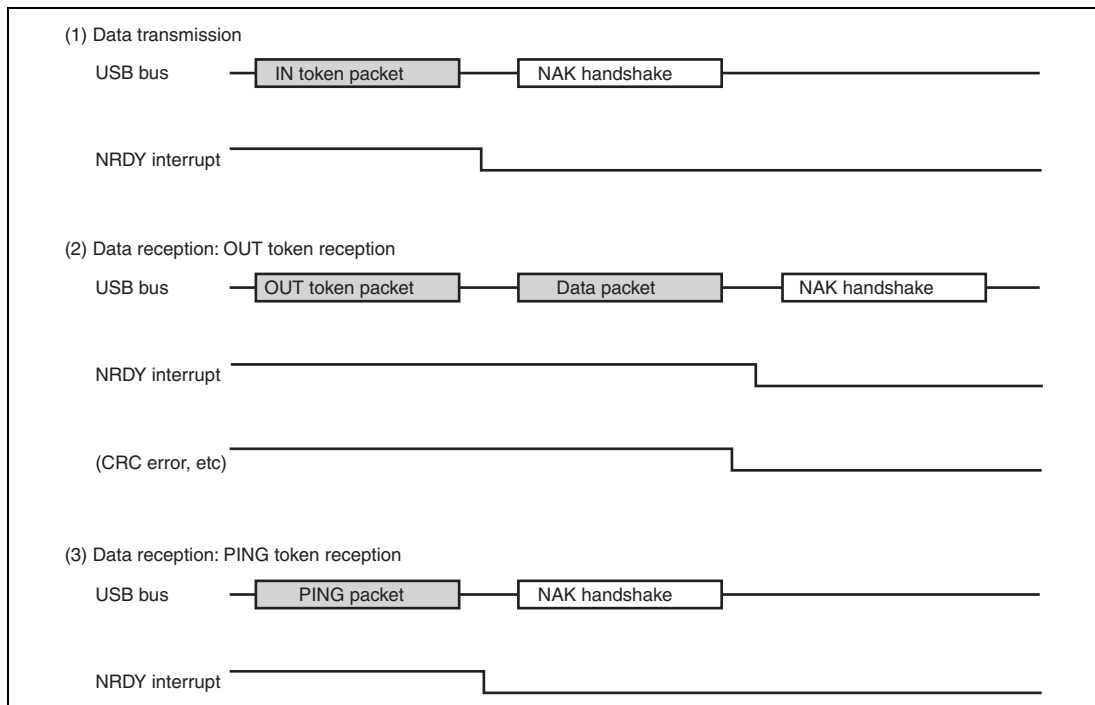
- On receiving a PING token when there is no space available in the FIFO buffer.

In this case, this module generates a NRDY interrupt request at the reception of the PING token, setting the PIPENRDY bit to 1.

- For the pipe for isochronous transfers, when a token is not received normally within an interval frame.

In this case, this module generates a NRDY interrupt request, setting the PIPENRDY bit to 1.

Figure 21.4 shows the timing at which an NRDY interrupt is generated when the function controller function is selected.



**Figure 21.4 Timing at which NRDY Interrupt is Generated when Function Controller Function is Selected**

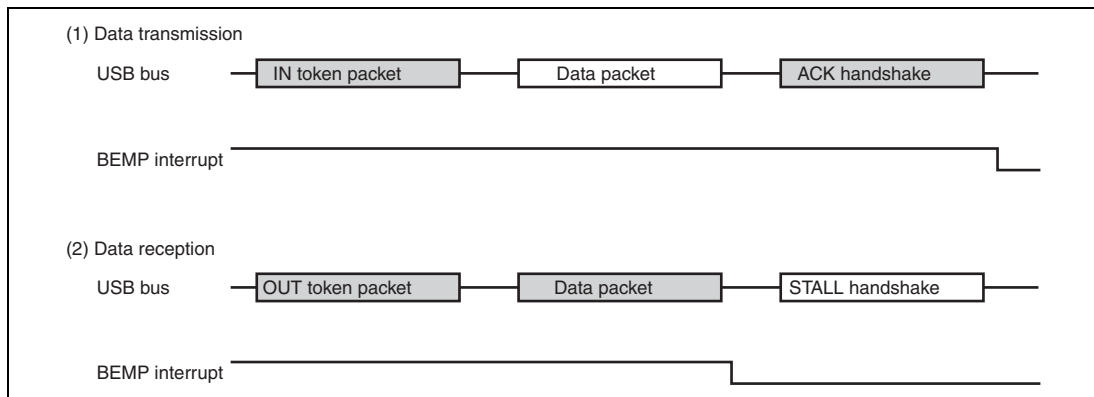
### (3) BEMP Interrupt

On generating the BEMP interrupt for the pipe whose PID bits are set to BUF by software, this module sets the corresponding PIPEBEMP bit in BEMPSTS to 1. If the corresponding bit in BEMPENB is set to 1 by software, this module sets the BEMP bit in INTSTS0 to 1, allowing the USB interrupt to be generated.

The following describes the conditions on which this module generates the internal BEMP interrupt request.

- (a) For the pipe in the transmitting direction, when the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission). In single buffer mode, the internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.
  - When software (DMAC) has already started writing data to the FIFO buffer of the CPU on completion of transmitting data of one plane in double buffer mode.
  - When the buffer is cleared (emptied) by setting the ACLRM or BCLR bit to 1.
  - When IN transfer (zero-length packet transmission) is performed during the control transfer status stage in function controller mode.
- (b) For the pipe in the receiving direction:
  - When the successfully-received data packet size exceeds the specified maximum packet size. In this case, this module generates the BEMP interrupt request, setting the corresponding PIPEBEMP bit to 1, and discards the received data and modifies the setting of the PID bits of the corresponding pipe to STALL (11).
  - Here, this module returns no response when used as the host controller, and returns STALL response when used as the function controller.
  - However, the internal BEMP interrupt request is not generated on any of the following conditions.
    - When a CRC error or bit stuffing error is detected in the received data.
    - When a setup transaction is being performed. Writing 0 to the PIPEBEMP bit clears the status; writing 1 to the PIPEBEMP bit has no effect.

Figure 21.5 shows the timing at which a BEMP interrupt is generated when the function controller function has been selected.



**Figure 21.5 Timing at which BEMP Interrupt is Generated when Function Controller Function is Selected**

#### (4) Device State Transition Interrupt

Figure 21.6 shows a diagram of this module 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 using 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.

Device state can be controlled only when the function controller function is selected. Also, the device state transition interrupts can be generated only when the function controller function is selected.

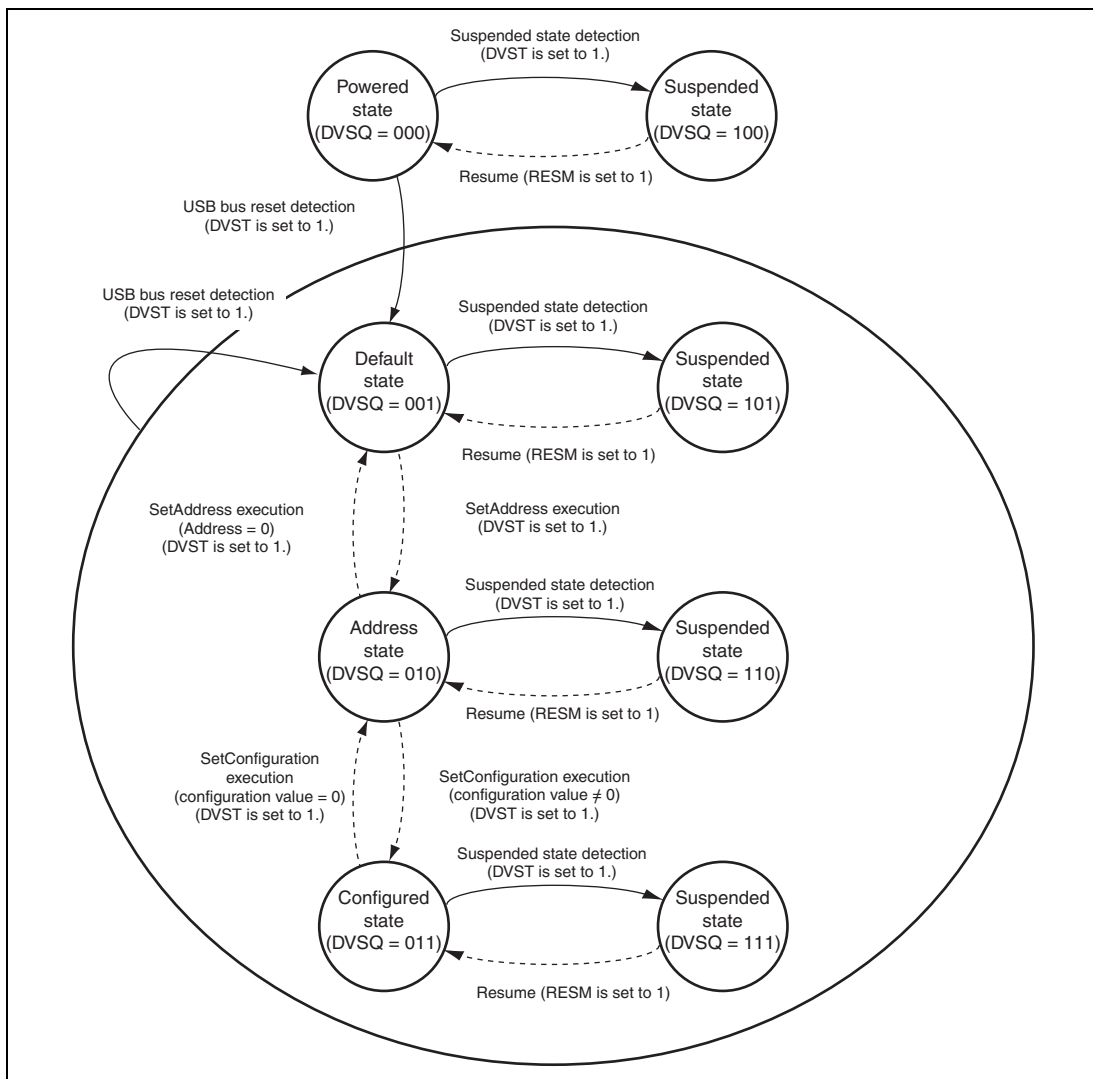


Figure 21.6 Device State Transitions

## (5) Control Transfer Stage Transition Interrupt

Figure 21.7 shows a diagram of how this module handles the control transfer stage transition. 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 using INTENB0. The transfer stage that made a transition can be confirmed using the CTSQ bit in INTSTS0.

The control transfer stage transition interrupts are generated only when the function controller function is selected.

The control transfer sequence errors are described below. If an error occurs, the PID bit in DCPCTR is set to B'1x (STALL).

### (a) 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 is received at the status stage for which the data packet is DATAPID = DATA0

### (b) During control write transfers

- At the OUT token of the data stage, an IN token is received when there have been no ACK response at all
- A packet is received at the data stage for which the first data packet is DATAPID = DATA0
- At the status stage, an OUT or PING token is received

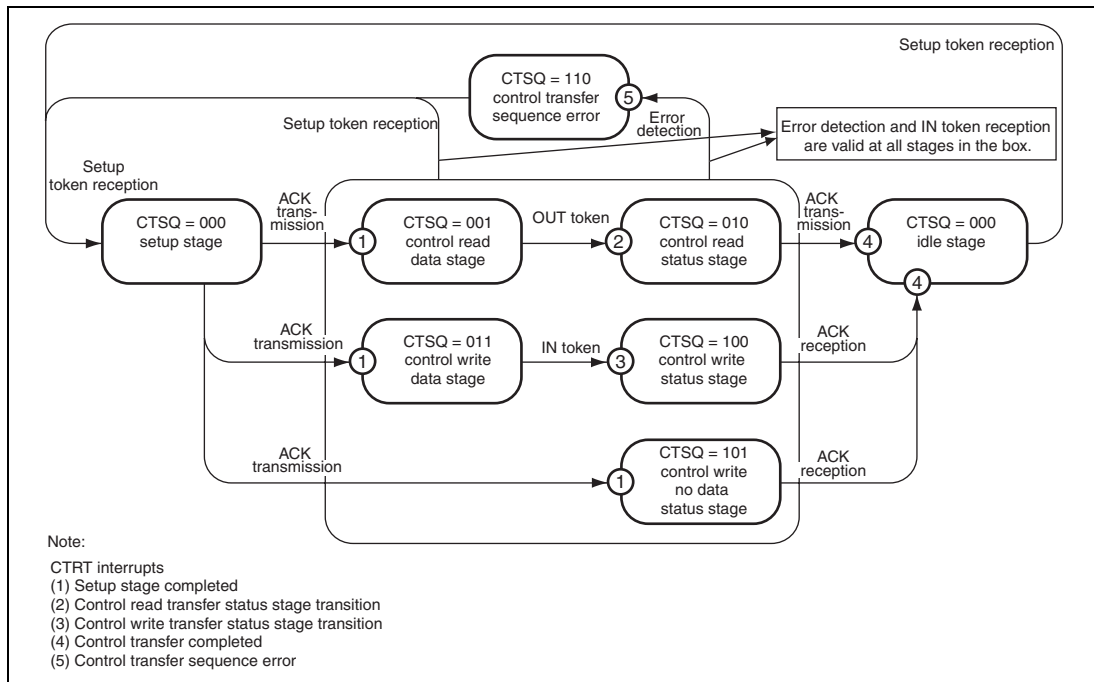
### (c) During no-data control transfers

- At the status stage, an OUT or PING token is received

At the control write transfer stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, 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 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 21.7 Control Transfer Stage Transitions**

## (6) Frame Update Interrupt

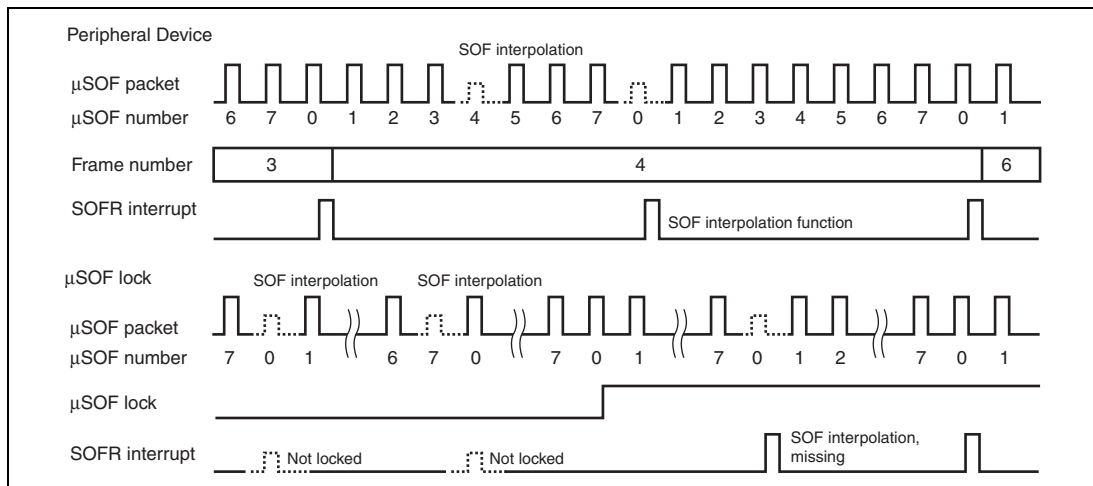
Figure 21.8 shows an example of the SOFR interrupt output timing of this module. With the host controller function selected, an interrupt is generated at the timing at which the frame number is updated. With the function controller function selected, the SOFR interrupt is generated when the frame number is updated.

When the function controller function is selected, this module updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation. During high-speed operation, however, this module does not update the frame number, or generates no SOFR interrupt until the module enters the  $\mu$ SOF locked state. Also, the SOF interpolation function is not activated. The  $\mu$ SOF lock state is the state in which  $\mu$ SOF packets with different frame numbers are received twice continuously without error occurrence.

The conditions under which the  $\mu$ SOF lock monitoring begins and stops are as follows.



1. Conditions under which  $\mu$ SOF lock monitoring begins  
USBE = 1
2. Conditions under which  $\mu$ SOF lock monitoring stops  
USBE = 0, a USB bus reset is received, or suspended state is detected.



**Figure 21.8 Example of SOFR Interrupt Output Timing**

## (7) VBUS Interrupt

If there has been a change in the VBUS pin, the VBUS interrupt is generated. The level of the VBUS pin can be checked with the VBSTS bit in INTSTS0. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the VBUS pin.

## (8) Resume Interrupt

The RESM interrupt is generated when the device state is the suspended state, and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.

## (9) BCHG Interrupt

The BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether or not the peripheral device is connected when the host controller function has been selected and can also be used to detect a remote wakeup. The BCHG interrupt is

generated regardless of whether the host controller function or function controller function has been selected.

#### **(10) DTCH Interrupt**

The DTCH interrupt is generated if disconnection of the USB bus is detected when the host controller function has been selected. This module detects bus disconnection based on USB Specification 2.0.

After detecting the DTCH interrupt, this module controls hardware as described below (irrespective of the set value of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried out for the pertinent port and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- (a) Modifies the UACT bit for the port in which a DTCH interrupt has been detected to 0.
- (b) Puts the port in which a DTCH interrupt has been generated into the idle state.

#### **(11) SACK Interrupt**

The SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller function selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

#### **(12) SIGN Interrupt**

The SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller function selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

#### **(13) ATTCH Interrupt**

The ATTCH interrupt is generated when J-state or K-state of the full-speed level signal is detected on the USB port for 2.5  $\mu$ s in host controller mode. To be more specific, the ATTCH interrupt is detected on any of the following conditions.

- (a) When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5  $\mu$ s.
- (b) When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5  $\mu$ s.

## (14) EOFERR Interrupt

The EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0.

After detecting the EOFERR interrupt, this module controls hardware as described below (irrespective of the set value of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried out for the pertinent port and perform re-enumeration of the pertinent port.

- (a) Modifies the UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- (b) Puts the port in which an EOFERR interrupt has been generated into the idle state.

### 21.4.3 Pipe Control

Table 21.18 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 system.

**Table 21.18 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 transfer direction	IN or OUT can be set
	EPNUM	Endpoint number	PIPE1 to PIPE9: Can be set A value other than 0000 should be set when the pipe is used.
PIPEBUF	BUFSIZE	Buffer memory size	PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected)
			PIPE3 to PIPE5: Can be set
	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'7F) PIPE6 to PIPE9: Cannot be set (areas fixed at H'4 to H'7)

Register Name	Bit Name	Setting Contents	Remarks
DCPMAXP PIPEMAXP	DEVSEL	Selects a device	Referenced only when the host controller function is selected.
	MXPS	Maximum packet size	Compliant with the USB standard.
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected)  PIPE3 to PIPE5: Cannot be set PIPE6 to PIPE9: Can be set (only when the host controller function has been selected)
	IITV	Interval counter	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected)  PIPE3 to PIPE5: Cannot be set PIPE6 to PIPE9: Can be set (only when the host controller function has been selected)
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Mounted for PIPE3 to PIPE5.
	SUREQ	SETUP request	Can be set only for the DCP.  Can be controlled only when the host controller function has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP.  Can be controlled only when the host controller function has been selected.
	CSCLR	CSSTS clear	Can be controlled only when the host controller function has been selected.
	CSSTS	SPLIT status indication	Can be referenced only when the host controller function has been selected.
	ATREPM	Auto response mode	PIPE1 to PIPE5: Can be set  Can be controlled only when the function controller function has been selected.

Register Name	Bit Name	Setting Contents	Remarks
DCPCTR PIPEnCTR	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	Monitors the data toggle bit
	PBUSY	Pipe busy status	
	PID	Response PID	See section 21.4.3 (6), Response PID
PIPEnTRE	TRENB	Transaction counter enable	PIPE1 to PIPE5: Can be set
	TRCLR	Current transaction counter clear	PIPE1 to PIPE5: Can be set
PIPEnTRN	TRNCNT	Transaction counter	PIPE1 to PIPE5: Can be set

### (1) Pipe control register switching procedures

The following bits in the pipe control registers can be modified only when USB communication is disabled (PID = NAK):

Registers that Should Not be Set in the USB Communication Enabled (PID = BUF) State

- Bits in DCPCFG and DCPMAXP
- The SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI
- The ATREPM, ACLRM, SQCLR and SQSET bits in PIPExCTR
- Bits in PIPExTRE and PIPExTRN

In order to modify the above bits from the USB communication enabled (PID = BUF) state, follow the procedure shown below:

1. Generate a bit modification request with the pipe control register.
2. Modify the PID corresponding to the pipe to NAK.
3. Wait until the corresponding CSSTS bit is cleared to 0 (only when the host controller function has been selected).
4. Wait until the corresponding PBUSY bit is cleared to 0.

## 5. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent information has not been set by the CURPIPE bits in CFIFOSEL, D0FIFOSEL and D1FIFOSEL.

Registers that Should Not be Set When CURPIPE in FIFO-PORT is set.

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI

In order to modify pipe information, the CURPIPE bits should be set to the pipes other than the pipe to be modified. For the DCP, the buffer should be cleared using BCLR after the pipe information is modified.

## (2) Transfer Types

The TYPE bit in PIPEPCFG is used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

1. DCP: No setting is necessary (fixed at control transfer).
2. PIPE1 and PIPE2: These should be set to bulk transfer or isochronous transfer.
3. PIPE3 to PIPE5: These should be set to bulk transfer.
4. PIPE6 to PIPE9: These should be set to interrupt transfer.

## (3) Endpoint Number

The EPNUM bit in PIPEPCFG is used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

1. DCP: No setting is necessary (fixed at end point 0).
2. PIPE1 to PIPE9: The endpoint numbers from 1 to 15 should be selected and set.  
These should be set so that the combination of the DIR bit and EPNUM bit is unique.

#### (4) Maximum Packet Size Setting

The MXPS bit in DCPMAXP and PIPEMAXP is used to specify the maximum packet size for each pipe. DCP and PIPE1 to PIPE5 can be set to any of the maximum pipe sizes defined by the USB specification. For PIPE6 to PIPE9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

1. DCP: 64 should be set when using high-speed operation.
2. DCP: Select and set 8, 16, 32, or 64 when using full-speed operation.
3. PIPE1 to PIPE5: 512 should be set when using high-speed bulk transfer.
4. PIPE1 to PIPE5: Select and set 8, 16, 32, or 64 when using full-speed bulk transfer.
5. PIPE1 and PIPE2: Set a value between 1 and 1024 when using high-speed isochronous transfer.
6. PIPE1 and PIPE2: Set a value between 1 and 1023 when using full-speed isochronous transfer.
7. PIPE6 to PIPE9: Set a value between 1 and 64.

The high bandwidth transfers used with interrupt transfers and isochronous transfers are not supported.

#### (5) Transaction Counter (For PIPE1 to PIPE5 in Reading Direction)

When the specified number of transactions have been completed in the data packet receiving direction, this module recognizes that the transfer has ended. The transaction counter function is available when the pipes assigned to the D0FIFO/D1FIFO port have been set in the direction of reading data from the buffer memory. Two transaction counters are provided: one is the TRNCNT register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. When the current counter value matches the number of the transactions specified in TRNCNT, reading the buffer memory is enabled. The current counter of the transaction counter function is initialized by the TRCLR bit, so that the transactions can be counted again starting from the beginning. The information read from TRNCNT differs depending on the setting of the TRENb bit.

- TRENb = 0: The specified transaction counter value can be read.
- TRENb = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.



## (6) Response PID

The PID bits in DCPCTR and PIPEnCTR are used to set the response PID for each pipe.

The following shows this module operation with various response PID settings:

### (a) Response PID settings when the host controller function is selected:

The response PID is used to specify the execution of transactions.

- (i) NAK setting: Using pipes is disabled. No transaction is executed.
- (ii) BUF setting: Transactions are executed based on the status of the buffer memory. For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued. For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
- (iii) STALL setting: Using pipes is disabled. No transaction is executed.

Setup transactions for the DCP are set with the SUREQ bit.

### (b) Response PID settings when the function controller function is selected:

The response PID is used to specify the response to transactions from the host.

- (i) NAK setting: The NAK response is always returned in response to the generated transaction.
- (ii) BUF setting: Responses are made to transactions based on the status of the buffer memory.
- (iii) STALL setting: The STALL response is always returned in response to the generated transaction.

For setup transactions, an ACK response is always returned, regardless of the PID setting, and the USB request is stored in the register.

This module may carry out writing to the PID bits, depending on the results of the transaction.

### (c) When the host controller function has been selected and the response PID is set by hardware:

- (i) NAK setting: In the following cases, PID = NAK is set and issuing of tokens is automatically stopped:

- When a transfer other than isochronous transfer has been performed and the NRDY interrupt is generated. (For details, see descriptions of the NRDY interrupt.)
  - If a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.
  - If the transaction counter ended when the SHTNAK bit has been set to 1 for bulk transfer.
- (ii) BUF setting: There is no BUF writing by this module.
- (iii) STALL setting: In the following cases, PID = STALL is set and issuing of tokens is automatically stopped:
- When STALL is received in response to the transmitted token.
  - When the size of the receive data packet exceeds the maximum packet size.
- (d) When the function controller function has been selected and the response PID is set by hardware:**
- (i) NAK setting: In the following cases, PID = NAK is set and NAK is always returned in response to transactions:
- When the SETUP token is received normally (DCP only).
  - If the transaction counter ended or a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.
- (ii) BUF setting: There is no BUF writing by this module.
- (iii) STALL setting: In the following cases, PID = STALL is set and STALL is always returned in response to transactions:
- When the size of the receive data packet exceeds the maximum packet size.
  - When a control transfer sequence error has been detected (DCP only).

## (7) Data PID Sequence Bit

This module automatically toggles the sequence bit in the data PID when data is transferred normally in the control transfer data stage, bulk transfer and interrupt transfer. The sequence bit of the data PID that was transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit switches at the timing at which the ACK handshake is received. When data is received, the sequence bit switches at the timing at which the ACK handshake is transmitted. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

When the function controller function has been selected and control transfer is used, this module automatically sets the sequence bit when a stage transition is made. DATA0 is returned when the setup stage is ended and DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller function has been selected and control transfer is used, the sequence bit should be set by software at the stage transition.

For the Clearfeature request transmission or reception, the data PID sequence bit should be set by software, regardless of whether the host controller function or function controller function is selected.

With pipes for which isochronous transfer has been set, sequence bit operation cannot be carried out using the SQSET bit.

## (8) Response PID = NAK Function

This module has a function that disables pipe operation (PID response = NAK) at the timing at which the final data packet of a transaction is received (this module automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the SHTNAK bit in PIPECFG to 1.

When a double buffer is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has disabled, the pipe has to be set to the enabled state again (PID response = BUF) using software.

This function can be used only when bulk transfers are used.

## **(9) Auto Transfer MODE**

With the pipes for bulk transfer (PIPE1 to PIPE5), when the ATREPM bit in PIPEnCTR is set to 1, a transition is made to auto response mode. During an OUT transfer (DIR = 0), OUT-NAK mode is entered, and during an IN transfer (DIR = 1), null auto response mode is entered.

### **(a) OUT-NAK Mode**

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT or PING token and an NRDY interrupt is output when the ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled and an ACK is returned in response to a PING token if the buffer is ready to receive data.

### **(b) Null Auto Response Mode**

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, the buffer should be emptied with the ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (full-speed: 10  $\mu$ s, high-speed: 3  $\mu$ s) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

## 21.4.4 FIFO Buffer Memory

### (1) FIFO Buffer Memory Allocation

Figure 21.9 shows an example of a FIFO buffer memory map for this module. The FIFO buffer memory is an area shared by the CPU and this module. In the FIFO buffer memory 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).

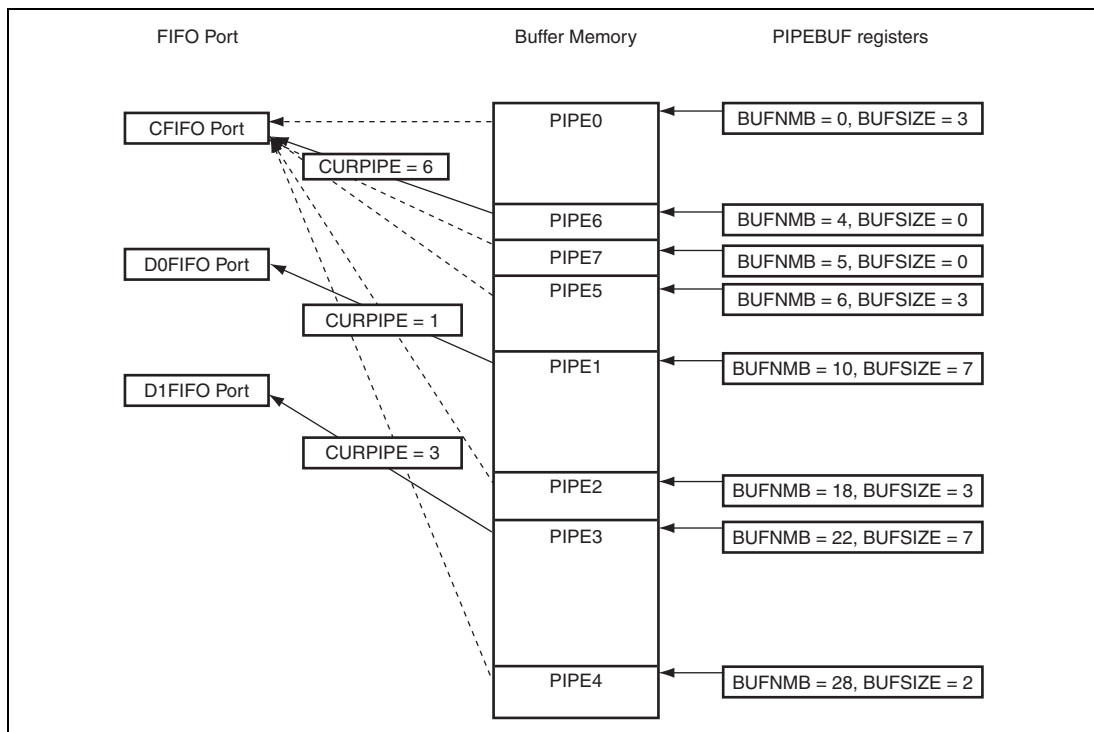
The buffer memory sets independent areas for each pipe. In the memory areas, 64 bytes comprise one block, and the memory areas are set using the first block number of the number of blocks (specified using the BUFNMB and BUFSIZE bits in PIPEBUF).

Independent buffer memory areas should be set for each pipe. Each memory area can be set using the first block number and the number of blocks (specified using the BUFNMB and BUFSIZE bits in PIPEBUF), where one block comprises 64 bytes.

When continuous transfer mode has been selected using the CNTMD bit in PIPEnCFG, the BUFSIZE bits should be set so that the buffer memory size should be an integral multiple of the maximum packet size. When double buffer mode has been selected using the DBLB bit in PIPEnCFG, two planes of the memory area specified using the BUFSIZE bits in PIPEBUF can be assigned to a single pipe.

Moreover, three FIFO ports are used for access to the buffer memory (reading and writing data). A pipe is assigned to the FIFO port by specifying the pipe number using the CURPIPE bit in C/DnFIFOSEL.

The buffer statuses of the various pipes can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. Also, the access right of the FIFO port can be confirmed using the FRDY bit in CFIFOCTR or DnFIFOCTR.



**Figure 21.9 Example of a Buffer Memory Map**

#### (a) Buffer Status

Tables 21.19 and 21.20 show the buffer status. The buffer memory status can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. The access direction for the buffer memory can be specified using either the DIR bit in PIPEnCFG or the ISEL bit in CFIFOSEL (when DCP is selected).

The INBUFM bit is valid for PIPE0 to PIPE5 in the sending direction.

For an IN pipe uses double buffer, software can refer the BSTS bit to monitor the buffer memory status of CPU side and the INBUFM bit to monitor the buffer memory status of SIE side. In the case like the BEMP interrupt may not shows the buffer empty status because the CPU (DMAC) writes data slowly, software can use the INBUFM bit to confirm the end of sending.

**Table 21.19 Buffer Status Indicated by the BSTS Bit**

<b>ISEL or DIR</b>	<b>BSTS</b>	<b>Buffer Memory State</b>
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is inhibited.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. However, because reading is not possible when a zero-length packet is received, the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been finished. Writing to the FIFO port is inhibited.
1 (transmitting direction)	1	The transmission has been finished. CPU write is allowed.

**Table 21.20 Buffer Status Indicated by the INBUFM Bit**

<b>IDIR</b>	<b>INBUFM</b>	<b>Buffer Memory State</b>
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been finished. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted

**(b) FIFO Buffer Clearing**

Table 21.21 shows the clearing of the FIFO buffer memory by this module. The buffer memory can be cleared using the three bits indicated below.

**Table 21.21 List of Buffer Clearing Methods**

Bit Name	BCLR	DCLRM	ACLRM
Register	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Function	Clears the buffer memory on the CPU side	In this mode, after the data of the specified pipe has been read, the buffer memory 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

**(c) Buffer Areas**

Table 21.22 shows the FIFO buffer memory map of this controller. The buffer memory has special fixed areas to which pipes are assigned in advance, and user areas that can be set by the user.

The buffer for the DCP is a special fixed area that is used both for control read transfers and control write transfers.

The PIPE6 to PIPE9 area is assigned in advance, but the area for pipes that are not being used can be assigned to PIPE1 to PIPE5 as a user area.

The settings should ensure that the various pipes do not overlap. Note that each area is twice as large as the setting value in the double buffer.

Also, the buffer size should not be specified using a value that is less than the maximum packet size.



**Table 21.22 Buffer Memory Map**

<b>Buffer Memory Number</b>	<b>Buffer Size</b>	<b>Pipe Setting</b>	<b>Note</b>
H'0	64 bytes	Fixed area only for the DCP	Single buffer, continuous transfers enabled
H'1 to H'3	—	Prohibited to be used	—
H'4	64 bytes	Fixed area for PIPE6	Single buffer
H'5	64 bytes	Fixed area for PIPE7	Single buffer
H'6	64 bytes	Fixed area for PIPE8	Single buffer
H'7	64 bytes	Fixed area for PIPE9	Single buffer
H'8 to H'4F	Up to 5120 bytes	PIPE1 to PIPE5 user area	Double buffer can be set, continuous transfers enabled

**(d) Auto Buffer Clear Mode Function**

With this module, all of the received data packets are discarded if the ACLRM bit in PIPEnCTR is set to 1. If a normal data packet has been received, the ACK response is returned to the host controller. This function can be set only in the buffer memory reading direction.

Also, if the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.

An access cycle of at least 100 ns is required between ACLRM = 1 and ACLRM = 0.

**(e) Buffer Memory Specifications (Single/Double Setting)**

Either a single or double buffer can be selected for PIPE1 to PIPE5, using the DBLB bit in PIPEnCFG. The double buffer is a function that assigns two memory areas specified with the BUFSIZE bit in PIPEBUF to the same pipe. Figure 21.10 shows an example of buffer memory settings for this module.

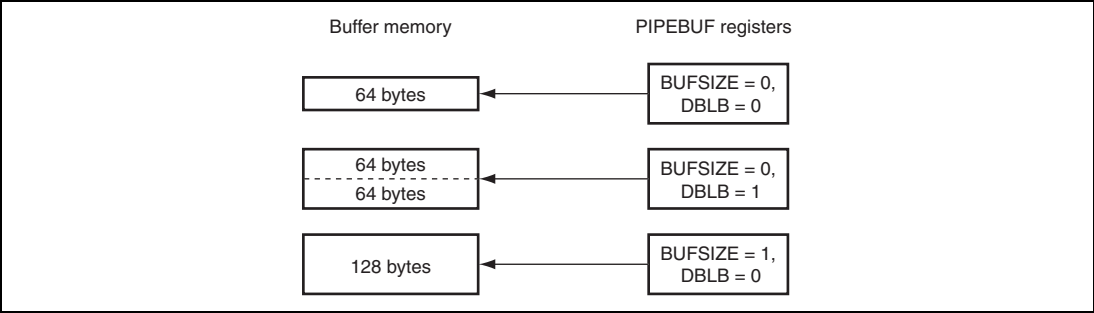


Figure 21.10 Example of Buffer Memory Settings

(f) Buffer Memory Operation (Continuous Transfer Setting)

Either the continuous transfer mode or the non-continuous transfer mode can be selected, using the CNTMD bit in PIPEnCFG. This selection is valid for PIPE1 to PIPE5.

The continuous transfer mode function is a function that sends and receives multiple transactions in succession. When the continuous transfer mode is set, data can be transferred without interrupts being issued to the CPU, up to the buffer sizes assigned for each of the pipes.

In the continuous sending mode, the data being written is divided into packets of the maximum packet size and sent. If the data being sent is less than the buffer size (short packet, or the integer multiple of the maximum packet size is less than the buffer size), BVAL = 1 must be set after the data being sent has been written.

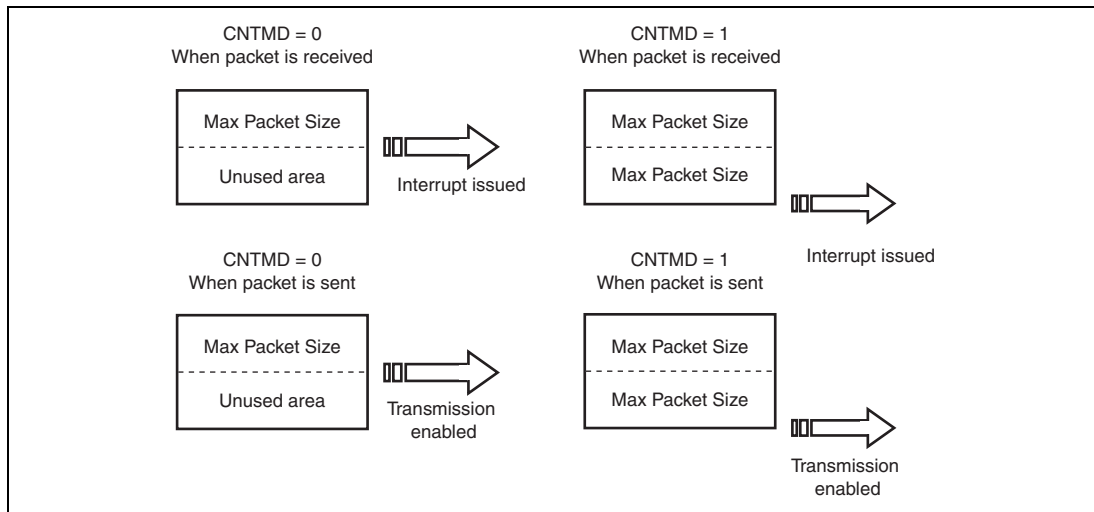
In the continuous reception mode, interrupts are not issued during reception of packets up to the buffer size, until the transaction counter has ended, or a short packet is received.

Table 21.23 describes the relationship between the transfer mode settings by CNTMD bit and the timings at which reading data or transmitting data from the FIFO buffer is enabled.

**Table 21.23 Relationship between Transfer Mode Settings by CNTMD Bit and Timings at which Reading Data or Transmitting Data from FIFO Buffer is Enabled**

<b>Continuous or Non-Continuous Transfer Mode</b>	<b>When Reading Data or Transmitting Data is Enabled</b>
Non-continuous transfer (CNTMD = 0)	<p>In the receiving direction (DIR = 0), reading data from the FIFO buffer is enabled when:</p> <ul style="list-style-type: none"> <li>• This module receives one packet.</li> </ul> <hr/> <p>In the transmitting direction (DIR = 1), transmitting data from the FIFO buffer is enabled when:</p> <ul style="list-style-type: none"> <li>• Software (or DMAC) writes data of the maximum packet size to the FIFO buffer.</li> <li>or</li> <li>• Software (or DMAC) writes data of the short packet size (including 0-byte data) to the FIFO buffer and then writes 1 to BVAL.</li> </ul>
Continuous transfer (CNTMD = 1)	<p>In the receiving direction (DIR = 0), reading data from the FIFO buffer is enabled when:</p> <ul style="list-style-type: none"> <li>• The number of the data bytes received in the FIFO buffer assigned to the selected pipe becomes the same as the number of assigned data bytes (<math>(\text{BUFSIZE} + 1) * 64</math>).</li> <li>• This module receives a short packet other than a zero-length packet.</li> <li>• This module receives a zero-length packet when data is already stored in the FIFO buffer assigned to the selected pipe.</li> <li>or</li> <li>• This module receives the number of packets equal to the transaction counter value specified for the selected pipe by software.</li> </ul> <hr/> <p>In the transmitting direction (DIR = 1), transmitting data from the FIFO buffer is enabled when:</p> <ul style="list-style-type: none"> <li>• The number of the data bytes written to the FIFO buffer by software (or DMAC) becomes the same as the number of data bytes in a single FIFO buffer plane assigned to the selected pipe.</li> <li>or</li> <li>• Software (or DMAC) writes to the FIFO buffer the number of data bytes less than the size of a single FIFO buffer plane (including 0-byte data) assigned to the selected pipe and then writes 1 to BVAL.</li> </ul>

Figure 21.11 shows an example of buffer memory operation for this module.



**Figure 21.11 Example of Buffer Memory Operation**

## (2) FIFO Port Functions

Table 21.24 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 of the data. To enable sending of data before the buffer is full (or before the maximum packet size for non-continuous transfers), the BVAL bit in C/DnFIFOCTR must be set to end the writing. Also, to send a zero-length packet, the BCLR bit in the same register must be used to clear the buffer and then the BVAL bit set in order to end the writing.

In read access, reception of new packets is automatically enabled if all of the data has been read. Data cannot be read when a zero-length packet is being received (DTLN = 0), so the BCLR bit in the register must be used to release the buffer. The length of the data being received can be confirmed using the DTLN bit in C/DnFIFOCTR.

**Table 21.24 FIFO Port Function Settings**

Register Name	Bit Name	Function	Note
C/DnFIFOSEL	RCNT	Selects 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 DnFIFO only
	DREQE	Enables DMA transfers	For DnFIFO only
	MBW	FIFO port access bit width	
	BIGEND	Selects FIFO port endian	
	ISEL	FIFO port access direction	
	CURPIPE	Selects the current pipe	For DCP only
C/DnFIFOCTR	BVAL	Ends writing to the buffer memory	
	BCLR	Clears the buffer memory on the CPU side	
	DTLN	Checks the length of received data	

**(a) FIFO Port Selection**

Table 21.25 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed is selected using the CURPIPE bit in C/DnFIFOSEL. After the pipe is selected, whether the CURPIPE value for the pipe which was written last can be correctly read should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by this module.) Then, the FIFO port can be accessed after FRDY = 1 is checked.

Also, the bus width to be accessed should be selected using the MBW bit. The buffer memory access direction conforms to the DIR bit in PIPEnCFG. The ISEL bit determines this only for the DCP.

**Table 21.25 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 D0FIFO/D1FIFO port register
	DMA access	D0FIFO/D1FIFO port register

## **(b) REW Bit**

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing using the current pipe once again. The REW bit in C/DnFIFOSEL is used for this.

If a pipe is selected when the REW bit is set to 1 and at the same time the CURPIPE bit in C/DnFIFOSEL is set, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte. Also, if a pipe is selected with 0 set for the REW bit, data can be read and written in continuation of the previous selection, without the pointer used for reading from and writing to the buffer memory being reset.

To access the FIFO port, FRDY = 1 must be ensured after selecting a pipe.

## **(3) DMA Transfers (D0FIFO/D1FIFO port)**

### **(a) Overview of DMA Transfers**

For pipes 1 to 9, the FIFO port can be accessed using the DMAC. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

The unit of transfer to the FIFO port should be selected using the MBW bit in DnFIFOSEL and the pipe targeted for the DMA transfer should be selected using the CURPIPE bit. The selected pipe should not be changed during the DMA transfer.

### **(b) Auto Recognition of DMA Transfer Completion**

With this module, it is possible to complete FIFO data writing through DMA transfer by controlling DMA transfer end signal input. When a transfer end signal is sampled, the module enables buffer memory transmission (the same condition as when BVAL = 1).

### **(c) DnFIFO Auto Clear Mode (D0FIFO/D1FIFO Port Reading Direction)**

If 1 is set for the DCLRM bit in DnFIFOSEL, the module automatically clears the buffer memory of the selected pipe when reading of the data from the buffer memory has been completed.

Table 21.26 shows the packet reception and buffer memory clearing processing for each of the various settings. As shown, the buffer clear conditions depend on the value set to the BFRE bit. 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 in the buffer memory reading direction.

**Table 21.26 Packet Reception and Buffer Memory Clearing Processing**

Buffer Status When Packet is Received	Register Setting			
	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

### 21.4.5 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 reading and control writing. The buffer memory can be accessed through the CFIFO port.

#### (1) Control Transfers when the Host Controller Function is Selected

##### (a) Setup Stage

USQREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the SUREQ bit in DCPCTR transmits the specified data for setup transactions. Upon completion of transactions, the SUREQ bit is cleared to 0. The above USB request registers should not be modified while SUREQ = 1. The device address for setup transactions is specified using the DEVSEL bits in DCPMAXP.

When the data for setup transactions has been sent, a SIGN or SACK interrupt request is generated according to the response received from the peripheral device (SIGN1 or SACK bits in INTSTS1), by means of which the result of the setup transactions can be confirmed.

A data packet of DATA0 (USB request) is transmitted as the data packet for the setup transactions regardless of the setting of the SQMON bit in DCPCTR.

**(b) Data Stage**

Data transfers are done using the DCP buffer memory.

The access direction of the DCP buffer memory should be specified using the ISEL bit in CFIFOSEL.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Transaction is done by setting the data PID = DATA1 and the PID bit = BUF using the SQSET bit in DCPCFG. Completion of data transfer is detected using the BRDY and BEMP interrupts.

Setting continuous transfer mode allows data transfers over multiple packets. Note that when continuous transfer mode is set for the receiving direction, the BRDY interrupt is not generated until the buffer becomes full or a short packet is received (the integer multiple of the maximum packet size, and less than 256 bytes).

For control write transfers, when the number of data bytes to be sent is the integer multiple of the maximum packet size, software must control so as to send a zero-length packet at the end.

**(c) Status Stage**

Zero-length packet data transfers are done in the direction opposite to that in the data stage. As with the data stage, data transfers are done using the DCP buffer memory. Transactions are done in the same manner as the data stage.

For the data packets of the status stage, the data PID must be transferred as DATA1. The data PID should be set to DATA1 using the SQSET bit in DCPCFG.

For reception of a zero-length packet, the received data length must be confirmed using the DTLN bits in CFIFOCTR after the BRDY interrupt is generated, and the buffer memory must then be cleared using the BCLR bit.



## **(2) Control Transfers when the Function Controller Function is Selected**

### **(a) Setup Stage**

This module always sends an ACK response in response to a setup packet that is normal with respect to this module. The operation of this module operates in the setup stage is noted below.

- (i) When a new USB request is received, this module sets the following registers:
  - Set the VALID bit in INTSTS0 to 1.
  - Set the PID bit in DCPCTR to NAK.
  - Set the CCPL bit in DCPCTR to 0.
- (ii) When a data packet is received right after the SETUP packet, the USB request parameters are stored 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 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 the 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 transition. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified. For information on the stage control of this module, see figure 21.7.

### **(b) Data Stage**

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 carried out based on the state of the buffer memory.

**(c) Status Stage**

Control transfers are terminated by setting the CCPL bit to 1 with the PID bit 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.

**(i) For control read transfers:**

This module sends a zero-length packet and receives an ACK response from the USB host.

**(ii) 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.

**(d) Control Transfer Auto Response Function**

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 the software is necessary.

**(i) Any transfer other than a control read transfer: bmRequestType  $\neq$  H'00****(ii) If a request error occurs: wIndex  $\neq$  H'00****(ii) For any transfer other than a no-data control transfer: wLength  $\neq$  H'00****(iv) If a request error occurs: wValue  $>$  H'7F****(v) Control transfer of a device state error: DVSQ = 011 (Configured)**

For all requests other than the SET\_ADDRESS request, a response is required from the corresponding software.

### 21.4.6 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.

#### (1) PING Packet Control when the Host Controller Function is Selected

This module automatically sends a PING packet in the OUT direction.

On receiving an ACK handshake in the initial state in which PING packet sending mode is set, this module sends an OUT packet as noted below. Reception of an NAK or NYET handshake returns this module to PING packet sending mode. This control also applies to the control transfers in the data stage and status stage.

1. Sets OUT data sending mode.
  2. Sends a PING packet.
  3. Receives an ACK handshake.
  4. Sends an OUT data packet.
  5. Receives an ACK handshake.
- (Repeats steps 4 and 5.)
6. Sends an OUT data packet.
  7. Receives an NAK/NYET handshake.
  8. Sends a PING packet.

This module is returned to PING packet sending mode by a power-on reset, receiving a NYET/NAK handshake, setting or clearing the sequence toggle bits (SQSET and SQCLR), and setting the buffer clear bit (ACLRM) in PIPEnCTR.

## (2) NYET Handshake Control when the Function Controller Function is Selected

Table 21.27 shows the NYET handshake responses of this module. The NYET response of this module is made in conformance with the conditions noted below. When a short packet is received, however, the response will be an ACK response instead of a NYET packet response. The same applies to the data stages of control write transfers.

**Table 21.27 NYET Handshake Responses**

Value Set for PID Bit in DCPCTR	Buffer Memory State	Token	Response	Note
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY1	OUT/PING	ACK	If an OUT token is received, a data packet is received.
	RCV-BRDY2	OUT	NYET	Notifies whether a data packet can be received
	RCV-BRDY2	OUT (Short)	ACK	Notifies whether a data packet can be received
	RCV-BRDY2	PING	ACK	Notifies that a data packet can be received
	RCV-NRDY	OUT/PING	NAK	Notifies that a data packet cannot be received
	TRN-BRDY	IN	DATA0/DATA1	A data packet is transmitted
	TRN-NRDY	IN	NAK	TRN-NRDY

### [Legend]

- RCV-BRDY1: When an OUT/PING token is received, there is space in the buffer memory for two or more packets.
- RCV-BRDY2: When an OUT token is received, there is only enough space in the buffer memory for one packet.
- RCV-NRDY: When a PING token is received, there is no space in the buffer memory.
- TRN-BRDY: When an IN token is received, there is data to be sent in the buffer memory.
- TRN-NRDY: When an IN token is received, there is no data to be sent in the buffer memory.

### 21.4.7 Interrupt Transfers (PIPE6 to PIPE9)

When the function controller function is selected, 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 carried out without an NYET handshake response being made.

When the host controller function is selected, this module can set the timing of issuing a token using the interval timer. At this time, this module issues an OUT token even in the OUT direction, without issuing a PING token.

This module does not support high bandwidth transfers of interrupt transfers.

#### (1) Interval Counter during Interrupt Transfers when the Host Controller Function is Selected

For interrupt transfers, intervals between transactions are set in the IITV bits in PIPEPERI. This controller issues an interrupt transfer token based on the specified intervals.

##### (a) Counter Initialization

This controller initializes the interval counter under the following conditions.

##### (i) Power-on reset:

The IITV bits are initialized.

##### (ii) Buffer memory initialization using the ACLRM bit:

The IITV bits are not initialized but the count value is. Setting the ACLRM bit to 0 starts counting from the value set in the IITV bits.

Note that the interval counter is not initialized in the following case.

##### (iii) USB bus reset, USB suspended:

The IITV bits are not initialized. Setting 1 to the UACT bit starts counting from the value before entering the USB bus reset state or USB suspended state.

**(b) Operation when Transmission/Reception is Impossible at Token Issuance Timing**

This module cannot issue tokens even at token issuance timing in the following cases. In such a case, this module attempts transactions at the subsequent interval.

- (i) When the PID is set to NAK or STALL.
- (ii) When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- (iii) When there is no data to be sent in the buffer memory at the token sending timing in the sending (OUT) direction.

**21.4.8 Isochronous Transfers (PIPE1 and PIPE2)**

1. This module has the following functions pertaining to isochronous transfers.
2. Notification of isochronous transfer error information
3. Interval counter (specified by the IITV bit)
4. Isochronous IN transfer data setup control (IDLY function)
5. Isochronous IN transfer buffer flush function (specified by the IFIS bit)

This module does not support the High Bandwidth transfers of isochronous transfers.

**(1) Error Detection with Isochronous Transfers**

This module has a function for detecting the error information noted below, so that when errors occur in isochronous transfers, software can control them. Tables 21.28 and 21.29 show the priority in which errors are confirmed and the interrupts that are generated.

- (i) PID errors
  - If the PID of the packet being received is illegal
- (ii) CRC errors and bit stuffing errors
  - If an error occurs in the CRC of the packet being received, or the bit stuffing is illegal
- (iii) Maximum packet size exceeded
  - The maximum packet size exceeded the set value.
- (iv) Overrun and underrun errors
  - When host controller function is selected:
    - When using isochronous IN transfers (reception), the IN token was received but the buffer memory is not empty.

- When using isochronous OUT transfers (transmission), the OUT token was transmitted, but the data was not in the buffer memory.
- When function controller function is selected:
  - When using isochronous IN transfers (transmission), the IN token was received but the data was not in the buffer memory.
  - When using isochronous OUT transfers (reception), the OUT token was received, but the buffer memory was not empty.

(v) Interval errors

- During an isochronous IN transfer, the token could not be received during the interval frame.
- During an isochronous OUT transfer, the OUT token was received during frames other than the interval frame.

**Table 21.28 Error Detection when a Token is Received**

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller function is selected and the function controller function is selected (ignored as a corrupted packet).
2	CRC error and bit stuffing errors	No interrupts generated in both cases when the host controller function is selected and the function controller function is selected (ignored as a corrupted packet).
3	Overflow and underflow errors	<p>An NRDY interrupt is generated to set the OVRN bit in both cases when host controller function is selected and function controller function is selected.</p> <p>When the host controller function is selected, no tokens are transmitted.</p> <p>When the function controller function is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.</p>
4	Interval errors	An NRDY interrupt is generated when the function controller function is selected. It is not generated when the host controller function is selected.

**Table 21.29 Error Detection when a Data Packet is Received**

<b>Detection Priority Order</b>	<b>Error</b>	<b>Generated Interrupt and Status</b>
1	PID errors	No interrupts are generated (ignored as a corrupted packet)
2	CRC error and bit stuffing errors	An NRDY interrupt is generated to set the CRCE bit in both cases when the host controller function is selected and the function controller function is selected.
3	Maximum packet size exceeded error	A BEMP interrupt is generated to set the PID bits to STALL in both cases when the host controller function is selected and the function controller function is selected.

**(2) DATA-PID**

This module does not support High Bandwidth transfers. When the function controller function is selected, this module operates as follows in response to the received PID.

**(a) IN direction**

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mData: Not sent

**(b) OUT direction (when using full-speed operation)**

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mData: Packets are ignored

**(c) OUT direction (when using high-speed operation)**

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Received normally as data packet PID
- mData: Received normally as data packet PID



### (3) Interval Counter

The isochronous interval can be set using the IITV bits in PIPEPERI. The interval counter enables the functions shown in table 21.30 when the function controller function is selected. When the host controller function is selected, this module generates the token issuance timing. When the host controller function is selected, the interval counter operation is the same as the interrupt transfer operation.

**Table 21.30 Functions of the Interval Counter when the Function Controller Function is Selected**

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 not being received	When an OUT token cannot be normally received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the  $2^{\text{IITV}}$  frame or  $2^{\text{IITV}}$   $\mu$  frames.

#### (a) Counter Initialization when the Function Controller Function is Selected

This module initializes the interval counter under the following conditions.

- (i) Power-on reset

The IITV bit is initialized.

- (ii) Buffer memory initialization using the ACLRM bit

The IITV bits are not initialized but the count 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 counter is started under the following conditions 1 or 2 when a packet has been transferred normally.

1. An SOF is received following transmission of data in response to an IN token, in the PID = BUF state.
2. An SOF 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.

3. When the PID bit is set to NAK or STALL

The interval timer does not stop. This module attempts the transactions at the subsequent interval.

4. The USB bus reset or the USB is suspended

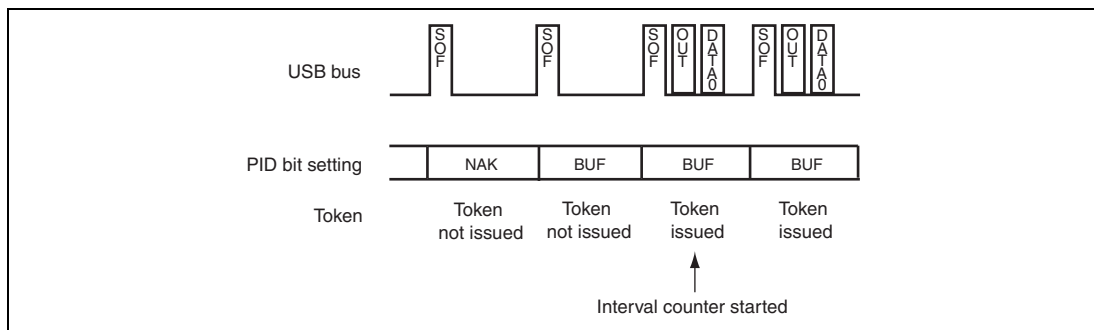
The IITV bit is not initialized. When the SOF has been received, the counter is restarted from the value prior to the reception of the SOF.

**(b) Interval Counting and Transfer Control when the Host Controller Function is Selected**

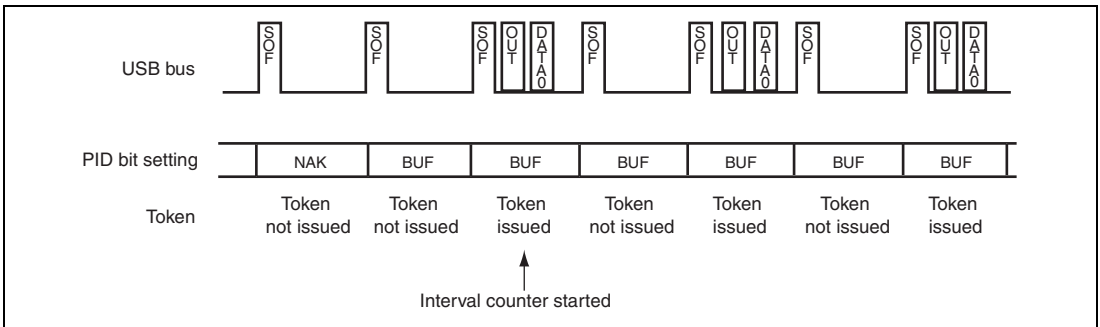
This module controls the interval between token issuance operations based on the IITV bit settings. Specifically, this module issues a token for a selected pipe once every 2IITV ( $\mu$ ) frames.

This module counts the interval every 1-ms frame for the pipes used for communications with the full-speed peripheral devices connected to a high-speed HUB.

This module starts counting the token issuance interval at the ( $\mu$ ) frame following the ( $\mu$ ) frame in which software has set the PID bits to BUF.



**Figure 21.12 Token Issuance when IITV = 0**



**Figure 21.13 Token Issuance when IITV = 1**

When the selected pipe is for isochronous transfers, this module carries out the operation below in addition to controlling token issuance interval. This module issues a token even when the NRDY interrupt generation condition is satisfied.

(i) When the selected pipe is for isochronous IN transfers

This module generates the NRDY interrupt when this module issues the IN token but does not receive a packet successfully from a peripheral device (no response or packet error).

This module sets the OVRN bit to 1 generating the NRDY interrupt when the time to issue an IN token comes in a state in which this module cannot receive data because the FIFO buffer is full (due to the fact that software (DMAC) is too slow to read data from the FIFO buffer),

(ii) When the selected pipe is for isochronous OUT transfers

This module sets the OVRN bit to 1 generating the NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes in a state in which there is no data to be transmitted in the FIFO buffer (because software (DMAC) is too slow to write data to the FIFO buffer).

The token issuance interval is reset on any of the following conditions.

- When a hardware-reset is applied to this module (here, the IITV bits are also cleared to 0).
- When software sets the ACLRM bit to 1.

### (c) Interval Counting and Transfer Control when the Function Controller Function is Selected

#### (i) When the selected pipe is for isochronous OUT transfers

This module generates the NRDY interrupt when this module fails to receive a data packet within the interval set by the IITV bits in terms of ( $\mu$ ) frames.

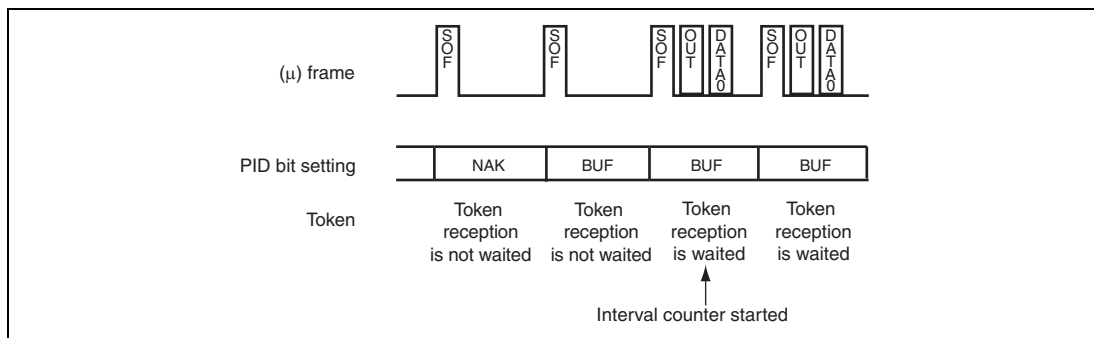
This module generates the NRDY interrupt when this module fails to receive a data packet because of a CRC error or other errors contained in the packet, or because of the FIFO buffer being full.

This module generates the NRDY interrupt on receiving an SOF packet. Even if the SOF packet is corrupted, the internal interpolation is used and allows the interrupt to be generated at the timing to receive the SOF packet.

However, when the IITV bits are set to the value other than 0, this module generates the NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation. When the PID bits are set to NAK by software after starting the interval timer, this module does not generate the NRDY interrupt on receiving an SOF packet.

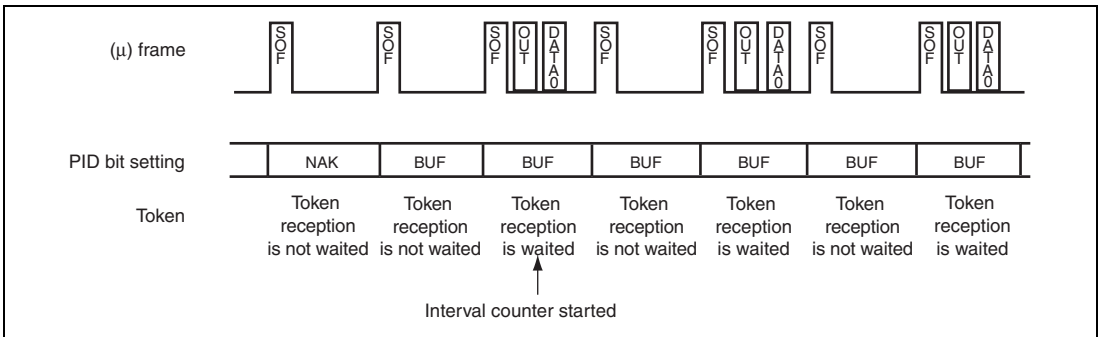
The interval counting starts at the different timing depending on the IITV bit setting as follows.

- When IITV = 0: The interval counting starts at the ( $\mu$ ) frame following the ( $\mu$ ) frame in which software has set the PID bits for the selected pipe to BUF.



**Figure 21.14 Relationship between ( $\mu$ ) Frames and Expected Token Reception when IITV = 0**

- When IITV  $\neq$  0: The interval counting starts on completion of successful reception of the first data packet after the PID bits for the selected pipe have been modified to BUF.



**Figure 21.15 Relationship between (μ) Frames and Expected Token Reception when IITV ≠ 0**

(ii) When the selected pipe is for isochronous IN transfers

The IFIS bit should be 1 for this use. When IFIS = 0, this module transmits a data packet in response to the received IN token irrespective of the IITV bit setting.

When IFIS = 1, this module clears the FIFO buffer when this module fails to receive an IN token within the interval set by the IITV bits in terms of (μ) frames in a state in which there is data to be transmitted in the FIFO buffer.

This module also clears the FIFO buffer when this module fails to receive an IN token successfully because of a bus error such as a CRC error contained in the token.

This module clears the FIFO buffer on receiving an SOF packet. Even if the SOF packet is corrupted, the internal interpolation is used and allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The interval counting starts at the different timing depending on the IITV bit setting (similar to the timing during OUT transfers).

The interval is counted on any of the following conditions in function controller mode.

- When a hardware-reset is applied to this module (here, the IITV bits are also cleared to 0).
- When software sets the ACLRM bit to 1.
- When this module detects a USB reset.

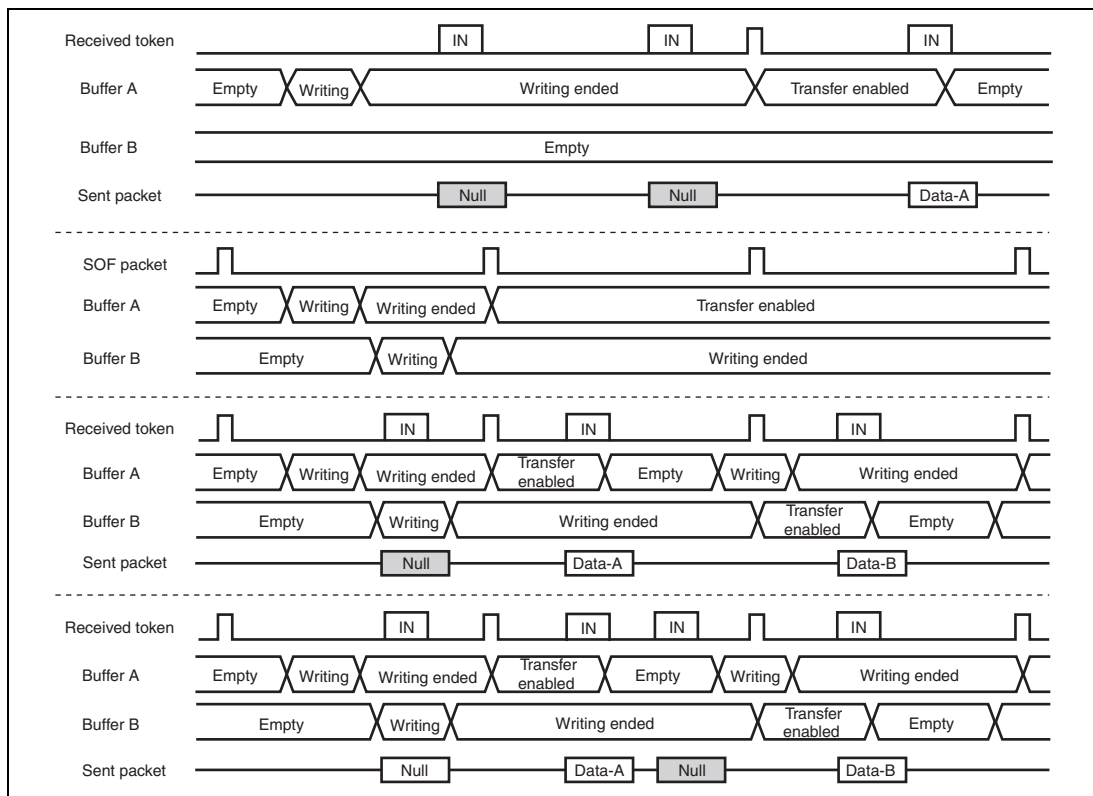
#### **(4) Setup of Data to be Transmitted using Isochronous Transfer when the Function Controller Function is Selected**

With isochronous data transmission using this module in function controller function, 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, that buffer memory being the one to which the data writing was completed first. For this reason, even if multiple IN tokens are received, the only buffer memory that can be sent is 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. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 21.16 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 21.16 Example of Data Setup Function Operation**

#### **(5) Isochronous Transfer Transmission Buffer Flush when the Function Controller Function is Selected**

If an SOF packet or a  $\mu$ SOF packet is received without receiving an IN token in the interval frame during isochronous data transmission, this module operates as if an 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 SOF or  $\mu$ SOF packets reception.

The timing at which the operation of the buffer flush function varies depending on the value set for the IITV bit.

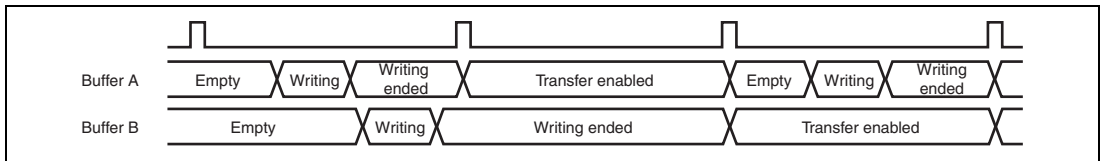
**(a) If IITV = 0**

The buffer flush operation starts from the next frame after the pipe becomes valid.

**(b) In any cases other than IITV = 0**

The buffer flush operation is carried out subsequent to the first normal transaction.

Figure 21.17 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 according to the buffer state.



**Figure 21.17 Example of Buffer Flush Function Operation**

Figure 21.18 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 IN buffer flush function is activated.

If an interval error occurs during an IN transfers, 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.

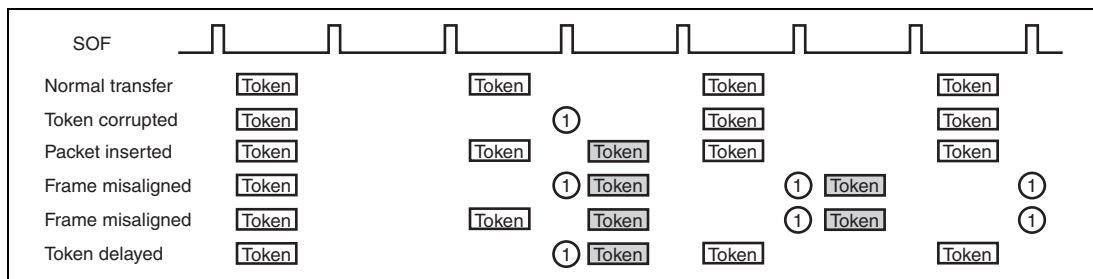
1. IN direction:

- 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.

2. 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 21.18 Example of an Interval Error Being Generated when IITV = 1**

### 21.4.9 SOF Interpolation Function

When the function controller function is selected and if data could not be received at intervals of 1 ms (when using full-speed operation) or 125  $\mu$ s (when using high-speed operation) because an SOF packet was corrupted or missing, this module interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in SYSCFG have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- USB bus reset
- Suspended state detected

Also, the SOF interpolation operates under the following specifications.

- 125  $\mu$ s/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  $\mu$ s or 1 ms is counted with an 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.)

This module supports the following functions based on the SOF detection. These functions also operate normally with SOF interpolation, if the SOF packet was corrupted.

- Refreshing of the frame number and the micro-frame number
- SOFR interrupt timing and  $\mu$ SOF lock
- Isochronous transfer interval count

If an SOF packet is missing when full-speed operation is being used, the FRNM bit in FRMNUM0 is not refreshed.

If a  $\mu$ SOF packet is missing during high-speed operation, the UFRNM bit in FRMNUM1 is refreshed.

However, if a  $\mu$ SOF packet for which the  $\mu$ FRNM = 000 is missing, the FRNM bit is not refreshed. In this case, the FRNM bit is not refreshed even if successive  $\mu$ SOF packets other than  $\mu$ FRNM = 000 are received normally.

## 21.4.10 Pipe Schedule

### (1) Conditions for Generating a Transaction

When the host controller function is selected and UACT has been set to 1, this module generates a transaction under the conditions noted in table 21.31.

**Table 21.31 Conditions for Generating a Transaction**

Transaction	Conditions for Generation				
	DIR	PID	IITV0	Buffer State	SUREQ
Setup	—* <sup>1</sup>	—* <sup>1</sup>	—* <sup>1</sup>	—* <sup>1</sup>	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—* <sup>1</sup>
	OUT	BUF	Invalid	Send data exists	—* <sup>1</sup>
Interrupt transfer	IN	BUF	Valid	Receive area exists	—* <sup>1</sup>
	OUT	BUF	Valid	Send data exists	—* <sup>1</sup>
Isochronous transfer	IN	BUF	Valid	* <sup>2</sup>	—* <sup>1</sup>
	OUT	BUF	Valid	* <sup>3</sup>	—* <sup>1</sup>

- Notes:
1. Symbols (—) in the table indicate that the condition is one that is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, the condition is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that the condition is generated regardless of the interval counter.
  2. This indicates that a transaction is generated regardless of whether or not there is a receive area. If there was no receive area, however, the received data is destroyed.
  3. This indicates that a transaction is generated regardless of whether or not there is any data to be sent. If there was no data to be sent, however, a zero-length packet is sent.

## (2) Transfer Schedule

This section describes the transfer scheduling within a frame of this module. After the module sends an SOF, the transfer is carried out in the sequence described below.

### (a) Execution of periodic transfers

A pipe is searched in the order of Pipe 1 → Pipe 2 → Pipe 6 → Pipe 7 → Pipe 8 → Pipe 9, and then, if the pipe is one for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.

### (b) Setup transactions for control transfers

The DCP is checked, and if a setup transaction is possible, it is sent.

### (c) Execution of bulk and control transfer data stages and status stages

A pipe is searched in the order of DCP → Pipe 1 → Pipe 2 → Pipe 3 → Pipe 4 → Pipe 5, and then, if the pipe is one for which a bulk or control transfer data stage or a control transfer status stage transaction can be generated, the transaction is generated.

If a transfer is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. Also, if there is time for the transfer to be done within the frame, step 3 is repeated.

## (3) USB Communication Enabled

Setting the UACT bit of the DVSTCTR register to 1 initiates sending of an SOF or  $\mu$ SOF, and makes it possible to generate a transaction.

Setting the UACT bit to 0 stops the sending of the SOF or  $\mu$ SOF and initiates a suspend state. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF or  $\mu$ SOF is sent.

## 21.5 Usage Notes

### 21.5.1 USB Startup and Stop Procedures

#### (1) Startup Procedure

The USB module should be started up in the following procedure.

1. In refresh standby mode, generate an IRQn or NMI interrupt, write 0 to the STBY bit in STBCR in power-down mode, and return to the normal operation.
2. Wait for 3 to 5 ms to secure the oscillation stabilization time.
3. Write 0 to the USB bit of MSTPCR0 in the CPG and write 1 to the SCKE bit in SYSCFG to cancel module standby state.

#### (2) Stop Procedure

The USB module should be stopped in the following procedure.

1. Write 0 to the SCKE bit in SYSCFG and write 1 to the USB bit of MSTPCR0 in the CPG to place the USB module in module standby state.
2. Wait for more than 40 ns (2 cycles in 48-MHz clock).
3. Write 1 to the STBY bit in STBCR to make a transition from power-down mode to refresh standby mode.

**Note:** The USB bit in MSTPCR0 should always be set in combination with the SCKE bit in SYSCFG.

## Section 22 LCD Controller (LCDC)

A unified memory architecture is adopted for the LCD controller (LCDC) so that the image data for display is stored in system memory. The LCDC module reads data from system memory, uses the palette memory to determine the colors, then puts the display on the LCD panel. It is possible to connect the LCDC to the LCD module\* other than microcomputer bus interface types and NTSC/PAL types and those that apply the LVDS interface.

Note: \* LCD module can be connected to the LVDS interface by using the LSI with LVDS conversion LSI.

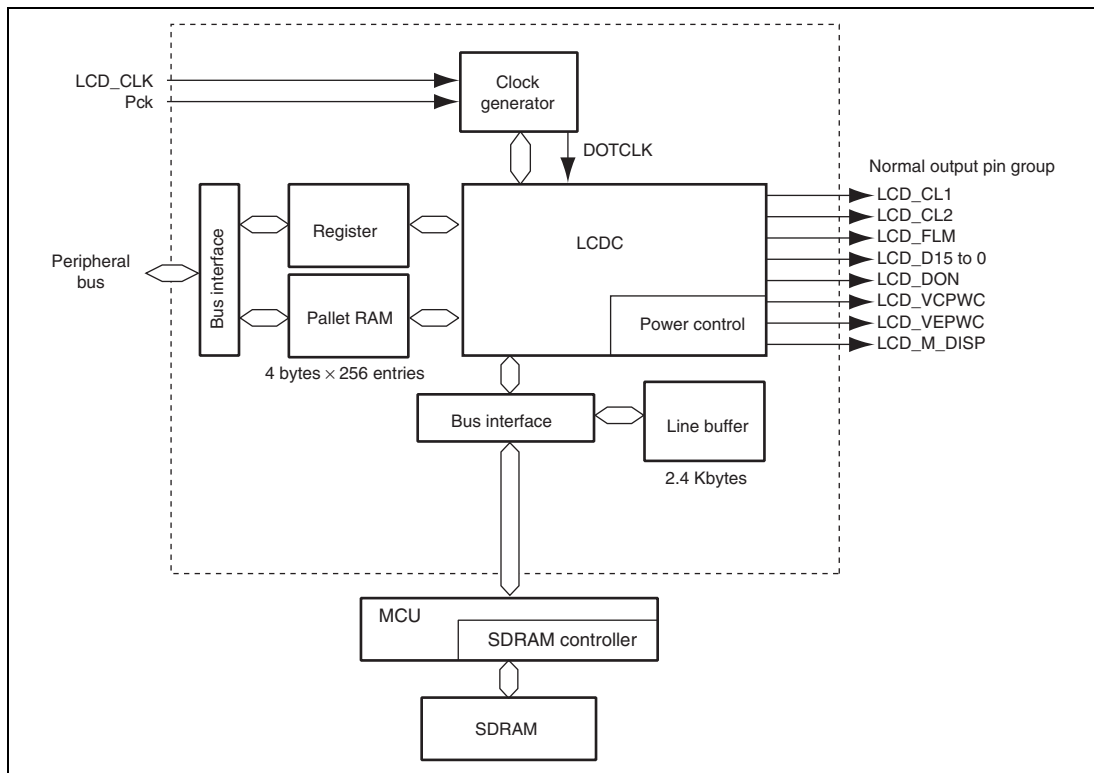
### 22.1 Features

The LCDC has the following features.

- Panel interface
  - Serial interface method
  - Supports data formats for STN/dual-STN/TFT panels (8/12/16/18-bit bus width)\*<sup>1</sup>
- Supports 4/8/15/16-bpp (bits per pixel) color modes
- Supports 1/2/4/6-bpp grayscale modes
- Supports LCD-panel sizes from  $16 \times 1$  to  $1024 \times 1024$ \*<sup>2</sup>
- 24-bit color palette memory (16 of the 24 bits are valid; R:5/G:6/B:5)
- STN/DSTN panels are prone to flicker and shadowing. The controller applies 65536-color control by 24-bit space-modulation FRC with 8-bit RGB values for reduced flicker.
- Dedicated display memory is unnecessary using part of the SDRAM (area 1, 2) as the VRAM to store display data of the LCDC.
- The display is stable because of the large 2.4-kbyte line buffer
- Supports the inversion of the output signal to suit the LCD panel's signal polarity
- Supports the selection of data formats (the endian setting for bytes, backed pixel method) by register settings
- An interrupt can be generated at the user specified position (controlling the timing of VRAM update start prevents flicker)

- Notes: 1. When connecting the LCDC to a TFT panel with an unwired 18-bit bus, the lower bit lines should be connected to GND or to the lowest bit from which data is output.
2. For details, see section 22.4.1, LCD Module Sizes which can be Displayed in this LCDC.

Figure 22.1 shows a block diagram of LCDC.



**Figure 22.1 LCDC Block Diagram**

## 22.2 Input/Output Pins

Table 22.1 summarizes the LCDC's pin configuration.

**Table 22.1 Pin Configuration**

Pin Name	I/O	Function
LCD_D15 to 0	Output	Data for LCD panel
LCD_DON	Output	Display-on signal (DON)
LCD_CL1	Output	Shift-clock 1 (STN/DSTN)/horizontal sync signal (HSYNC)
LCD_CL2	Output	Shift-clock 2 (STN/DSTN)/dot clock (DOTCLK)
LCD_M_DISP	Output	LCD current-alternating signal/DISP signal
LCD_FLM	Output	First line marker/vertical sync signal (VSYNC) (TFT)
LCD_VCPWC	Output	LCD-module power control (VCC)
LCD_VEPWC	Output	LCD-module power control (VEE)
LCD_CLK	Input	LCD clock-source input

Note: Check the LCD module specifications carefully in section 22.5, Clock and LCD Data Signal Examples, before deciding on the wiring specifications for the LCD module.

## 22.3 Register Configuration

Table 22.2 shows the LCDC register configuration. Table 22.3 shows the register status in each operating mode.

**Table 22.2 Register Configuration**

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
Palette data register 00 to FF	LDPR00 to LDPRFF	R/W	H'FFE3 0000 to H'FFE3 03FC	H'1FE3 0000 to H'1FE3 03FC	32
LCDC input clock register	LDICKR	R/W	H'FFE3 0400	H'1FE3 0400	16
LCDC module type register	LDMTR	R/W	H'FFE3 0402	H'1FE3 0402	16
LCDC data format register	LDDFR	R/W	H'FFE3 0404	H'1FE3 0404	16
LCDC data fetch start address register for upper display panel	LDSARU	R/W	H'FFE3 0408	H'1FE3 0408	32
LCDC data fetch start address register for lower display panel	LDSARL	R/W	H'FFE3 040C	H'1FE3 040C	32
LCDC fetch data line address offset register for display panel	LDLAOR	R/W	H'FFE3 0410	H'1FE3 0410	16
LCDC palette control register	LDPALCR	R/W	H'FFE3 0412	H'1FE3 0412	16
LCDC horizontal character number register	LDHCNR	R/W	H'FFE3 0414	H'1FE3 0414	16
LCDC horizontal synchronization signal register	LDHSYNR	R/W	H'FFE3 0416	H'1FE3 0416	16
LCDC vertical displayed line number register	LDVDLNR	R/W	H'FFE3 0418	H'1FE3 0418	16
LCDC vertical total line number register	LDVTLNR	R/W	H'FFE3 041A	H'1FE3 041A	16
LCDC vertical synchronization signal register	LDVSYNR	R/W	H'FFE3 041C	H'1FE3 041C	16
LCDC AC modulation signal toggle line number register	LDACLNR	R/W	H'FFE3 041E	H'1FE3 041E	16
LCDC interrupt control register	LDINTR	R/W	H'FFE3 0420	H'1FE3 0420	16
LCDC power management mode register	LDPMMR	R/W	H'FFE3 0424	H'1FE3 0424	16
LCDC power supply sequence period register	LDPSPR	R/W	H'FFE3 0426	H'1FE3 0426	16



Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
LCDC control register	LDCNTR	R/W	H'FFE3 0428	H'1FE3 0428	16
LCDC user specified interrupt control register	LDUINTR	R/W	H'FFE3 0434	H'1FE3 0434	16
LCDC user specified interrupt line number register	LDUINTLNR	R/W	H'FFE3 0436	H'1FE3 0436	16
LCDC memory access interval number register	LDLIRNR	R/W	H'FFE3 0440	H'1FE3 0440	16

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 22.3 Register State in Each Operating Mode**

Register Name	Abbreviation	Power-On Reset	Sleep	Standby
Palette data register 00 to FF	LDPR00 to LDPRFF	Undefined	Retained	Retained
LCDC input clock register	LDICKR	H'1101	Retained	Retained
LCDC module type register	LDMTR	H'0109	Retained	Retained
LCDC data format register	LDDFR	H'000C	Retained	Retained
LCDC data fetch start address register for upper display panel	LDSARU	H'04000000	Retained	Retained
LCDC data fetch start address register for lower display panel	LDSARL	H'04000000	Retained	Retained
LCDC fetch data line address offset register for display panel	LDLAOR	H'0280	Retained	Retained
LCDC palette control register	LDPALCR	H'0000	Retained	Retained
LCDC horizontal character number register	LDHCNR	H'4F52	Retained	Retained
LCDC horizontal synchronization signal register	LDHSYNR	H'0050	Retained	Retained
LCDC vertical displayed line number register	LDVDLNR	H'01DF	Retained	Retained
LCDC vertical total line number register	LDVTLNR	H'01DF	Retained	Retained

Register Name	Abbreviation	Power-On Reset	Sleep	Standby
LCDC vertical synchronization signal register	LDVSYNR	H'01DF	Retained	Retained
LCDC AC modulation signal toggle line number register	LDACLNR	H'000C	Retained	Retained
LCDC interrupt control register	LDINTR	H'0000	Retained	Retained
LCDC power management mode register	LDPMMR	H'0010	Retained	Retained
LCDC power supply sequence period register	LDPSPR	H'F60F	Retained	Retained
LCDC control register	LDCNTR	H'0000	Retained	Retained
LCDC user specified interrupt control register	LDUINTR	H'0000	Retained	Retained
LCDC user specified interrupt line number register	LDUINTLNR	H'004F	Retained	Retained
LCDC memory access interval number register	LDLIRNR	H'0000	Retained	Retained

### 22.3.1 LCDC Input Clock Register (LDICKR)

This LCDC can select bus clock, the peripheral clock, or the external clock as its operation clock source. The selected clock source can be divided using an internal divider into a clock of 1/1 to 1/32 and be used as the LCDC operating clock (DOTCLK). The clock output from the LCDC is used to generate the synchronous clock output (LCD\_CL2) for the LCD panel from the operating clock selected in this register. For a TFT panel, LCD\_CL2 = DOTCLK, and for an STN or DSTN panel, LCD\_CL2 = a clock with a frequency of (DOTCLK/data bus width of output to LCD panel). The LDICKR must be set so that the clock input to the LCDC is less than the peripheral clock regardless of the LCD\_CL2.

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

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	ICKSEL[1:0]	00	R/W	Input Clock Select Set the clock source for DOTCLK. 00: Setting prohibited 01: Peripheral clock (Pck) is selected 10: External clock (LCD_CLK) is selected 11: Setting prohibited
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	DCDR[5:0]	000001	R/W	Clock Division Ratio Set the input clock division ratio. For details on the setting, refer to table 22.4.

**Table 22.4 I/O Clock Frequency and Clock Division Ratio**

<b>DCCR[5:0]</b>	<b>Clock Division Ratio</b>	<b>I/O Clock Frequency (MHz)</b>	
		<b>50.000</b>	<b>54.000</b>
000001	1/1	50.000	54.000
000010	1/2	25.000	27.000
000011	1/3	16.667	18.000
000100	1/4	12.500	13.500
000110	1/6	8.333	9.000
001000	1/8	6.250	6.750
001100	1/12	4.167	4.500
010000	1/16	3.125	3.375
011000	1/24	2.083	2.250
100000	1/32	1.563	1.688

Note: Any setting other than above is handled as a clock division ratio of 1/1 (initial value).

### 22.3.2 LCDC Module Type Register (LDMTR)

LDMTR sets the control signals output from this LCDC and the polarity of the data signals, according to the polarity of the signals for the LCD module connected to the LCDC.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLM POL	CL1 POL	DISP POL	DPOL	—	MCNT	CL1CNT	CL2CNT	—	—	MIFTYP[5:0]					
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	FLMPOL	0	R/W	FLM (Vertical Sync Signal) Polarity Select Selects the polarity of the LCD_FLM (vertical sync signal, first line marker) for the LCD module. 0: LCD_FLM pulse is high active 1: LCD_FLM pulse is low active
14	CL1POL	0	R/W	CL1 (Horizontal Sync Signal) Polarity Select Selects the polarity of the LCD_CL1 (horizontal sync signal) for the LCD module. 0: LCD_CL1 pulse is high active 1: LCD_CL1 pulse is low active
13	DISPPOL	0	R/W	DISP (Display Enable) Polarity Select Selects the polarity of the LCD_M_DISP (display enable) for the LCD module. 0: LCD_M_DISP is high active 1: LCD_M_DISP is low active

Bit	Bit Name	Initial Value	R/W	Description
12	DPOL	0	R/W	<p>Display Data Polarity Select</p> <p>Selects the polarity of the LCD_D (display data) for the LCD module. This bit supports inversion of the LCD module.</p> <p>0: LCD_D is high active, transparent-type LCD panel</p> <p>1: LCD_D is low active, reflective-type LCD panel</p>
11	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
10	MCNT	0	R/W	<p>M Signal Control</p> <p>Sets whether or not to output the LCD's current-alternating signal of the LCD module.</p> <p>0: M (AC line modulation) signal is output</p> <p>1: M signal is not output</p>
9	CL1CNT	0	R/W	<p>CL1 (Horizontal Sync Signal) Control</p> <p>Sets whether or not to enable CL1 output during the vertical retrace period.</p> <p>0: CL1 is output during vertical retrace period</p> <p>1: CL1 is not output during vertical retrace period</p>
8	CL2CNT	1	R/W	<p>CL2 (Dot Clock of LCD Module) Control</p> <p>Sets whether or not to enable CL2 output during the vertical and horizontal retrace period.</p> <p>0: CL2 is output during vertical and horizontal retrace period</p> <p>1: CL2 is not output during vertical and horizontal retrace period</p>
7, 6	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	MIFTYP [5:0]	001001	R/W	<p>Module Interface Type Select</p> <p>Set the LCD panel type and data bus width to be output to the LCD panel. There are three LCD panel types: STN, DSTN, and TFT. There are four data bus widths for output to the LCD panel: 4, 8, 12, and 16 bits. When the required data bus width for a TFT panel is 16 bits or more, connect the LCDC and LCD panel according to the data bus size of the LCD panel. Unlike in a TFT panel, in an STN or DSTN panel, the data bus width setting does not have a 1:1 correspondence with the number of display colors and display resolution, e.g., an 8-bit data bus can be used for 16 bpp, and a 12-bit data bus can be used for 4 bpp. This is because the number of display colors in an STN or DSTN panel is determined by how data is placed on the bus, and not by the number of bits. For data specifications for an STN or DSTN panel, see the specifications of the LCD panel used. The output data bus width should be set according to the mechanical interface specifications of the LCD panel.</p> <p>If an STN or DSTN panel is selected, display control is performed using a 24-bit space-modulation FRC consisting of the 8-bit R, G, and B included in the LCDC, regardless of the color and gradation settings. Accordingly, the color and gradation specified by DSPCOLOR is selected from 16 million colors in an STN or DSTN panel. If a palette is used, the color specified in the palette is displayed.</p> <p>000000: STN monochrome 4-bit data bus module  000001: STN monochrome 8-bit data bus module  001000: STN color 4-bit data bus module  001001: STN color 8-bit data bus module  001010: STN color 12-bit data bus module  001011: STN color 16-bit data bus module  010001: DSTN monochrome 8-bit data bus module  010011: DSTN monochrome 16-bit data bus module  011001: DSTN color 8-bit data bus module  011010: DSTN color 12-bit data bus module  011011: DSTN color 16-bit data bus module  101011: TFT color 16-bit data bus module</p> <p>Settings other than above: Setting prohibited</p>

### 22.3.3 LCDC Data Format Register (LDDFR)

LDDFR sets the bit alignment for pixel data in one byte and selects the data type and number of colors used for display so as to match the display driver software specifications.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PABD	—	DSPCOLOR[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	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
15 to 9	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
8	PABD	0	R/W	Byte Data Pixel Alignment  Sets the pixel data alignment type in one byte of data. The contents of aligned data per pixel are the same regardless of this bit's setting. For example, data H'05 should be expressed as B'0101 which is the normal style handled by a MOV instruction of the this CPU, and should not be selected between B'0101 and B'1010.  0: Big endian for byte data 1: Little endian for byte data
7	—	0	R	Reserved  This bit is always read as 0. The write value should always be 0.



Bit	Bit Name	Initial Value	R/W	Description
6 to 0	DSPCOLOR [6:0]	0001100	R/W	<p>Display Color Select</p> <p>Set the number of display colors for the display (0 is written to upper bits of 4 to 6 bpp). For display colors to which the description (via palette) is added below, the color set by the color palette is actually selected by the display data and displayed.</p> <p>The number of colors that can be supported in rotation mode is restricted by the display resolution.</p> <p>0000000: Monochrome, 2 grayscales, 1 bpp (via palette)</p> <p>0000001: Monochrome, 4 grayscales, 2 bpp (via palette)</p> <p>0000010: Monochrome, 16 grayscales, 4 bpp (via palette)</p> <p>0000100: Monochrome, 64 grayscales, 6 bpp (via palette)</p> <p>0001010: Color, 16 colors, 4 bpp (via palette)</p> <p>0001100: Color, 256 colors, 8 bpp (via palette)</p> <p>0011101: Color, 32K colors (RGB: 555), 15 bpp</p> <p>0101101: Color, 64K colors (RGB: 565), 16 bpp</p> <p>Settings other than above: Setting prohibited</p>

### 22.3.4 LCDC Start Address Register for Upper Display Data Fetch (LDSARU)

LDSARU sets the start address from which data is fetched by the LCDC for display of the LCDC panel. When a DSTN panel is used, this register specifies the fetch start address for the upper side of the panel.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	SAU[27:16]											
Initial value:	0	0	0	0	0	1	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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAU[15:4]												—	—	—	—
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 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	SAU[27]	0	R/W	Start Address for Upper Display Data Fetch
26	SAU[26]	1	R/W	The start address for data fetch of the display data must be set within the SDRAM area of area 1 or 2.
25 to 4	SAU[25:4]	All 0	R/W	
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: The minimum alignment unit of LDSARU is 512 bytes. Write 0 to the lower nine bits.

### 22.3.5 LCDC Start Address Register for Lower Display Data Fetch (LDSARL)

When a DSTN panel is used, LDSARL specifies the fetch start address for the lower side of the panel.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	SAL[27:16]											
Initial value:	0	0	0	0	0	1	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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAL[15:4]												—	—	—	—
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 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	SAL[27]	0	R/W	Start Address for Lower Panel Display Data Fetch The start address for data fetch of the display data must be set within the SDRAM area of area 1 or 2.
26	SAL[26]	1	R/W	STN and TFT: Cannot be used
25 to 4	SAL[25:4]	All 0	R/W	DSTN: Start address for fetching display data corresponding to the lower panel
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: The minimum alignment unit of LDSARU is 32 bytes. Write 0 to the lower five bits.

### 22.3.6 LCDC Line Address Offset Register for Display Data Fetch (LDLAOR)

LDLAOR sets the address width of the Y-coordinates increment used for LCDC to read the image recognized by the graphics driver. This register specifies how many bytes the address from which data is to be read should be moved when the Y coordinates have been incremented by 1. This register does not have to be equal to the horizontal width of the LCD panel. When the memory address of a point (X, Y) in the two-dimensional image is calculated by  $Ax + By + C$ , this register becomes equal to B in this equation.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	LAO [15:0]	H'0280	R/W	<p>Line Address Offset</p> <p>The minimum alignment unit of LDLAOR is 32 bytes. Because the LCDC handles these values as 32-byte data, the values written to the lower five bits of the register are always treated as 0. The lower five bits of the register are always read as 0. The initial values (<math>\times</math> resolution = 640) will continuously and accurately place the VGA (<math>640 \times 480</math> dots) display data without skipping an address between lines.</p> <p>A binary exponential at least as large as the horizontal width of the image is recommended for the LDLAOR value, taking into consideration the software operation speed.</p>

### 22.3.7 LCDC Palette Control Register (LDPALCR)

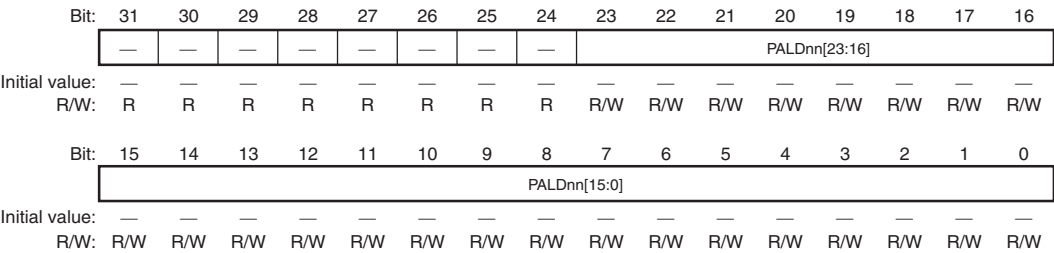
LDPALCR selects whether the CPU or LCDC accesses the palette memory. When the palette memory is being used for display operation, display mode should be selected. When the palette memory is being written to, color-palette setting mode should be selected.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PALS	—	—	—	PALEN
Initial value:	0	0	0	0	0	0	0	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 5	—	All 0	R	Reserved These bits always read as 0. The write value should always be 0.
4	PALS	0	R	Palette State Indicates the access right state of the palette. 0: Normal display mode: LCDC uses the palette 1: Color-palette setting mode: The host (CPU) uses the palette
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PALEN	0	R/W	Palette Read/Write Enable Requests the access right to the palette. 0: Request for transition to normal display mode 1: Request for transition to color palette setting mode

22.3.8 Palette Data Registers 00 to FF (LDPR00 to LDPRFF)

LDPR registers are for accessing palette data directly allocated (4 bytes x 256 addresses) to the memory space. To access the palette memory, access the corresponding register among this register group (LDPR00 to LDPRFF). Each palette register is a 32-bit register including three 8-bit areas for R, G, and B. For details on the color palette specifications, see section 22.4.2, Color Palette Specification.



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved
23 to 0	PALDnn[23:0]	—	R/W	Palette Data
				Bits 18 to 16, 9, 8, and 2 to 0 are reserved within each RGB palette and cannot be set. However, these bits can be extended according to the upper bits.

Note: nn = H'00 to H'FF

### 22.3.9 LCDC Horizontal Character Number Register (LDHCNR)

LDHCNR specifies the LCD module's horizontal size (in the scan direction) and the entire scan width including the horizontal retrace period.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HDCN[7:0]								HTCN[7:0]							
Initial value:	0	1	0	0	1	1	1	1	0	1	0	1	0	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	HDCN [7:0]	01001111	R/W	<p>Horizontal Display Character Number</p> <p>Set the number of horizontal display characters (unit: character = 8 dots).</p> <p>Specify to the value of (the number of display characters) -1.</p> <p>Example: For a LCD module with a width of 640 pixels. HDCN = (640/8) -1 = 79 = H'4F</p>
7 to 0	HTCN [7:0]	01010010	R/W	<p>Horizontal Total Character Number</p> <p>Set the number of total horizontal characters (unit: character = 8 dots).</p> <p>Specify to the value of (the number of total characters) - 1.</p> <p>However, the minimum horizontal retrace period is three characters (24 dots).</p> <p>Example: For a LCD module with a width of 640 pixels. HTCN = [(640/8)-1] +3 = 82 = H'52 In this case, the number of total horizontal dots is 664 dots and the horizontal retrace period is 24 dots.</p>

- Notes:
- The values set in HDCN and HTCN must satisfy the relationship of  $HTCN \geq HDCN$ . Also, the total number of characters of HTCN must be an even number. (The set value will be an odd number, as it is one less than the actual number.)
  - Set HDCN according to the display resolution as follows:
    - 1 bpp: (multiplex of 16) - 1 [1 line is multiplex of 128 pixel]
    - 2 bpp: (multiplex of 8) - 1 [1 line is multiplex of 64 pixel]
    - 4 bpp: (multiplex of 4) - 1 [1 line is multiplex of 32 pixel]
    - 6 bpp/8 bpp: (multiplex of 2) - 1 [1 line is multiplex of 16 pixel]

### 22.3.10 LDCD Horizontal Sync Signal Register (LDHSYNR)

LDHSYNR specifies the timing of the generation of the horizontal (scan direction) sync signals for the LCD module.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	HSYNW [3:0]	0000	R/W	<p>Horizontal Sync Signal Width</p> <p>Set the width of the horizontal sync signals (CL1 and Hsync) (unit: character = 8 dots).</p> <p>Specify to the value of (the number of horizontal sync signal width) -1.</p> <p>Example: For a horizontal sync signal width of 8 dots.  <math>HSYNW = (8 \text{ dots} / 8 \text{ dots/character}) - 1 = 0 = H'0</math></p>
11 to 8	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
7 to 0	HSYNP [7:0]	01010000	R/W	<p>Horizontal Sync Signal Output Position</p> <p>Set the output position of the horizontal sync signals (unit: character = 8 dots).</p> <p>Specify to the value of (the number of horizontal sync signal output position) -1.</p> <p>Example: For a LCD module with a width of 640 pixels.  <math>HSYNP = [(640/8) + 1] - 1 = 80 = H'50</math>            In this case, the horizontal sync signal is active from the 648th through the 655th dot.</p>

Note: The following conditions must be satisfied:

$$HTCN \geq HSYNP + HSYNW + 1$$

$$HSYNP \geq HDCN + 1$$



### 22.3.11 LCDC Vertical Display Line Number Register (LDVDLNR)

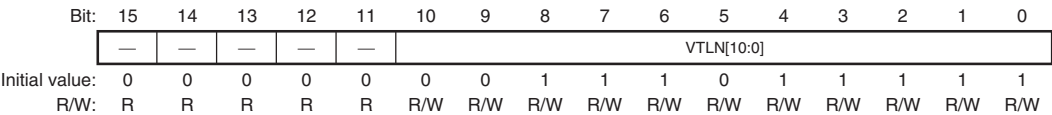
LDVDLNR specifies the LCD module's vertical size (for both scan direction and vertical direction). For a DSTN panel, specify an even number at least as large as the LCD panel's vertical size regardless of the size of the upper and lower panels, e.g. 480 for a 640 x 480 panel.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	VDLN[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
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 to 0	VDLN[10:0]	00111011111	R/W	Vertical Display Line Number  Set the number of vertical display lines (unit: line). Specify to the value of (the number of display line) -1.  Example: For an 480-line LCD module VDLN = 480-1 = 479 = H'1DF

22.3.12 LCDC Vertical Total Line Number Register (LDVTLNR)

LDVTLNR specifies the LCD panel's entire vertical size including the vertical retrace period.



Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
10 to 0	VTLN[10:0]	0011101111	R/W	Vertical Total Line Number  Set the total number of vertical display lines (unit: line).  Specify to the value of (the number of total line) -1.  The minimum for the total number of vertical lines is 2 lines. The following conditions must be satisfied: VTLN>=VDLN, VTLN>=1.  Example: For an 480-line LCD module and a vertical period of 0 lines. VTLN = (480+0) –1 = 479 = H'1DF

### 22.3.13 LCDC Vertical Sync Signal Register (LDVSYNR)

LDVSYNR specifies the vertical (scan direction and vertical direction) sync signal timing of the LCD module.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	VSYNW[3:0]	0000	R/W	<p>Vertical Sync Signal Width</p> <p>Set the width of the vertical sync signals (FLM and Vsync) (unit: line).</p> <p>Specify to the value of (the vertical sync signal width) -1.</p> <p>Example: For a vertical sync signal width of 1 line.  <math>VSYNW = (1-1) = 0 = H'0</math></p>
11	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
10 to 0	VSYNP[10:0]	0011101111	R/W	<p>Vertical Sync Signal Output Position</p> <p>Set the output position of the vertical sync signals (FLM and Vsync) (unit: line).</p> <p>Specify to the value of (the number of vertical sync signal output position) -2.</p> <p>DSTN should be set to an odd number value. It is handled as (setting value+1)/2.</p> <p>Example: For an 480-line LCD module and a vertical retrace period of 0 lines (in other words, VTLN=479 and the vertical sync signal is active for the first line):</p> <ul style="list-style-type: none"> <li>Single display <math display="block">VSYNP = [(1-1)+VTLN] \bmod (VTLN+1)</math> <math display="block">= [(1-1)+479] \bmod (479+1)</math> <math display="block">= 479 \bmod 480 = 479 = H'1DF</math> </li> <li>Dual displays <math display="block">VSYNP = [(1-1) \times 2 + VTLN] \bmod (VTLN+1)</math> <math display="block">= [(1-1) \times 2 + 479] \bmod (479+1)</math> <math display="block">= 479 \bmod 480 = 479 = H'1DF</math> </li> </ul>

22.3.14 LCDC AC Modulation Signal Toggle Line Number Register (LDACLNR)

LDACLNR specifies the timing to toggle the AC modulation signal (LCD current-alternating signal) of the LCD module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	ACLN[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	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	R/W	Description
15 to 5	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
4 to 0	ACLN[4:0]	01100	R/W	AC Line Number  Set the number of lines where the LCD current-alternating signal of the LCD module is toggled (unit: line).  Specify to the value of (the number of toggle line) - 1.  Example: For toggling every 13 lines. ACLN = 13-1 = 12= H'0C

Note: When the total line number of the LCD panel is even, set an even number so that toggling is performed at an odd line.

### 22.3.15 LCDC Interrupt Control Register (LDINTR)

LDINTR specifies where to control the Vsync interrupt of the LCD module. See also section 22.3.19, LCDC User Specified Interrupt Control Register (LDUINTR) and section 22.3.20, LCDC User Specified Interrupt Line Number Register (LDUINTLNR) for interrupts. Note that operations by this register setting and LCDC user specified interrupt control register (LDUINTR) setting are independent.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MINT EN	FINT EN	VSINT EN	VEINT EN	MINTS	FINTS	VSINTS	VEINTS	—	—	—	—	—	—	—	—
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	Bit Name	Initial Value	R/W	Description
15	MINTEN	0	R/W	<p><b>Memory Access Interrupt Enable</b></p> <p>Enables or disables an interrupt generation at the start point of each vertical retrace line period for VRAM access by LCDC.</p> <p>0: Disables an interrupt generation at the start point of each vertical retrace line period for VRAM access</p> <p>1: Enables an interrupt generation at the start point of each vertical retrace line period for VRAM access</p>
14	FINTEN	0	R/W	<p><b>Frame End Interrupt Enable</b></p> <p>Enables or disables the generation of an interrupt after the last pixel of a frame is output to LDC panel.</p> <p>0: Disables an interrupt generation when the last pixel of the frame is output</p> <p>1: Enables an interrupt generation when the last pixel of the frame is output</p>
13	VSINTEN	0	R/W	<p><b>Vsync Starting Point Interrupt Enable</b></p> <p>Enables or disables the generation of an interrupt at the start point of LCDC's Vsync.</p> <p>0: Interrupt at the start point of the Vsync is disabled</p> <p>1: Interrupt at the start point of the Vsync is enabled</p>

Bit	Bit Name	Initial Value	R/W	Description
12	VEINTEN	0	R/W	<p>Vsync Ending Point Interrupt Enable</p> <p>Enables or disables the generation of an interrupt at the end point of LCDC's Vsync.</p> <p>0: Interrupt at the end point of the Vsync signal is disabled</p> <p>1: Interrupt at the end point of the Vsync signal is enabled</p>
11	MINTS	0	R/W	<p>Memory Access Interrupt State</p> <p>Indicates the memory access interrupt handling state.</p> <p>This bit indicates 1 when the LCDC memory access interrupt is generated (set state). During the memory access interrupt handling routine, this bit should be cleared by writing 0.</p> <p>0: LCDC did not generate a memory access interrupt or has been informed that the generated memory access interrupt has completed</p> <p>1: LCDC has generated a memory access end interrupt and not yet been informed that the generated memory access interrupt has completed</p>
10	FINTS	0	R/W	<p>Flame End Interrupt State</p> <p>Indicates the flame end interrupt handling state.</p> <p>This bit indicates 1 at the time when the LCDC flame end interrupt is generated (set state). During the flame end interrupt handling routine, this bit should be cleared by writing 0.</p> <p>0: LCDC did not generate a flame end interrupt or has been informed that the generated flame end interrupt has completed</p> <p>1: LCDC has generated a flame end interrupt and not yet been informed that the generated flame end interrupt has completed</p>

Bit	Bit Name	Initial Value	R/W	Description
9	VSINTS	0	R/W	<p>Vsync Start Interrupt State</p> <p>Indicates the LCDC's Vsync start interrupt handling state. This bit is set to 1 at the time a Vsync start interrupt is generated. During the Vsync start interrupt handling routine, this bit should be cleared by writing 0 to it.</p> <p>0: LCDC did not generate a Vsync start interrupt or has been informed that the generated Vsync start interrupt has completed</p> <p>1: LCDC has generated a Vsync start interrupt and has not yet been informed that the generated Vsync start interrupt has completed</p>
8	VEINTS	0	R/W	<p>Vsync End Interrupt State</p> <p>Indicates the LCDC's Vsync end interrupt handling state. This bit is set to 1 at the time a Vsync end interrupt is generated. During the Vsync end interrupt handling routine, this bit should be cleared by writing 0.</p> <p>0: LCDC did not generate a Vsync end interrupt or has been informed that the generated Vsync end interrupt has completed</p> <p>1: LCDC has generated a Vsync end interrupt and has not yet been informed that the generated Vsync interrupt has completed</p>
7 to 0	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

### 22.3.16 LCDC Power Management Mode Register (LDPMMR)

LDPMMR controls the power supply circuit that provides power to the LCD module. The usage of two types of power-supply control pins, LCD\_VCPWC and LCD\_VEPWC, and turning on or off the power supply function are selected.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONC[3:0]				OFFD[3:0]				—	VCPE	VEPE	DONE	—	—	LPS[1:0]	
Initial value:	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	ONC[3:0]	0000	R/W	<p>LCDC Power-On Sequence Period</p> <p>Set the period from LCD_VEPWC assertion to LCD_DON assertion in the power-on sequence of the LCD module in frame units.</p> <p>Specify to the value of (the period) -1.</p> <p>This period is the (c) period in figures 22.4 to 22.7, Power-Supply Control Sequence and States of the LCD Module. For details on setting this register, see table 22.5, Available Power-Supply Control-Sequence Periods at Typical Frame Rates. (The setting method is common for ONA, ONB, OFFD, OFFE, and OFFF.)</p>
11 to 8	OFFD[3:0]	0000	R/W	<p>LCDC Power-Off Sequence Period</p> <p>Set the period from LCD_DON negation to LCD_VEPWC negation in the power-off sequence of the LCD module in frame units.</p> <p>Specify to the value of (the period) -1.</p> <p>This period is the (d) period in figures 22.4 to 22.7, Power-Supply Control Sequence and States of the LCD Module.</p>
7	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>



Bit	Bit Name	Initial Value	R/W	Description
6	VCPE	0	R/W	<p>LCD_VCPWC Pin Enable</p> <p>Sets whether or not to enable a power-supply control sequence using the LCD_VCPWC pin.</p> <p>0: Disabled: LCD_VCPWC pin is masked and fixed low</p> <p>1: Enabled: LCD_VCPWC pin output is asserted and negated according to the power-on or power-off sequence</p>
5	VEPE	0	R/W	<p>LCD_VEPWC Pin Enable</p> <p>Sets whether or not to enable a power-supply control sequence using the LCD_VEPWC pin.</p> <p>0: Disabled: LCD_VEPWC pin is masked and fixed low</p> <p>1: Enabled: LCD_VEPWC pin output is asserted and negated according to the power-on or power-off sequence</p>
4	DONE	1	R/W	<p>LCD_DON Pin Enable</p> <p>Sets whether or not to enable a power-supply control sequence using the LCD_DON pin.</p> <p>0: Disabled: LCD_DON pin is masked and fixed low</p> <p>1: Enabled: LCD_DON pin output is asserted and negated according to the power-on or power-off sequence</p>
3, 2	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
1, 0	LPS[1:0]	00	R	<p>LCD Module Power-Supply Input State</p> <p>Indicates the power-supply input state of the LCD module when using the power-supply control function.</p> <p>0: LCD module power off</p> <p>1: LCD module power on</p>

### 22.3.17 LDCD Power-Supply Sequence Period Register (LDPSPR)

LDPSPR controls the power supply circuit that provides power to the LCD module. The timing to start outputting the timing signals to the LCD\_VEPWC and LCD\_VCPWC pins is specified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONA[3:0]				ONB[3:0]				OFFE[3:0]				OFFF[3:0]			
Initial value:	1	1	1	1	0	1	1	0	0	0	0	0	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 12	ONA[3:0]	1111	R/W	<p>LDCD Power-On Sequence Period</p> <p>Set the period from LCD_VCPWC assertion to starting output of the display data (LCD_D) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) in the power-on sequence of the LCD module in frame units.</p> <p>Specify to the value of (the period)-1.</p> <p>This period is the (a) period in figures 22.4 to 22.7, Power-Supply Control Sequence and States of the LCD Module.</p>
11 to 8	ONB[3:0]	0110	R/W	<p>LDCD Power-On Sequence Period</p> <p>Set the period from starting output of the display data (LCD_D) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) to the LCD_VEPWC assertion in the power-on sequence of the LCD module in frame units.</p> <p>Specify to the value of (the period)-1.</p> <p>This period is the (b) period in figures 22.4 to 22.7, Power-Supply Control Sequence and States of the LCD Module.</p>

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	OFFE[3:0]	0000	R/W	<p>LCDC Power-Off Sequence Period</p> <p>Set the period from LCD_VEPWC negation to stopping output of the display data (LCD_D) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) in the power-off sequence of the LCD module in frame units.</p> <p>Specify to the value of (the period)-1.</p> <p>This period is the (e) period in figures 22.4 to 22.7, Power-Supply Control Sequence and States of the LCD Module.</p>
3 to 0	OFFF[3:0]	1111	R/W	<p>LCDC Power-Off Sequence Period</p> <p>Set the period from stopping output of the display data (LCD_D) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) to LCD_VCPWC negation to in the power-off sequence of the LCD module in frame units.</p> <p>Specify to the value of (the period)-1.</p> <p>This period is the (f) period in figures 22.4 to 22.7, Power-Supply Control Sequence and States of the LCD Module.</p>

### 22.3.18 LCDC Control Register (LDCNTR)

LDCNTR specifies start and stop of display by the LCDC.

When 1s are written to the DON2 bit and the DON bit, the LCDC starts display. Turn on the LCD module following the sequence set in the LDPMMR and LDCNTR. The sequence ends when the LPS[1:0] value changes from B'00 to B'11. Do not make any action to the DON bit until the sequence ends.

When 0 is written to the DON bit, the LCDC stops display. Turn off the LCD module following the sequence set in the LDPMMR and LDCNTR. The sequence ends when the LPS[1:0] value changes from B'11 to B'00. Do not make any action to the DON bit until the sequence ends.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	DON2	—	—	—	DON
Initial value:	0	0	0	0	0	0	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	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	DON2	0	R/W	Display On 2 Specifies the start of the LCDC display operation. 0: LCDC is being operated or stopped 1: LCDC starts operation When this bit is read, always read as 0. Write 1 to this bit only when starting display. If a value other than 0 is written when starting display, the operation is not guaranteed. When 1 is written to, it resumes automatically to 0. Accordingly, this bit does not need to be cleared by writing 0.
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	R/W	Description
0	DON	0	R/W	Display On Specifies the start and stop of the LCDC display operation. The control sequence state can be checked by referencing the LPS[1:0] of LDPMMR. 0: Display-off mode: LCDC is stopped 1: Display-on mode: LCDC operates

- Notes: 1. Write H'0011 to LDCNTR when starting display and H'0000 when completing display. Data other than H'0011 and H'0000 must not be written to.
2. Setting bit DON2 to 1 makes the contents of the palette RAM undefined. Before writing to the palette RAM, set bit DON2 to 1.

### 22.3.19 LCDC User Specified Interrupt Control Register (LDUINTR)

LDUINTR sets whether the user specified interrupt is generated, and indicates its processing state. This interrupt is generated at the time when image data which is set by the line number register (LDUINTLNR) in LCDC is read from VRAM.

This LCDC issues the interrupts (LCDCI): user specified interrupt by this register, memory access interrupt by the LCDC interrupt control register (LDINTR), and OR of Vsync interrupt output. This register and LCDC interrupt control register (LDINTR) settings affect the interrupt operation independently.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	UINTEN	0	R/W	User Specified Interrupt Enable Sets whether generate an LCDC user specified interrupt. 0: LCDC user specified interrupt is not generated 1: LCDC user specified interrupt is generated

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
0	UINTS	0	R/W	User Specified Interrupt State This bit is set to 1 at the time an LCDC user specified interrupt is generated (set state). During the user specified interrupt handling routine, this bit should be cleared by writing 0 to it. 0: LCDC did not generate a user specified interrupt or has been informed that the generated user specified interrupt has completed 1: LCDC has generated a user specified interrupt and has not yet been notified that the generated user specified interrupt has completed

Note: Interrupt processing flow:

1. Interrupt signal is input
2. LDINTR is read
3. If MINTS, FINTS, VSINTS, or VEINTS is 1, a generated interrupt is memory access interrupt, flame end interrupt, Vsync rising edge interrupt, or Vsync falling edge interrupt. Processing for each interrupt is performed.
4. If MINTS, FINTS, VSINTS, or VEINTS is 0, a generated interrupt is not memory access interrupt, flame end interrupt, Vsync rising edge interrupt, or Vsync falling edge interrupt.
5. UINTS is read.
6. If UINTS is 1, a generated interrupt is a user specified interrupt. Process for user specified interrupt is carried out.
7. If UINTS is 0, a generated interrupt is not a user specified interrupt. Other processing is performed.

### 22.3.20 LCDC User Specified Interrupt Line Number Register (LDUINTLNR)

LDUINTLNR sets the point where the user specified interrupt is generated. Setting is done in horizontal line units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	UINTLN[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1
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 to 0	UINTLN [10:0]	00001001111	R/W	User Specified Interrupt Generation Line Number Specifies the line in which the user specified interrupt is generated (line units). Set (the number of lines in which interrupts are generated) – 1 Example: Generate the user specified interrupt in the 80th line. $UINTLN = 160/2 - 1 = 79 = H'04F$

- Notes:
1. When using the LCD module with STN/TFT display, the setting value of this register should be equal to lower than the vertical display line number (VDLN) in LDVDLNR.
  2. When using the LCD module with DSTN display, the setting value of this register should be equal to or lower than half the vertical display line number (VDLN) in LDVDLNR. The user specified interrupt is generated at the point when the LCDC read the specified piece of image data in lower display from VRAM.

### 22.3.21 LCDC Memory Access Interval Number Register (LDLIRNR)

LDLIRNR controls the bus cycle interval when the LCDC reads VRAM. When LDLIRNR is set to a value other than H'00, the LCDC does not access VRAM until clock count of the SDRAM matches the value set in LDLIRNR. When LDLIRNR is set to H'00 (initial value), the LCDC accesses VRAM one clock after the LCDC accessed VRAM.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	LIRN[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
15 to 8	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
7 to 0	LIRN[7:0]	All 0	R/W	VRAM Read Clock Cycle Interval  Specifies the number of the SDRAM clock cycles which can be performed during burst clock cycles to read VRAM by LCDC.  H'00: 1 clock cycle H'01: 1 clock cycle H'02: 2 clock cycle  : H'FE: 254 clock cycles H'FF: 255 clock cycles



## 22.4 Operation

### 22.4.1 LCD Module Sizes which can be Displayed in this LCDC

This LCDC is capable of controlling displays with up to  $1024 \times 1024$  dots and 16 bpp (bits per pixel). The image data for display is stored in VRAM, which is shared with the CPU. This LCDC should read the data from VRAM before display.

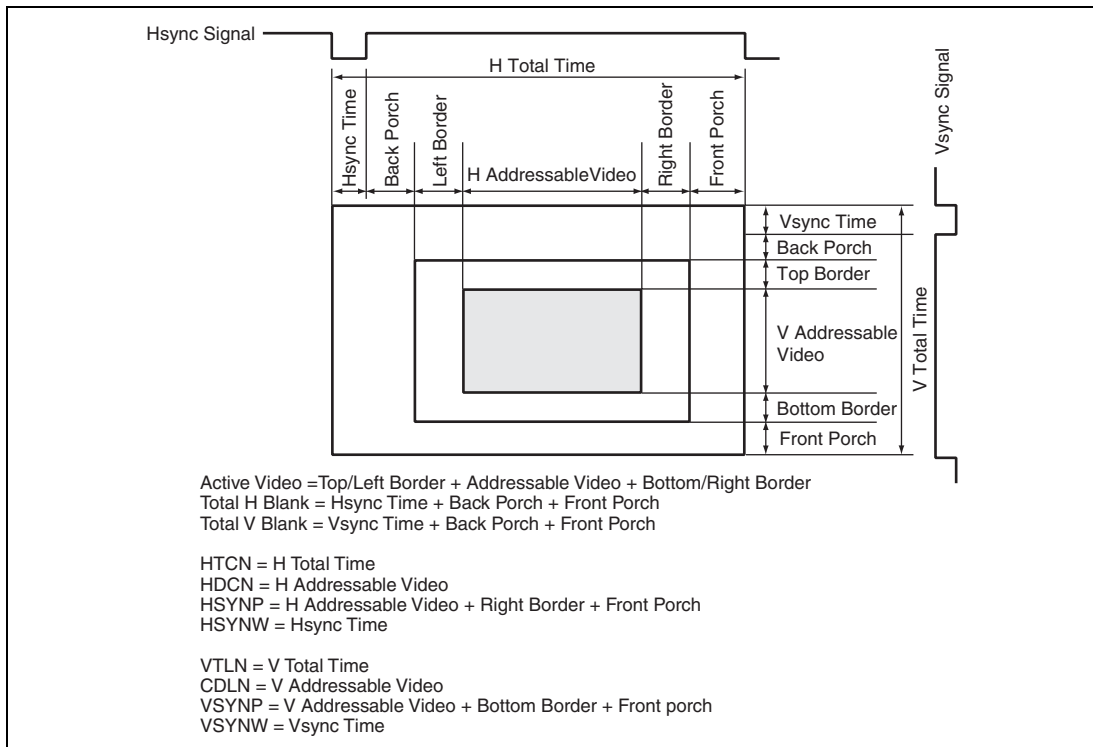
This LSI has a maximum 16-burst (32-bit bus width) memory read operation and a 2.4-Kbyte line buffer, so although a complete breakdown of the display is unlikely, there may be some problems with the display depending on the combination. A recommended size at the frame rate of 60 Hz is  $320 \times 240$  dots in 16 bpp or  $640 \times 480$  dots in 8 bpp.

As a rough standard, the bus occupation ratio shown below should not exceed 40%.

$$\text{Bus occupation ratio (\%)} = \frac{\text{Overhead coefficient} \times \text{Total number of display pixels ((HDCN + 1) \times 8 \times (VDLN + 1))}}{\text{CLKOUT (Hz)} \times \text{Bus width (32 bits)}} \times 100$$

The overhead coefficient becomes 2.000 when the CL2 SDRAM is connected to a 32-bit data bus and 1.825 when connected to a 64-bit data bus. The each value is ideal value under the best condition.

Figure 22.2 shows the valid display and the retrace period.



**Figure 22.2 Valid Display and the Retrace Period**

## 22.4.2 Color Palette Specification

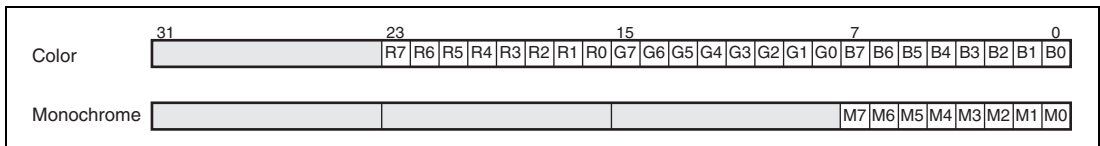
### (1) Color Palette Register

This LCDC has a color palette which outputs 24 bits of data per entry and is able to simultaneously hold 256 entries. The color palette thus allows the simultaneous display of 256 colors chosen from among 16-M colors.

The procedure below may be used to set up color palettes at any time.

1. The PALEN bit in the LDPALCR is 0 (initial value); normal display operation
2. Access LDPALCR and set the PALEN bit to 1; enter color-palette setting mode after three cycles of peripheral clock.
3. Access LDPALCR and confirm that the PALS bit is 1.
4. Access LDPR00 to LDPRFF and write the required values to the PALD00 to PALDFF bits.
5. Access LDPALCR and clear the PALEN bit to 0; return to normal display mode after a cycle of peripheral clock.

A 0 is output on the LCDC display data output (LCD\_D) while the PALS bit in LDPALCR is set to 1.



**Figure 22.3 Color-Palette Data Format**

PALDnn color and gradation data should be set as above.

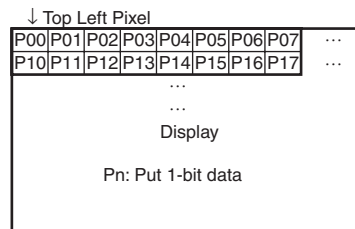
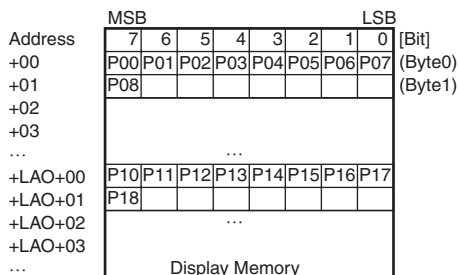
For a color display, PALDnn[23:16], PALDnn[15:8], and PALDnn[7:0] respectively hold the R, G, and B data. Although the bits PALDnn[18:16], PALDnn[9:8], and PALDnn[2:0] exist, no memory is associated with these bits. PALDnn[18:16], PALDnn[9:8], and PALDnn[2:0] are thus not available for storing palette data. The numbers of valid bits are thus R: 5, G: 6, and B: 5. A 24-bit (R: 8 bits, G: 8 bits, and B: 8 bits) data should, however, be written to the palette-data registers. When the values for PALDnn[23:19], PALDnn[15:10], or PALDnn[7:3] are not 0, 1 or 0 should be written to PALDnn[18:16], PALDnn[9:8], or PALDnn[2:0], respectively. When the values of PALDnn[23:19], PALDnn[15:10], or PALDnn[7:3] are 0, 0s should be written to PALDnn[18:16], PALDnn[9:8], or PALDnn[2:0], respectively. Then 24 bits are extended.


Grayscale data for a monochromatic display should be set in PALDnn[7:3]. PALDnn[23:8] are all "don't care". When the value in PALDnn[7:3] is not 0, 1s should be written to PALDnn[2:0].

When the value in PALDnn[7:3] is 0, 0s should be written to PALDnn[2:0]. Then 8 bits are extended.

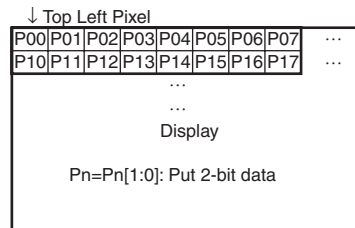
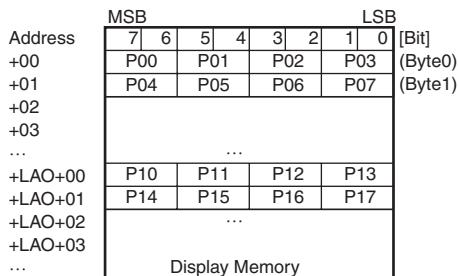
### 22.4.3 Data Format


#### 1. Packed 1bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



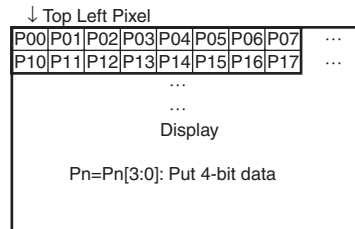
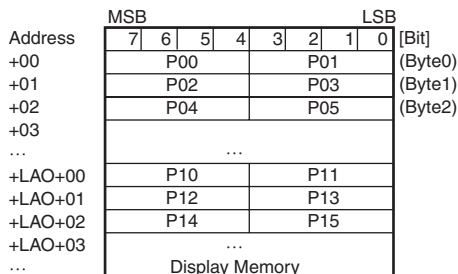
LAO: Line Address Offset  
 —Unused bits should be 0


#### 2. Packed 2bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



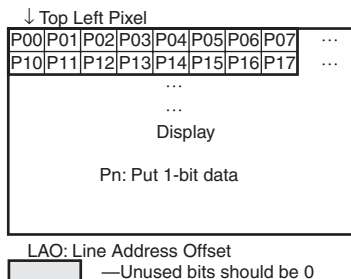
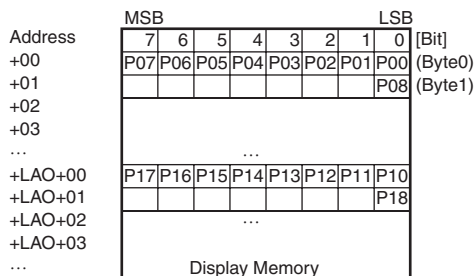
LAO: Line Address Offset  
 —Unused bits should be 0

#### 3. Packed 4bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]

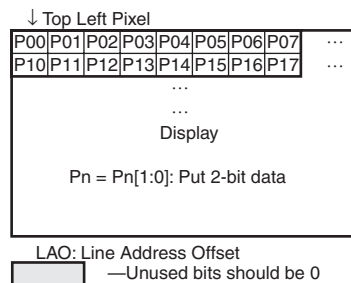
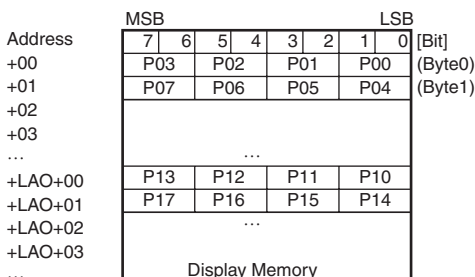


LAO: Line Address Offset  
 —Unused bits should be 0

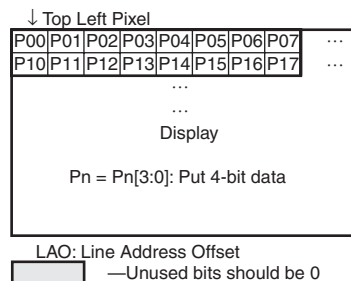
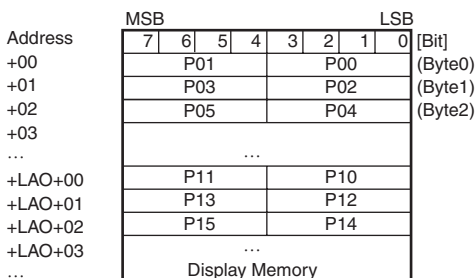
## 4. Packed 1bpp (Pixel Alignment in Byte is Little Endian)



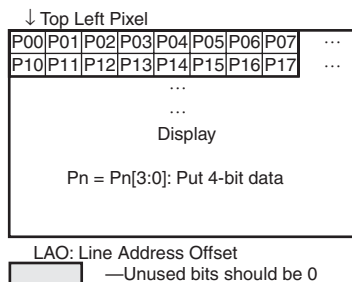
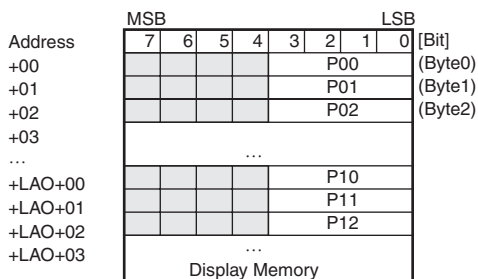
## 5. Packed 2bpp (Pixel Alignment in Byte is Little Endian)



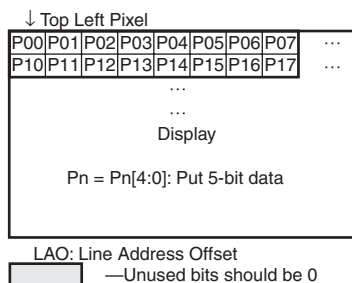
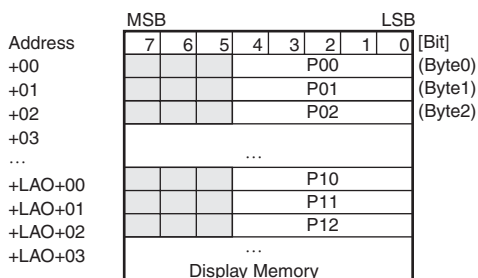
## 6. Packed 4bpp (Pixel Alignment in Byte is Little Endian)



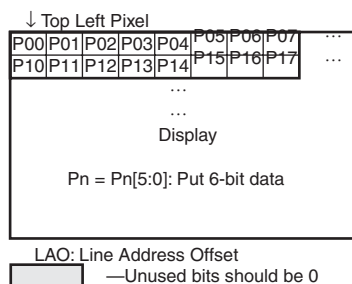
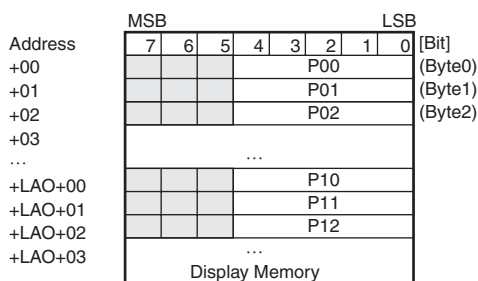
## 7. Unpacked 4bpp [Windows CE Recommended Format]



## 8. Unpacked 5bpp [Windows CE Recommended Format]

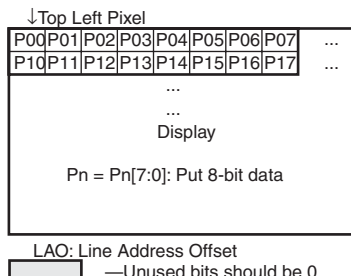


## 9. Unpacked 6bpp [Windows CE Recommended Format]



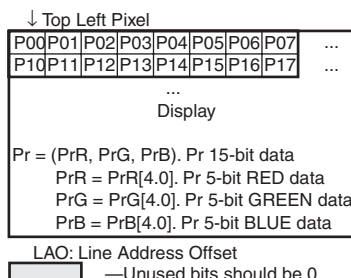
## 10. Packed 8bpp [Windows CE Recommended Format]

	MSB							LSB		
Address	7	6	5	4	3	2	1	0	[Bit]	
+00	P00								(Byte0)	
+01	P01								(Byte1)	
+02	P02								(Byte2)	
+03										
...	...									
+LAO+00	P10									
+LAO+01	P11									
+LAO+02	P12									
+LAO+03	...									
...	Display Memory									



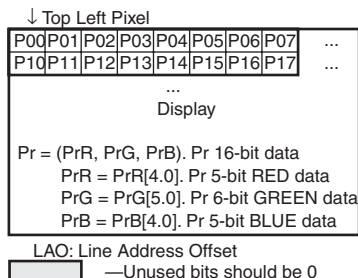
## 11. Unpacked color 15bpp (RGB 555) [Windows CE Recommended Format]

	MSB															LSB		
Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	[Bit]	
+00			P00R				P00G				P00B						(Word0)	
+02			P01R				P01G				P01B						(Word2)	
+04			P02R				P02G				P02B						(Word4)	
+06																		
...	...																	
+LAO+00			P10R				P10G				P10B							
+LAO+02			P11R				P11G				P11B							
+LAO+04			P12R				P12G				P12B							
+LAO+06																		
...	Display Memory																	



## 12. Packed color 16bpp (RGB 565) [Windows CE Recommended Format]

	MSB															LSB		
Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	[Bit]	
+00	P00R					P00G					P00B						(Word0)	
+02	P01R					P01G					P01B						(Word2)	
+04	P02R					P02G					P02B						(Word4)	
+06																		
...	...																	
+LAO+00	P10R					P10G					P10B							
+LAO+02	P11R					P11G					P11B							
+LAO+04	P12R					P12G					P12B							
+LAO+06	...																	
...	Display Memory																	



#### 22.4.4 Setting the Display Resolution

The display resolution is set up in LDHCNR, LDHSYNR, LDVDLNR, LDVTLNR, and LDVSYNR. The LCD current-alternating period for an STN or DSTN display is set by using the LDACLNR. The initial values in these registers are typical settings for VGA (640 × 480 dots) on an STN or DSTN display.

The clock to be used is set with the LDICKR. The LCD module frame rate is determined by the display interval + retrace line interval (non-display interval) for one screen set in a size related register and the frequency of the clock used.

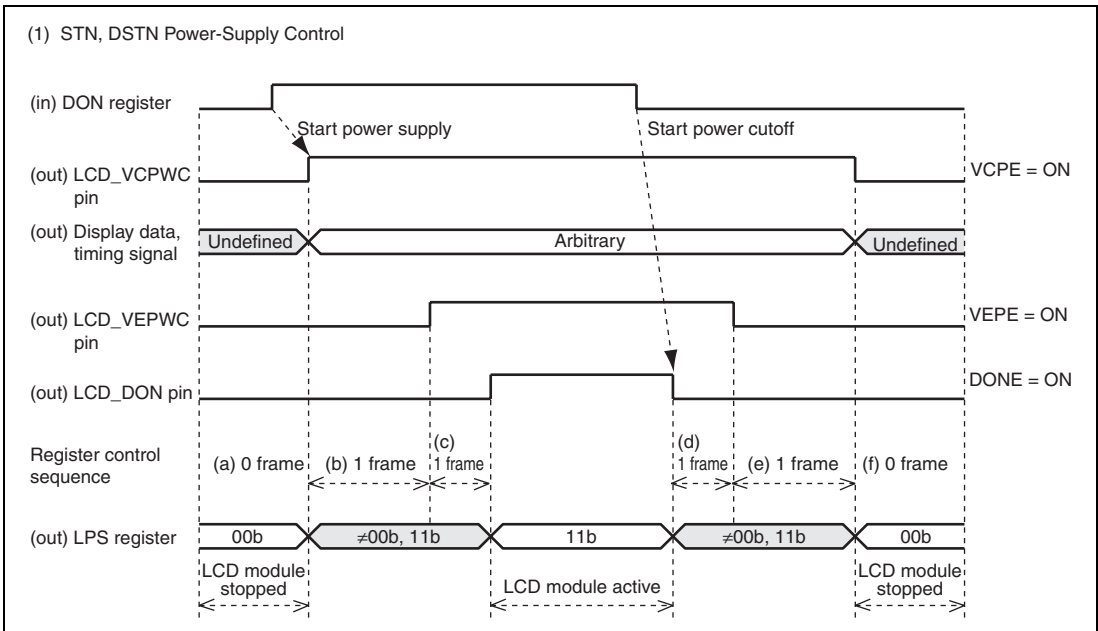
This LCDC has a Vsync interrupt function so that it is possible to issue an interrupt at the beginning of each vertical retrace line period (to be exact, at the beginning of the line after the last line of the display). This function is set up by using the LDINTR.

#### 22.4.5 Power-Supply Control Sequence

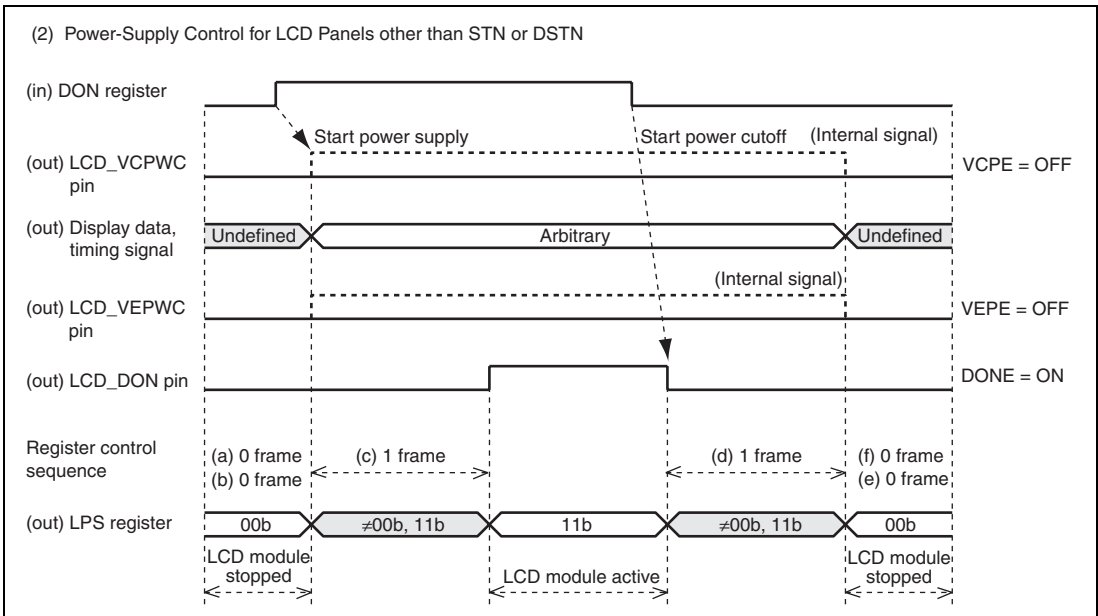
An LCD module normally requires a specific sequence for processing to do with the cutoff of the input power supply. Settings in LDPMMR, LDPSPR, and LDCNTR, in conjunction with the LCD power-supply control pins (LCD\_VCPWC, LCD\_VEPWC, and LCD\_DON), are used to provide processing of power-supply control sequences that suits the requirements of the LCD module.

Figures 22.4 to 22.7 are timing charts that show outlines of power-supply control sequences and table 22.5 is a summary of available power-supply control sequence periods.

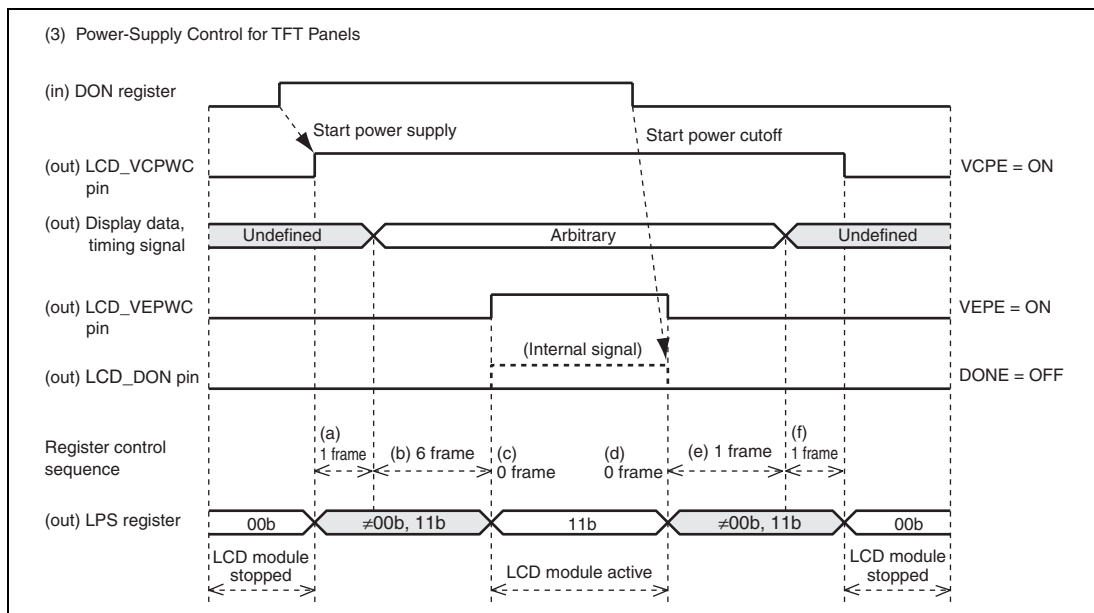




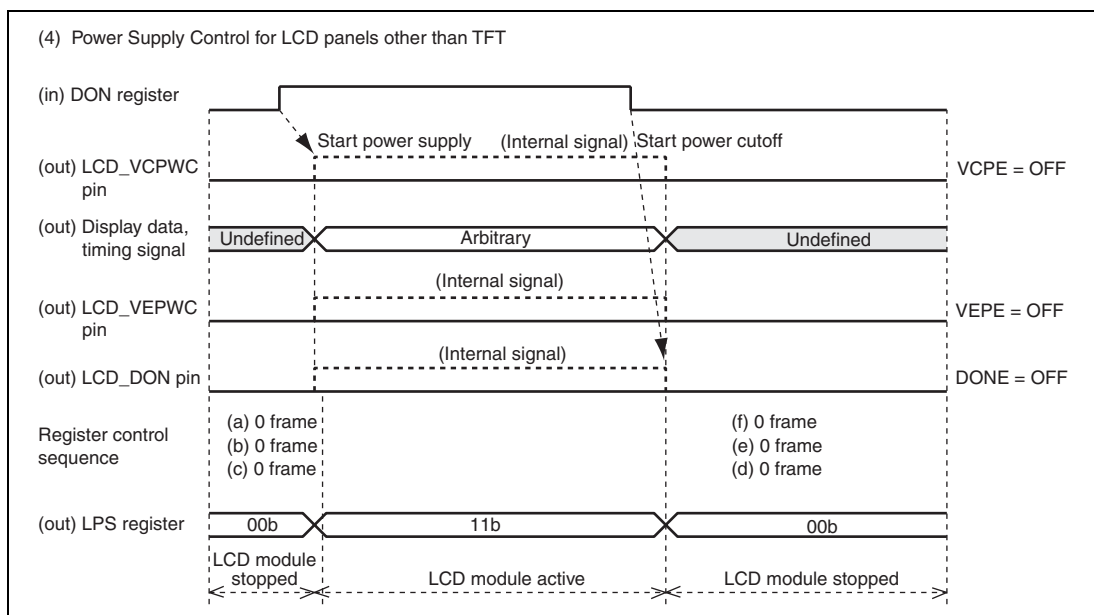
**Figure 22.4 Power-Supply Control Sequence and States of the LCD Module**



**Figure 22.5 Power-Supply Control Sequence and States of the LCD Module**



**Figure 22.6 Power-Supply Control Sequence and States of the LCD Module**



**Figure 22.7 Power-Supply Control Sequence and States of the LCD Module**

**Table 22.5 Available Power-Supply Control-Sequence Periods at Typical Frame Rates**

ONX, OFFX Register Value	Frame Rate	
	120 Hz	60 Hz
H'F	$(-1+1)/120 = 0.00 \text{ (ms)}$	$(-1+1)/60 = 0.00 \text{ (ms)}$
H'0	$(0+1)/120 = 8.33 \text{ (ms)}$	$(0+1)/60 = 16.67 \text{ (ms)}$
H'1	$(1+1)/120 = 16.67 \text{ (ms)}$	$(1+1)/60 = 33.33 \text{ (ms)}$
H'2	$(2+1)/120 = 25.00 \text{ (ms)}$	$(2+1)/60 = 50.00 \text{ (ms)}$
H'3	$(3+1)/120 = 33.33 \text{ (ms)}$	$(3+1)/60 = 66.67 \text{ (ms)}$
H'4	$(4+1)/120 = 41.67 \text{ (ms)}$	$(4+1)/60 = 83.33 \text{ (ms)}$
H'5	$(5+1)/120 = 50.00 \text{ (ms)}$	$(5+1)/60 = 100.00 \text{ (ms)}$
H'6	$(6+1)/120 = 58.33 \text{ (ms)}$	$(6+1)/60 = 116.67 \text{ (ms)}$
H'7	$(7+1)/120 = 66.67 \text{ (ms)}$	$(7+1)/60 = 133.33 \text{ (ms)}$
H'8	$(8+1)/120 = 75.00 \text{ (ms)}$	$(8+1)/60 = 150.00 \text{ (ms)}$
H'9	$(9+1)/120 = 83.33 \text{ (ms)}$	$(9+1)/60 = 166.67 \text{ (ms)}$
H'A	$(10+1)/120 = 91.67 \text{ (ms)}$	$(10+1)/60 = 183.33 \text{ (ms)}$
H'B	$(11+1)/120 = 100.00 \text{ (ms)}$	$(11+1)/60 = 200.00 \text{ (ms)}$
H'C	$(12+1)/120 = 108.33 \text{ (ms)}$	$(12+1)/60 = 216.67 \text{ (ms)}$
H'D	$(13+1)/120 = 116.67 \text{ (ms)}$	$(13+1)/60 = 233.33 \text{ (ms)}$
H'E	$(14+1)/120 = 125.00 \text{ (ms)}$	$(14+1)/60 = 250.00 \text{ (ms)}$

ONA, ONB, ONC, OFFD, OFFE, and OFFF are used to set the power-supply control-sequence periods, in units of frames, from 0 to 15. 1 is subtracted from each register. H'0 to H'E settings select from 1 to 15 frames. The setting H'F selects 0 frames.

Actual sequence periods depend on the register values and the frame frequency of the display. The following table gives power-supply control-sequence periods for display frame frequencies used by typical LCD modules.

- When ONB is set to H'6 and display's frame frequency is 120 Hz  
 The display's frame frequency is 120 Hz. 1 frame period is thus  $8.33 \text{ (ms)} = 1/120 \text{ (sec)}$ .  
 The power-supply input sequence period is 7 frames because ONB setting is subtracted by 1.  
 As a result, the sequence period is  $58.33 \text{ (ms)} = 8.33 \text{ (ms)} \times 7$ .

**Table 22.6 LCDC Operating Modes**

<b>Mode</b>		<b>Function</b>
Display on (LCDC active)	Register setting: DON = 1	Fixed resolution, the format of the data for display is determined by the number of colors, and timing signals are output to the LCD module.
Display off (LCDC stopped)	Register setting: DON = 0	Register access is enabled.  Fixed resolution, the format of the data for display is determined by the number of colors, and timing signals are not output to the LCD module.

**Table 22.7 LCD Module Power-Supply States**

(STN, DSTN module)

<b>State</b>	<b>Power Supply for Logic</b>	<b>Display Data, Timing Signal</b>	<b>Power Supply for High-Voltage Systems</b>	<b>DON Signal</b>
Control Pin	LCD_VCPWC	LCD_CL2, LCD_CL1, LCD_FLM, LCD_M_DISP, LCD_D	LCD_VEPWC	LCD_DON
Operating State	Supply	Supply	Supply	Supply
(Transitional State)	Supply	Supply	Supply	
	Supply	Supply		
	Supply			
Stopped State				

(TFT module)

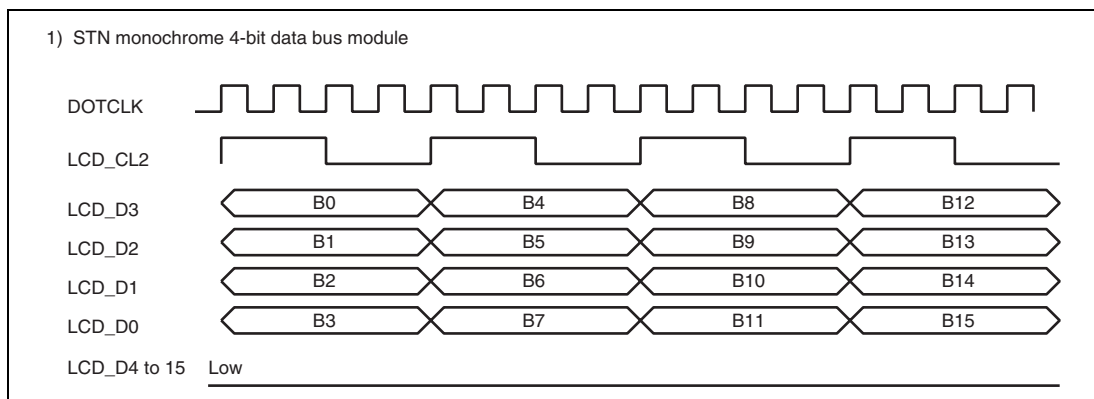
State	Power Supply for Logic	Display Data, Timing Signal	Power Supply for High-Voltage Systems
Control Pin	LCD_VCPWC	LCD_CL2, LCD_CL1, LCD_FLM, LCD_M_DISP, LCD_D	LCD_VEPWC
Operating State	Supply	Supply	Supply
(Transitional State)	Supply	Supply	
	Supply		
Stopped State			

The table above shows the states of the power supply, display data, and timing signals for the typical LCD module in its active and stopped states. Some of the supply voltages described may not be necessary, because some modules internally generate the power supply required for high-voltage systems from the logic-level power-supply voltage.

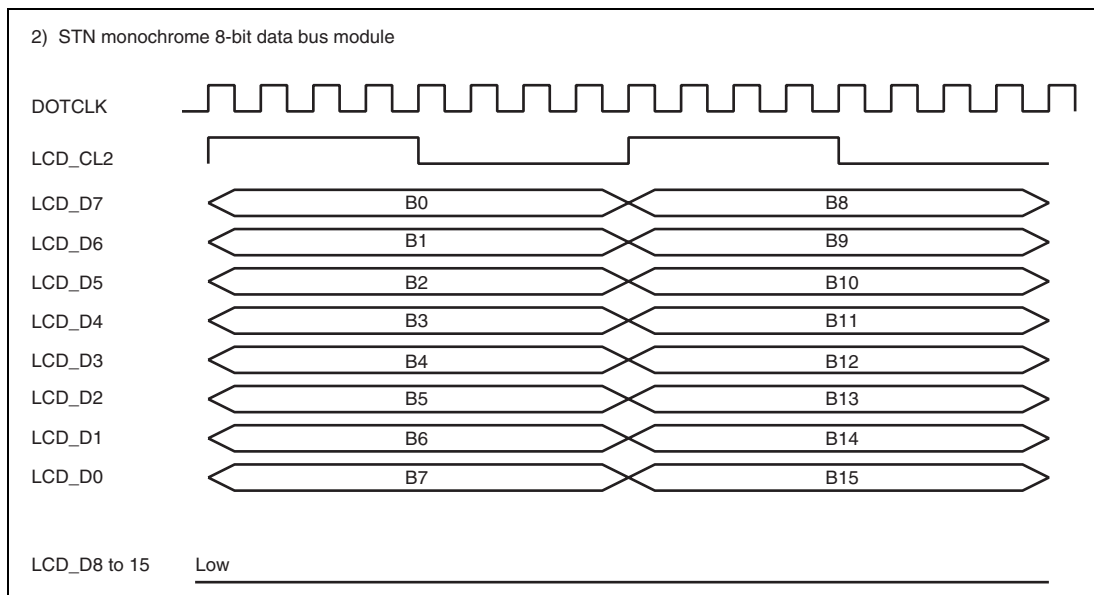
Notes on display-off mode (LCDC stopped):

If LCD module power-supply control-sequence processing is in use by the LCDC or the supply of power is cut off while the LCDC is in its display-on mode, normal operation is not guaranteed. In the worst case, the connected LCD module may be damaged.

## 22.5 Clock and LCD Data Signal Examples

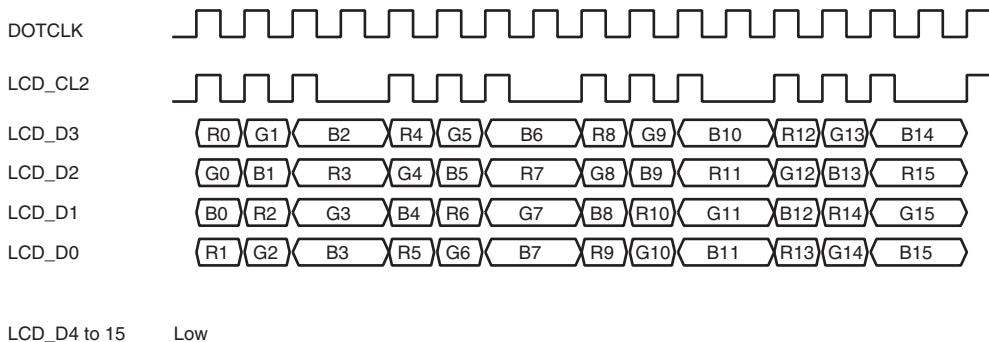


**Figure 22.8 Clock and LCD Data Signal Example**

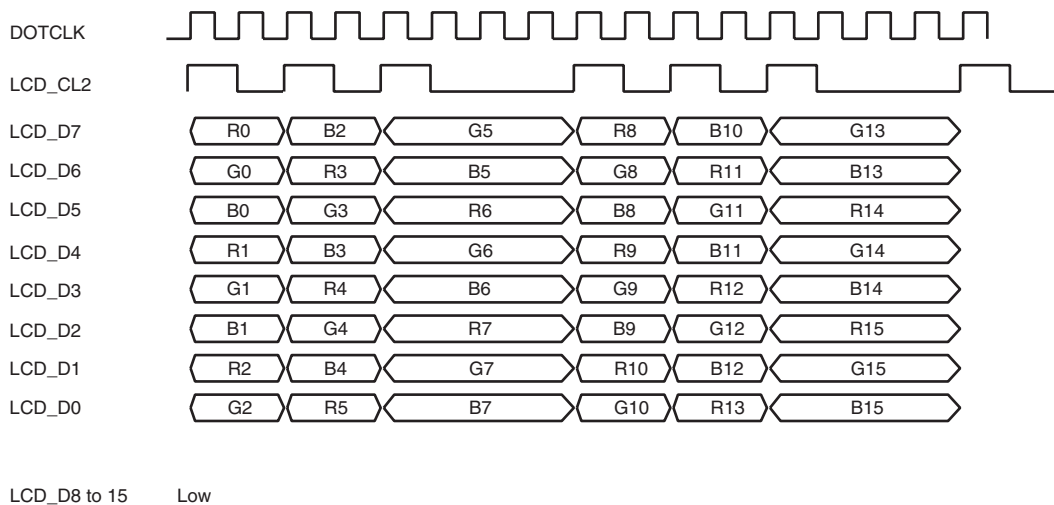


**Figure 22.9 Clock and LCD Data Signal Example  
(STN Monochrome 8-Bit Data Bus Module)**

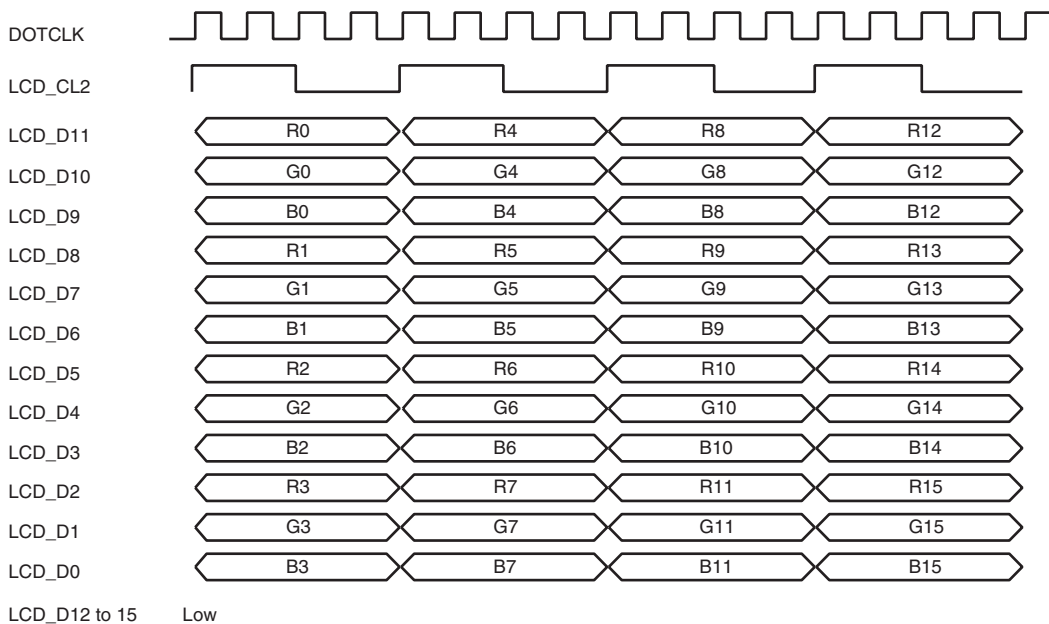
## 3) STN color 4-bit data bus module

**Figure 22.10 Clock and LCD Data Signal Example (STN Color 4-Bit Data Bus Module)**

## 4) STN color 8-bit data bus module

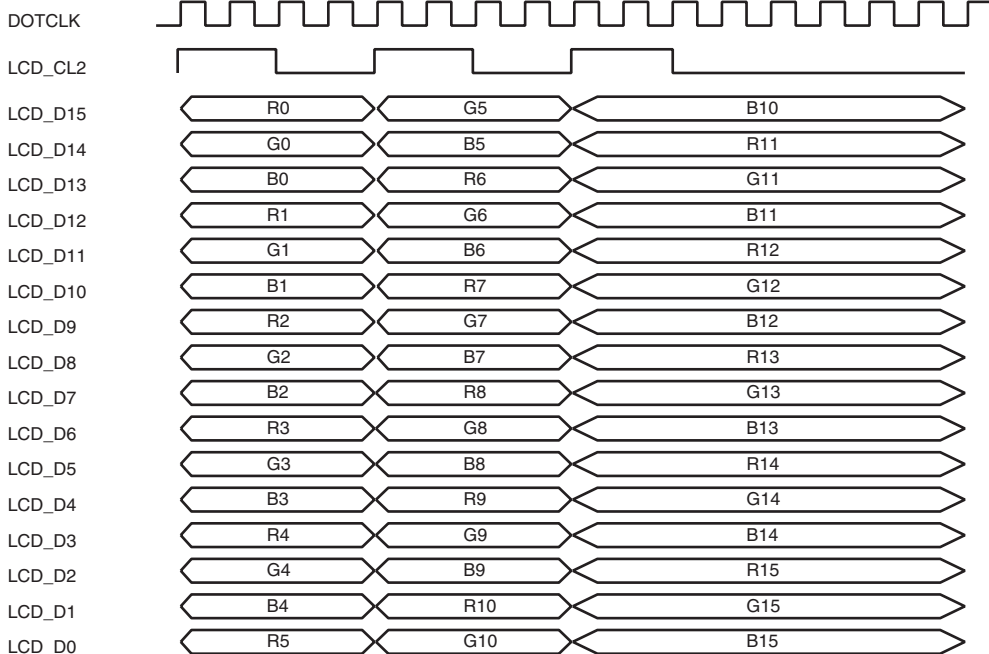
**Figure 22.11 Clock and LCD Data Signal Example (STN Color 8-Bit Data Bus Module)**

## 5) STN color 12-bit data bus module

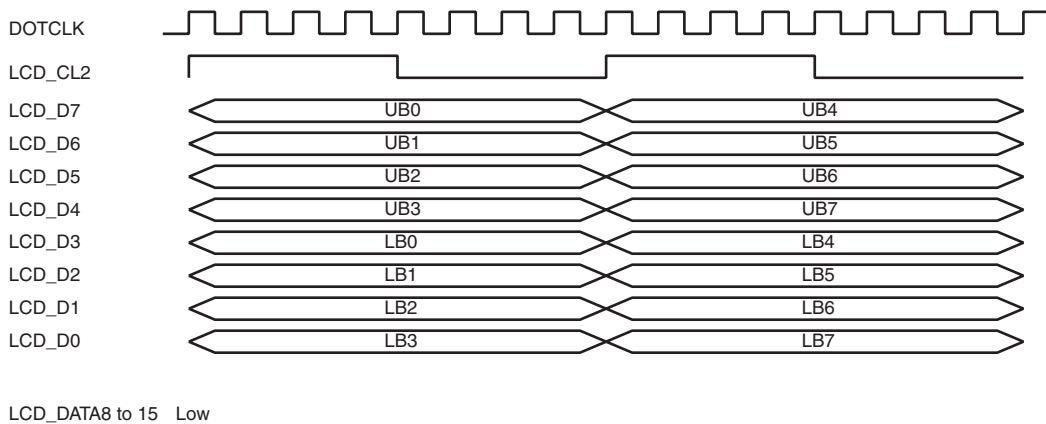
**Figure 22.12 Clock and LCD Data Signal Example (STN Color 12-Bit Data Bus Module)**



## 6) STN color 16-bit data bus module

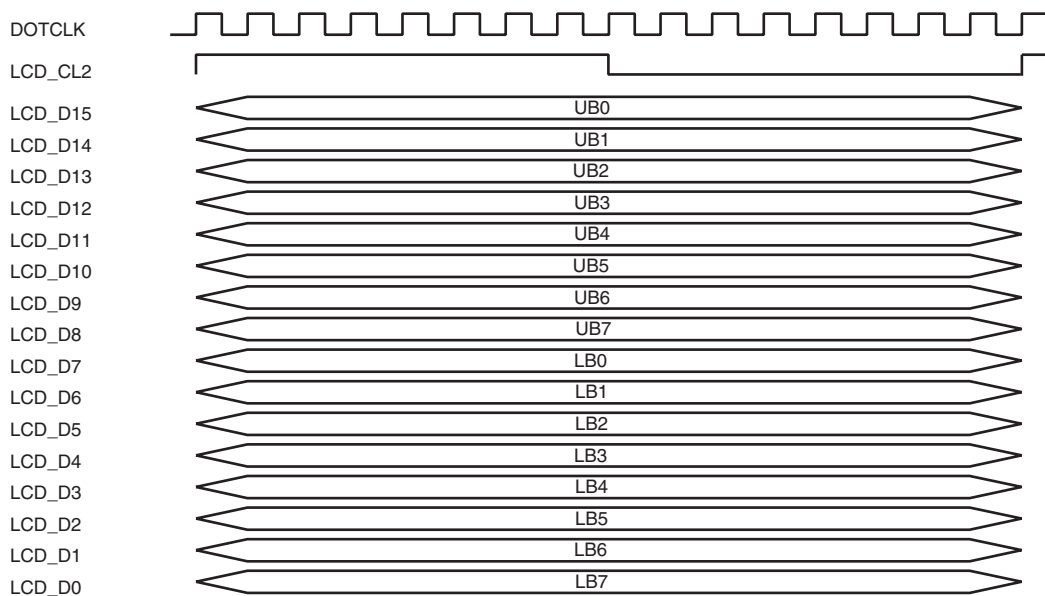
**Figure 22.13 Clock and LCD Data Signal Example (STN Color 16-Bit Data Bus Module)**

## 7) DSTN monochrome 8-bit data bus module



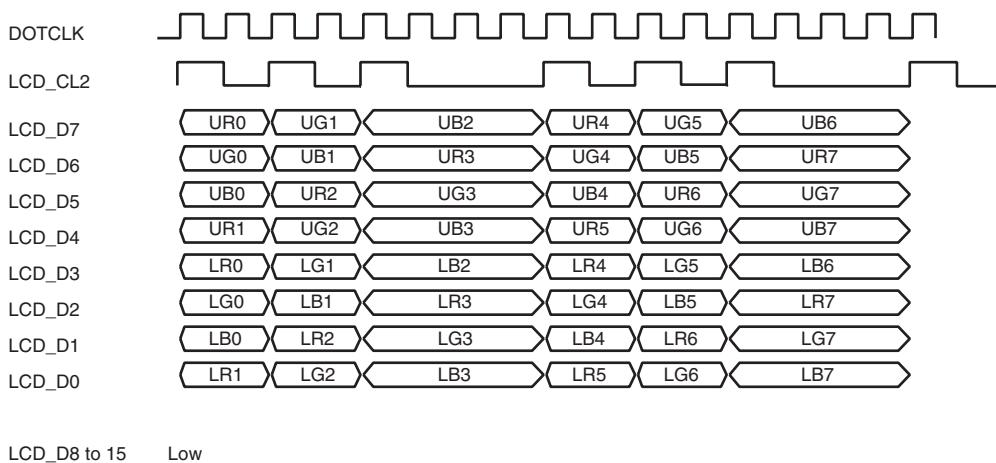
**Figure 22.14 Clock and LCD Data Signal Example  
(DSTN Monochrome 8-Bit Data Bus Module)**

## 8) DSTN monochrome 16-bit data bus module

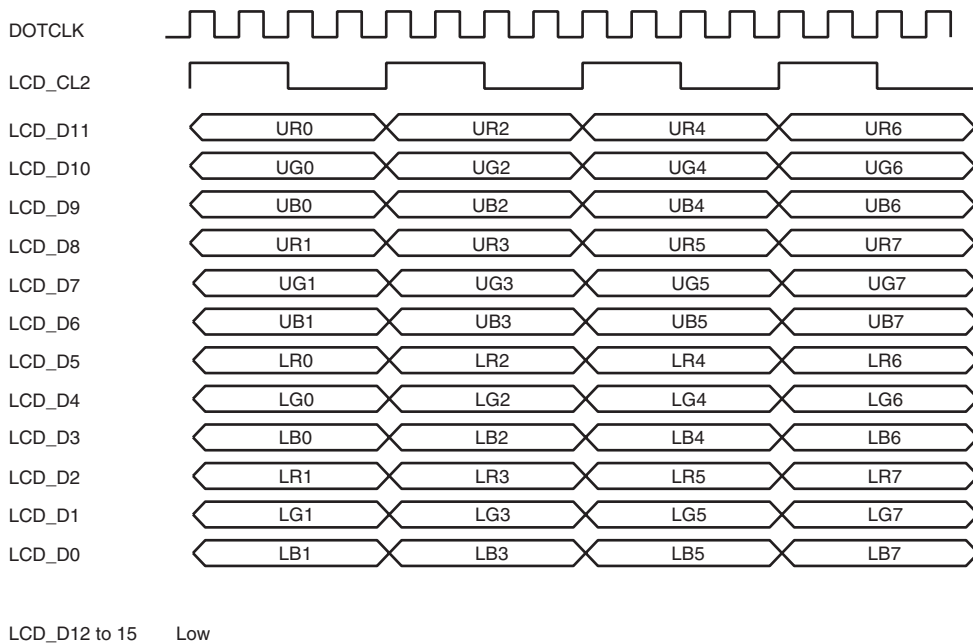


**Figure 22.15 Clock and LCD Data Signal Example  
(DSTN Monochrome 16-Bit Data Bus Module)**

## 9) DSTN color 8-bit data bus module

**Figure 22.16 Clock and LCD Data Signal Example (DSTN Color 8-Bit Data Bus Module)**

10) DSTN color 12-bit data bbus module

**Figure 22.17 Clock and LCD Data Signal Example (DSTN Color 12-Bit Data Bus Module)**

## 11) DSTN color 16-bit data bus module

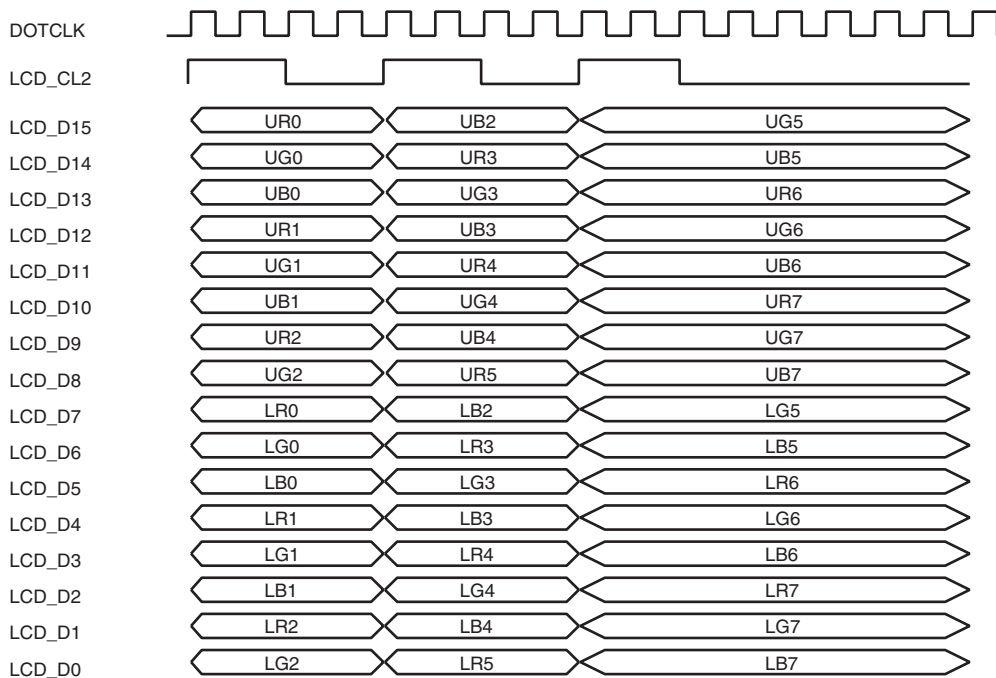
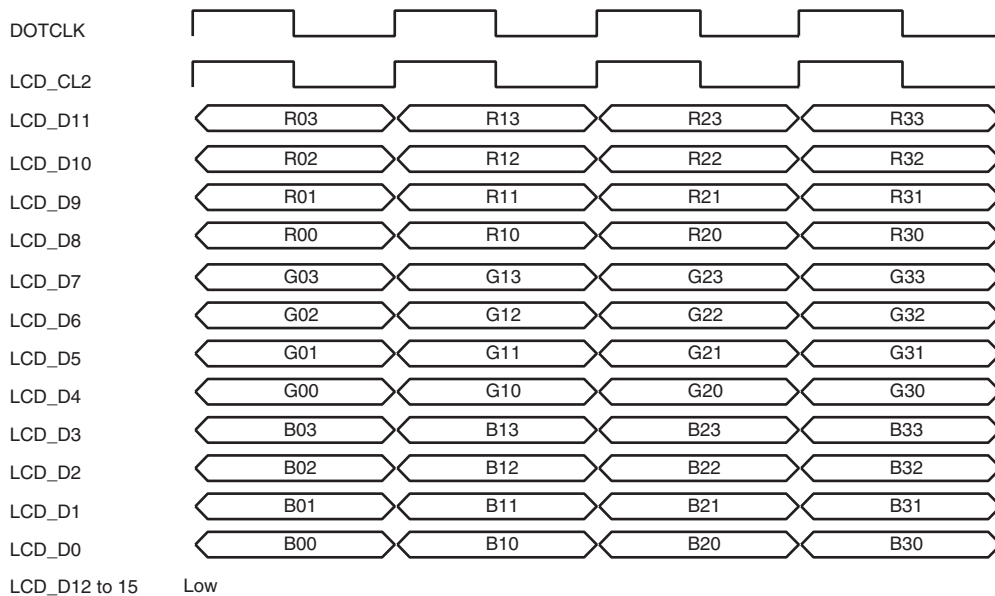


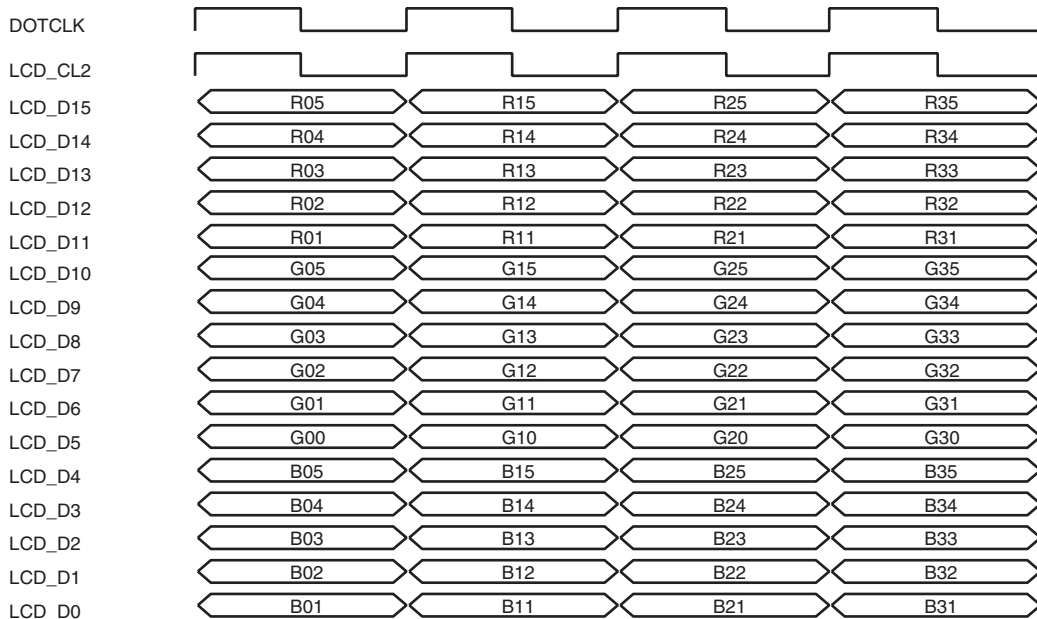
Figure 22.18 Clock and LCD Data Signal Example (DSTN Color 16-Bit Data Bus Module)

12) TFT color 12-bit data bus module



**Figure 22.19** Clock and LCD Data Signal Example (TFT Color 12-Bit Data Bus Module)

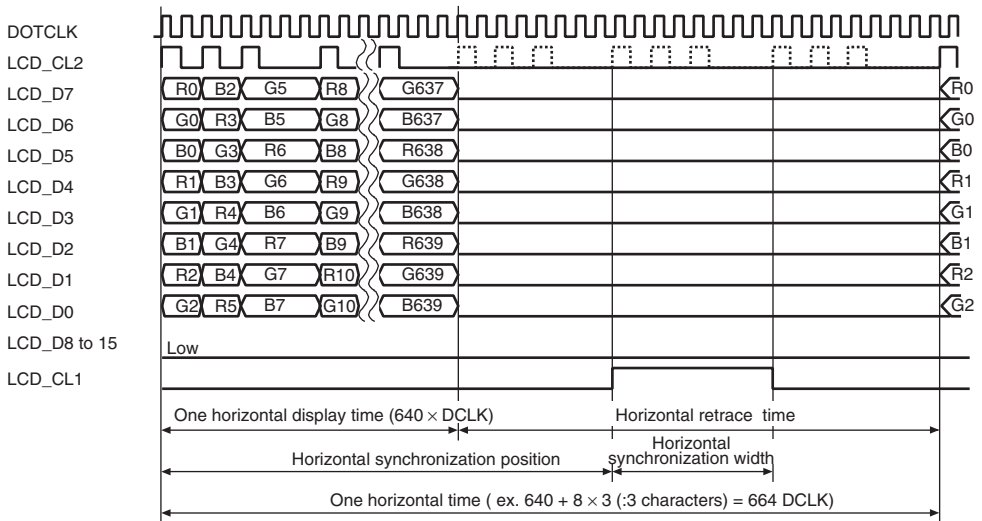
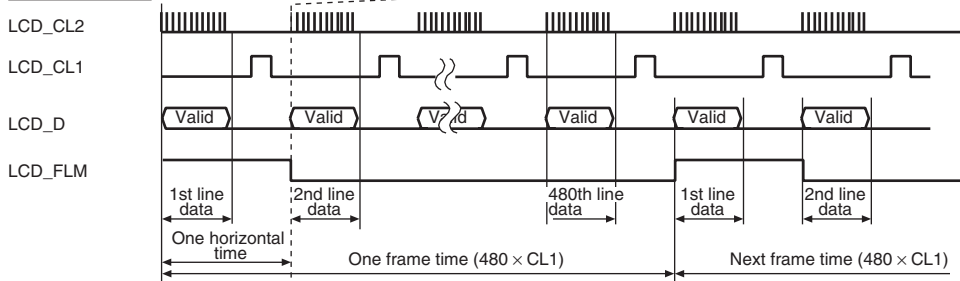
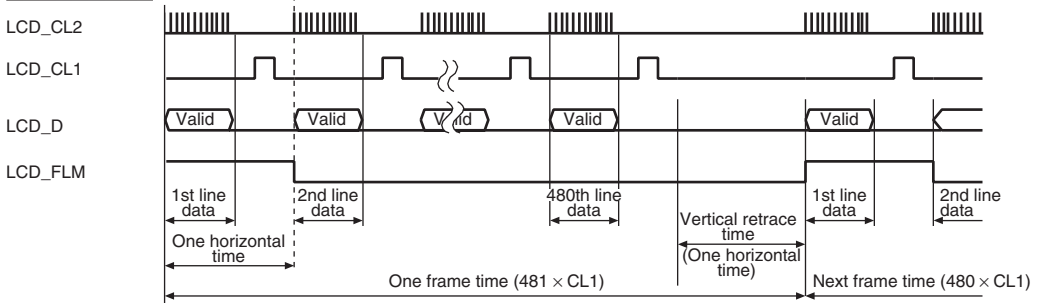
## 13) TFT color 16-bit data bus module

**Figure 22.20 Clock and LCD Data Signal Example (TFT Color 16-Bit Data Bus Module)**



14) 8-bit interface color  $640 \times 840$ 

STN-LCD

Horizontal waveNo vertical retraceOne vertical retraceFigure 22.21 Clock and LCD Data Signal Example (8-Bit Interface Color  $640 \times 480$ )



## 22.6 Usage Notes

### 22.6.1 Procedure for Halting Access to Display Data Storage VRAM (SDRAM in Area 1 or 2)

Follow the procedure below to halt access to VRAM for storing display data (SDRAM in area 1 or 2).

Procedure for Halting Access to Display Data Storage VRAM:

1. Confirm that the LPS1 and LPS0 bits in LDPMMR are currently set to 1.
2. Clear the DON bit in LDCNTR to 0 (display-off mode).
3. Confirm that the LPS1 and LPS0 bits in LDPMMR have changed to 0.
4. Wait for the display time for a single frame to elapse.

This halting procedure is required before selecting self-refreshing for the display data storage VRAM (SDRAM in area 1 or 2) or making a transition to standby mode or module standby mode.

### 22.6.2 Notes on Holding the Access Request by MCU

If the NMIFL bit in the NMIFCR register is set to 1 by an NMI interrupt when both the NMIME (bit 24) bit and the LCDM (bit 16) bit are 1 in the request mask setting register (RQM) of MCU, the LCDC cannot access the VRAM that is used for the display data storage (SDRAM in area 1 or 2). The access request is held, if the LCDC attempts access the VRAM.

As the LCDC continues to output data stored in the line buffer to the LCD panel data pin, the LCD display will be stopped if the line buffer becomes empty. Accordingly, NMI interrupts should be disabled and the NMIFL bit should be cleared to 0 before the line buffer becomes empty.

If a bus release request is accepted from the an external device, the LCDC also cannot access the VRAM and the access request is held. Acquire the bus mastership again before the line buffer becomes empty as in the case of the NMI interrupt.



## Section 23 G2D

### 23.1 Basic Functions

#### 23.1.1 List of Commands and Rendering Attributes

**Table 23.1 Commands and Rendering Attributes**

										Draw Mode									
										b15	b14	b13	b12	b11	b10	b9	b8		
OP CODE																		DTRANS	WORK
Command	b31	b30	b29	b28	b27	b26	b25	b24	MTRE	Reserved	CLIP	RCLIP	STRANS	/LINKE	/LREL	SS			
POLYGON4A	1	0	0	0	0	0	1	0	MTRE		CLIP	RCLIP	STRANS		WORK	SS			
POLYGON4B					0	0	0	1	MTRE		CLIP	RCLIP	STRANS		WORK	SS			
POLYGON4C					0	0	0	0	MTRE		CLIP	RCLIP			WORK				
LINEA	1	0	1	1	0	0	1	0	MTRE		CLIP	RCLIP	STRANS			SS (0)			
LINEB					0	0	0	1	MTRE		CLIP	RCLIP	STRANS			SS (0)			
LINEC					0	0	0	0	MTRE		CLIP	RCLIP			LREL				
LINED					0	0	1	1	MTRE		CLIP	RCLIP		LINKE	LREL				
RLINEA					0	1	1	0	MTRE		CLIP	RCLIP	STRANS			SS (0)			
RLINEB					0	1	0	1	MTRE		CLIP	RCLIP	STRANS			SS (0)			
RLINEC					0	1	0	0	MTRE		CLIP	RCLIP		LINKE	LREL				
RLINED					0	1	1	1	MTRE		CLIP	RCLIP		LINKE	LREL				
FTRAPC	1	1	0	1	0	0	0	0	MTRE		CLIP	RCLIP		LINKE	LREL				
RFTRAPC					0	1	0	0	MTRE		CLIP	RCLIP		LINKE	LREL				
CLRWC	1	1	1	0	0	0	0	0	MTRE		CLIP	RCLIP							
LINEWC	1	1	1	1	0	0	0	0	MTRE		CLIP	RCLIP							
RLINEWC					0	1	0	0	MTRE		CLIP	RCLIP							
BITBLTA	1	0	1	0	0	0	1	0	MTRE		CLIP	RCLIP	STRANS	DTRANS	WORK	SS			
BITBLTB					0	0	0	1	MTRE		CLIP	RCLIP	STRANS	DTRANS	WORK	SS			
BITBLTC					0	0	0	0	MTRE		CLIP	RCLIP		DTRANS	WORK				
Test mode	1	0	1	0	1	0	0	0	This is for internal verification only and should not be set. Even if this setting is made, the command error (CER) flag is not set.										

										Draw Mode							
										b7	b6	b5	b4	b3	b2	b1	b0
OP CODE										STYLE	BLKE	NET/EDG	EOS		AA	CLKW	
Command	b31	b30	b29	b28	b27	b26	b25	b24	REL	/SRCDIRX	/SRCDIRY	/DSTDIRX	/DSTDIRY	COOF	/αE	/SαE	
POLYGON4A	1	0	0	0	0	0	1	0	REL	STYLE	BLKE	NET	EOS	COOF	αE	SαE	
POLYGON4B					0	0	0	1	REL	STYLE	BLKE	NET	EOS	COOF	αE		
POLYGON4C					0	0	0	0			BLKE	NET	EOS	COOF	αE		
LINEA	1	0	1	1	0	0	1	0	REL	STYLE (1)		NET	EOS	COOF	AA		
LINEB					0	0	0	1	REL	STYLE (1)		NET	EOS	COOF	AA		
LINEC					0	0	0	0				NET	EOS	COOF	AA		
LINED					0	0	1	1							AA (1)	CLKW	
RLINEA					0	1	1	0	REL	STYLE (1)		NET	EOS	COOF	AA		
RLINEB					0	1	0	1	REL	STYLE (1)		NET	EOS	COOF	AA		
RLINEC					0	1	0	0				NET	EOS	COOF	AA		
RLINED					0	1	1	1							AA (1)	CLKW	
FTRAPC	1	1	0	1	0	0	0	0			BLKE (1)	EDG	EOS				
RFTRAPC					0	1	0	0			BLKE (1)	EDG	EOS				
CLRWC	1	1	1	0	0	0	0	0			BLKE (1)						
LINEWC	1	1	1	1	0	0	0	0					EOS				
RLINEWC					0	1	0	0					EOS				
BITLETA	1	0	1	0	0	0	1	0	REL	SRCDIRX	SRCDIRY	DSTDIRX	DSTDIRY	COOF	αE	SαE	
BITLETB					0	0	0	1	REL	SRCDIRX	SRCDIRY	DSTDIRX	DSTDIRY	COOF	αE		
BITLETC					0	0	0	0				DSTDIRX	DSTDIRY	COOF	αE		
Test mode	1	0	1	0	1	0	0	0	This is for internal verification only and should not be set. Even if this setting is made, the command error (CER) flag is not set.								

**REL:** Valid only when SS = 0. Clear this bit to 0 when SS = 1.

**COOF:** Valid only in 16-bit/pixel mode (GBM = 1). Clear this bit to 0 in 8-bit/pixel mode (GBM = 0).

**SαE:** Valid only for the ARGB format (SPF = DPF = 1). Clear this bit to 0 for the RGB format (SPF = DPF = 0) and 8-bit/pixel mode (GBM = 0).  
Clear this bit to 0 when αE = 0.

**αE:** Valid only in 16-bit/pixel mode (GBM = 1). Clear this bit to 0 in 8-bit/pixel mode (GBM = 0).  
In the POLYGON4A, POLYGON4B, or POLYGON4C command, valid only when BLKE = 1. Clear this bit to 0 when BLKE = 0.  
In the BITBLTA, BITBLTB, or BITBLTC command, valid only when the ROP code = H'CC. Clear this bit to 0 for other codes.

**LREL:** Valid only when LINKE = 1. The LREL bit should be cleared to 0 when LINKE = 0.

**STYLE:** Set this bit to 1 when BLKE = 1. In the LINEA, LINEB, RLINEA, or RLINEB command, set this bit to 1.

**AA:** Clear this bit to 0 when NET = 1. Valid only in 16-bit/pixel mode (GBM = 1). Clear this bit to 0 in 8-bit/pixel mode (GBM = 0).

In the LINED or RLINED command, set this bit to 1.

**SS:** In the LINEA, LINEB, RLINEA, or RLINEB command, clear this bit to 0.

**BLKE:** In the FTRAPC, RFTRAPC, or CLRWC command, set this bit to 1.

Shaded bit: Cannot be used (clear this bit to 0).

**Table 23.2 Commands and Rendering Attributes.**

Command	OP CODE								Draw Mode							
	b31	b30	b29	b28	b27	b26	b25	b24	b15	b14	b13	b12	b11	b10	b9	B8
TRAP	0	0	0	0	0	0	0	0								
NOP/INT	0	0	0	0	1	0	0	0	INT							
VBKEM	0	0	0	1	0	0	0	0								
WPR	0	0	0	1	1	0	0	0						LINKE	LREL	
JUMP	0	0	1	0	1	0	0	0								
GOSUB	0	0	1	1	0	0	0	0								
RET	0	0	1	1	1	0	0	0								
LCOFS	0	1	0	0	0	0	0	0								
RLCOFS	0	1	0	0	0	1	0	0								
MOVE	0	1	0	0	1	0	0	0								
RMOVE	0	1	0	0	1	1	0	0								
Test mode	0	1	0	1	0	0	0	0	This is for internal verification only and should not be set. Even if this setting is made, the command error (CER) flag is not set.							

Command	OP CODE								Draw Mode							
	b31	b30	b29	b28	b27	b26	b25	b24	b7	b6	b5	b4	b3	b2	b1	b0
TRAP	0	0	0	0	0	0	0	0			Flip5	Flip4	Flip3	Flip2	Flip1	Flip0
NOP/INT	0	0	0	0	1	0	0	0								
VBKEM	0	0	0	1	0	0	0	0								
WPR	0	0	0	1	1	0	0	0						ByteM3	ByteM2	ByteM1
JUMP	0	0	1	0	1	0	0	0	REL							
GOSUB	0	0	1	1	0	0	0	0	REL							No
RET	0	0	1	1	1	0	0	0								No
LCOFS	0	1	0	0	0	0	0	0								
RLCOFS	0	1	0	0	0	1	0	0								
MOVE	0	1	0	0	1	0	0	0								
RMOVE	0	1	0	0	1	1	0	0								
Test mode	0	1	0	1	0	0	0	0	This is for internal verification only and should not be set. Even if this setting is made, the command error (CER) flag is not set.							

### 23.1.2 Basic Functions

#### (1) Features

- On-chip geometry engine for coordinate transformation  
Hardware that performs coordinate transformation ( $4 \times 4$  matrix operation + Z clipping + perspective W division) for the input vertex
- Extended 2D functions  
High-functional bold line drawing, antialias line drawing, and BITBLT type commands with ROP/alpha blending
- Upgraded control command functions  
Two command systems: GOSUB/RET and INT command, and upgraded WPR and TRAP command functions
- Upper compatibility with Q2SD on a functional level

#### (2) $4 \times 4$ Matrix Operation

A  $4 \times 4$  matrix is used to transform the input vertex. Setting the coordinate transformation enable bit (GTE) in the coordinate transformation control register (GTRCR) to 1 and then setting the rendering attribute MTRE bit of each command to 1 executes matrix operation for the input vertex.

Generally, the following matrix equation is used for transforming the coordinates of the input vertex.



$$\begin{aligned}
 \begin{pmatrix} TX \\ TY \\ TZ \\ W \end{pmatrix} &= \begin{pmatrix} 1 & 0 & P02 & P03 \\ 0 & 1 & P12 & P13 \\ 0 & 0 & P22 & P23 \\ 0 & 0 & P32 & P33 \end{pmatrix} \begin{pmatrix} m00 & m01 & m02 & m03 \\ m10 & m11 & m12 & m13 \\ m20 & m21 & m22 & m23 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \\
 &= \begin{pmatrix} m00 + P02m20 & m01 + P02m21 & m02 + P02m22 & m03 + P02m23 \\ m10 + P12m20 & m11 + P12m21 & m12 + P12m22 & m13 + P12m23 \\ P22m20 & P22m21 & P22m22 & P22m23 \\ P32m20 & P32m21 & P32m22 & P32m23 + P33 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}
 \end{aligned}$$

Here the input vertex Z is 0 and the TZ output is not used, so the synthesized matrix becomes as follows:

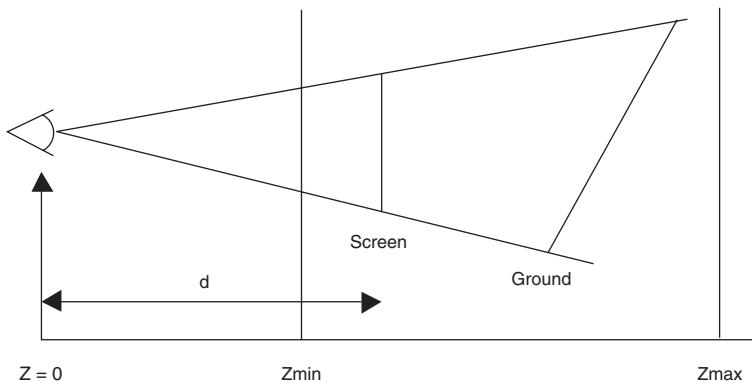
$$\begin{pmatrix} m00 + P02m20 & m01 + P02m21 & 0 & m03 + P02m23 \\ m10 + P12m20 & m11 + P12m21 & 0 & m13 + P12m23 \\ 0 & 0 & 0 & 0 \\ P32m20 & P32m21 & 0 & P32m20 + P33 \end{pmatrix}$$

The remaining nine parameters are set to the matrix parameter A to I registers (MTRAR to MTRIR), which are coordinate transformation control registers.

The relationship between the matrix parameter registers and matrix parameters is shown below.

$$\begin{pmatrix} A & B & 0 & C \\ D & E & 0 & F \\ 0 & 0 & 0 & 0 \\ G & H & 0 & I \end{pmatrix}$$

For example, when the view point is  $Z = 0$  as in the figure below, the parse transformation matrix becomes (1). Therefore, the synthesized matrix obtained by combining the parse transformation matrix with the affine transformation matrix becomes (2).



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} m00 & m01 & 0 & m03 \\ m10 & m11 & 0 & m13 \\ 0 & 0 & 0 & 0 \\ m20/d & m21/d & 0 & m23/d \end{bmatrix} \quad (2)$$

The above nine parameters are set to the matrix parameter A to I registers (MTRAR to MTRIR), which are coordinate transformation control registers.

MTRAR to MTRIR are set in the single-precision floating-point format defined by the IEEE 754 standard. Since internal operation is executed in the 32-bit fixed-point mode (16-bit integer portion and 16-bit fractional portion), parameters A to I should be set within the range of  $-2^{15} \leq \text{MTRAR to MTRIR} < 2^{15}$ . If a setting exceeds the above range when being transformed from a single-precision floating-point value into a 32-bit fixed-point value, saturation processing is executed. Note that parameters A to I must be set so that the matrix operation results TX, TY, and

W are within the ranges of  $-H'7FFF\ FFFF \leq TX, TY \leq H'7FFF\ FFFF$  and  $H'0000\ 0001 \leq W \leq H'7FFF\ FFFF$ . If TX, TY, or W exceeds the range of  $-H'7FFF\ FFFF \leq TX, TY, W \leq H'7FFF\ FFFF$ , the matrix operation error bit (MTRER) in the status register is set to 1 and saturation processing executed.

- Notes:
1. Before matrix operation, internally add the local offset to the input coordinates X and Y in the G2D.
  2. For a command with a relative coordinate specification, the coordinates are internally modified to the absolute coordinates in the G2D before matrix operation.
  3. Matrix operation is performed for only the center coordinates (BXC, BYC) in a BITBLT type command and for only the starting and final coordinate points in bold line drawing.
  4. The matrix operation error bit (MTRER) is not masked by the coordinate transformation enable bit (GTE) in the coordinate transformation control register (GTRCR) or the rendering attribute MTRE bit. Thus, when  $GTE = 0$  or when  $GTE = 1$  with  $MTRE = 0$ , the MTRER bit may be set to 1 even when coordinate transformation is not performed.

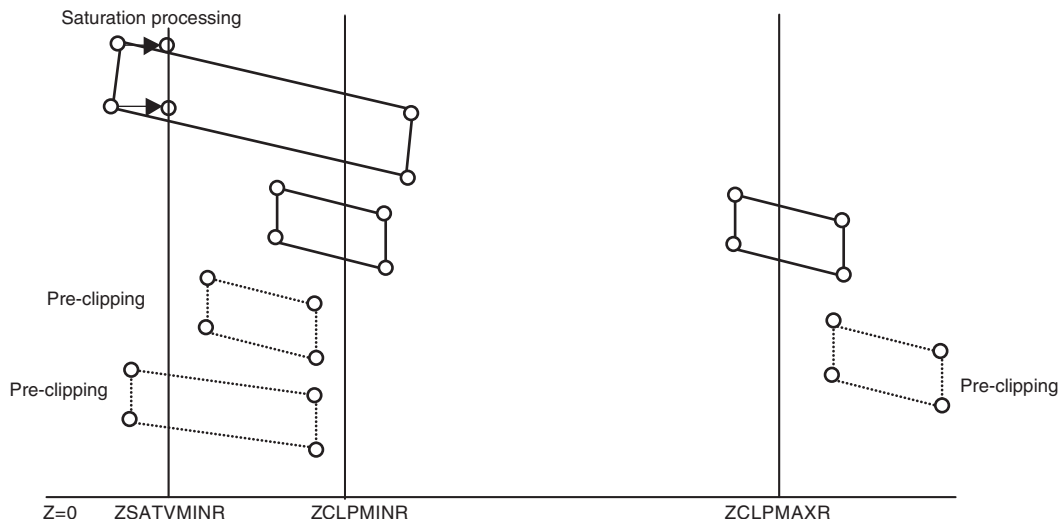
Therefore, throughout the period from rendering start to TRAP command issuance, do not use the MTRER bit unless both the GTE and MTRE bits are set to 1.

### (3) Z Clipping

Pre-clipping is performed when all vertices of the drawn figure (see the note below) are smaller than the value in the Z clipping area MIN register (ZCLPMINR) or greater than the value in the Z clipping area MAX register (ZCLPMAXR). Since the W value is used for comparison in Z clipping, set values corresponding to W in ZCLPMINR and ZCLPMAXR ( $Z_{min}/d$  and  $Z_{max}/d$  in the above example). If pre-clipping is not performed and the Z coordinate value is not greater than the value set in the Z saturation value MIN register (ZSATVMINR), saturation processing is performed to become the value set in ZSATVMINR. A value corresponding to W should also be set in ZSATVMINR. The figure will be deformed by saturation processing. Therefore, either set the matrix parameters to prevent the Z coordinate value from becoming equal to or lower than the ZSATVMINR value or divide the drawn figure beforehand and turn it into a display list so that it does not appear at screen coordinates. ZCLPMINR, ZCLPMAXR, and ZSATVMINR must be set in the single-precision floating-point format defined by the IEEE 754 standard. However, since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), the registers should be set within the range of  $2^{-16} \leq ZSATVMINR \leq ZCLPMINR \leq ZCLPMAXR < 2^{15}$ . If one of the registers is set outside this range, saturation processing is performed.

Z pre-clipping is not performed in a LINE, RLINE, LINEW, or RLINEW command to maintain the continuity of the pattern. As a result, a figure which does not appear on the screen coordinates may be drawn, thus degrading the performance. To prevent degraded performance, set the line pre-clipping enable bit (LPCE) in the rendering control register (RCLR) to 1. When the LPCE bit is set to 1, pre-clipping is performed in line-segment units in the 2-dimensional clipping areas (system clipping, user clipping, and relative user clipping areas) and the performance improved. However, if a line segment in the middle is pre-clipped, the pattern continuity is broken (the pattern starts from the final point of the line segment previously drawn).

**Note:** When enabling edge drawing ( $EDG = 1$ ) in the FTRAPC or RFTRAPC command, Z pre-clipping is not performed for the edge line.



Note : POLYGON4 type and CLRWC : Four vertices after coordinate transformation

BITBLT type, AAFA and AAFC : One vertex for center coordinates after coordinate transformation

(R)FTRAPC : Four vertices of circumscribed quadrangle after coordinate transformation

#### (4) Perspective W Division

TX and TY after matrix operation are processed as follows, according to the affine transformation enable bit (AFE) in the coordinate transformation control register (GTRCR).

- AFE = 0

The output X coordinate X' and output Y coordinate Y' becomes:  $X' = TX/WC + GTROFSX$ ,  $Y' = TY/WC + GTROFSY$ .

GTROFSX and GTROFSY are set in the coordinate transformation offset X register (GTROFSX) and coordinate transformation offset Y register (GTROFSY), respectively. GTROFSX and GTROFSY are set as 16-bit integers (two's complement).

- AFE = 1

The output X coordinate X' and output Y coordinate Y' becomes:  $X' = TX$ ,  $Y' = TY$ . Z clipping, perspective W division, and offset addition are not performed.

When AFE = 0, if the W division result TX/WC or TY/WC exceeds the corresponding range of  $-H'7FFF \leq TX/WC$ ,  $TY/WC \leq H'7FFF$ , saturation processing is performed. Then after adding the offset (GTROFSX or GTROFSY), if  $TX/WC + GTROFSX$  or  $TY/WC + GTROFSY$  exceeds the corresponding range of  $-H'7FFF \leq TX/WC + GTROFSX$ ,  $TY/WC + GTROFSY \leq H'7FFF$ , saturation processing is performed likewise. Even if saturation processing has been performed, the command is continuously executed at the vertex coordinates which have been saturated.

Note: In a BITBLT type command, the four vertices are obtained by adding the width (LW, RW) and height (TH, BH) to the center coordinate values after they have been transformed. In bold line drawing, the four vertices are obtained from the final and starting points of the transformed coordinates and the line width (W). Therefore, be careful the rendering coordinates are not exceeded in these two commands because saturation processing is not performed for vertices obtained from the reference points (center coordinates in a BITBLT type command, and starting and final coordinate points in bold line drawing).

## (5) Coordinate Transformation Flow and Saturation Processing

Coordinate transformation is performed in the following sequence.

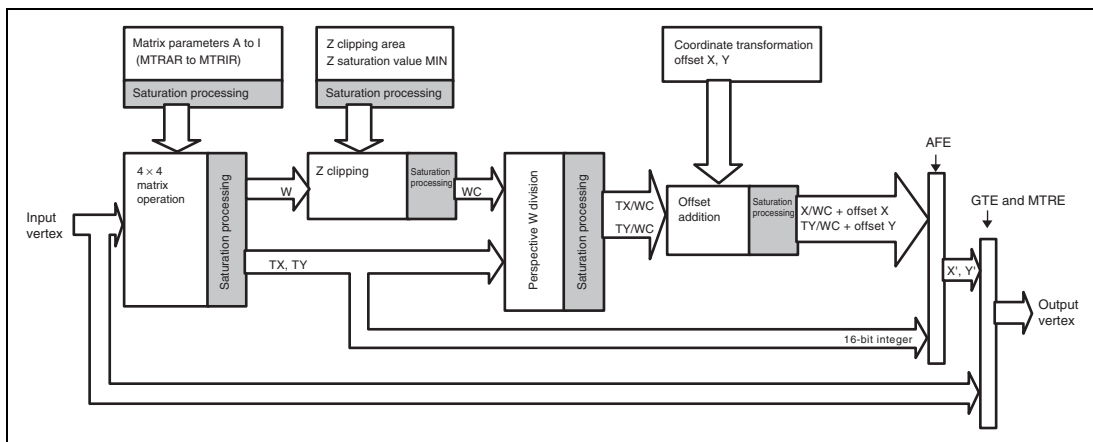


Figure 23.1 Coordinate Transformation Flow Image

**Table 23.3 Setting Ranges of Parameters Set by Registers and Saturation Processing**

Parameters Set by Registers	Setting Range	Saturation Processing
Matrix parameters A to I	$-2^{15} \leq$ Matrix parameters A to I $< 2^{15}$ (IEEE 754 standard single-precision floating-point)	When matrix parameters A to I $\geq 2^{15}$ : H'7FFF FFFF When matrix parameters A to I $< -2^{15}$ : -H'8000 0000 (32-bit fixed-point)
Z clipping area and Z saturation value MIN	$2^{-16} \leq$ Z clipping area, Z saturation value MIN $< 2^{15}$ (IEEE 754 standard single-precision floating-point)	When Z clipping area, Z saturation value MIN $\geq 2^{15}$ : H'7FFF FFFF When Z clipping area, Z saturation value MIN $< 2^{-16}$ : H'0000 0001 (32-bit fixed-point)
Coordinate transformation offset X, Y	$-2^{15} \leq$ Coordinate transformation offset X, Y $\leq 2^{15} - 1$ (16-bit integer (two's complement))	—

**Table 23.4 Vertex Ranges after Operation and Saturation Processing**

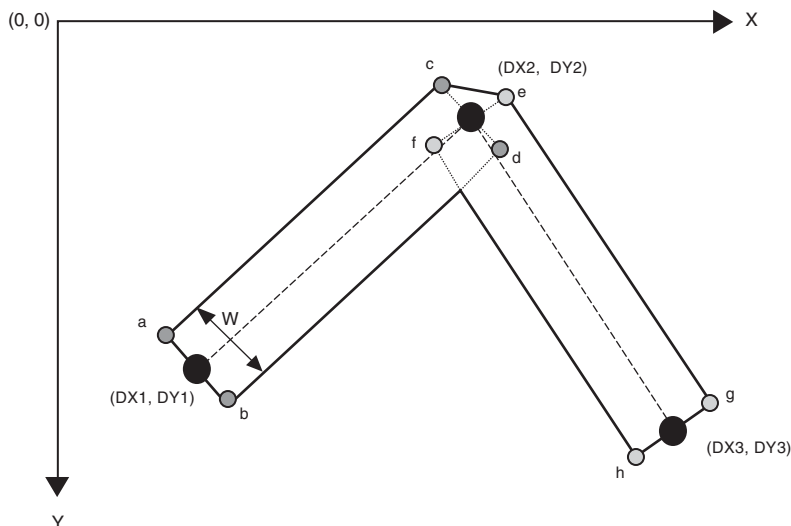
Vertex Coordinates after Operation	Range	Saturation Processing
TX, TY, W	$-H'7FFF\ FFFF \leq TX, TY, W \leq H'7FFF\ FFFF$ (32-bit fixed-point)	When TX, TY, W $> H'7FFF\ FFFF$ : H'7FFF FFFF When TX, TY, W $< -H'7FFF\ FFFF$ : -H'7FFF FFFF (32-bit fixed-point)
WC	Z saturation value MIN $\leq WC \leq H'7FFF\ FFFF$ (32-bit fixed-point)	When WC $< Z$ saturation value MIN: Z saturation value MIN (32-bit fixed-point)

Vertex Coordinates after Operation	Range	Saturation Processing
TX/WC, TY/WC and TX/WC + offset X, TY/WC + offset Y	$-H'7FFF \leq TX/WC, TY/WC \leq H'7FFF$ $-H'7FFF \leq TX/WC + \text{offset X},$ $TY/WC + \text{offset Y} \leq H'7FFF$ (16-bit integer)	When TX/WC, TY/WC > H'7FFF: H'7FFF  When TX/WC, TY/WC < -H'7FFF: -H'7FFF  When TX/WC + offset X, TY/WC + offset Y > H'7FFF: H'7FFF  When TX/WC + offset X, TY/WC + offset Y < -H'7FFF: -H'7FFF (16-bit integer)
X', Y'	$-H'7FFF \leq X', Y' \leq H'7FFF$ (16-bit integer)	—
Output vertex	$-H'8000 \leq \text{Output vertex} \leq H'7FFF$ (16-bit integer)	—

## (6) Bold Line Drawing

A bold line can be drawn by setting a value greater than 0 as line width W in a LINE type or RLINE type command. The bold line coordinates a, b, c, and d are obtained from the starting and final coordinate points and line width W, and the bold line drawn. W is set in the 6-bit integer part. When 0 is set in W, a line of line width 1 is drawn. The connection drawing mask bit (COM) in the rendering control register (RCLR) is used to select whether the linkage parts of bold lines are drawn or not. When the starting and final coordinate points of a line segment match in bold line drawing, nothing is drawn. In bold line drawing, specify the starting and final coordinate points in the range of  $-2^{15} + (W + 2) \leq x, y \leq 2^{15} - 1 - (W + 2)$  in the logical space.





## (7) Antialiasing

Antialiasing which reduces alias can be used in a LINE type or RLINE type command. Setting the rendering attribute antialias enable bit (AA) to 1 performs antialiasing.

When antialiasing is specified, specify the starting and final coordinate points in the range of  $-2^{15} + 1 \leq x, y \leq 2^{15} - 2$  in the logical space. In bold line drawing, specify the starting and final coordinate points in the range of  $-2^{15} + 1 + (W + 2) \leq x, y \leq 2^{15} - 2 - (W + 2)$ .

- For a dashed line, antialiasing is not performed for the gaps in the dashed line.
- When the starting and final coordinate points of a line segment match in the LINEA, LINEB, LINEC, RLINEA, RLINEB, or RLINEC command, a single dot is drawn for a 1-bit-wide line ( $W = 0$ ) without antialiasing and nothing is drawn for bold line drawing.
- When the starting and final coordinate points of a line segment match in the LINED, or RLINED command, nothing is drawn.
- Antialiasing is not performed for horizontal, vertical, and 45-degree diagonal line segments.



**Figure 23.2 Example of Antialias Specification**

### 23.1.3 Coordinate Systems

The G2D has four 2-dimensional coordinate systems (screen coordinates, rendering coordinates, 2-dimensional source coordinates, and work coordinates), and one 1-dimensional coordinate system (1-dimensional source coordinates).

Screen coordinates are the display control coordinates. Screen coordinate X corresponds to the horizontal dimension of the display screen and Y to the vertical dimension. The origin is the top-left corner in the display screen. The screen coordinate positive directions are right for the X-axis and down for the Y-axis. Either 16 bits (16 bits/pixel) or 8 bits (8 bits/pixel) can be selected as the data width of one screen coordinate.

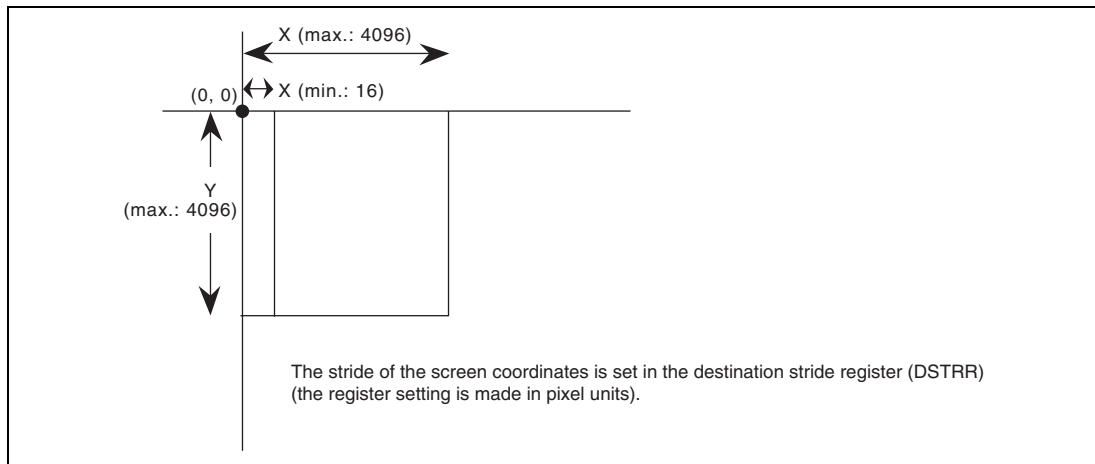
Rendering coordinates are drawing control coordinates. Rendering coordinates are shifted horizontally and vertically with respect to screen coordinates by the offset amounts specified in drawing commands. According to the drawing commands, the G2D performs drawing operations using these coordinates. Either 16 bits (16 bits/pixel) or 8 bits (8 bits/pixel) can be selected as the data width of one rendering coordinate.

2-dimensional source coordinates are drawing control coordinates. When a drawing command is executed with  $SS = 1$ , these are the source data (rectangle) coordinates specified by the drawing command. Either 16 bits (16 bits/pixel) or 8 bits (8 bits/pixel) can be selected as the data width of one 2-dimensional source coordinate.

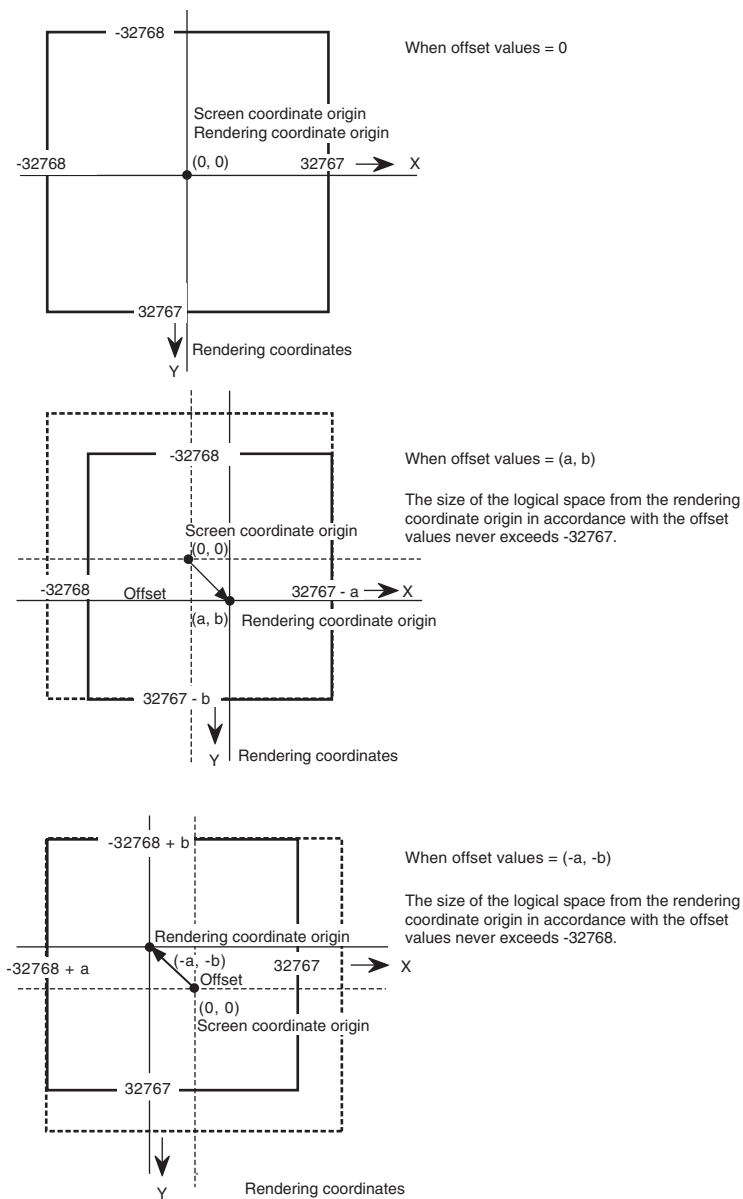
1-dimensional source coordinates are drawing control coordinates. When a drawing command is executed with  $SS = 0$ , these are the source data (1-dimensional) coordinates specified by the drawing command. 1 bit (1 bit/pixel), 16 bits (16 bits/pixel), or 8 bits (8 bits/pixel) can be selected as the data width of one 1-dimensional source coordinate. For one 1-dimensional source, one physical address (top-left) and the horizontal width and vertical height of the 1-dimensional source are specified.

Work coordinates are drawing control coordinates that correspond one-to-one with the rendering coordinates. When a drawing command is executed, these are the work coordinates specified by the drawing command. The data width of one work coordinate is 1 bit.

The maximum values of the screen coordinates are  $X = 4095$ ,  $Y = 4095$ .



**Figure 23.3 Screen Coordinates**



**Figure 23.4 Rendering Coordinates**

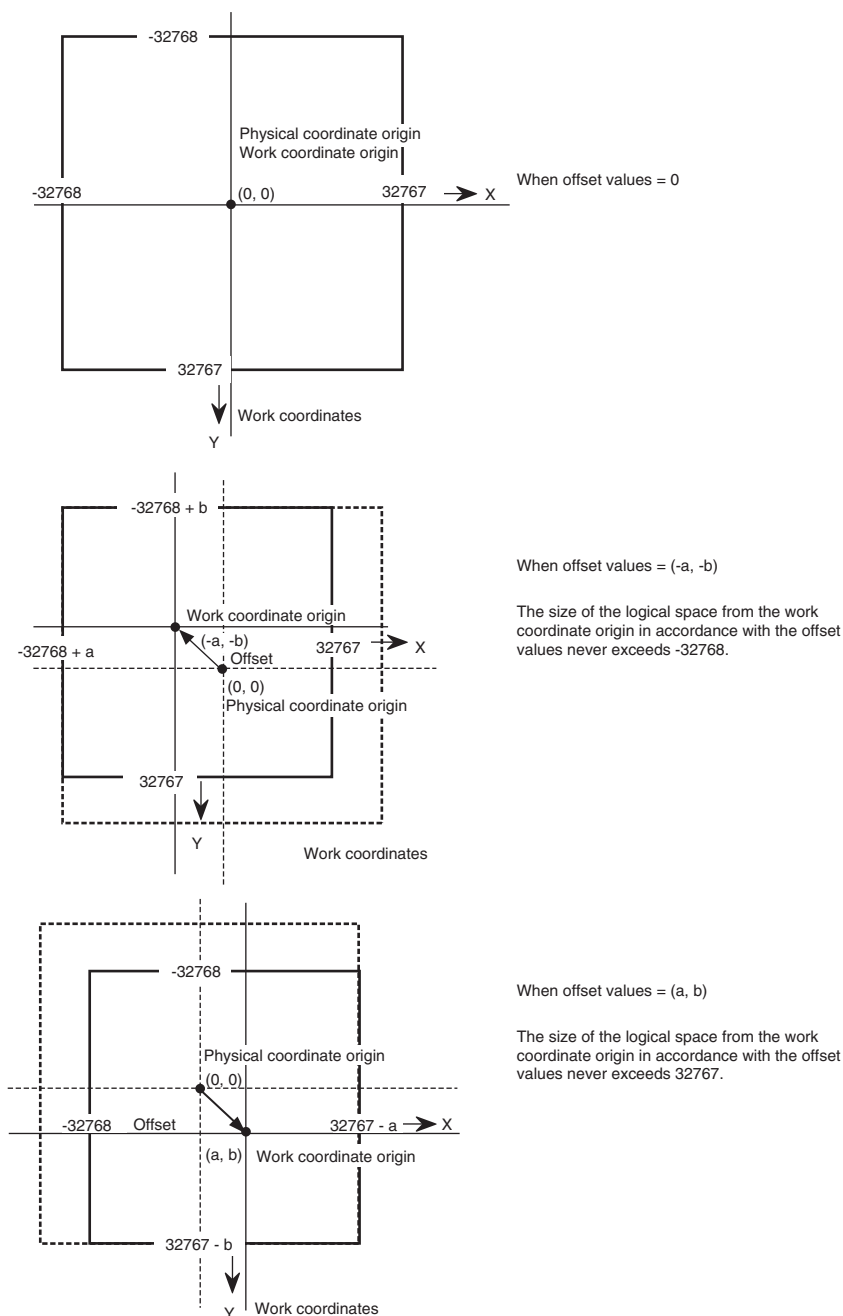
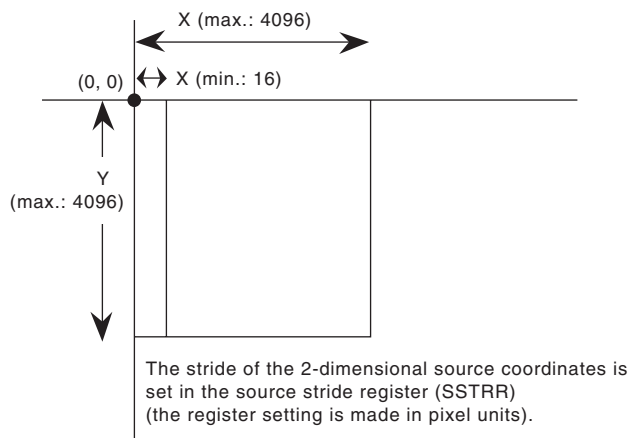
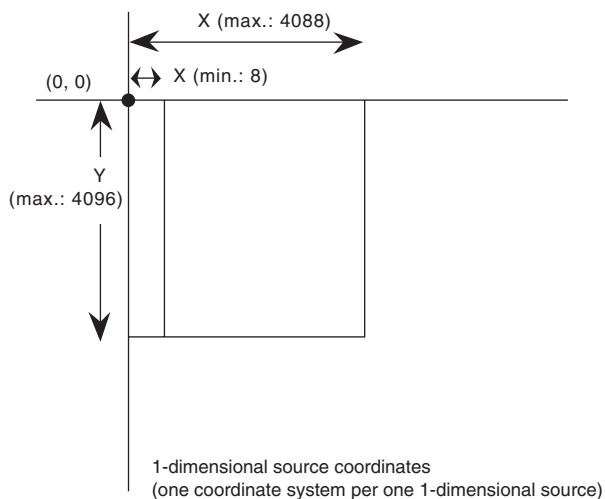


Figure 23.5 Work Coordinates



**Figure 23.6 2-Dimensional Source Coordinates (SS = 1)**



**Figure 23.7 1-Dimensional Source Coordinates (SS = 0)**

## 23.1.4 Data Formats

- 1-bit/pixel data

Bit	63	56 55	48 47	40 39	32 31	24 23	16 15	8 7	0
Pixel number	63	56 55	48 47	40 39	32 31	24 23	16 15	8 7	0

The pixel number is 0 at the left side of the screen, and increments as it shifts right.

- 8-bit/pixel data

Bit	63	56 55	48 47	40 39	32 31	24 23	16 15	8 7	0
Pixel number	7	6	5	4	3	2	1	0	

The pixel number is 0 at the left side of the screen, and increments as it shifts right.

- 16-bit/pixel data (RGB)

Bit	63	59 58	53 52	48 47	43 42	37 36	32 31	27 26	21 20	16 15	11 10	5 4	0
Pixel number	R3	G3	B3	R2	G2	B2	R1	G1	B1	R0	G0	B0	
	3			2			1				0		

The pixel number is 0 at the left side of the screen, and increments as it shifts right.

- 16-bit/pixel data (ARGB)

Bit	63 62	58 57	53 52	48 47	46 45	42 41	37 36	32 31	30 29	26 25	21 20	16 15	14 13	10 9	5 4	0
Pixel number	A3	R3	G3	B3	A2	R2	G2	B2	A1	R1	G1	B1	A0	R0	G0	B0
	3				2				1					0		

The pixel number is 0 at the left side of the screen, and increments as it shifts right.

- 32-bit data (display list)

Bit	63	32 31	0
	Adress 8n+4	Adress 8n	

### 23.1.5 Rendering Attributes

#### (1) Source Transparency Specification (STRANS)

When referencing source data, the STRANS bit can be used to select transparency or non-transparency on an individual drawing command basis. If transparency is selected, the source color becomes transparent at register value = source color when the source transparent color polarity bit (STP) in the rendering control register (RCLR) is 0, and the source color becomes transparent at register value  $\neq$  source color when the STP bit is 1, and the pixels are not drawn in either case. The source transparency specification can be used with the POLYGON4A, POLYGON4B, LINEA, LINEB, RLINEA, RLINEB, BITBLTA, and BITBLTB commands. The STRANS bit should be cleared to 0 in other commands. When the source pixel format is ARGB, the A value is not compared. Note that when the STRANS bit is set to 1, the source data is always read in the BITBLTA or BITBLTB command, regardless of the ROP code.

#### (2) Destination Transparency Specification (DTRANS)

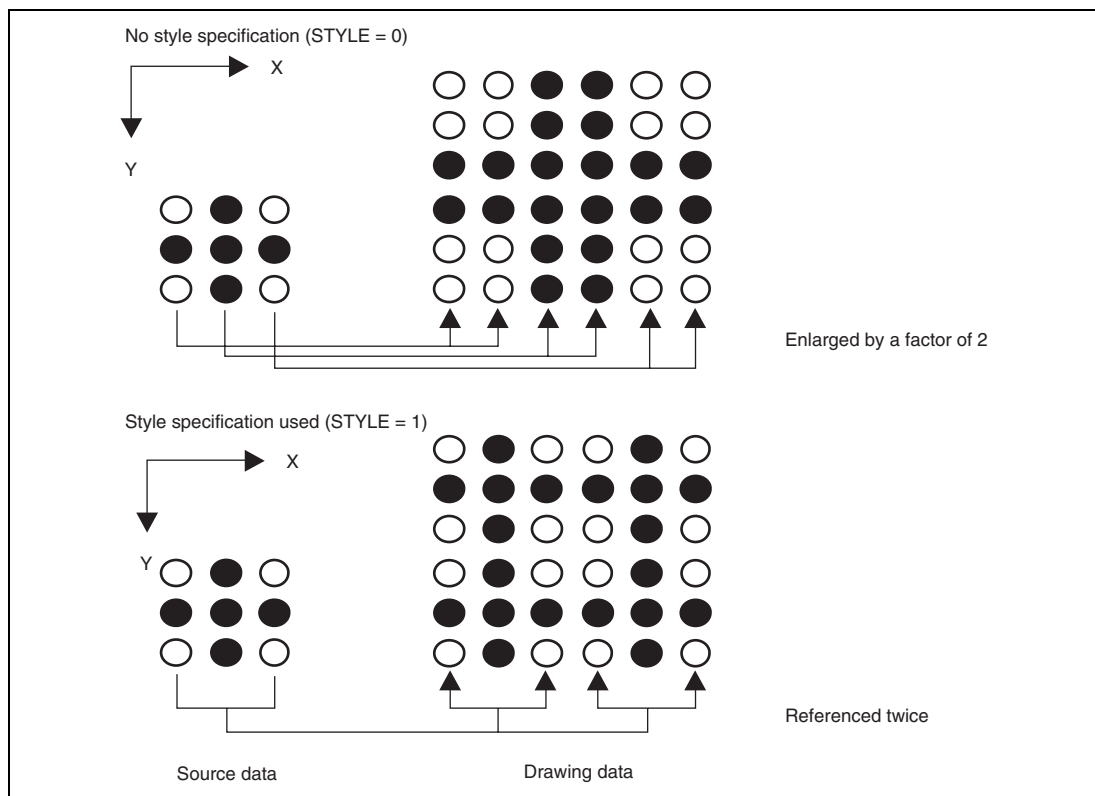
When referencing destination data, the DTRANS bit can be used to select transparency or non-transparency on an individual drawing command basis. If transparency is selected, the destination color becomes transparent at register value = destination color when the destination transparent color polarity bit (DTP) in the rendering control register (RCLR) is 0, and the destination color becomes transparent at register value  $\neq$  destination color when the DTP bit is 1, and the pixels are not drawn in either case. The destination transparency specification can be used with the BITBLTA, BITBLTB, and BITBLTC commands. The DTRANS bit should be cleared to 0 in other commands. When the destination pixel format is ARGB, the A value is not compared. Note that when the DTRANS bit is set to 1, the destination data is always read, regardless of the ROP code.

#### (3) Source Style Specification (STYLE)

The STYLE bit can be used to select, on an individual drawing command basis, whether to enlarge or reduce the source data or repeatedly reference it. If no style specification is made, the source data is enlarged or reduced in proportion to the size of the rendering area. When a style specification is made, the source data is referenced repeatedly in proportion to the size of the rendering area. This attribute is therefore used when drawing repeated patterns such as hatch patterns. The source style specification can be used with the POLYGON4A, POLYGON4B, LINEA, LINEB, RLINEA, and RLINEB commands. The STYLE bit should be cleared to 0 in other commands. The STYLE bit must be set to 1 when BLKE = 1 in the POLYGON4A, POLYGON4B, LINEA, LINEB, RLINEA, or RLINEB command.



In the LINEA, LINEB, RLINEA, and RLINEB commands, the source data is repeatedly referenced in only the X direction of the source data. The source data is enlarged or reduced in proportion to the line width in the Y direction of the source data.



**Figure 23.8 Example of Source Style Specification**

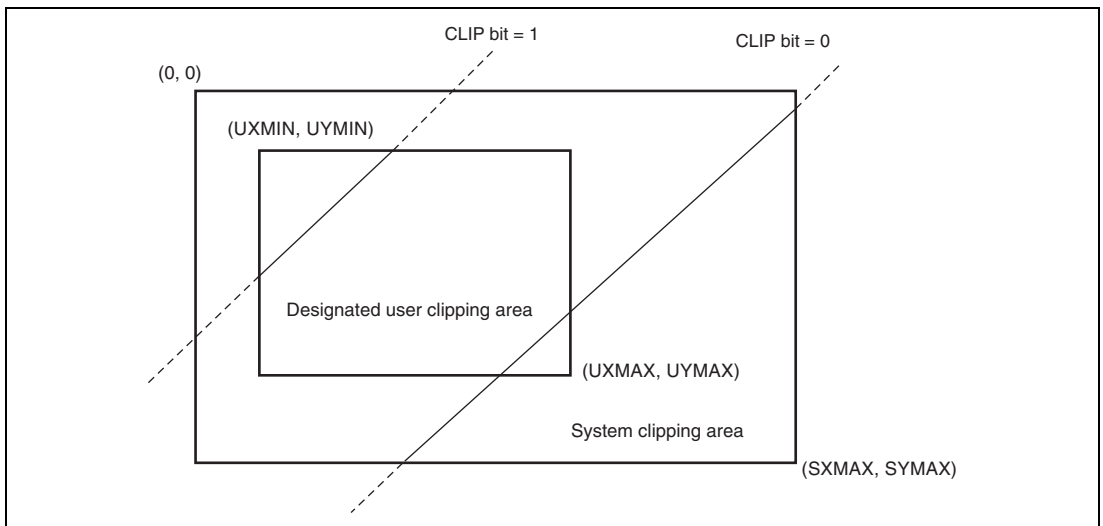
#### (4) Clipping Specification (CLIP)

The G2D can perform clipping area management. There are three kinds of clipping areas: system clipping area, user clipping area, and relative user clipping area.

The system clipping area has a fixed drawing range. The system clipping area is always valid, regardless of attribute specifications.

A user clipping area can be designated as desired within the system clipping area. Whether or not clipping is performed in that area can be selected on an individual command basis with the rendering attribute CLIP bit. The boundary is drawn. The local offset values specified by the LCOFS or RLCOFS command are not added. When setting a user clipping area, the following ranges must be satisfied:  $XMIN < XMAX$ ,  $YMIN < YMAX$ .

Clipping is set with screen coordinates. Since the clipping area is undefined after the power is turned on, set the clipping area by the WPR command at the top of the display list that is executed first. XMAX must be set to a value less than the value set in the destination stride register (DSTRR).



**Figure 23.9 Example of Clipping Specification**

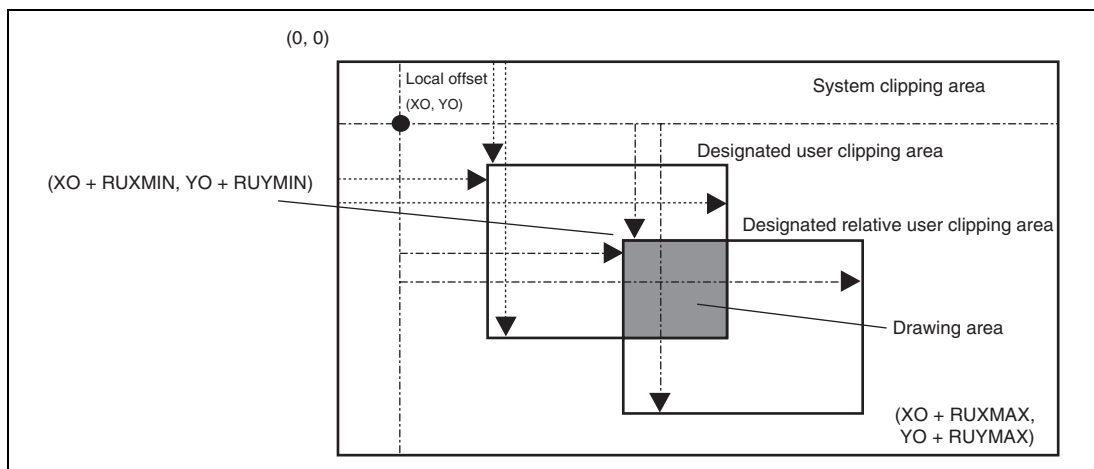
### (5) Relative Clipping Specification (RCLIP)

The G2D can perform clipping area management. There are three kinds of clipping areas: system clipping area, user clipping area, and relative user clipping area.

The system clipping area has a fixed drawing range. The system clipping area is always valid, regardless of attribute specifications.

A relative user clipping area can be designated as desired within the system clipping area at a relative setting with respect to the local offset. Whether or not clipping is performed in that area can be selected on an individual command basis with the rendering attribute RCLIP bit. The boundary is drawn. The local offset values specified by the LCOFS or RLCOFS command are added.

When setting a relative user clipping area, the following ranges must be satisfied:  $XMIN < XMAX$ ,  $YMIN < YMAX$ . Clipping is set with screen coordinates. Since the clipping area is undefined after the power is turned on, set the clipping area by the WPR command at the top of the display list that is executed first.  $XMAX$  must be set to a value less than the value set in the destination stride register (DSTRR). If both the RCLIP and CLIP bits are set to 1 simultaneously, the region where the two clipping areas overlap is drawn.



**Figure 23.10 Example of Relative User Clipping Specification**

When a relative user clipping area ((XO + RUXMIN, YO + RUYMIN) - (XO + RUXMAX, YO + RUYMAX)) intersects with the system clipping area, saturation processing is performed as follows:

$$XO + RUXMIN < 0 \rightarrow XO + RUXMIN = 0$$

$$XO + RUXMAX > SXMAX \rightarrow XO + RUXMAX = SXMAX$$

$$YO + RUYMIN < 0 \rightarrow YO + RUYMIN = 0$$

$$YO + RUYMAX > SYMAX \rightarrow YO + RUYMAX = SYMAX$$

Note: Set the local offset values and relative user clipping area without exceeding the following ranges:

$$-4096 \leq XO + RUXMIN \leq 4095$$

$$-4096 \leq YO + RUYMIN \leq 4095$$

$$0 \leq XO + RUXMAX \leq 8191$$

$$0 \leq YO + RUYMAX \leq 8191$$

When RCLIP = 1 and the relative user clipping area satisfies one of the following conditions, the relative user clipping area is disabled internally by the G2D (same operation as RCLIP = 0).

$$4095 < XO + RUXMIN$$

$$4095 < YO + RUYMIN$$

$$XO + RUXMAX < 0$$

$$YO + RUYMAX < 0$$

## (6) Net Drawing Specification (NET)

The NET bit can be used to select, on an individual drawing command basis, whether or not net drawing is to be performed. Net drawing is a function for drawing only pixels at coordinates for which the condition "rendering coordinates  $X + Y = \text{EOS}$  (0: even number, 1: odd number)" is true. For example, if  $\text{EOS} = 0$ , drawing is only performed on the pixels at coordinates  $Y = 0, X = 0, 2, 4, 6, 8, \dots$  and  $Y = 1, X = 1, 3, 5, 7, 9, \dots$

This function enables the drawn figure and ground to be mutually semi-composed.

The net drawing specification can be used with the POLYGON4 type, LINEA, LINEB, LINEC, RLINEA, RLINEB, and RLINEC commands. The NET bit should be cleared to 0 in other commands. The NET bit cannot be used together with the antialias enable bit (AA).

## (7) Even/Odd Select Specification (EOS)

Even pixels are selected when  $\text{EOS} = 0$ , and odd pixels when  $\text{EOS} = 1$ .

The even/odd select specification is used together with the net drawing specification (NET). With the LINEWC and RLINWC commands, drawing is performed at the work coordinates with 0 when  $\text{EOS} = 0$ , and with 1 when  $\text{EOS} = 1$ .

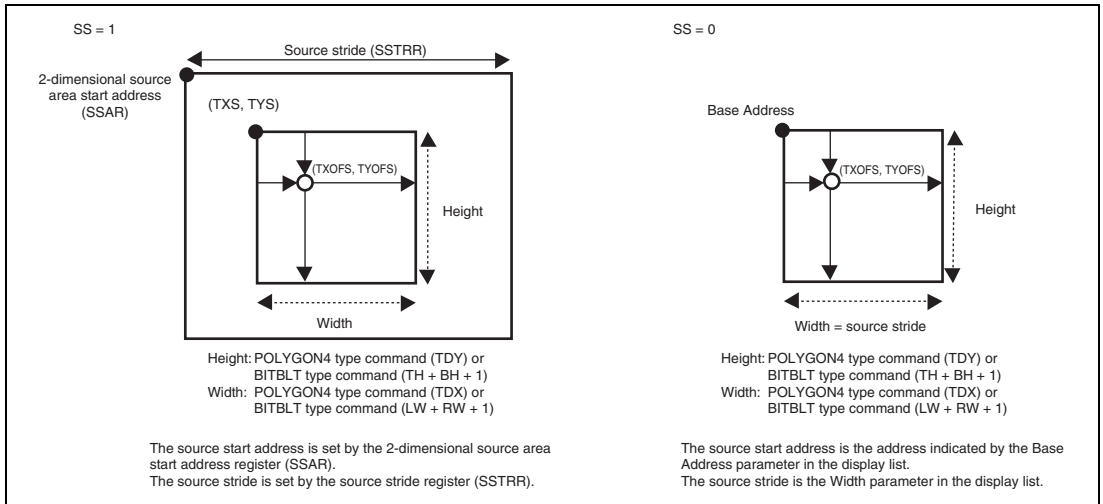
## (8) Work Specification (WORK)

When drawing is performed at rendering coordinates with the POLYGON4 type or BITBLT type command, the WORK bit can be used to select, on an individual drawing command basis, whether or not binary work data is to be referenced.

When binary work data referencing is selected, drawing is performed if the work data for the pixel corresponding to the rendering coordinates is 1, but not if the work data is 0. The same shape as that drawn at work coordinates can thus be drawn at rendering coordinates. Drawing at work coordinates can be performed either by means of the FTRAPC, RFTRAPC, LINEWC, RLINWC, or CLRWC command or else by the CPU. Ensure that memory drawing access by a command and memory drawing access by the CPU are not performed simultaneously. The work specification can be used with the POLYGON4 type, and BITBLT type, commands. The WORK bit should be cleared to 0 in the other commands.

## (9) Source Address Specification (SS)

The SS bit is used to select whether the source is to be referenced at a 2-dimensional source area address or at the address indicated by the Base Address parameter in the display list. The source address specification can be used with the POLYGON4A, POLYGON4B, BITBLTA, and BITBLTB commands. The SS bit should be clear to 0 in the other commands. If the offset values are set, the source is referenced from (TXOFS, TYOFS).



**Figure 23.11 Example of Source Address Specification**

**Note:** When SS = 1, settings must be made within the ranges of  $0 \leq \text{TXS} \leq \text{SSTRR} - \text{Width}$  ( $\text{TDX}, \text{LW} + \text{RW} + 1$ ),  $0 \leq \text{TYS} \leq 4096 - \text{Height}$  ( $\text{TDY}, \text{TH} + \text{BH} + 1$ ).

## (10) Source Coordinate Relative Address Specification (REL)

Setting the REL bit to 1 in the POLYGON4A, POLYGON4B, BITBLTA, BITBLTB, LINEA, LINEB, RLINEA, RLINEB, JUMP, and GOSUB commands enables source referencing and branching to be performed at an address relative to (before or after) the command code. Clear the SS bit to 0 in the POLYGON4A or BITBLTA command; correct operation is not guaranteed when the SS bit is set to 1.

The command code address is the origin of the relative address (longword address).

Note: With the POLYGON4A, POLYGON4B, BITBLTA, BITBLTB, LINEA, LINEB, RLINEA, and RLINEB commands, adding the address (longword: 32-bit units) where the command code is located to the source start relative address (longword: 32-bit units) must result in a quad word address (64-bit units).

## (11) Edge Drawing (EDG)

With the FTRAP and RFTRAP commands, setting the EDG bit to 1 enables edge lines to be drawn after completion of trapezoid painting to the work area. Whether edge line drawing is performed with 0 or with 1 is specified by the EOS bit.

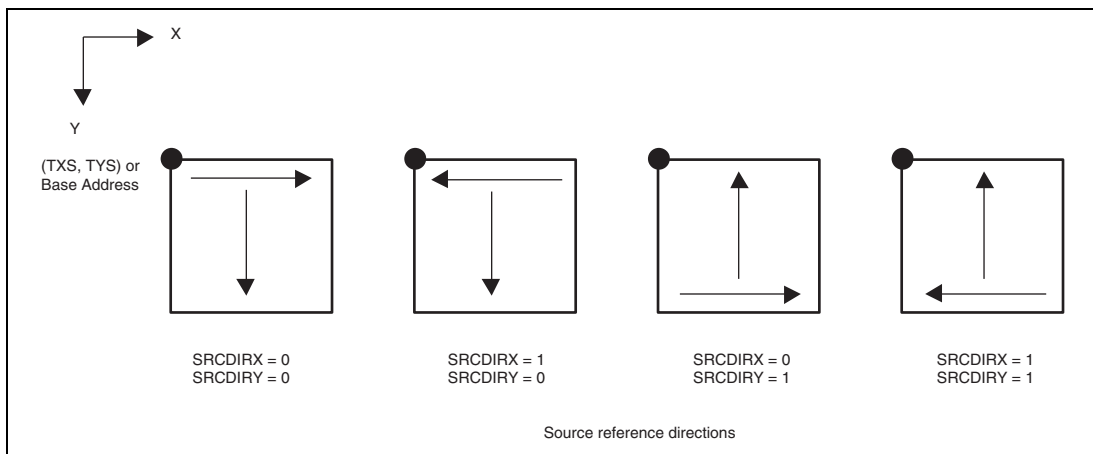
## (12) Color Offset (COOF)

The color offset specification can be used with the POLYGON4 type, LINEA, LINEB, LINEC, RLINEA, RLINEB, RLINEC, BITBLT type, and AAFA commands. In 16-bit/pixel drawing, if the rendering attribute COOF bit is set to 1, the result of adding the value in COFSR to the value of the source data (color expanded data for a binary source and the specified color for the monochrome specification) is drawn. In 8-bit/pixel drawing, the COOF bit should be cleared to 0. When the source pixel format is ARGB, the A value is not used in operation.

## (13) Source Direction X, Y (SRCDIRX, SRCDIRY)

The source direction X, Y specification can be used with the BITBLTA and BITBLTB commands. The directions in which to scan the source data are selected.

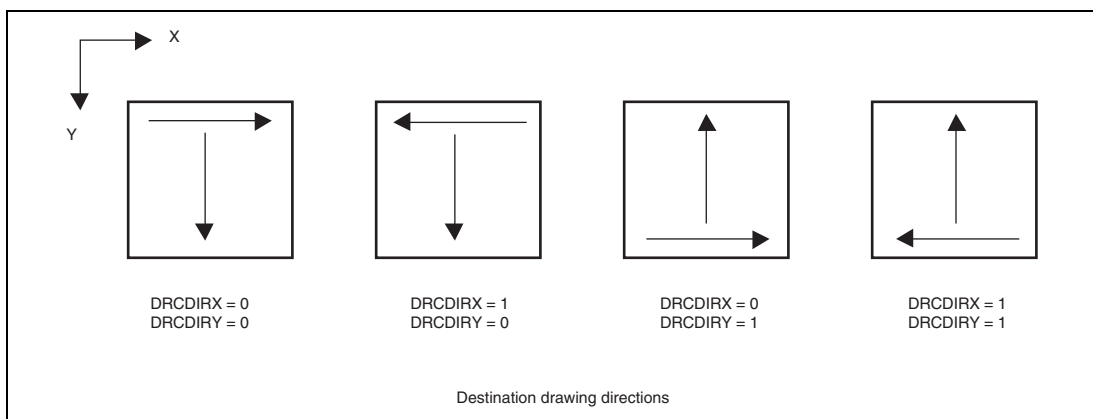
(TXS, TYS) or the Base Address specifies the top-left corner of the rectangle source, regardless of the source scan directions.



**Figure 23.12 Example of Source Direction Specification**

#### (14) Destination Direction X, Y (DSTDIRX, DSTDIRY)

The destination direction X, Y specification can be used with the BITBLTA, BITBLTB, and BITBLTC commands. The directions in which to draw the destination data are selected.



**Figure 23.13 Example of Destination Direction Specification**



### (15) Antialias Enable (AA)

The antialias enable specification can be used with the LINE type and RLINE type commands to reduce alias. The antialias enable specification is enabled only in 16-bit/pixel drawing. For 8-bit/pixel drawing, the AA bit should be cleared to 0. The AA bit should be set to 1 with the LINED and RLINED commands. The antialias enable specification cannot be used together with the net drawing specification (NET).

### (16) Alpha Blend Enable ( $\alpha$ E)

The alpha blend enable specification can be used with the POLYGON4 type and BITBLT type commands. The source data (color expanded data for a binary source and the specified color for the monochrome specification) and ground data are alpha blended and drawn. The alpha value is set in the alpha value register (ALPHR). The alpha blend enable specification is enabled only in 16-bit/pixel drawing. For 8-bit/pixel drawing, the  $\alpha$ E bit should be cleared to 0. In the POLYGON4 type commands, the alpha blend enable specification is enabled only when BLKE = 1. The  $\alpha$ E bit should be cleared to 0 when BLKE = 0. In the BITBLT type commands, the alpha blend enable specification is enabled only when the ROP code is H'CC (source copy). For other ROP codes, the  $\alpha$ E bit should be cleared to 0. The A value in the ARGB format is not alpha blended. The A value is drawn according to the source A value use (SAU) and A value (AVALUE) bits in the rendering control register (RCLR).

### (17) Source Alpha Enable (S $\alpha$ E)

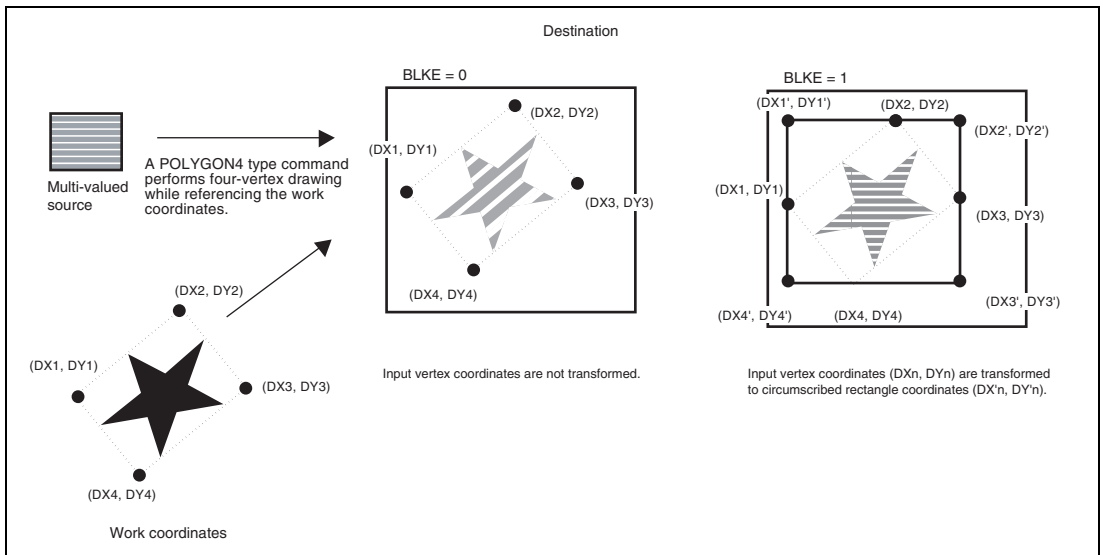
The source alpha enable specification can be used with the POLYGON4A, and BITBLTA commands. The S $\alpha$ E bit is used together with the alpha blend enable bit ( $\alpha$ E). When  $\alpha$ E = 0, the S $\alpha$ E bit should be cleared to 0. When the source pixel format bit (SPF) is 1 (ARGB format), only the pixels whose source A value is 1 are alpha blended. Pixels whose source A value is 0 are not alpha blended and the source data is drawn as it is. The source alpha enable specification is enabled only when SPF = 1. The S $\alpha$ E bit should be cleared to 0 when SPF = 0.

### (18) Block Enable (BLKE)

The block enable specification can be used with the POLYGON4 type commands. When BLKE = 1, the input vertex coordinates (DX<sub>n</sub>, DY<sub>n</sub>) are internally transformed to circumscribed rectangle coordinates (DX'<sub>n</sub>, DY'<sub>n</sub>) and four-vertex drawing performed. When coordinate transformation is to be performed, the transformed vertices are internally converted into a rectangle and drawn. This is effective for vertically pasting the pattern even after coordinate transformation. When BLKE = 1, the fixed drawing direction is from the upper-left corner to the lower-right corner (up-and-down and right-and-left directions cannot be reversed).

When coordinate transformation is performed by the CLRWC command, the four vertices are internally obtained from the input left and right X coordinate values and upper and lower Y coordinate values, and the coordinates for these four vertices are transformed. The transformed four vertices are then internally converted into a circumscribed rectangle drawn.

When coordinate transformation is performed by the FTRAPC or RFTRAPC command, the four vertices are internally obtained from the coordinate values for the circumscribed quadrangle of the input polygon, and the coordinates for these four vertices are transformed. The transformed four vertices are then internally converted into a circumscribed rectangle, the left edge obtained, and the polygon drawn. The BLKE bit should be set to 1 with the CLRWC, FTRAPC, and RFTRAPC commands.



**Figure 23.14 Example of Block Enable Specification**

### (19) Coordinate Transformation Enable (MTRE)

The coordinate transformation enable specification can be used with all drawing commands. Setting the MTRE bit to 1 when the coordinate transformation enable bit (GTE) in the coordinate transformation control register (GTRCR) is 1 performs coordinate transformation for the input vertex.

## **(20) Link Specification Enable (LINKE)**

The link specification enable specification can be used with the LINEC, LINED, RLINEC, RLINED, FTRAPC, RFTRAPC, and WPR commands. From the memory address specified by the LINK Address, the vertex coordinates are read with the LINEC, LINED, RLINEC, RLINED, FTRAPC, and RFTRAPC commands, and the register write data is read with the WPR command.

The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by the LINK Address.

## **(21) Link Address Relative Specification (LREL)**

The link address relative specification can be used with the LINEC, LINED, RLINEC, RLINED, FTRAPC, RFTRAPC, and WPR commands. The LREL bit is used together with the link specification enable bit (LINKE). The LREL bit should be cleared to 0 when LINKE = 0. The link destination address is specified as a relative address. The command code address is the origin of the relative address.

The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by the LINK Address.

## **(22) Clockwise (CLKW)**

The clockwise specification can be used with the LINED and RLINED commands. The CLKW bit is used to specify whether the order in giving the n vertices is clockwise or counterclockwise. The order is clockwise when CLKW = 1 and counterclockwise when CLKW = 0.

## **(23) Raster Operation (ROP)**

The raster operation specification can be used with the BITBLT type commands. The ROP code is specified in the ROP field, which is a BITBLT command parameter.

ROP Code	Operation
H'00	0
H'11	$\neg(S \mid D)$
H'22	$\neg S \& D$
H'33	$\neg S$
H'44	$S \& \neg D$
H'55	$\neg D$
H'66	$S \wedge D$
H'77	$\neg(S \& D)$
H'88	$S \& D$
H'99	$\neg(S \wedge D)$
H'AA	D
H'BB	$\neg S \mid D$
H'CC	S
H'DD	$S \mid \neg D$
H'EE	$S \mid D$
H'FF	1

Set the ROP code to H'CC when alpha blending is enabled ( $\alpha E = 1$ ). Neither alpha blending nor raster operation is performed for the A value in the ARGB format. The A value is drawn according to the source A value use (SAU) and A value (AVALUE) bits in the rendering control register (RCLR).

## 23.2 Display List

### 23.2.1 4-Vertex Screen Drawing Commands

#### (1) POLYGON4A

##### (a) Function

Performs any four-vertex drawing in the destination area while referencing a multi-valued (8- or 16-bit/pixel) source.

##### (b) Command Format

- SS = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1000_0010								Reserve (all 0)								Draw Mode															
0	0	0	0	TXS ( $0 \leq \text{TXS} \leq 4088$ )								0	0	0	0	TYS ( $0 \leq \text{TYS} \leq 4095$ )															
0	0	0	0	TDX ( $8 \leq \text{TDX} \leq 4095$ )								0	0	0	0	TDY ( $1 \leq \text{TDY} \leq 4095$ )															
0	0	0	0	TXOFS ( $0 \leq \text{TXOFS} \leq \text{TDX} - 1$ )								0	0	0	0	TYOFS ( $0 \leq \text{TYOFS} \leq \text{TDY} - 1$ )															
Sign				DX1 ( $-32768 \leq \text{DX1} \leq 32767$ )								Sign				DY1 ( $-32768 \leq \text{DY1} \leq 32767$ )															
Sign				DX2 ( $-32768 \leq \text{DX2} \leq 32767$ )								Sign				DY2 ( $-32768 \leq \text{DY2} \leq 32767$ )															
Sign				DX3 ( $-32768 \leq \text{DX3} \leq 32767$ )								Sign				DY3 ( $-32768 \leq \text{DY3} \leq 32767$ )															
Sign				DX4 ( $-32768 \leq \text{DX4} \leq 32767$ )								Sign				DY4 ( $-32768 \leq \text{DY4} \leq 32767$ )															

Note:  $0 \leq \text{TXS} \leq \text{SSTRR} - \text{TDX}$ ,  $0 \leq \text{TYS} \leq 4096 - \text{TDY}$  (SSTRR: Source stride register setting)

- SS = 0 and REL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1000_0010								Reserve (all 0)								Draw Mode															
0 0 0			Base Address (quad word address)																										0 0 0		
0 0 0 0				TDX ( $8 \leq \text{TDX} \leq 4088$ )								0 0 0 0 0 0 0 0				TDY ( $1 \leq \text{TDY} \leq 4095$ )															
0 0 0 0				TXOFS ( $0 \leq \text{TXOFS} \leq \text{TDX} - 1$ )												0 0 0 0				TYOFS ( $0 \leq \text{TYOFS} \leq \text{TDY} - 1$ )											
Sign				DX1 ( $-32768 \leq \text{DX1} \leq 32767$ )												Sign				DY1 ( $-32768 \leq \text{DY1} \leq 32767$ )											
Sign				DX2 ( $-32768 \leq \text{DX2} \leq 32767$ )												Sign				DY2 ( $-32768 \leq \text{DY2} \leq 32767$ )											
Sign				DX3 ( $-32768 \leq \text{DX3} \leq 32767$ )												Sign				DY3 ( $-32768 \leq \text{DY3} \leq 32767$ )											
Sign				DX4 ( $-32768 \leq \text{DX4} \leq 32767$ )												Sign				DY4 ( $-32768 \leq \text{DY4} \leq 32767$ )											

- SS = 0 and REL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
OP CODE = 1000_0010								Reserve (all 0)								Draw Mode																							
Sign extended		Sign		Base Address (longword address)																										0		0							
0		0		0		0		TDX (8 ≤ TDX ≤ 4088)								0		0		0		0		TDY (1 ≤ TDY ≤ 4095)															
0		0		0		0		TXOFS (0 ≤ TXOFS ≤ TDX – 1)								0		0		0		0		TYOFS (0 ≤ TYOFS ≤ TDY – 1)															
Sign				DX1 (-32768 ≤ DX1 ≤ 32767)												Sign				DY1 (-32768 ≤ DY1 ≤ 32767)																			
Sign				DX2 (-32768 ≤ DX2 ≤ 32767)												Sign				DY2 (-32768 ≤ DY2 ≤ 32767)																			
Sign				DX3 (-32768 ≤ DX3 ≤ 32767)												Sign				DY3 (-32768 ≤ DY3 ≤ 32767)																			
Sign				DX4 (-32768 ≤ DX4 ≤ 32767)												Sign				DY4 (-32768 ≤ DY4 ≤ 32767)																			

**Note:** Adding the address (longword: 32-bit units) where the command code is located to the Base Address (longword: 32-bit units) must result in a quad word address (64-bit units).

## 1. Code

B'10000010

## 2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
O		O (only WORK = 1)		O	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	STRANS	Fixed to 0	WORK	SS	REL	STYLE	BLKE	NET	EOS	COOF	αE	SαE

## 3. Command Parameters

TXS, TYS: Source starting point. Write 0 to the unused bits.

Base Address: Source start absolute address (Quad word address. Write 0 to bits A31 to A29 and A2 to A0.)

Source start relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

TDX, TDY: Source size. Write 0 to the unused bits.

DXn, DYn (n = 1 to 4): Rendering coordinates (absolute coordinates). Negative numbers expressed as two's complement.

TXOFS, TYOFS: Source offset. Write 0 to the unused bits.

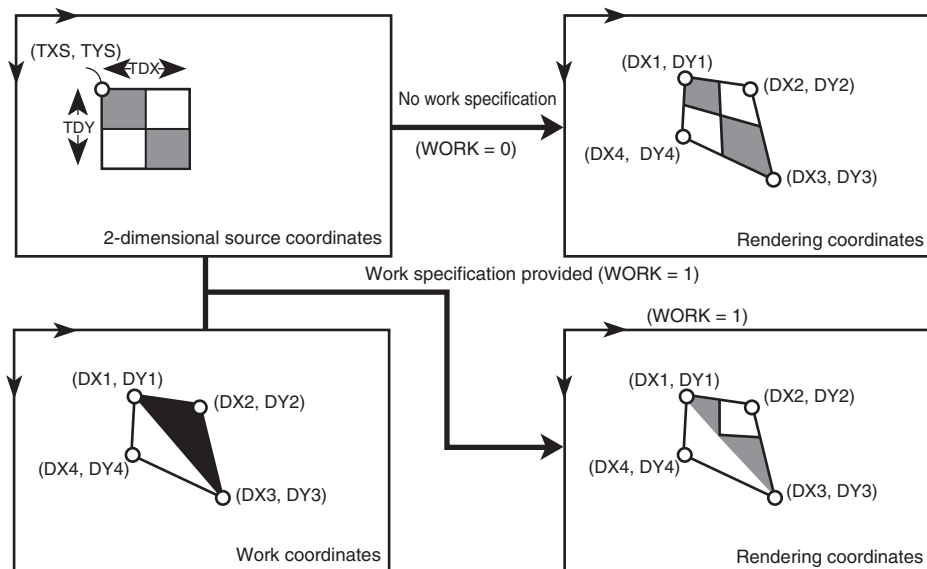
**(c) Description**

Transfers multi-valued (8- or 16-bit/pixel) source data to any quadrilateral rendering coordinates. The source data is always scanned horizontally, but diagonal scanning may be used in the drawing, depending on the shape. In diagonally-scanned drawing, double-writing occurs to fill in gaps.

When  $SS = 0$ , set a multiple of 8 pixels as the TDX value. When  $SS = 1$ , set 8 or more pixels as the TDX value. If the TDX setting is less than 8 pixels, multi-valued source references will not be performed normally. If TXOFS or TYOFS is set, the source at a location shifted by the offset amount is referenced. Make the TXOFS and TYOFS settings in pixel units.

1. When source style specification is selected as a rendering attribute ( $STYLE = 1$ ), the source data is not enlarged or reduced, but is referenced repeatedly.
2. When work specification is selected as a rendering attribute ( $WORK = 1$ ), only places where the work coordinate pixel is 1 are drawn at rendering coordinates while referencing work coordinates for the same coordinates as the rendering coordinates.
3. When  $SS = 1$ , the source data is referenced from the 2-dimensional source area. When  $SS = 0$ , the source data is referenced from the Base Address in the display list. When  $REL = 0$ , the source address can be specified as an absolute address. When  $REL = 1$ , the source address can be specified as a relative address with respect to the memory address at which the POLYGON4A command code is located.
4. In 16-bit/pixel drawing, if the rendering attribute COOF bit is set to 1, the result of adding the value in COFSR to the value of the multi-valued source data is drawn. The operation is performed by saturation processing. In 8-bit/pixel drawing, the COOF bit should be cleared to 0.



**(d) Example**

## (2) POLYGON4B

### (a) Function

Performs any four-vertex drawing in the destination area while referencing a binary (1-bit/pixel) source.

### (b) Command Format

- SS = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1000_0001								Reserve (all 0)								Draw Mode															
Color1																Color0															
0	0	0	0	TXS ( $0 \leq \text{TXS} \leq 4088$ )												0	0	0	0	TYS ( $0 \leq \text{TYS} \leq 4095$ )											
0	0	0	0	TDX ( $8 \leq \text{TDX} \leq 4088$ )								0	0	0	0	0	0	0	0	TDY ( $1 \leq \text{TDY} \leq 4095$ )											
0	0	0	0	TXOFS ( $0 \leq \text{TXOFS} \leq \text{TDX} - 1$ )												0	0	0	0	TYOFS ( $0 \leq \text{TYOFS} \leq \text{TDY} - 1$ )											
Sign				DX1 ( $-32768 \leq \text{DX1} \leq 32767$ )												Sign				DY1 ( $-32768 \leq \text{DY1} \leq 32767$ )											
Sign				DX2 ( $-32768 \leq \text{DX2} \leq 32767$ )												Sign				DY2 ( $-32768 \leq \text{DY2} \leq 32767$ )											
Sign				DX3 ( $-32768 \leq \text{DX3} \leq 32767$ )												Sign				DY3 ( $-32768 \leq \text{DY3} \leq 32767$ )											
Sign				DX4 ( $-32768 \leq \text{DX4} \leq 32767$ )												Sign				DY4 ( $-32768 \leq \text{DY4} \leq 32767$ )											

Note:  $0 \leq \text{TXS} \leq \text{SSTRR} - \text{TDX}$ ,  $0 \leq \text{TYS} \leq 4096 - \text{TDY}$  (SSTRR: Source stride register setting)

- SS = 0 and REL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
OP CODE = 1000_0001									Reserve (all 0)								Draw Mode															
Color1																Color0																
0	0	0	Base Address (quad word address)																										0	0	0	
0	0	0	0	TDX ( $8 \leq \text{TDX} \leq 4088$ )								0	0	0	0	0	0	0	TDY ( $1 \leq \text{TDY} \leq 4095$ )													
0	0	0	0	TXOFS ( $0 \leq \text{TXOFS} \leq \text{TDX} - 1$ )												0	0	0	0	TYOFS ( $0 \leq \text{TYOFS} \leq \text{TDY} - 1$ )												
Sign				DX1 ( $-32768 \leq \text{DX1} \leq 32767$ )												Sign				DY1 ( $-32768 \leq \text{DY1} \leq 32767$ )												
Sign				DX2 ( $-32768 \leq \text{DX2} \leq 32767$ )												Sign				DY2 ( $-32768 \leq \text{DY2} \leq 32767$ )												
Sign				DX3 ( $-32768 \leq \text{DX3} \leq 32767$ )												Sign				DY3 ( $-32768 \leq \text{DY3} \leq 32767$ )												
Sign				DX4 ( $-32768 \leq \text{DX4} \leq 32767$ )												Sign				DY4 ( $-32768 \leq \text{DY4} \leq 32767$ )												

- SS = 0 and REL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE = 1000_0001								Reserve (all 0)								Draw Mode																	
Color1																Color0																	
Sign extended		Sign		Base Address (longword address)																												0	0
0	0	0	0	TDX (8 ≤ TDX ≤ 4088)								0	0	0	0	0	0	0	TDY (1 ≤ TDY ≤ 4095)														
0	0	0	0	TXOFS (0 ≤ TXOFS ≤ TDX – 1)								0	0	0	0	TYOFS (0 ≤ TYOFS ≤ TDY – 1)																	
Sign		DX1 (-32768 ≤ DX1 ≤ 32767)														Sign		DY1 (-32768 ≤ DY1 ≤ 32767)															
Sign		DX2 (-32768 ≤ DX2 ≤ 32767)														Sign		DY2 (-32768 ≤ DY2 ≤ 32767)															
Sign		DX3 (-32768 ≤ DX3 ≤ 32767)														Sign		DY3 (-32768 ≤ DY3 ≤ 32767)															
Sign		DX4 (-32768 ≤ DX4 ≤ 32767)														Sign		DY4 (-32768 ≤ DY4 ≤ 32767)															

Note: Adding the address (longword: 32-bit units) where the command code is located to the Base Address (longword: 32-bit units) must result in a quad word address (64-bit units).

## 1. Code

B'10000001

## 2. Rendering Attributes

Reference Data					Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color		Rendering	Work
	0	0			0	
		(only WORK = 1)				

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	STRANS	Fixed to 0	WORK	SS	REL	STYLE	BLKE	NET	EOS	COOF	$\alpha E$	Fixed to 0

### 3. Command Parameters

TXS, TYS:	Source starting point. Write 0 to the unused bits.
Base Address:	Source start absolute address (Quad word address. Write 0 to bits A31 to A29 and A2 to A0.) Source start relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

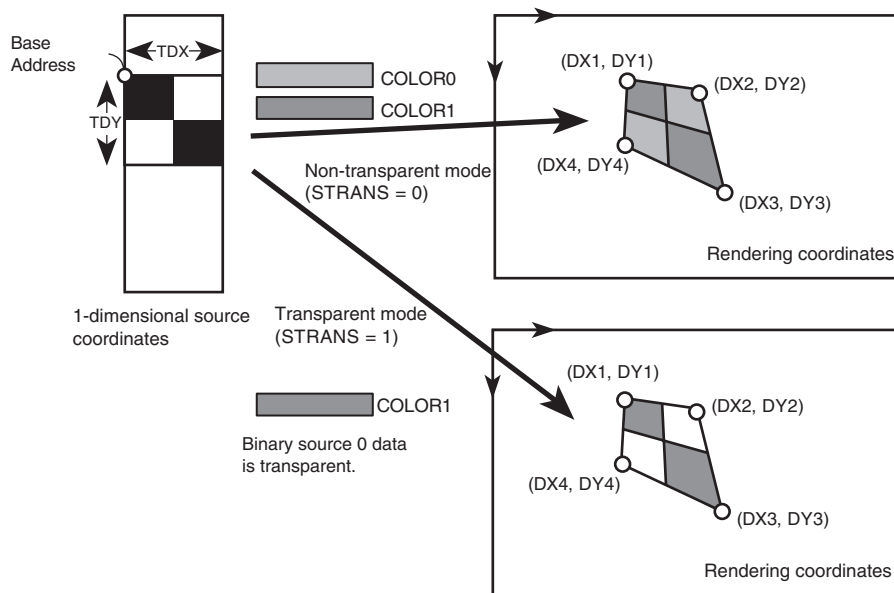
Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

TDX, TDY:	Source size. Write 0 to the unused bits.
DXn, DYn (n = 1 to 4):	Rendering coordinates (absolute coordinates). Negative numbers expressed as two's complement.
TXOFS, TYOFS:	Source offset. Write 0 to the unused bits.
Color0, Color1:	8- or 16-bit/pixel color specification. For 16-bit/pixel drawing, the color specification should match the destination pixel format. For 8-bit/pixel drawing, the same value should be set in the upper and lower bytes.

#### (c) Description

Draws binary (1-bit/pixel) source data in any quadrilateral rendering area, using the colors specified by parameters Color0 and Color1. For the color specifications (Color0 and Color1) in 8-bit/pixel drawing, set the same 8-bit data in the upper and lower bytes. The source data is always scanned horizontally, but diagonal scanning may be used in the drawing, depending on the shape. In diagonally-scanned drawing, double-writing occurs to fill in gaps. A multiple of 8 pixels must be set as the TDX value, regardless of the SS bit value. If TXOFS or TYOFS is set, the source at a location shifted by the offset amount is referenced. Make the TXOFS and TYOFS settings in pixel units.

1. When source style specification is selected as a rendering attribute (STYLE = 1), the source data is not enlarged or reduced, but is referenced repeatedly.
2. When work specification is selected as a rendering attribute (WORK = 1), only places where the work coordinate pixel is 1 are drawn at rendering coordinates while referencing work coordinates for the same coordinates as the rendering coordinates.
3. When REL = 0, the source address can be specified as an absolute address. When REL = 1, the source address can be specified as a relative address with respect to the memory address at which the POLYGON4B command code is located.

**(d) Example**

(3) POLYGON4C

(a) Function

Performs any four-vertex drawing at rendering coordinates with a monochrome specification.

(b) Command Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1000_0000								Reserve (all 0)								Draw Mode															
All 0																Color															
Sign				DX1 (-32768 ≤ DX1 ≤ 32767)												Sign				DY1 (-32768 ≤ DY1 ≤ 32767)											
Sign				DX2 (-32768 ≤ DX2 ≤ 32767)												Sign				DY2 (-32768 ≤ DY2 ≤ 32767)											
Sign				DX3 (-32768 ≤ DX3 ≤ 32767)												Sign				DY3 (-32768 ≤ DY3 ≤ 32767)											
Sign				DX4 (-32768 ≤ DX4 ≤ 32767)												Sign				DY4 (-32768 ≤ DY4 ≤ 32767)											

1. Code

B'10000000

2. Rendering Attributes

Reference Data						Drawing Destination	
Multi-Valued Source	Binary Source		Binary Work		Specified Color	Rendering	Work
			O		O	O	
			(only WORK = 1)				

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	Fixed to 0	WORK	Fixed to 0	Fixed to 0	Fixed to 0	BLKE	NET	EOS	COOF	αE	Fixed to 0

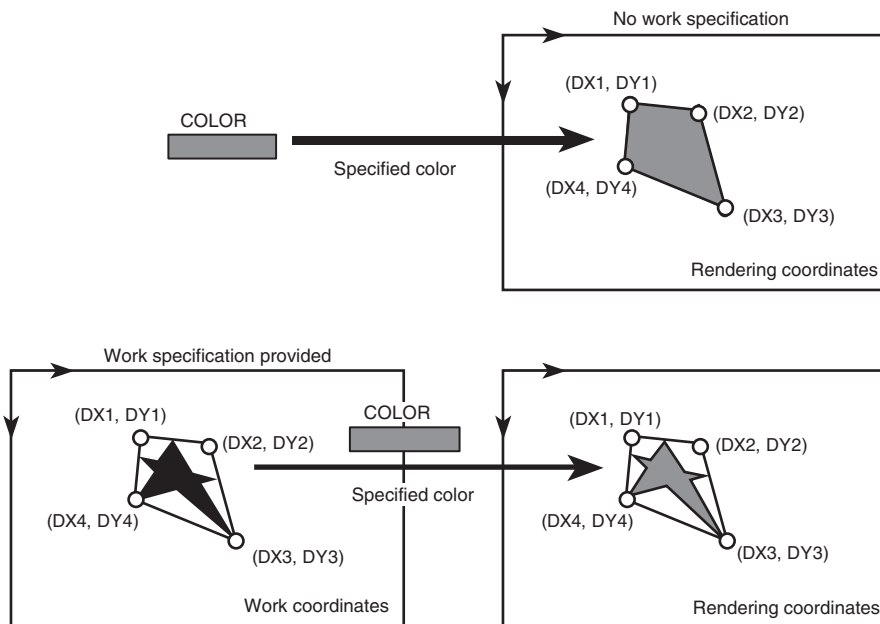
3. Command Parameters

DXn, DYn (n = 1 to 4): Rendering coordinates (absolute coordinates). Negative numbers expressed as two's complement.

Color: 8- or 16-bit/pixel color specification. For 16-bit/pixel drawing, the color specification should match the destination pixel format.  
For 8-bit/pixel drawing, the same value should be set in the upper and lower bytes.

**(c) Description**

Draws any quadrilateral in the rendering area in the single color specified by the Color parameter. When work specification is selected as a rendering attribute ( $WORK = 1$ ), only places where the work coordinate pixel is 1 are drawn at rendering coordinates while referencing work coordinates for the same coordinates as the rendering coordinates.

**(d) Example**

23.2.2 Line Drawing Commands

(1) LINEA

(a) Function

Draws a polygonal line with any width in the destination area while referencing a multi-valued (8- or 16-bit/pixel) source.

(b) Command Format

- REL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
OP CODE = 1011_0010								Reserve (all 0)								Draw Mode																		
0 0 0			Base Address (quad word address)																										0 0 0					
0 0 0			0			TDX (8 ≤ TDX ≤ 4088)								0 0 0			0 0 0			0 0 0			TDY (1 ≤ TDY ≤ 4095)											
0 0 0			0			TXOFS (0 ≤ TXOFS ≤ TDX – 1)																n (2 ≤ n ≤ 65535)												
Reserve (all 0)												0 0 0			0 0 0			0 0 0			0 0 0			0 0 0			W (0,2 ≤ W ≤ 63)							
Sign				DX1 (-32768 ≤ DX1 ≤ 32767)												Sign				DY1 (-32768 ≤ DY1 ≤ 32767)														
Sign				:												Sign				:														
Sign				:												Sign				:														
Sign				DXn (-32768 ≤ DXn ≤ 32767)												Sign				DYn (-32768 ≤ DYn ≤ 32767)														

Notes: 1. When W = 0, set TDY to 1.  
2. When n = 0 or 1, correct operation is not guaranteed.



- REL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE = 1011_0010									Reserve (all 0)							Draw Mode																	
Sign extended		Sign		Base Address (longword address)																												0	0
0	0	0	0	TDX ( $8 \leq TDX \leq 4088$ )									0	0	0	0	0	0	0	TDY ( $1 \leq TDY \leq 4095$ )													
0	0	0	0	TXOFS ( $0 \leq TXOFS \leq TDX - 1$ )																n ( $2 \leq n \leq 65535$ )													
Reserve (all 0)															0	0	0	0	0	0	0	0	0	0	W ( $0,2 \leq W \leq 63$ )								
Sign		DX1 ( $-32768 \leq DX1 \leq 32767$ )														Sign		DY1 ( $-32768 \leq DY1 \leq 32767$ )															
Sign		:														Sign		:															
Sign		:														Sign		:															
Sign		DXn ( $-32768 \leq DXn \leq 32767$ )														Sign		DYn ( $-32768 \leq DYn \leq 32767$ )															

- Notes:
- Adding the address (longword: 32-bit units) where the command code is located to the Base Address (longword: 32-bit units) must result in a quad word address (64-bit units).
  - When W = 0, set TDY to 1.
  - When n = 0 or 1, correct operation is not guaranteed.

## 1. Code

B'10110010

## 2. Rendering Attributes

Reference Data					Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color		Rendering	Work
0					0	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	STRANS	Fixed to 0	Fixed to 0	SS (0)	REL	STYLE (1)	Fixed to 0	NET	EOS	COOF	AA	Fixed to 0

- Notes:
- Clear the SS bit to 0.
  - Set the STYLE bit to 1.

### 3. Command Parameters

**Base Address:** Source start absolute address (Quad word address. Write 0 to bits A31 to A29 and A2 to A0.)  
Source start relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

**Note:** Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

**TDX, TDY:** Source size. Write 0 to the unused bits.  
**TXOFS:** Source offset. Write 0 to the unused bits.  
**n (n = 2 to 65,535):** Number of vertices  
**W:** Line width. Set a 6-bit integer. Write 0 to the unused bits.  
When 0 is set in W, a polygonal line of line width 1 is drawn. Setting 1 in W is prohibited.  
**DXn (n = 2 to 65,535):** Rendering coordinate (absolute coordinate). Negative number expressed as two's complement.  
**DYn (n = 2 to 65,535):** Rendering coordinate (absolute coordinate). Negative number expressed as two's complement.

#### (c) Description

Draws a polygonal line from vertex 1 (DX1, DY1), through vertex 2 (DX2, DY2), ..., vertex n – 1 (DXn – 1, DYn – 1), to vertex n (DXn, DYn). Set a multiple of 8 pixels as the TDX value. If TXOFS is set, the source at a location shifted by the offset amount is referenced. Make the TXOFS setting in pixel units. Pattern repetition selected by the STYLE bit is only performed in the X direction of the source data. The source data is enlarged or reduced in proportion to the line width in the Y direction.

When a value greater than 1 is set in W, a bold line can be drawn.

- Notes: 1. 8-point drawing is used for a line width of 1. Both of the 8-point drawing and 4-point drawing are used for bold line drawing.
2. The final point of each line segment is drawn. When the starting and final coordinate points of a line segment match, a single dot is drawn for a line width of 1 and nothing is drawn for bold line drawing
3. When AA = 1, note the following:
- For a dashed line, antialiasing is not performed for the gaps in the dashed line.
  - When the starting and final coordinate points of a line segment match, antialiasing is not performed.
  - Antialiasing is not performed for horizontal, vertical, and 45-degree diagonal line segments.

## (2) LINEB

### (a) Function

Draws a polygonal line with any width in the destination area while referencing a binary (1-bit/pixel) source.

### (b) Command Format

- REL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1011_0001								Reserve (all 0)								Draw Mode															
Color1																Color0															
0 0 0			Base Address (quad word address)																										0 0 0		
0 0 0 0				TDX (8 ≤ TDX ≤ 4088)								0 0 0 0 0 0 0 0								TDY (1 ≤ TDY ≤ 4095)											
0 0 0 0				TXOFS (0 ≤ TXOFS ≤ TDX – 1)																n (2 ≤ n ≤ 65535)											
Reserve (all 0)																0 0 0 0 0 0 0 0								W (0,2 ≤ W ≤ 63)							
Sign		DX1 (-32768 ≤ DX1 ≤ 32767)														Sign		DY1 (-32768 ≤ DY1 ≤ 32767)													
Sign		:														Sign		:													
Sign		:														Sign		:													
Sign		DXn (-32768 ≤ DXn ≤ 32767)														Sign		DYn (-32768 ≤ DYn ≤ 32767)													

- Notes: 1. When W = 0, set TDY to 1.
2. When n = 0 or 1, correct operation is not guaranteed.

- REL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE = 1011_0001								Reserve (all 0)								Draw Mode																			
Color1																Color0																			
Sign extended		Sign		Base Address (longword address)																		0		0											
0		0		0		0		TDX ( $8 \leq TDX \leq 4088$ )								0		0		0		0		0		0		TDY ( $1 \leq TDY \leq 4095$ )							
0		0		0		0		TXOFS ( $0 \leq TXOFS \leq TDX - 1$ )																n ( $2 \leq n \leq 65535$ )											
Reserve (all 0)																0		0		0		0		0		0		0		0		0		W ( $0,2 \leq W \leq 63$ )	
Sign		DX1 ( $-32768 \leq DX1 \leq 32767$ )														Sign		DY1 ( $-32768 \leq DY1 \leq 32767$ )																	
Sign		:														Sign		:																	
Sign		:														Sign		:																	
Sign		DXn ( $-32768 \leq DXn \leq 32767$ )														Sign		DYn ( $-32768 \leq DYn \leq 32767$ )																	

- Notes:
- Adding the address (longword: 32-bit units) where the command code is located to the Base Address (longword: 32-bit units) must result in a quad word address (64-bit units).
  - When W = 0, set TDY to 1.
  - When n = 0 or 1, correct operation is not guaranteed.

- Code  
B'10110001
- Rendering Attributes

Reference Data								Drawing Destination							
Multi-Valued Source		Binary Source				Binary Work		Specified Color		Rendering		Work			
		O								O					

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	STRANS	Fixed to 0	Fixed to 0	SS (0)	REL	STYLE (1)	Fixed to 0	NET	EOS	COOF	AA	Fixed to 0

- Notes:
- Clear the SS bit to 0.
  - Set the STYLE bit to 1.

### 3. Command Parameters

Color0, Color1:	8- or 16-bit/pixel color specification. For 16-bit/pixel drawing, the color specification should match the destination pixel format. For 8-bit/pixel drawing, the same value should be set in the upper and lower bytes.
Base Address:	Source start absolute address (Quad word address. Write 0 to bits A31 to A29 and A2 to A0. Source start relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

TDX, TDY:	Source size. Write 0 to the unused bits.
TXOFS:	Source offset. Write 0 to the unused bits.
n (n = 2 to 65,535):	Number of vertices
W:	Line width. Set a 6-bit integer. Write 0 to the unused bits. When 0 is set in W, a polygonal line of line width 1 is drawn. Setting 1 in W is prohibited.
DXn (n = 2 to 65,535):	Rendering coordinate (absolute coordinate). Negative number expressed as two's complement.
DYn (n = 2 to 65,535):	Rendering coordinate (absolute coordinate). Negative number expressed as two's complement.

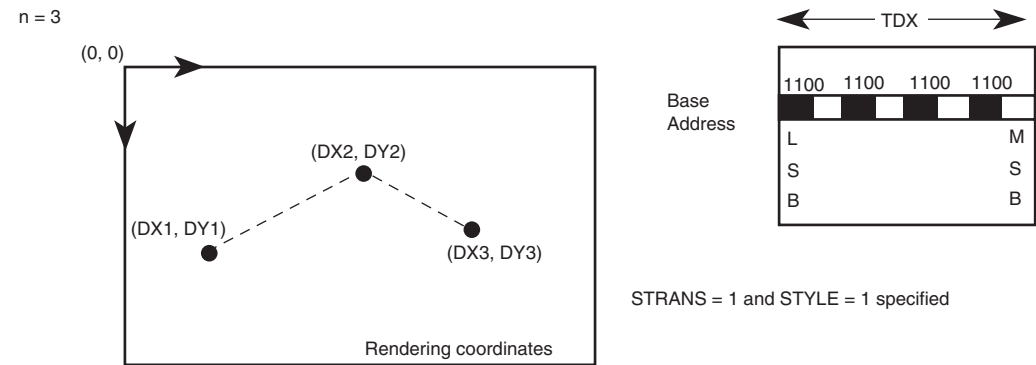
#### (c) Description

Draws a polygonal line from vertex 1 (DX1, DY1), through vertex 2 (DX2, DY2), ..., vertex  $n - 1$  (DX $n - 1$ , DY $n - 1$ ), to vertex n (DXn, DYn). Set a multiple of 8 pixels as the TDX value. If TXOFS is set, the source at a location shifted by the offset amount is referenced. Make the TXOFS setting in pixel units. Pattern repetition selected by the STYLE bit is only performed in the X direction of the source data. The source data is enlarged or reduced in proportion to the line width in the Y direction.

When a value greater than 1 is set in W, a bold line can be drawn.

- Notes:
- 1. 8-point drawing is used for a line width of 1. Both of the 8-point drawing and 4-point drawing are used for bold line drawing.
  - 2. The final point of each line segment is drawn. When the starting and final coordinate points of a line segment match, a single dot is drawn for a line width of 1 and nothing is drawn for bold line drawing.
  - 3. When AA = 1, note the following:
    - For a dashed line, antialiasing is not performed for the gaps in the dashed line.
    - When the starting and final coordinate points of a line segment match, antialiasing is not performed.
    - Antialiasing is not performed for horizontal, vertical, and 45-degree diagonal line segments.

(d) Example



### (3) LINEC

#### (a) Function

Draws a polygonal line with any width in the destination area with a monochrome specification.

#### (b) Command Format

- LINKE = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE = 1011_0000									Reserve (all 0)							Draw Mode																	
Color																n (2 ≤ n ≤ 65535)																	
Reserve (all 0)																0	0	0	0	0	0	0	0	0	0	0	W (0,2 ≤ W ≤ 63)						
Sign		DX1 (-32768 ≤ DX1 ≤ 32767)														Sign		DY1 (-32768 ≤ DY1 ≤ 32767)															
Sign		:														Sign		:															
Sign		:														Sign		:															
Sign		DXn (-32768 ≤ DXn ≤ 32767)														Sign		DYn (-32768 ≤ DYn ≤ 32767)															

Note: When  $n = 0$  or  $1$ , correct operation is not guaranteed.

- LINKE = 1 and LREL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
OP CODE = 1011_0000									Reserve (all 0)								Draw Mode															
Color																n (2 ≤ n ≤ 65535)																
Reserve (all 0)																0	0	0	0	0	0	0	0	0	0	0	W (0,2 ≤ W ≤ 63)					
0	0	0	LINK Address (longword address)																												0	0

- Notes:
1. When  $n = 0$  or  $1$ , correct operation is not guaranteed.
  2. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by the LINK Address.

- LINKE = 1 and LREL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1011_0000									Reserve (all 0)								Draw Mode														
Color																n (2 ≤ n ≤ 65535)															
Reserve (all 0)									0 0 0 0 0 0 0 0 0 0 0 0									W (0,2 ≤ W ≤ 63)													
Sign extended		Sign		LINK Address (longword address)																		0 0									

- Notes:
1. When  $n = 0$  or  $1$ , correct operation is not guaranteed.
  2. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by adding the address where the command code is located to the LINK Address.

### 1. Code

B'10110000

### 2. Rendering Attributes

Reference Data					Drawing Destination	
Multi-Valued Source	Binary Source		Binary Work		Specified Color	Rendering Work
					0	0

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	LINKE	LREL	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	NET	EOS	COOF	AA	Fixed to 0



### 3. Command Parameters

Color:	8- or 16-bit/pixel color specification. For 16-bit/pixel drawing, the color specification should match the destination pixel format. For 8-bit/pixel drawing, the same value should be set in the upper and lower bytes.
n (n = 2 to 65,535):	Number of vertices
W:	Line width. Set a 6-bit integer. Write 0 to the unused bits. When 0 is set in W, a polygonal line of line width 1 is drawn. Setting 1 in W is prohibited.
DXn (n = 2 to 65,535):	Rendering coordinate (absolute coordinate). Negative number expressed as two's complement.
DYn (n = 2 to 65,535):	Rendering coordinate (absolute coordinate). Negative number expressed as two's complement.
LINK Address:	LINK absolute address (Longword address. Write 0 to bits A31 to A29, A1, and A0.) LINK relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

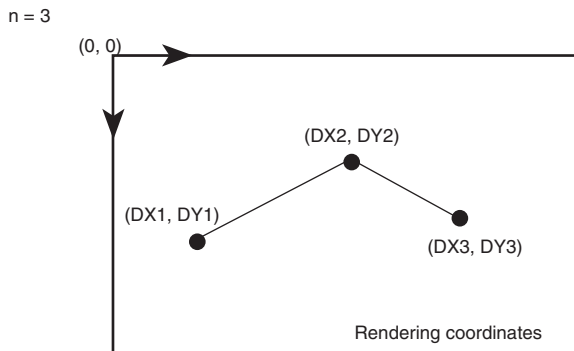
Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

#### (c) Description

Draws a polygonal line from vertex 1 (DX1, DY1), through vertex 2 (DX2, DY2), ..., vertex n – 1 (DXn – 1, DYn – 1), to vertex n (DXn, DYn). When a value greater than 1 is set in W, a bold line can be drawn. When LINKE = 1, the vertex coordinates are read from the memory address specified by the LINK Address. The LINK Address can be specified through the LREL bit as an absolute address or a relative address with respect to the memory address at which the LINEC command code is located.

- Notes:
1. 8-point drawing is used for a line width of 1. Both of the 8-point drawing and 4-point drawing are used for bold line drawing.
  2. The final point of each line segment is drawn. When the starting and final coordinate points of a line segment match, a single dot is drawn for a line width of 1 and nothing is drawn for bold line drawing.
  3. When  $AA = 1$ , note the following:
    - When the starting and final coordinate points of a line segment match, antialiasing is not performed.
    - Antialiasing is not performed for horizontal, vertical, and 45-degree diagonal line segments.

**(d) Example**



## (4) LINED

### (a) Function

Performs antialiasing for the exterior frame of a polygon. This command can only be executed for a 16-bit/pixel destination.

### (b) Command Format

- LINKE = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1011_0011								Reserve (all 0)								Draw Mode															
Reserve (all 0)																n (2 ≤ n ≤ 65535)															
Sign								DX1 (-32768 ≤ DX1 ≤ 32767)								Sign								DY1 (-32768 ≤ DY1 ≤ 32767)							
Sign								:								Sign								:							
Sign								:								Sign								:							
Sign								DXn (-32768 ≤ DXn ≤ 32767)								Sign								DYn (-32768 ≤ DYn ≤ 32767)							

Note: When  $n = 0$  or  $1$ , correct operation is not guaranteed.

- LINKE = 1 and LREL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1011_0011								Reserve (all 0)								Draw Mode															
Reserve (all 0)																n (2 ≤ n ≤ 65535)															
0	0	0	LINK Address (longword address)																										0	0	

- Notes:
1. When  $n = 0$  or  $1$ , correct operation is not guaranteed.
  2. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by the LINK Address.

- LINKE = 1 and LREL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1011_0011								Reserve (all 0)								Draw Mode															
Reserve (all 0)																n (2 ≤ n ≤ 65535)															
Sign extended		Sign		LINK Address (longword address)																		0		0							

- Notes:
1. When  $n = 0$  or  $1$ , correct operation is not guaranteed.
  2. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by adding the address where the command code is located to the LINK Address.

### 1. Code

B'10110011

### 2. Rendering Attributes

Reference Data					Drawing Destination	
Multi-Valued Source	Binary Source		Binary Work		Specified Color	
					Rendering	Work
						○

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	LINKE	LREL	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	AA (1)	CLKW

Note: Set the AA bit to 1.

### 3. Command Parameters

- $n$  ( $n = 2$  to  $65,535$ ): Number of vertices
- $DXn$  ( $n = 2$  to  $65,535$ ): Rendering coordinate (absolute coordinate). Negative number expressed as two's complement.
- $DYn$  ( $n = 2$  to  $65,535$ ): Rendering coordinate (absolute coordinate). Negative number expressed as two's complement.
- LINK Address: LINK absolute address (Longword address. Write 0 to bits A31 to A29, A1, and A0).  
LINK relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

#### (c) Description

Performs antialiasing for the exterior frame of a polygon drawn using work reference.

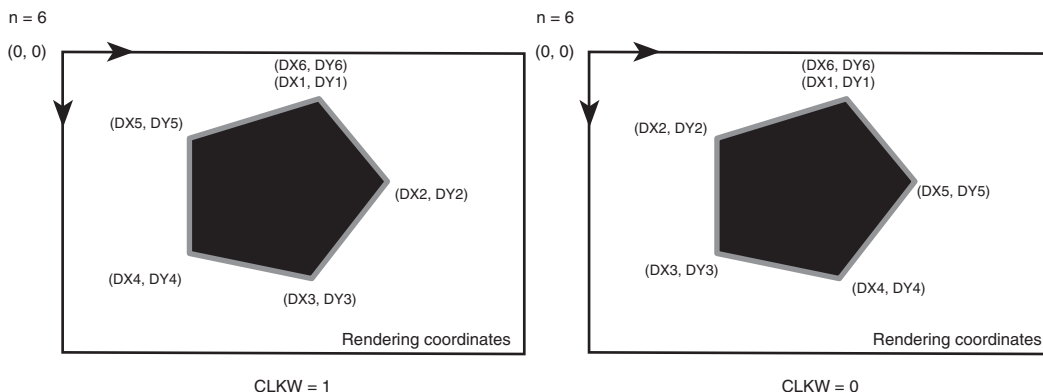
The CLKW bit specifies whether the order to give the  $n$  vertices is clockwise or counterclockwise: CLKW = 1 selects clockwise and CLKW = 0 selects counterclockwise. When clockwise is specified, the left image with respect to the drawing direction is referenced by antialiasing. On the other hand, the right image is referenced when counterclockwise is selected. When LINKE = 1, the vertex coordinates are read from the memory address specified by the LINK Address. The LINK Address can be specified through the LREL bit as an absolute address or a relative address with respect to the memory address at which the LINED command code is located.

This command can only be executed for a 16-bit/pixel destination.

When a polygon used in work reference is drawn by the FTRAPC (RFTRAPC) command, perform drawing with both the EDG and EOS bits set to 1.

- Notes:
1. 8-point drawing is used.
  2. The final point of each line segment is not drawn. When antialiasing is performed for the exterior frame of a polygon drawn by a POLYGON4 type command, the paths may not match.
  3. When the starting and final coordinate points of a line segment match, nothing is drawn.
  4. Antialiasing is not performed for horizontal, vertical, and 45-degree diagonal line segments, which are pre-clipped inside the G2D.
  5. Clipping is performed on a pixel basis when either the referenced pixel or the pixel to be drawn is outside the clipping area, and antialiasing is not performed in such a case.

**(d) Example**



**(5) RLINEA****(a) Function**

Draws a polygonal line with any width in the destination area with a relative coordinate specification from the current pointer value while referencing a multi-valued (8- or 16-bit/pixel) source.

**(b) Command Format**

- REL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
OP CODE = 1011_0110								Reserve (all 0)								Draw Mode																		
0 0 0			Base Address (quad word address)																												0 0 0			
0 0 0 0			TDX (8 ≤ TDX ≤ 4088)												0 0 0 0 0 0 0 0				TDY (1 ≤ TDY ≤ 4095)															
0 0 0 0			TXOFS (0 ≤ TXOFS ≤ TDX – 1)																n (1 ≤ n ≤ 65535)															
Reserve (all 0)																0 0 0 0 0 0 0 0				0 0 0 0 0 0 0 0				W (0,2 ≤ W ≤ 63)										
Sign		DX2 (-128 ≤ DX2 ≤ 127)						Sign		DY2 (-128 ≤ DY2 ≤ 127)						Sign		DX1 (-128 ≤ DX1 ≤ 127)						Sign		DY1 (-128 ≤ DY1 ≤ 127)								
Sign		:						Sign		:						Sign		:						Sign		:								
Sign		:						Sign		:						Sign		:						Sign		:								
Sign		DXn (-128 ≤ DXn ≤ 127)						Sign		DYn (-128 ≤ DYn ≤ 127)						Sign		DXn-1 (-128 ≤ DXn-1 ≤ 127)						Sign		DYn-1 (-128 ≤ DYn-1 ≤ 127)								

- Notes:
- When W = 0, set TDY to 1.
  - When n = 0, correct operation is not guaranteed.
  - When n is an odd number, insert a dummy word of 0 at the end.

- REL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																								
OP CODE = 1011_0110									Reserve (all 0)								Draw Mode																																																						
Sign extended				Sign				Base Address (longword address)																									0		0																																				
0				0				0				0				TDX (8 ≤ TDX ≤ 4088)								0				0				0				0				0				0				TDY (1 ≤ TDY ≤ 4095)																							
0				0				0				0				TXOFS (0 ≤ TXOFS ≤ TDX – 1)																n (1 ≤ n ≤ 65535)																																							
Reserve (all 0)																0								0								0								0								0								0								W (0,2 ≤ W ≤ 63)							
Sign				DX2 (-128 ≤ DX2 ≤ 127)												Sign				DY2 (-128 ≤ DY2 ≤ 127)												Sign				DX1 (-128 ≤ DX1 ≤ 127)												Sign				DY1 (-128 ≤ DY1 ≤ 127)																			
Sign				:												Sign				:												Sign				:												Sign				:																			
Sign				:												Sign				:												Sign				:												Sign				:																			
Sign				DXn (-128 ≤ DXn ≤ 127)												Sign				DYn (-128 ≤ DYn ≤ 127)												Sign				DXn-1 (-128 ≤ DXn-1 ≤ 127)												Sign				DYn-1 (-128 ≤ DYn-1 ≤ 127)																			

- Notes:
- Adding the address (longword: 32-bit units) where the command code is located to the Base Address (longword: 32-bit units) must result in a quad word address (64-bit units).
  - When  $W = 0$ , set TDY to 1.
  - When  $n = 0$ , correct operation is not guaranteed.
  - When  $n$  is an odd number, insert a dummy word of 0 at the end.

## 1. Code

B'10110110

## 2. Rendering Attributes

Reference Data								Drawing Destination							
Multi-Valued Source		Binary Source				Binary Work		Specified Color		Rendering		Work			
O										O					

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	STRANS	Fixed to 0	Fixed to 0	SS (0)	REL	STYLE (1)	Fixed to 0	NET	EOS	COOF	AA	Fixed to 0

- Notes:
- Clear the SS bit to 0.
  - Set the STYLE bit to 1.



### 3. Command Parameters

Base Address:	Source start absolute address (Quad word address. Write 0 to bits A31 to A29 and A2 to A0.)
	Source start relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

TDX, TDY:	Source size. Write 0 to the unused bits.
TXOFS:	Source offset. Write 0 to the unused bits.
n (n = 1 to 65,535):	Number of vertices
W:	Line width. Set a 6-bit integer. Write 0 to the unused bits. When 0 is set in W, a polygonal line of line width 1 is drawn. Setting 1 in W is prohibited.
DXn (n = 1 to 65,535):	Rendering coordinate (relative coordinate). Negative number expressed as two's complement.
DYn (n = 1 to 65,535):	Rendering coordinate (relative coordinate). Negative number expressed as two's complement.

#### (c) Description

Draws a polygonal line comprising line segments  $(XC, YC) - (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) - (XC + DX1 + DX2, YC + DY1 + DY2), \dots, (XC + \dots + DXn - 1, YC + \dots + DYn - 1) - (XC + \dots + DXn - 1 + DXn, YC + \dots + DYn - 1 + DYn)$  to the coordinates specified by the relative shift (DX, DY) from the current pointer values (XC, YC).

The final coordinate point is stored as the current pointer values (XC, YC). Set a multiple of 8 pixels as the TDX value. If TXOFS is set, the source at a location shifted by the offset amount is referenced. Make the TXOFS setting in pixel units.

Pattern repetition selected by the STYLE bit is only performed in the X direction of the source data. The source data is enlarged or reduced in proportion to the line width in the Y direction.

When a value greater than 0 is set in W, a bold line can be drawn.

- Notes:
1. 8-point drawing is used for a line width of 1. Both of the 8-point drawing and 4-point drawing are used for bold line drawing.
  2. The final point of each line segment is drawn. When the starting and final coordinate points of a line segment match, a single dot is drawn for a line width of 1 and nothing is drawn for bold line drawing.
  3. When AA = 1, note the following:
    - For a dashed line, antialiasing is not performed for the gaps in the dashed line.
    - When the starting and final coordinate points of a line segment match, antialiasing is not performed.
    - Antialiasing is not performed for horizontal, vertical, and 45-degree diagonal line segments
  4. The final coordinate point before coordinate transformation is stored as the current pointer values (XC, YC).

**(6) RLINEB****(a) Function**

Draws a polygonal line with any width in the destination area with a relative coordinate specification from the current pointer value while referencing a binary (1-bit/pixel) source.

**(b) Command Format**

- REL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
OP CODE = 1011_0101								Reserve (all 0)								Draw Mode																															
Color1																Color0																															
0 0 0			Base Address (quad word address)																											0 0 0																	
0 0 0 0				TDX (8 ≤ TDX ≤ 4088)										0 0 0 0 0 0 0 0				TDY (1 ≤ TDY ≤ 4095)																													
0 0 0 0				TXOFS (0 ≤ TXOFS ≤ TDX – 1)										n (1 ≤ n ≤ 65535)																																	
Reserve (all 0)																0 0 0 0 0 0 0 0				0 0 0 0 0 0 0 0				W (0,2 ≤ W ≤ 63)																							
Sign		DX2 (-128 ≤ DX2 ≤ 127)										Sign		DY2 (-128 ≤ DY2 ≤ 127)										Sign		DX1 (-128 ≤ DX1 ≤ 127)										Sign		DY1 (-128 ≤ DY1 ≤ 127)									
Sign		:										Sign		:										Sign		:										Sign		:									
Sign		:										Sign		:										Sign		:										Sign		:									
Sign		DXn (-128 ≤ DXn ≤ 127)										Sign		DYn (-128 ≤ DYn ≤ 127)										Sign		DXn-1 (-128 ≤ DXn-1 ≤ 127)										Sign		DYn-1 (-128 ≤ DYn-1 ≤ 127)									

- Notes:
1. When W = 0, set TDY to 1.
  2. When n = 0, correct operation is not guaranteed.
  3. When n is an odd number, insert a dummy word of 0 at the end.

- REL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
OP CODE = 1011_0101								Reserve (all 0)								Draw Mode																															
Color1																Color0																															
Sign extended		Sign		Base Address (longword address)																		0		0																							
0	0	0	0	TDX (8 ≤ TDX ≤ 4088)								0								0								TDY (1 ≤ TDY ≤ 4095)																			
0	0	0	0	TXOFS (0 ≤ TXOFS ≤ TDX – 1)																n (1 ≤ n ≤ 65535)																											
Reserve (all 0)																0								0								0								W (0,2 ≤ W ≤ 63)							
Sign		DX2 (-128 ≤ DX2 ≤ 127)								Sign		DY2 (-128 ≤ DY2 ≤ 127)								Sign		DX1 (-128 ≤ DX1 ≤ 127)								Sign		DY1 (-128 ≤ DY1 ≤ 127)															
Sign		:								Sign		:								Sign		:								Sign		:															
Sign		:								Sign		:								Sign		:								Sign		:															
Sign		DXn (-128 ≤ DXn ≤ 127)								Sign		DYn (-128 ≤ DYn ≤ 127)								Sign		DXn-1 (-128 ≤ DXn-1 ≤ 127)								Sign		DYn-1 (-128 ≤ DYn-1 ≤ 127)															

- Notes:
- Adding the address (longword: 32-bit units) where the command code is located to the Base Address (longword: 32-bit units) must result in a quad word address (64-bit units).
  - When W = 0, set TDY to 1.
  - When n = 0, correct operation is not guaranteed.
  - When n is an odd number, insert a dummy word of 0 at the end.

- Code  
B'10110101
- Rendering Attributes

Reference Data								Drawing Destination							
Multi-Valued Source				Specified Color				Rendering				Work			
Binary Source				Binary Work											
O								O							

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	STRANS	Fixed to 0	Fixed to 0	SS (0)	REL	STYLE (1)	Fixed to 0	NET	EOS	COOF	AA	Fixed to 0

- Notes:
- Clear the SS bit to 0.
  - Set the STYLE bit to 1.

### 3. Command Parameters

Color0, Color1:	8- or 16-bit/pixel color specification. For 16-bit/pixel drawing, the color specification should match the destination pixel format. For 8-bit/pixel drawing, the same value should be set in the upper and lower bytes.
Base Address:	Source start absolute address (Quad word address. Write 0 to bits A31 to A29 and A2 to A0.) Source start relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

TDX, TDY:	Source size. Write 0 to the unused bits.
TXOFS:	Source offset. Write 0 to the unused bits.
n (n = 1 to 65,535):	Number of vertices
W:	Line width. Set a 6-bit integer. Write 0 to the unused bits. When 0 is set in W, a polygonal line of line width 1 is drawn. Setting 1 in W is prohibited.
DXn (n = 1 to 65,535):	Rendering coordinate (relative coordinate). Negative number expressed as two's complement.
DYn (n = 1 to 65,535):	Rendering coordinate (relative coordinate). Negative number expressed as two's complement.

#### (c) Description

Draws a polygonal line comprising line segments  $(XC, YC) - (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) - (XC + DX1 + DX2, YC + DY1 + DY2), \dots, (XC + \dots + DXn - 1, YC + \dots + DYn - 1) - (XC + \dots + DXn - 1 + DXn, YC + \dots + DYn - 1 + DYn)$  to the coordinates specified by the relative shift (DX, DY) from the current pointer values (XC, YC).

The final coordinate point is stored as the current pointer values (XC, YC).

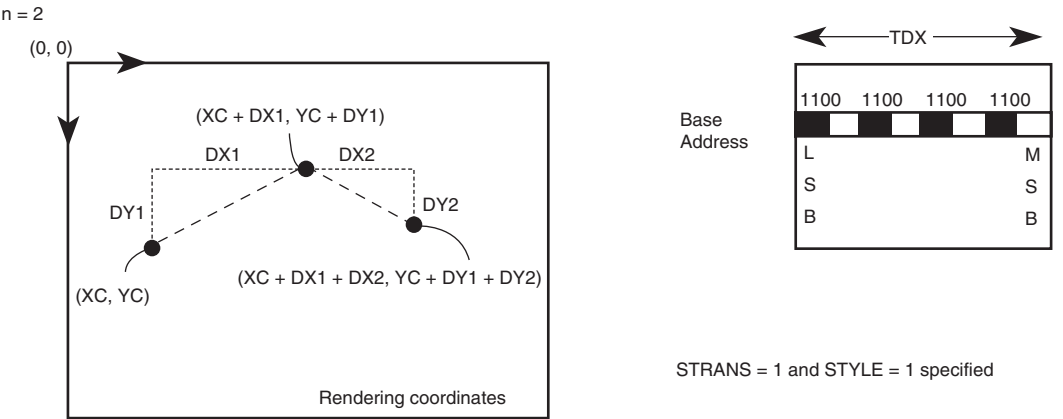
Set a multiple of 8 pixels as the TDX value. If TXOFS is set, the source at a location shifted by the offset amount is referenced. Make the TXOFS setting in pixel units.

Pattern repetition selected by the STYLE bit is only performed in the X direction of the source data. The source data is enlarged or reduced in proportion to the line width in the Y direction.

When a value greater than 1 is set in W, a bold line can be drawn.

- Notes:
1. 8-point drawing is used for a line width of 1. Both of the 8-point drawing and 4-point drawing are used for bold line drawing.
  2. The final point of each line segment is drawn. When the starting and final coordinate points of a line segment match, a single dot is drawn for a line width of 1 and nothing is drawn for bold line drawing.
  3. When AA = 1, note the following:
    - For a dashed line, antialiasing is not performed for the gaps in the dashed line.
    - When the starting and final coordinate points of a line segment match, antialiasing is not performed.
    - Antialiasing is not performed for horizontal, vertical, and 45-degree diagonal line segments.
  4. The final coordinate point before coordinate transformation is stored as the current pointer values (XC, YC).

(d) Example



## (7) RLINEC

### (a) Function

Draws a polygonal line with any width in the destination area with a monochrome specification and with a relative coordinate specification from the current pointer value.

### (b) Command Format

- LINKE = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE = 1011_0100								Reserve (all 0)								Draw Mode																	
Color																n (1 ≤ n ≤ 65535)																	
Reserve (all 0)																0	0	0	0	0	0	0	0	0	0	0	0	W (0,2 ≤ W ≤ 63)					
Sign	DX2 (-128 ≤ DX2 ≤ 127)							Sign	DY2 (-128 ≤ DY2 ≤ 127)							Sign	DX1 (-128 ≤ DX1 ≤ 127)							Sign	DY1 (-128 ≤ DY1 ≤ 127)								
Sign	⋮							Sign	⋮							Sign	⋮							Sign	⋮								
Sign	⋮							Sign	⋮							Sign	⋮							Sign	⋮								
Sign	DXn (-128 ≤ DXn ≤ 127)							Sign	DYn (-128 ≤ DYn ≤ 127)							Sign	DXn-1 (-128 ≤ DXn-1 ≤ 127)							Sign	DYn-1 (-128 ≤ DYn-1 ≤ 127)								

- Notes:
1. When  $n = 0$ , correct operation is not guaranteed.
  2. When  $n$  is an odd number, insert a dummy word of 0 at the end.

- LINKE = 1 and LREL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1011_0100									Reserve (all 0)							Draw Mode															
Color																n (1 ≤ n ≤ 65535)															
Reserve (all 0)																0	0	0	0	0	0	0	0	0	0	0	W (0,2 ≤ W ≤ 63)				
0	0	0	LINK Address (longword address)																										0	0	

- Notes:
1. When  $n = 0$ , correct operation is not guaranteed.
  2. When  $n$  is an odd number, insert a dummy word of 0 at the end.
  3. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by the LINK Address.

- LINKE = 1 and LREL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1011_0100								Reserve (all 0)								Draw Mode															
Color																n (1 ≤ n ≤ 65535)															
Reserve (all 0)																0	0	0	0	0	0	0	0	0	0	0	0	W (0,2 ≤ W ≤ 63)			
Sign extended		Sign		LINK Address (longword address)																										0	0

Notes: 1. When n = 0, correct operation is not guaranteed.  
2. When n is an odd number, insert a dummy word of 0 at the end.  
3. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by adding the address where the command code is located to the LINK Address.

- 1. Code  
B'10110100
- 2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
			0	0	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	LINKE	LREL	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	NET	EOS	COOF	AA	Fixed to 0



### 3. Command Parameters

Color:	8- or 16-bit/pixel color specification. For 16-bit/pixel drawing, the color specification should match the destination pixel format. For 8-bit/pixel drawing, the same value should be set in the upper and lower bytes.
n (n = 1 to 65,535):	Number of vertices
W:	Line width. Set a 6-bit integer. Write 0 to the unused bits. When 0 is set in W, a polygonal line of line width 1 is drawn. Setting 1 in W is prohibited.
DXn (n = 1 to 65,535):	Rendering coordinate (relative coordinate). Negative number expressed as two's complement.
DYn (n = 1 to 65,535):	Rendering coordinate (relative coordinate). Negative number expressed as two's complement.
LINK Address:	LINK absolute address (Longword address. Write 0 to bits A31 to A29, A1, and A0.) LINK relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

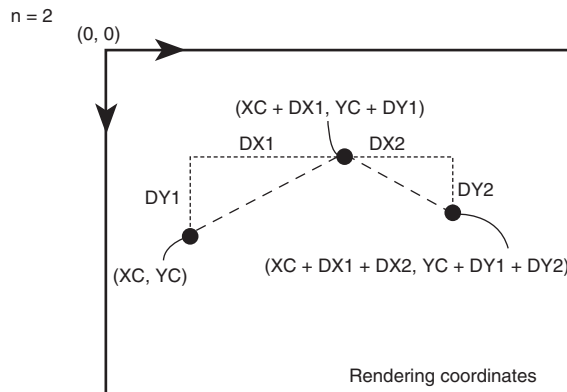
#### (c) Description

Draws a polygonal line comprising line segments  $(XC, YC) - (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) - (XC + DX1 + DX2, YC + DY1 + DY2), \dots, (XC + \dots + DXn - 1, YC + \dots + DYn - 1) - (XC + \dots + DXn - 1 + DXn, YC + \dots + DYn - 1 + DYn)$  to the coordinates specified by the relative shift (DX, DY) from the current pointer values (XC, YC). When a value greater than 1 is set in W, a bold line can be drawn. When LINKE = 1, the vertex coordinates are read from the memory address specified by the LINK Address. The LINK Address can be specified through the LREL bit as an absolute address or a relative address with respect to the memory address at which the RLINEC command code is located.

The final coordinate point is stored as the current pointer values (XC, YC).

- Notes:
1. 8-point drawing is used for a line width of 1. Both of the 8-point drawing and 4-point drawing are used for bold line drawing.
  2. The final point of each line segment is drawn. When the starting and final coordinate points of a line segment match, a single dot is drawn for a line width of 1 and nothing is drawn for bold line drawing.
  3. When  $AA = 1$ , note the following:
    - When the starting and final coordinate points of a line segment match, antialiasing is not performed.
    - Antialiasing is not performed for horizontal, vertical, and 45-degree diagonal line segments.
  4. The final coordinate point before coordinate transformation is stored as the current pointer values (XC, YC).

**(d) Example**



**(8) RLINED****(a) Function**

Performs antialiasing for the exterior frame of a polygon with a relative coordinate specification from the current pointer value. This command can only be executed for a 16-bit/pixel destination.

**(b) Command Format**

- LINKE = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE = 1011_0111									Reserve (all 0)							Draw Mode																			
Reserve (all 0)																n (1 ≤ n ≤ 65535)																			
Sign		DX2 (-128 ≤ DX2 ≤ 127)							Sign		DY2 (-128 ≤ DY2 ≤ 127)							Sign		DX1 (-128 ≤ DX1 ≤ 127)							Sign		DY1 (-128 ≤ DY1 ≤ 127)						
Sign		⋮							Sign		⋮							Sign		⋮							Sign		⋮						
Sign		⋮							Sign		⋮							Sign		⋮							Sign		⋮						
Sign		DXn (-128 ≤ DXn ≤ 127)							Sign		DYn (-128 ≤ DYn ≤ 127)							Sign		DXn-1 (-128 ≤ DXn-1 ≤ 127)							Sign		DYn-1 (-128 ≤ DYn-1 ≤ 127)						

- Notes:
1. When n = 0, correct operation is not guaranteed.
  2. When n is an odd number, insert a dummy word of 0 at the end.

- LINKE = 1 and LREL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1011_0111								Reserve (all 0)								Draw Mode															
Reserve (all 0)																n (1 ≤ n ≤ 65535)															
0	0	0	LINK Address (longword address)																										0	0	

- Notes:
1. When n = 0, correct operation is not guaranteed.
  2. When n is an odd number, insert a dummy word of 0 at the end.
  3. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by the LINK Address.

- LINKE = 1 and LREL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE = 1011_0111								Reserve (all 0)								Draw Mode																	
Reserve (all 0)																n (1 ≤ n ≤ 65535)																	
Sign extended		Sign		LINK Address (longword address)																										0		0	

- Notes:
1. When  $n = 0$ , correct operation is not guaranteed.
  2. When  $n$  is an odd number, insert a dummy word of 0 at the end.
  3. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by adding the address where the command code is located to the LINK Address.

## 1. Code

B'10110111

## 2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
				O	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	LINKE	LREL	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	AA (1)	CLKW

Note: Set the AA bit to 1.

### 3. Command Parameters

- n (n = 1 to 65,535): Number of vertices
- DXn (n = 1 to 65,535): Rendering coordinate (relative coordinate). Negative number expressed as two's complement.
- DYn (n = 1 to 65,535): Rendering coordinate (relative coordinate). Negative number expressed as two's complement.
- LINK Address: LINK absolute address (Longword address. Write 0 to bits A31 to A29, A1, and A0.)  
LINK relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

#### (c) Description

Performs antialiasing for the exterior frame of a polygon drawn using work reference, with a relative coordinate specification from the current pointer value.

The CLKW bit specifies whether the order to give the n vertices is clockwise or counterclockwise: CLKW = 1 selects clockwise and CLKW = 0 selects counterclockwise. When clockwise is specified, the left image with respect to the drawing direction is referenced by antialiasing. On the other hand, the right image is referenced when counterclockwise is selected. When LINKE = 1, the vertex coordinates are read from the memory address specified by the LINK Address. The LINK Address can be specified through the LREL bit as an absolute address or a relative address with respect to the memory address at which the RLINED command code is located.

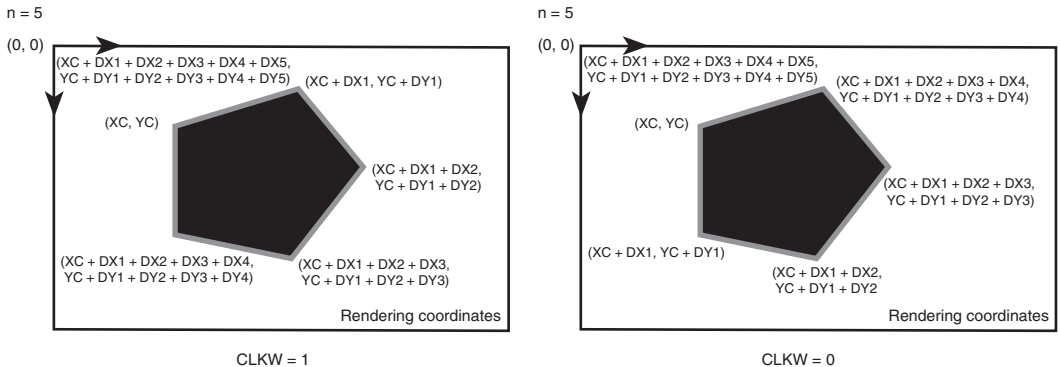
This command can only be executed for a 16-bit/pixel destination.

When a polygon used in work reference is drawn by the FTRAPC (RFTRAPC) command, perform drawing with both the EDG and EOS bits set to 1.

The final coordinate point is stored as the current pointer values (XC, YC).

- Notes:
1. 8-point drawing is used.
  2. The final point of each line segment is not drawn. When antialiasing is performed for the exterior frame of a polygon drawn by a POLYGON4 type command, the paths may not match.
  3. When the starting and final coordinate points of a line segment match, nothing is drawn.
  4. Antialiasing is not performed for horizontal, vertical, and 45-degree diagonal line segments, which are pre-clipped inside the G2D.
  5. Clipping is performed on a pixel basis when either the referenced pixel or the pixel to be drawn is outside the clipping area, and antialiasing is not performed in such a case.
  6. The final coordinate point before coordinate transformation is stored as the current pointer values (XC, YC).

**(d) Example**





- LINKE = 1 and LREL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE = 1101_0000								Reserve (all 0)								Draw Mode																			
Reserve (all 0)																n (2 ≤ n ≤ 65535)																			
Sign		Xmin (-32768 ≤ Xmin ≤ 32767)														Sign		Ymin (-32768 ≤ Ymin ≤ 32767)																	
Sign		Xmax (-32768 ≤ Xmax ≤ 32767)														Sign		Ymax (-32768 ≤ Ymax ≤ 32767)																	
Sign extended		Sign		LINK Address (longword address)																												0		0	

- Notes:
1. When n = 0 or 1, correct operation is not guaranteed.
  2. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by adding the address where the command code is located to the LINK Address.

1. Code  
B'11010000
2. Rendering Attributes

Reference Data					Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work	
						O

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	LINKE	LREL	Fixed to 0	Fixed to 0	Fixed to 0	BLKE (1)	EDG	EOS	Fixed to 0	Fixed to 0	Fixed to 0

Note: Set the BLKE bit to 1.



### 3. Command Parameters

n (n = 2 to 65,535):	Number of vertices
Xmin:	Xmin value in the circumscribed quadrangle of the polygon. Work coordinate (absolute coordinate). Negative number expressed as two's complement.
Ymin:	Ymin value in the circumscribed quadrangle of the polygon. Work coordinate (absolute coordinate). Negative number expressed as two's complement.
Xmax:	Xmax value in the circumscribed quadrangle of the polygon. Work coordinate (absolute coordinate). Negative number expressed as two's complement.
Ymax:	Ymax value in the circumscribed quadrangle of the polygon. Work coordinate (absolute coordinate). Negative number expressed as two's complement.
DXn (n = 2 to 65,535):	Work coordinate (absolute coordinate). Negative number expressed as two's complement.
DYn (n = 2 to 65,535):	Work coordinate (absolute coordinate). Negative number expressed as two's complement.
LINK Address:	<p>LINK absolute address (Longword address. Write 0 to bits A31 to A29, A1, and A0.)</p> <p>LINK relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)</p>

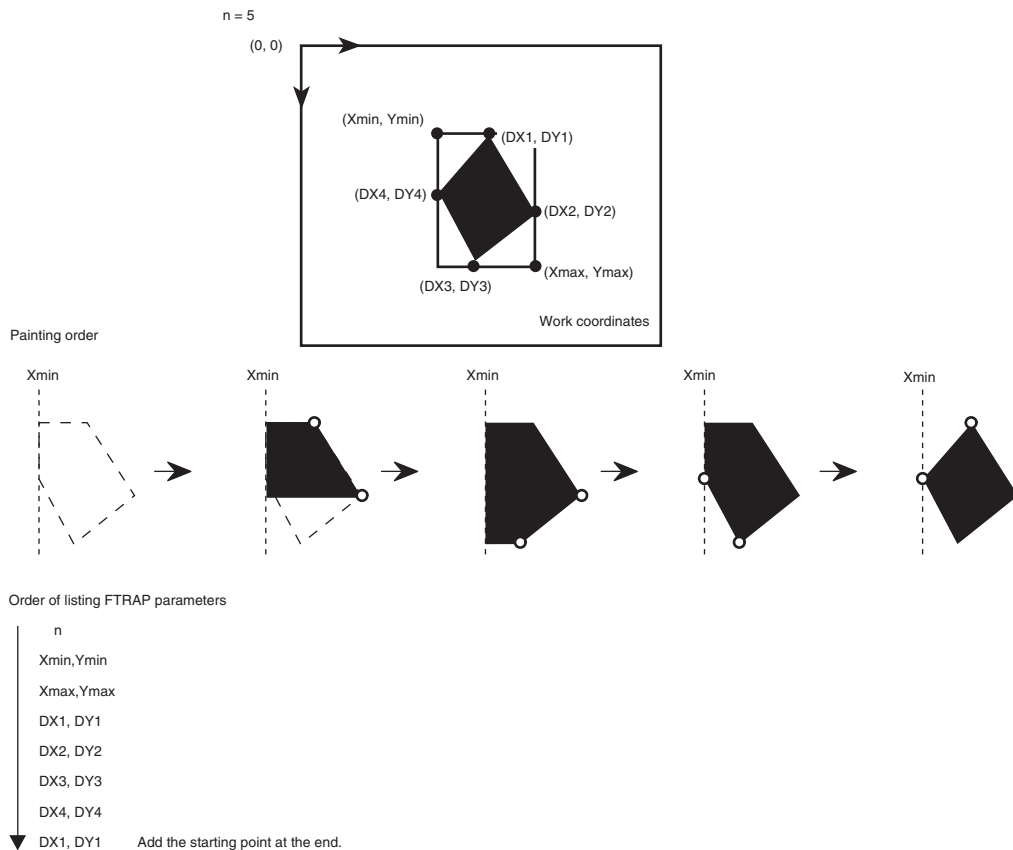
Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

**(c) Description**

Draws a polygon with  $n - 1$  vertices at work coordinates. Paints  $n - 1$  trapezoids at work coordinates using binary EOR, with  $X = X_{\min}$  as the left-hand side, and line segments  $(DX1, DY1) - (DX2, DY2)$ ,  $(DX2, DY2) - (DX3, DY3)$ , ...,  $(DX_{n-1}, DY_{n-1}) - (DX_n, DY_n)$  as the right-hand sides, and with the top and bottom bases parallel to the X-axis. Bottom base drawing is not performed. Set  $(DX_n, DY_n) = (DX1, DY1)$  to give a closed figure. If the rendering attribute EDG bit is set to 1, an edge line is drawn after the paint operation. The line drawing data is selected with the EOS bit.

The FTRAPC command performs coordinate transformation by internally obtaining the four vertices from the coordinates for the circumscribed quadrangle of the input polygon and then transforming the coordinates for these four vertices. The transformed four vertices are then internally converted into a circumscribed rectangle, the left edge obtained, and the polygon drawn.

Note: When enabling edge drawing ( $EDG = 1$ ), Z pre-clipping is not performed for the edge line.

**(d) Example**

## (2) RFTRAPC

### (a) Function

Draws a polygon at work coordinates with a relative coordinate specification from the current pointer value.

### (b) Command Format

- LINKE = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1101_0100								Reserve (all 0)								Draw Mode															
Reserve (all 0)																n (1 ≤ n ≤ 65535)															
Sign		Xmin (-32768 ≤ Xmin ≤ 32767)														Sign		Ymin (-32768 ≤ Ymin ≤ 32767)													
Sign		Xmax (-32768 ≤ Xmax ≤ 32767)														Sign		Ymax (-32768 ≤ Ymax ≤ 32767)													
Sign		DX2 (-128 ≤ DX2 ≤ 127)					Sign		DY2 (-128 ≤ DY2 ≤ 127)					Sign		DX1 (-128 ≤ DX1 ≤ 127)					Sign		DY1 (-128 ≤ DY1 ≤ 127)								
Sign		⋮					Sign		⋮					Sign		⋮					Sign		⋮								
Sign		⋮					Sign		⋮					Sign		⋮					Sign		⋮								
Sign		DXn (-128 ≤ DXn ≤ 127)					Sign		DYn (-128 ≤ DYn ≤ 127)					Sign		DXn-1 (-128 ≤ DXn-1 ≤ 127)					Sign		DYn-1 (-128 ≤ DYn-1 ≤ 127)								

Notes: 1. When n = 0, correct operation is not guaranteed.  
 2. When n is an odd number, insert a dummy word of 0 at the end.

- LINKE = 1 and LREL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
OP CODE = 1101_0100									Reserve (all 0)								Draw Mode															
Reserve (all 0)																n (1 ≤ n ≤ 65535)																
Sign		Xmin (-32768 ≤ Xmin ≤ 32767)														Sign		Ymin (-32768 ≤ Ymin ≤ 32767)														
Sign		Xmax (-32768 ≤ Xmax ≤ 32767)														Sign		Ymax (-32768 ≤ Ymax ≤ 32767)														
0	0	0	LINK Address (longword address)																												0	0

Notes: 1. When n = 0, correct operation is not guaranteed.  
 2. When n is an odd number, insert a dummy word of 0 at the end.  
 3. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by the LINK Address.

- LINKE = 1 and LREL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1101_0100									Reserve (all 0)							Draw Mode															
Reserve (all 0)																n (1 ≤ n ≤ 65535)															
Sign		Xmin (-32768 ≤ Xmin ≤ 32767)														Sign		Ymin (-32768 ≤ Ymin ≤ 32767)													
Sign		Xmax (-32768 ≤ Xmax ≤ 32767)														Sign		Ymax (-32768 ≤ Ymax ≤ 32767)													
Sign extended		Sign		LINK Address (longword address)																		0		0							

Notes: 1. When  $n = 0$ , correct operation is not guaranteed.  
 2. When  $n$  is an odd number, insert a dummy word of 0 at the end.  
 3. The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by the address where the command code is located plus the LINK Address.

#### 1. Code

B'11010100

#### 2. Rendering Attributes

Reference Data			Drawing Destination		
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
					○

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	LINKE	LREL	Fixed to 0	Fixed to 0	Fixed to 0	BLKE (1)	EDG	EOS	Fixed to 0	Fixed to 0	Fixed to 0

Note: Set the BLKE bit to 1.

### 3. Command Parameters

n (n = 1 to 65,535):	Number of vertices
Xmin:	Xmin value in the circumscribed quadrangle of the polygon. Work coordinate (absolute coordinate). Negative number expressed as two's complement.
Ymin:	Ymin value in the circumscribed quadrangle of the polygon. Work coordinate (absolute coordinate). Negative number expressed as two's complement.
Xmax:	Xmax value in the circumscribed quadrangle of the polygon. Work coordinate (absolute coordinate). Negative number expressed as two's complement.
Ymax:	Ymax value in the circumscribed quadrangle of the polygon. Work coordinate (absolute coordinate). Negative number expressed as two's complement.
DXn (n = 1 to 65,535):	Work coordinate (relative coordinate). Negative number expressed as two's complement.
DYn (n = 1 to 65,535):	Work coordinate (relative coordinate). Negative number expressed as two's complement.
LINK Address:	LINK absolute address (Longword address. Write 0 to bits A31 to A29, A1, and A0.) LINK relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

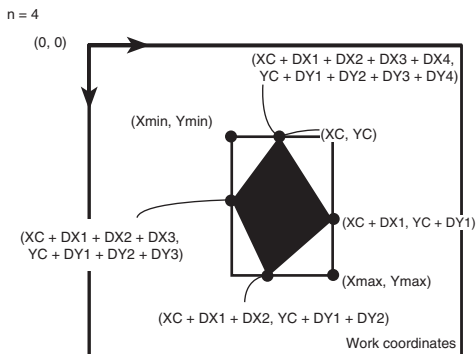
### (c) Description

Draws a polygon with  $n - 1$  vertices at work coordinates. Paints  $n - 1$  trapezoids at work coordinates using binary EOR, with  $X = X_{\min}$  as the left-hand side, and line segments specified by the relative shift (DX, DY) from the current pointer values (XC, YC)  $((XC, YC) - (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) - (XC + DX1 + DX2, YC + DY1 + DY2), \dots, (XC + \dots + DX_{n-1}, YC + \dots + DY_{n-1}) - (XC + \dots + DX_{n-1} + DX_n, YC + \dots + DY_{n-1} + DY_n))$  as the right-hand sides, and with the top and bottom bases parallel to the X-axis. Bottom base drawing is not performed.

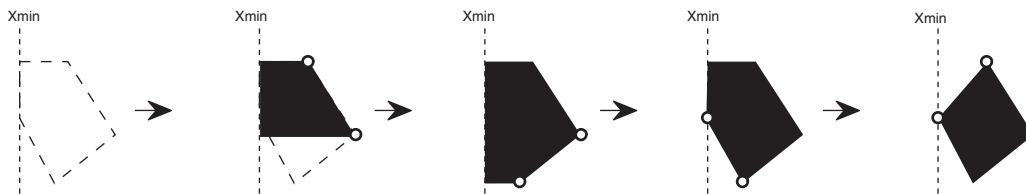
The final coordinate point is stored as the current pointer values (XC, YC). Set  $(DX1 + DX2 + \dots + DX_n = 0, DY1 + DY2 + \dots + DY_n = 0)$  to give a closed figure. If the rendering attribute EDG bit is set to 1, an edge line is drawn after the paint operation. The line drawing data is selected with the EOS bit.

The RFTRAPC command performs coordinate transformation by internally obtaining the four vertices from the coordinates for the circumscribed quadrangle of the input polygon and then transforming the coordinates for these four vertices. The transformed four vertices are then internally converted into a circumscribed rectangle, the left edge obtained, and the polygon drawn.

- Notes:
1. The final coordinate point before coordinate transformation is stored as the current pointer values (XC, YC).
  2. When enabling edge drawing ( $EDG = 1$ ), Z pre-clipping is not performed for the edge line.

**(d) Example**

Painting order





### (3) CLRWC

#### (a) Function

Clear the work coordinates to 0.

#### (b) Command Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1110_0000								Reserve (all 0)								Draw Mode															
Sign		Xmin (-32768 ≤ Xmin ≤ 32767)										Sign		Ymin (-32768 ≤ Ymin ≤ 32767)																	
Sign		Xmax (-32768 ≤ Xmax ≤ 32767)										Sign		Ymax (-32768 ≤ Ymax ≤ 32767)																	

#### 1. Code

B'11100000

#### 2. Rendering Attributes

Reference Data			Drawing Destination		
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
					0

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	BLKE (1)	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0

Note: Set the BLKE bit to 1.

#### 3. Command Parameters

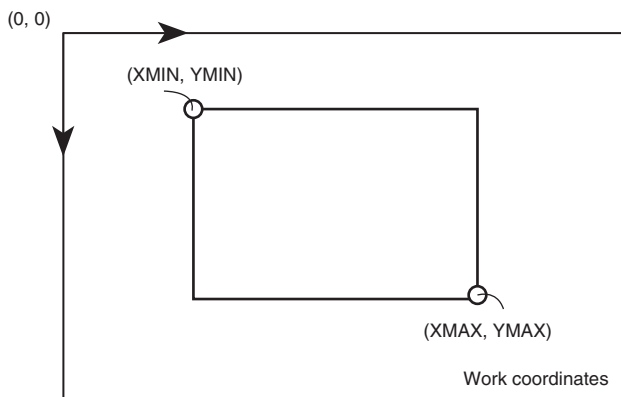
Xmin, Xmax: Left and right X coordinate values. Work coordinates (absolute coordinates. Negative numbers expressed as two's complement.

Ymin, Ymax: Upper and lower Y coordinate values. Work coordinates (absolute coordinates. Negative numbers expressed as two's complement.

**(c) Description**

Zero-clears the area specified by upper-left coordinates (Xmin, Ymin) and lower-right coordinates (Xmax, Ymax) at work coordinates.

The CLRWC command performs coordinate transformation by internally obtaining the four vertices from the left and right X coordinate values and upper and lower Y coordinate values, and then transforming the coordinates for these four vertices. The transformed four vertices are then internally converted into a circumscribed rectangle and the polygon drawn.

**(d) Example**

## 23.2.4 Work Line Drawing Commands

### (1) LINEWC

#### (a) Function

Draws a polygon at work coordinates.

#### (b) Command Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1111_0000								Reserve (all 0)								Draw Mode															
Reserve (all 0)																n (2 ≤ n ≤ 65535)															
Sign								DX1 (-32768 ≤ DX1 ≤ 32767)								Sign								DY1 (-32768 ≤ DY1 ≤ 32767)							
Sign								:								Sign								:							
Sign								:								Sign								:							
Sign								DXn (-32768 ≤ DXn ≤ 32767)								Sign								DYn (-32768 ≤ DYn ≤ 32767)							

Note: When  $n = 0$  or  $1$ , correct operation is not guaranteed.

#### 1. Code

B'11110000

#### 2. Rendering Attributes

Reference Data			Drawing Destination		
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
			0 (EOS of binary work)		0

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	EOS	Fixed to 0	Fixed to 0	Fixed to 0

### 3. Command Parameters

$n$  ( $n = 2$  to  $65,535$ ): Number of vertices

$DXn$  ( $n = 2$  to  $65,535$ ): Work coordinate (absolute coordinate). Negative number expressed as two's complement.

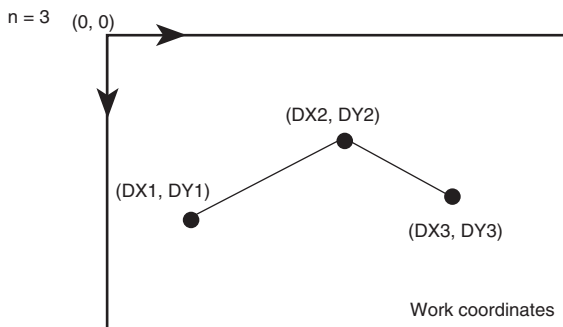
$DYn$  ( $n = 2$  to  $65,535$ ): Work coordinate (absolute coordinate). Negative number expressed as two's complement.

#### (c) Description

Performs binary drawing at work coordinates of a polygonal line from vertex 1 ( $DX1, DY1$ ), through vertex 2 ( $DX2, DY2$ ), ..., vertex  $n - 1$  ( $DXn - 1, DYn - 1$ ), to vertex  $n$  ( $DXn, DYn$ ). 0 drawing or 1 drawing is selected with the drawing mode EOS bit. Drawing is performed at work coordinates with 0 when  $EOS = 0$ , and at work coordinates with 1 when  $EOS = 1$  (Used for edge drawing at work coordinates for a polygonal painted figure).

Note: 8-point drawing is used. The final point of each line segment is drawn.

#### (d) Example



**(2) RLINEWC****(a) Function**

Draws a 1-bit-wide solid line at work coordinates with a relative coordinate specification from the current pointer value.

**(b) Command Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1111_0100								Reserve (all 0)								Draw Mode															
Reserve (all 0)																n (1 ≤ n ≤ 65535)															
Sign				DX2 (-128 ≤ DX2 ≤ 127)				Sign				DY2 (-128 ≤ DY2 ≤ 127)				Sign				DX1 (-128 ≤ DX1 ≤ 127)				Sign				DY1 (-128 ≤ DY1 ≤ 127)			
Sign				⋮				Sign				⋮				Sign				⋮				Sign				⋮			
Sign				⋮				Sign				⋮				Sign				⋮				Sign				⋮			
Sign				DXn (-128 ≤ DXn ≤ 127)				Sign				DYn (-128 ≤ DYn ≤ 127)				Sign				DXn-1 (-128 ≤ DXn-1 ≤ 127)				Sign				DYn-1 (-128 ≤ DYn-1 ≤ 127)			

Notes: 1. When  $n = 0$ , correct operation is not guaranteed.  
 2. When  $n$  is an odd number, insert a dummy word of 0 at the end.

## 1. Code

B'11110100

## 2. Rendering Attributes

Reference Data				Drawing Destination		
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work	
			O (EOS of binary work)		O	

Draw Mode																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	EOS	Fixed to 0	Fixed to 0	Fixed to 0	

### 3. Command Parameters

$n$  ( $n = 1$  to 65,535): Number of vertices

$DX_n$  ( $n = 1$  to 65,535): Work coordinate (relative coordinate). Negative number expressed as two's complement.

$DY_n$  ( $n = 1$  to 65,535): Work coordinate (relative coordinate). Negative number expressed as two's complement.

#### (c) Description

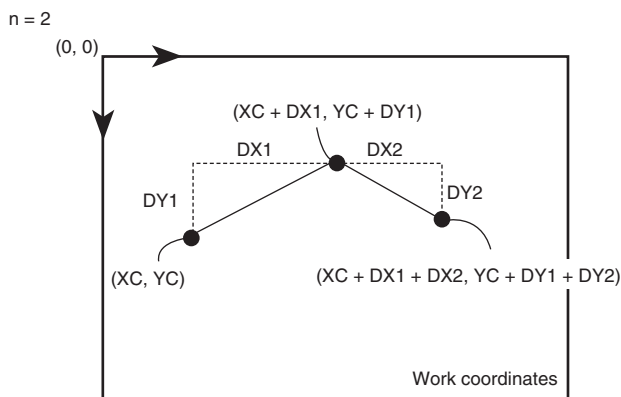
Performs binary drawing at work coordinates of a polygonal line comprising line segments  $(XC, YC) - (XC + DX1, YC + DY1)$ ,  $(XC + DX1, YC + DY1) - (XC + DX1 + DX2, YC + DY1 + DY2)$ , ...,  $(XC + \dots + DX_{n-1}, YC + \dots + DY_{n-1}) - (XC + \dots + DX_n, YC + \dots + DY_n)$  to the coordinates specified by the relative shift  $(DX, DY)$  from the current pointer values  $(XC, YC)$ . 0 drawing or 1 drawing is selected with the drawing mode EOS bit. Drawing is performed at work coordinates with 0 when  $EOS = 0$ , and at work coordinates with 1 when  $EOS = 1$ . (Used for edge drawing at work coordinates for a polygonal painted figure.)

The final coordinate point is stored as the current pointer values  $(XC, YC)$ .

Notes: 1. 8-point drawing is used. The end of a line is drawn.

2. The final coordinate point before coordinate transformation is stored as the current pointer values  $(XC, YC)$ .

#### (d) Example



## 23.2.5 Rectangle Drawing Commands

### (1) BITBLTA

#### (a) Function

Transfers multi-valued (8- or 16-bit/pixel) rectangle source data to the destination area.

#### (b) Command Format

- SS = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1010_0010								Reserve (all 0)								Draw Mode															
Reserve (all 0)																0	0	0	0	0	0	0	0	ROP							
0	0	0	0	TXS ( $0 \leq \text{TXS} \leq 4088$ )												0	0	0	0	TYS ( $0 \leq \text{TYS} \leq 4095$ )											
0	0	0	0	LW ( $0 \leq \text{LW} \leq 4094$ )												0	0	0	0	RW ( $0 \leq \text{RW} \leq 4094$ )											
0	0	0	0	TH ( $0 \leq \text{TH} \leq 4094$ )												0	0	0	0	BH ( $0 \leq \text{BH} \leq 4094$ )											
Sign				BXC ( $-32768 \leq \text{BXC} \leq 32767$ )												Sign				BYC ( $-32768 \leq \text{BYC} \leq 32767$ )											

- Notes: 1.  $0 \leq \text{TXS} \leq \text{SSTRR} - (\text{LW} + \text{RW} + 1)$ ,  $0 \leq \text{TYS} \leq 4096 - (\text{TH} + \text{BH} + 1)$  (SSTRR: Source stride register setting)
2.  $8 \leq \text{LW} + \text{RW} + 1 \leq 4095$ ,  $1 \leq \text{TH} + \text{BH} + 1 \leq 4095$
3.  $-32768 \leq \text{BXC} - \text{LW} \leq 32767$ ,  $-32768 \leq \text{BYC} - \text{TH} \leq 32767$ ,  $-32768 \leq \text{BXC} + \text{RW} \leq 32767$ ,  $-32768 \leq \text{BYC} + \text{BH} \leq 32767$

- SS = 0 and REL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE = 1010_0010								Reserve (all 0)								Draw Mode																	
Reserve (all 0)																0	0	0	0	0	0	0	0	ROP									
0	0	0	Base Address (quad word address)																												0	0	0
0	0	0	0	LW (0 ≤ LW ≤ 4087)												0	0	0	0	RW (0 ≤ RW ≤ 4087)													
0	0	0	0	TH (0 ≤ TH ≤ 4094)												0	0	0	0	BH (0 ≤ BH ≤ 4094)													
Sign				BXC (-32768 ≤ BXC ≤ 32767)												Sign				BYC (-32768 ≤ BYC ≤ 32767)													

- Notes: 1.  $8 \leq \text{LW} + \text{RW} + 1 \leq 4088$  (multiple of 8),  $1 \leq \text{TH} + \text{BH} + 1 \leq 4095$
2.  $-32768 \leq \text{BXC} - \text{LW} \leq 32767$ ,  $-32768 \leq \text{BYC} - \text{TH} \leq 32767$ ,  $-32768 \leq \text{BXC} + \text{RW} \leq 32767$ ,  $-32768 \leq \text{BYC} + \text{BH} \leq 32767$

- SS = 0 and REL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE = 1010_0010								Reserve (all 0)								Draw Mode																	
Reserve (all 0)																0	0	0	0	0	0	0	0	ROP									
Sign extended		Sign		Base Address (longword address)																												0	0
0	0	0	0	LW (0 ≤ LW ≤ 4087)												0	0	0	0	RW (0 ≤ RW ≤ 4087)													
0	0	0	0	TH (0 ≤ TH ≤ 4094)												0	0	0	0	BH (0 ≤ BH ≤ 4094)													
Sign		BXC (-32768 ≤ BXC ≤ 32767)														Sign		BYC (-32768 ≤ BYC ≤ 32767)															

- Notes:
- $8 \leq LW + RW + 1 \leq 4088$  (multiple of 8),  $1 \leq TH + BH + 1 \leq 4095$
  - $-32768 \leq BXC - LW \leq 32767$ ,  $-32768 \leq BYC - TH \leq 32767$ ,  $-32768 \leq BXC + RW \leq 32767$ ,  $-32768 \leq BYC + BH \leq 32767$
  - Adding the address (longword: 32-bit units) where the command code is located to the Base Address (longword: 32-bit units) must result in a quad word address (64-bit units).

- Code  
B'10100010
- Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
0		0 (only WORK = 1)		0	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	STRANS	DTRANS	WORK	SS	REL	SRCDIRX	SRCDIRY	DSTDIRX	DSTDIRY	COOF	αE	SαE



### 3. Command Parameters

TXS, TYS:	Source starting point. Write 0 to the unused bits.
Base Address:	Source start absolute address (Quad word address. Write 0 to bits A31 to A29 and A2 to A0.) Source start relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

BXC, BYC:	Center X and Y coordinate values. Rendering coordinates (absolute coordinates). Negative numbers expressed as two's complement.
LW, RW:	Left and right widths. Relative value from (BXC, BYC). Rendering coordinates. Make the setting in pixel units. Write 0 to the unused bits.
TH, BH:	Top and bottom heights. Relative value from (BXC, BYC). Rendering coordinates. Make the setting in pixel units. Write 0 to the unused bits.
ROP:	Raster operation code

#### (c) Description

Transfers multi-valued (8- or 16-bit/pixel) rectangle source data to rendering coordinates.

When SS = 0, set the (LW + RW + 1) value to be a multiple of 8 pixels. When SS = 1, set the (LW + RW + 1) value to be 8 or more pixels.

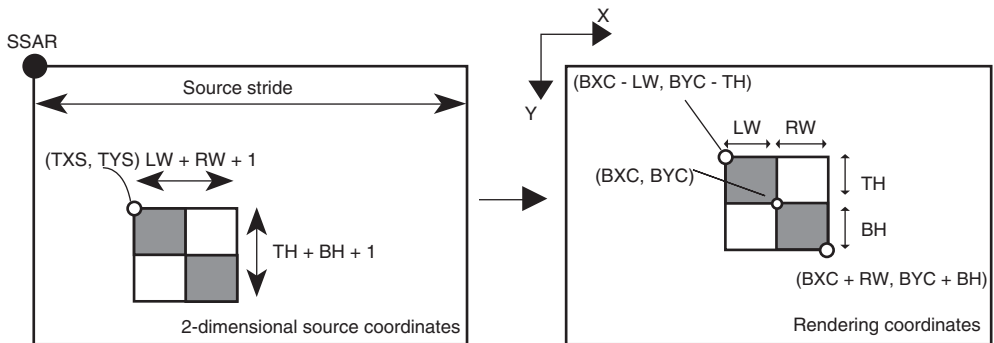
1. When work specification is selected as a rendering attribute (WORK = 1), only places where the work coordinate pixel is 1 are drawn at rendering coordinates while referencing work coordinates for the same coordinates as the rendering coordinates.
2. When SS = 1, the source data is referenced from the 2-dimensional source area. When SS = 0, the source data is referenced from the Base Address in the display list. When REL = 0, the source address can be specified as an absolute address. When REL = 1, the source address can be specified as a relative address with respect to the memory address at which the BITBLTA command code is located.
3. In 16-bit/pixel drawing, if the rendering attribute COOF bit is set to 1, the result of adding the value in COFSR to the value of the multi-valued source data is drawn. The operation is performed by saturation processing. In 8-bit/pixel drawing, the COOF bit should be cleared to 0.

4. The direction to reference the source data can be selected by the SRCDIRX and SRCDIRY bits.
5. The drawing direction can be selected by the DSTDIRX and DSTDIRY bits.
6. When  $\alpha E = 1$ , the source data and ground data are alpha blended before drawing. When setting  $\alpha E = 1$ , also set the ROP code = H'CC (source copy). The A value in the ARGB format is not alpha blended. The A value is drawn according to the source A value use (SAU) and A value (AVALUE) bits in the rendering control register (RCLR). Alpha blending is valid only in 16-bit/pixel drawing.
7. 16 raster operations are possible. The A value in the ARGB format is not subject to raster operations. The A value is drawn according to the source A value use (SAU) and A value (AVALUE) bits in the rendering control register (RCLR).

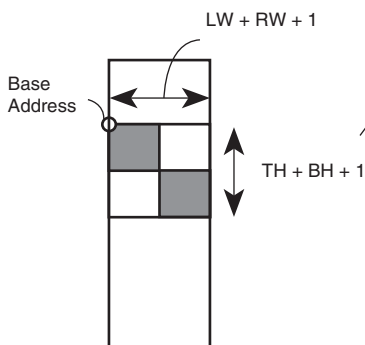
Note: System clipping or (relative) user clipping is performed when drawing a rectangle. Z clipping is performed only at the center coordinates.

#### (d) Example

SS = 1



SS = 0



## (2) BITBLTB

### (a) Function

Transfers binary (1-bit/pixel) rectangle source data that has been color expanded to the destination area.

### (b) Command Format

- SS = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1010_0001								Reserve (all 0)								Draw Mode															
Reserve (all 0)																0	0	0	0	0	0	0	0	ROP							
Color1																Color0															
0	0	0	0	TXS (0 ≤ TXS ≤ 4088)												0	0	0	0	TYS (0 ≤ TYS ≤ 4095)											
0	0	0	0	LW (0 ≤ LW ≤ 4087)												0	0	0	0	RW (0 ≤ RW ≤ 4087)											
0	0	0	0	TH (0 ≤ TH ≤ 4094)												0	0	0	0	BH (0 ≤ BH ≤ 4094)											
Sign BXC (-32768 ≤ BXC ≤ 32767)								Sign BYC (-32768 ≤ BYC ≤ 32767)																							

- Notes:
- $0 \leq \text{TXS} \leq \text{SSTR} - (\text{LW} + \text{RW} + 1)$ ,  $0 \leq \text{TYS} \leq 4096 - (\text{TH} + \text{BH} + 1)$  (SSTR: Source stride register setting)
  - $8 \leq \text{LW} + \text{RW} + 1 \leq 4088$  (multiple of 8),  $1 \leq \text{TH} + \text{BH} + 1 \leq 4095$
  - $-32768 \leq \text{BXC} - \text{LW} \leq 32767$ ,  $-32768 \leq \text{BYC} - \text{TH} \leq 32767$ ,  $-32768 \leq \text{BXC} + \text{RW} \leq 32767$ ,  $-32768 \leq \text{BYC} + \text{BH} \leq 32767$

- SS = 0 and REL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE = 1010_0001									Reserve (all 0)								Draw Mode																		
Reserve (all 0)																0	0	0	0	0	0	0	0	ROP											
Color1																Color0																			
0	0	0	Base Address (quad word address)																											0	0	0			
0	0	0	0	LW ( $0 \leq \text{LW} \leq 4087$ )												0	0	0	0	RW ( $0 \leq \text{RW} \leq 4087$ )															
0	0	0	0	TH ( $0 \leq \text{TH} \leq 4094$ )												0	0	0	0	BH ( $0 \leq \text{BH} \leq 4094$ )															
Sign								BXC ( $-32768 \leq \text{BXC} \leq 32767$ )								Sign								BYC ( $-32768 \leq \text{BYC} \leq 32767$ )											

- Notes:
- $8 \leq \text{LW} + \text{RW} + 1 \leq 4088$  (multiple of 8),  $1 \leq \text{TH} + \text{BH} + 1 \leq 4095$
  - $-32768 \leq \text{BXC} - \text{LW} \leq 32767$ ,  $-32768 \leq \text{BYC} - \text{TH} \leq 32767$ ,  $-32768 \leq \text{BXC} + \text{RW} \leq 32767$ ,  $-32768 \leq \text{BYC} + \text{BH} \leq 32767$

- SS = 0 and REL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE = 1010_0001								Reserve (all 0)								Draw Mode																	
Reserve (all 0)																0	0	0	0	0	0	0	0	ROP									
Color1																Color0																	
Sign extended		Sign		Base Address (longword address)																												0	0
0	0	0	0	LW (0 ≤ LW ≤ 4087)												0	0	0	0	RW (0 ≤ RW ≤ 4087)													
0	0	0	0	TH (0 ≤ TH ≤ 4094)												0	0	0	0	BH (0 ≤ BH ≤ 4094)													
Sign		BXC (-32768 ≤ BXC ≤ 32767)														Sign		BYC (-32768 ≤ BYC ≤ 32767)															

- Notes:
- $8 \leq LW + RW + 1 \leq 4088$  (multiple of 8),  $1 \leq TH + BH + 1 \leq 4095$
  - $-32768 \leq BXC - LW \leq 32767$ ,  $-32768 \leq BYC - TH \leq 32767$ ,  $-32768 \leq BXC + RW \leq 32767$ ,  $-32768 \leq BYC + BH \leq 32767$
  - Adding the address (longword: 32-bit units) where the command code is located to the Base Address (longword: 32-bit units) must result in a quad word address (64-bit units).

- Code  
B'10100001
- Rendering Attributes

Reference Data								Drawing Destination			
Multi-Valued Source		Binary Source		Binary Work		Specified Color		Rendering		Work	
		O		O				O			
				(only WORK = 1)							

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	STRANS	DTRANS	WORK	SS	REL	SRCDIRX	SRCDIRY	DSTDIRX	DSTDIRY	COOF	$\alpha E$	Fixed to 0

### 3. Command Parameters

TXS, TYS:	Source starting point. Write 0 to the unused bits.
Base Address:	Source start absolute address (Quad word address. Write 0 to bits A31 to A29 and A2 to A0.) Source start relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

BXC, BYC:	Center X and Y coordinate values. Rendering coordinates (absolute coordinates). Negative numbers expressed as two's complement.
LW, RW:	Left and right widths. Relative value from (BXC, BYC). Rendering coordinates. Make the setting in pixel units. Write 0 to the unused bits.
TH, BH:	Top and bottom heights. Relative value from (BXC, BYC). Rendering coordinates. Make the setting in pixel units. Write 0 to the unused bits.
ROP:	Raster operation code
Color0, Color1:	8- or 16-bit/pixel color specification. For 16-bit/pixel drawing, the color specification should match the destination pixel format. For 8-bit/pixel drawing, the same value should be set in the upper and lower bytes.

#### (c) Description

Transfers binary (1-bit/pixel) rectangle source data to rendering coordinates.

A multiple of 8 pixels must be set as the  $(LW + RW + 1)$  value, regardless of the SS bit value.

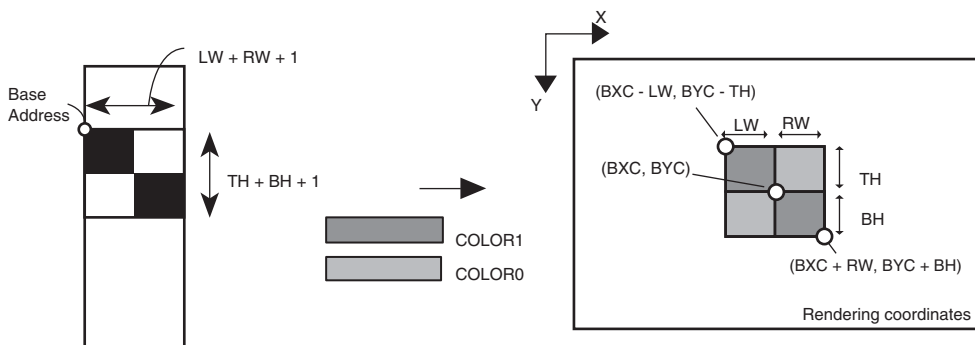
1. When work specification is selected as a rendering attribute ( $WORK = 1$ ), only places where the work coordinate pixel is 1 are drawn at rendering coordinates while referencing work coordinates for the same coordinates as the rendering coordinates.
2. The binary source data is arranged in memory in linear fashion. When  $REL = 0$ , the source address can be specified as an absolute address. When  $REL = 1$ , the source address can be specified as a relative address with respect to the memory address at which the BITBLTB command code is located.
3. In 16-bit/pixel drawing, if the rendering attribute COOF bit is set to 1, the result of adding the value in COFSR to the value of the binary source data that has been color expanded is drawn.

The operation is performed by saturation processing. In 8-bit/pixel drawing, the COOF bit should be cleared to 0.

4. The direction to reference the source data can be selected by the SRCDIRX and SRCDIRY bits.
5. The drawing direction can be selected by the DSTDIRX and DSTDIRY bits.
6. When  $\alpha E = 1$ , the data obtained by color expanding the binary source data and the ground data are alpha blended before drawing. When setting  $\alpha E = 1$ , also set the ROP code = H'CC (source copy). The A value in the ARGB format is not alpha blended. The A value is drawn according to the source A value use (SAU) and A value (AVALUE) bits in the rendering control register (RCLR). Alpha blending is valid only in 16-bit/pixel drawing.
7. 16 raster operations are possible. The A value in the ARGB format is not subject to raster operations. The A value is drawn according to the source A value use (SAU) and A value (AVALUE) bits in the rendering control register (RCLR).

Note: System clipping or (relative) user clipping is performed when drawing a rectangle. Z clipping is performed only at the center coordinates.

#### (d) Example



### (3) BITBLTC

#### (a) Function

Draws a rectangle with a monochrome specification to the destination area.

#### (b) Command Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1010_0000								Reserve (all 0)								Draw Mode															
Reserve (all 0)																0	0	0	0	0	0	0	0	ROP							
Reserve (all 0)																Color															
0	0	0	0	LW ( $0 \leq LW \leq 4094$ )												0	0	0	0	RW ( $0 \leq RW \leq 4094$ )											
0	0	0	0	TH ( $0 \leq TH \leq 4094$ )												0	0	0	0	BH ( $0 \leq BH \leq 4094$ )											
Sign				BXC ( $-32768 \leq BXC \leq 32767$ )												Sign				BYC ( $-32768 \leq BYC \leq 32767$ )											

Notes: 1.  $1 \leq LW + RW + 1 \leq 4095$ ,  $1 \leq TH + BH + 1 \leq 4095$   
 2.  $-32768 \leq BXC - LW \leq 32767$ ,  $-32768 \leq BYC - TH \leq 32767$ ,  $-32768 \leq BXC + RW \leq 32767$ ,  $-32768 \leq BYC + BH \leq 32767$

## 1. Code

B'10100000

## 2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
		0 (only WORK = 1)	0	0	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	Fixed to 0	CLIP	RCLIP	Fixed to 0	DTRANS	WORK	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	DSTDIRX	DSTDIRY	COOF	αE	Fixed to 0

### 3. Command Parameters

BXC, BYC:	Center X and Y coordinate values. Rendering coordinates (absolute coordinates). Negative numbers expressed as two's complement.
LW, RW:	Left and right widths. Relative value from (BXC, BYC). Rendering coordinates. Make the setting in pixel units. Write 0 to the unused bits.
TH, BH:	Top and bottom heights. Relative value from (BXC, BYC). Rendering coordinates. Make the setting in pixel units. Write 0 to the unused bits.
Color:	8- or 16-bit/pixel color specification. For 16-bit/pixel drawing, the color specification should match the destination pixel format. For 8-bit/pixel drawing, the same value should be set in the upper and lower bytes.
ROP:	Raster operation code

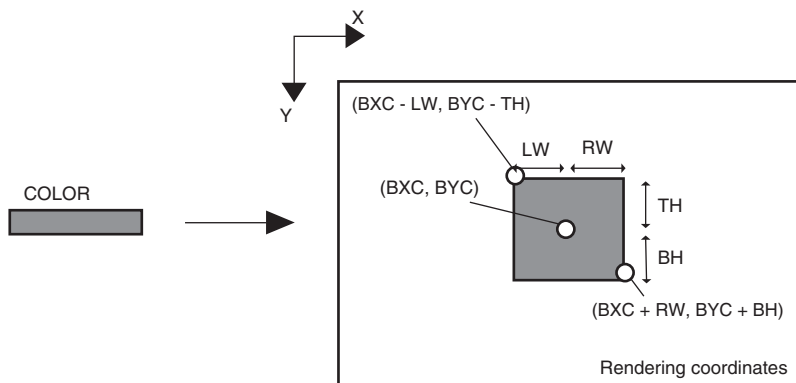
#### (c) Description

Draws a rectangle in the destination area in the single color specified by the Color parameter.

1. When work specification is selected as a rendering attribute (WORK = 1), only places where the work coordinate pixel is 1 are drawn at rendering coordinates while referencing work coordinates for the same coordinates as the rendering coordinates.
2. In 16-bit/pixel drawing, if the rendering attribute COOF bit is set to 1, the result of adding the value in COFSR to the value of the specified color is drawn. The operation is performed by saturation processing. In 8-bit/pixel drawing, the COOF bit should be cleared to 0.
3. The drawing direction can be selected by the DSTDIRX and DSTDIRY bits.
4. When  $\alpha E = 1$ , the specified color data and ground data are alpha blended before drawing. When setting  $\alpha E = 1$ , also set the ROP code = H'CC (source copy). The A value in the ARGB format is not alpha blended. The A value is drawn according to the source A value use (SAU) and A value (AVALUE) bits in the rendering control register (RCLR). Alpha blending is valid only in 16-bit/pixel drawing.
5. 16 raster operations are possible. The A value in the ARGB format is not subject to raster operations. The A value is drawn according to the source A value use (SAU) and A value (AVALUE) bits in the rendering control register (RCLR).

**Note:** System clipping or (relative) user clipping is performed when drawing a rectangle. Z clipping is performed only at the center coordinates.



**(d) Example**

23.2.6 Control Commands

(1) MOVE

(a) Function

Sets the current pointer.

(b) Command Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0100_1000								Reserve (all 0)								Draw Mode															
XC (-32768 ≤ XC ≤ 32767)																YC (-32768 ≤ YC ≤ 32767)															

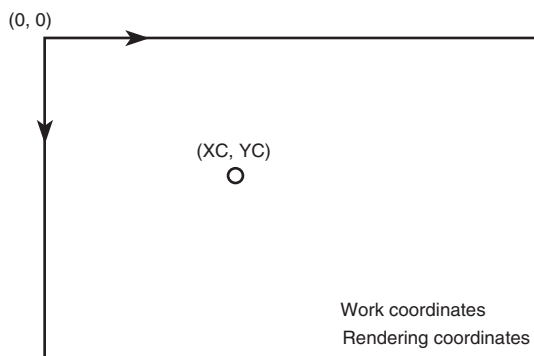
- 1. Code  
B'01001000
- 2. Rendering Attributes

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0

- 3. Command Parameters
  - XC: Rendering coordinate (absolute coordinate) or work coordinate (absolute coordinate). Negative number expressed as two's complement.
  - YC: Rendering coordinate (absolute coordinate) or work coordinate (absolute coordinate). Negative number expressed as two's complement.

**(c) Description**

Sets the values obtained by adding the local offset values to XC and YC in the current pointers. XC and YC are set as absolute coordinates. The current pointers are used by relative drawing commands only. After issuing a MOVE command, use relative drawing commands in succession. If an absolute drawing command is used during this sequence, the current pointers will be used as registers for internal computation, and the current pointer values will be lost. A MOVE command must be therefore be issued before using relative drawing commands again.

**(d) Example****(2) RMOVE****(a) Function**

Adds XC and YC to the current pointers.

**(b) Command Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0100_1100								Reserve (all 0)								Draw Mode															
XC (-32768 ≤ XC ≤ 32767)																YC (-32768 ≤ YC ≤ 32767)															

**1. Code**

B'01001100

## 2. Rendering Attributes

### Draw Mode

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0

## 3. Command Parameters

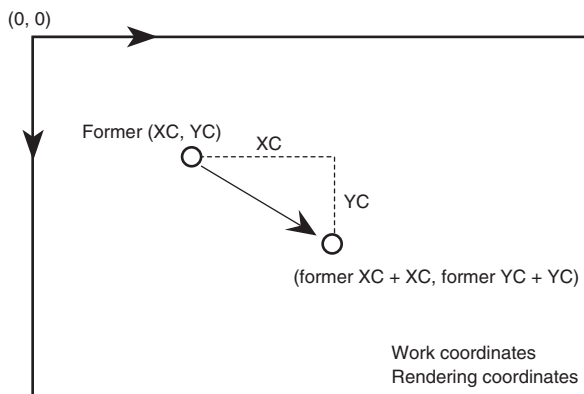
**XC:** Rendering coordinate (relative coordinate) or work coordinate (relative coordinate). Negative number expressed as two's complement.

**YC:** Rendering coordinate (relative coordinate) or work coordinate (relative coordinate). Negative number expressed as two's complement.

### (c) Description

Adds XC and YC to the current pointers.

### (d) Example



### (3) LCOFS

#### (a) Function

Sets the offset values (local offset) of the destination area and work area.

#### (b) Command Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0100_0000								Reserve (all 0)								Draw Mode															
XO (-32768 ≤ XO ≤ 32767)																YO (-32768 ≤ YO ≤ 32767)															

#### 1. Code

B'01000000

#### 2. Rendering Attributes

##### Draw Mode

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0

#### 3. Command Parameters

**XO:** Local offset value. Rendering coordinate (absolute coordinate) or work coordinate (absolute coordinate). Negative number expressed as two's complement.

**YO:** Local offset value. Rendering coordinate (absolute coordinate) or work coordinate (absolute coordinate). Negative number expressed as two's complement.

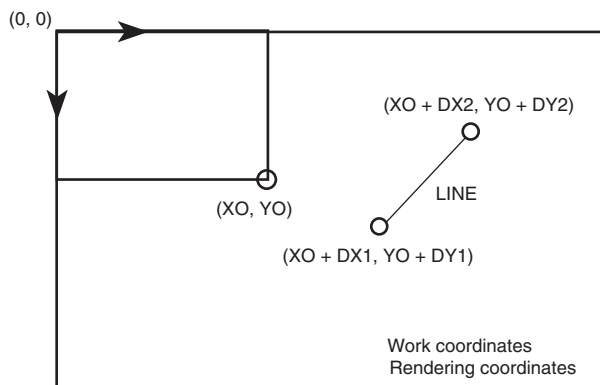
### (c) Description

After the local offset values are set, these offset values are added in all subsequent coordinate specifications made in drawing commands.

These settings must be made at the start of the display list (the initial values are undefined).

To reflect the local offset values in the current pointers, issue a MOVE command after the LCOFS command.

### (d) Example



#### (4) RLCOFS

##### (a) Function

Adds XO and YO to the local offset.

##### (b) Command Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0100_0100								Reserve (all 0)								Draw Mode															
XO (-32768 ≤ XO ≤ 32767)																YO (-32768 ≤ YO ≤ 32767)															

##### 1. Code

B'01000100

##### 2. Rendering Attributes

##### Draw Mode

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0

##### 3. Command Parameters

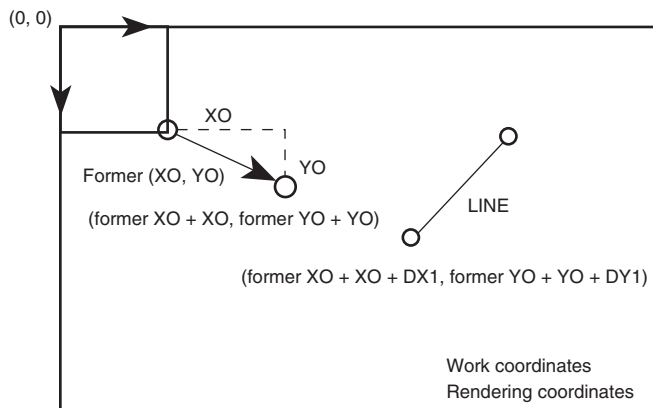
**XO:** Local offset value. Rendering coordinate (relative coordinate) or work coordinate (relative coordinate). Negative number expressed as two's complement.

**YO:** Local offset value. Rendering coordinate (relative coordinate) or work coordinate (relative coordinate). Negative number expressed as two's complement.

##### (c) Description

Adding XO and YO to the local offset makes the local offset values. After the local offset values are set, these offset values are added in all subsequent coordinate specifications made in drawing commands.

To reflect the local offset values in the current pointers, issue a MOVE command after setting the local offset with the LCOFS or RLCOFS command.

**(d) Example**



**(5) WPR****(a) Function**

Sets a value in a specific address-mapped register.

**(b) Command Format**

- LINKE = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																				
OP CODE = 0001_1000								Reserve (all 0)								Draw Mode																																			
Reserve (all 0)								n – 1 (0 ≤ n – 1 ≤ 255)								0	0	0	0	W Reg No																															
Data0																																																			
:																																																			
:																																																			
Data n–1																																																			

- LINKE = 1 and LREL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE = 0001_1000								Reserve (all 0)								Draw Mode																			
Reserve (all 0)								n - 1 (0 ≤ n - 1 ≤ 255)								0 0 0 0				W Reg No															
0 0 0			LINK Address (longword address)																												0 0 0				

Note: The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by the LINK Address.

- LINKE = 1 and LREL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE = 0001_1000									Reserve (all 0)							Draw Mode																	
Reserve (all 0)									n - 1 (0 ≤ n - 1 ≤ 255)							0 0 0 0				W Reg No													
Sign extended		Sign		LINK Address (longword address)																												0 0	

Note: The longword address following the LINK Address is handled as the next command code. Therefore, do not specify the longword address following the address where the LINK Address is to be assigned as the link destination address specified by the address where the command code is located plus the LINK Address.

## 1. Code

B'00011000

## 2. Rendering Attributes

**Draw Mode**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	LINKE	LREL	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	ByteM3	ByteM2	ByteM1	ByteM0

## 3. Command Parameters

W reg No: Register number

Data n (n = 1 to 256): Write data

n – 1: The number of write data

LINK Address: LINK absolute address (Longword address. Write 0 to bits A31 to A29, A1, and A0.)

LINK relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

**(c) Description**

Writes data to the address-mapped registers. The register number is set in W reg No, and the write data in Data n.

Also ensure that there is no conflict with access by the CPU.

1. When the LINKE bit is set to 1, data is read from the memory address specified by the LINK Address and written to a register.
2. The LINK Address can be specified through the LREL bit as an absolute address or a relative address with respect to the memory address at which the WPR command code is located.
3. Setting the ByteM3 to ByteM0 bits to 1 allows writing to a register to be masked in byte units.

**(6) JUMP****(a) Function**

Changes the display list fetch destination.

**(b) Command Format**

- REL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0010_1000								Reserve (all 0)								Draw Mode															
0	0	0	JUMP Address (longword address)																											0	0

- REL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
OP CODE = 0010_1000								Reserve (all 0)								Draw Mode																		
Sign extended		Sign		JUMP Address (longword address)																											0		0	

## 1. Code

B'00101000

## 2. Rendering Attributes

**Draw Mode**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	REL	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0

## 3. Command Parameter

**JUMP Address:** Jump destination absolute address (Longword address. Write 0 to bits A31 to A29, A1, and A0.)

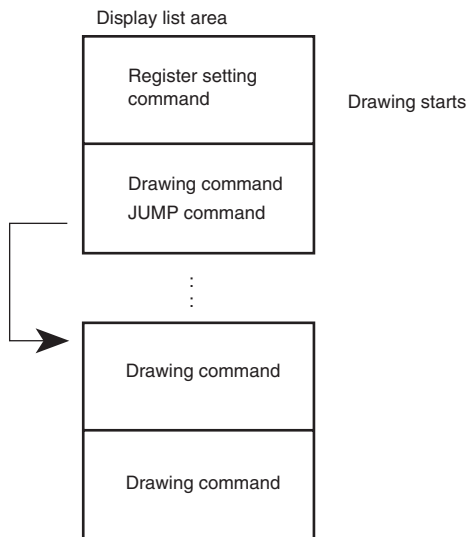
Jump destination relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

**Note:** Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

**(c) Description**

Changes the display list fetch destination to the specified address.

When REL = 0, the jump destination address can be specified as an absolute address. When REL = 1, the jump destination address can be specified as a relative address with respect to the memory address at which the command code is located.

**(d) Example**

## (7) GOSUB

### (a) Function

Makes a subroutine call for the display list.

### (b) Command Format

- REL = 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
OP CODE = 0011_0000								Reserve (all 0)								Draw Mode																
0	0	0	GOSUB Address (longword address)																												0	0

- REL = 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE = 0011_0000								Reserve (all 0)								Draw Mode																			
Sign extended		Sign		GOSUB Address (longword address)																												0		0	

#### 1. Code

B'00110000

#### 2. Rendering Attributes

##### Draw Mode

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	REL	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	No

#### 3. Command Parameter

GOSUB Address: Subroutine absolute address (Longword address. Write 0 to bits A31 to A29, A1, and A0.)

Subroutine relative address (Longword address. Negative number expressed as two's complement. Bits A31 to A29 are used to extend the sign in bit A28. Write 0 to bits A1 and A0.)

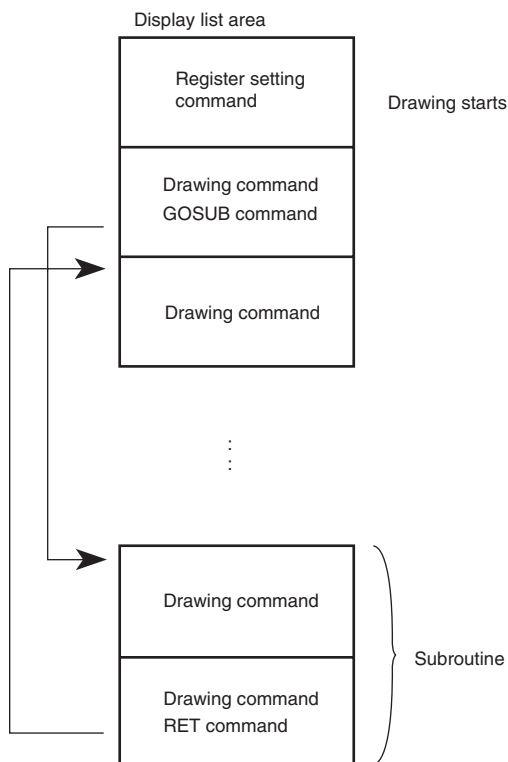
Note: Even in 32-bit addressing mode, write the values in bits 28 to 3 of the specified 32-bit address to bits A28 to A3.

**(c) Description**

Changes the display list fetch destination to the specified subroutine address. The fetch address is restored by an RET instruction. As only one level of nesting is permitted, it will not be possible to return if a subroutine call is issued within the subroutine.

When REL = 0, the subroutine address can be specified as an absolute address. When REL = 1, the jump destination address can be specified as a relative address with respect to the memory address at which the command code is located.

When the No bit is 0, the return address is set in the return address 0 register (RTN0R). When the No bit is 1, the return address is set in the return address 1 register (RTN1R).

**(d) Example**

**(8) RET****(a) Function**

Returns from a subroutine call made by the GOSUB command.

**(b) Command Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0011_1000								Reserve (all 0)								Draw Mode															

## 1. Code

B'00111000

## 2. Rendering Attributes

**Draw Mode**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	No

**(c) Description**

Restores the display list fetch destination to the address following the source of the subroutine call.

When the No bit is 0, the return address is set in the return address 0 register (RTN0R). When the No bit is 1, the return address is set in the return address 1 register (RTN1R).

**(9) NOP/INT****(a) Function**

Executes no operation.

**(b) Command Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0000_1000								Reserve (all 0)								Draw Mode															

## 1. Code

B'00001000

## 2. Rendering Attributes

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
INT	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	INT No							

**(c) Description**

This command does not perform any operation. This command simply fetches the next instruction. However when the INT bit is set to 1 in this command, after this command has been fetched, the INT bit in the status register (SR) is set to 1, INT No is saved in the interrupt command ID register (ICIDR), and the drawing operation is halted. Clearing the INT bit in the status register (SR) restarts the drawing operation from the next command.



**(10) VBKEM****(a) Function**

Performs synchronization with the frame change timing.

**(b) Command Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0001_0000								Reserve (all 0)								Draw Mode															

## 1. Code

B'00010000

## 2. Rendering Attributes

**Draw Mode**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0

**(c) Description**

When this command is executed, the drawing operation is kept waiting until the timing for a frame change. As soon as the frame change timing has elapsed, control passes to the next command. The frame change timing is the next VSYNC in non-interlace mode display or interlace sync & video mode display, and the starting point of the next frame in interlace sync mode display.

Note: This command can only be used in manual display charge mode or auto-rendering mode.

## (11) TRAP

### (a) Function

Informs the end of the display list.

### (b) Command Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0000_0000									Reserve (all 0)								Draw Mode														

#### 1. Code

B'00000000

#### 2. Rendering Attributes

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0	Flip5	Flip4	Flip3	Flip2	Flip1	Flip0

### (c) Description

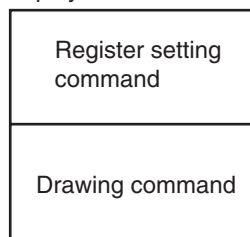
Halts the drawing operation and sets the TRA bit in the status register (SR) to 1. If the TRE bit in the interrupt enable register (IER) is set to 1, an interrupt is sent to the CPU.

This command must be placed at the end of the display list.

If the Flip5 to Flip0 bits are set, the corresponding plane is flipped (only valid in auto-rendering mode). The flip timing is the next VSYNC in non-interlace mode display or interlace sync & video mode display, and the starting point of the next frame in interlace sync mode display.

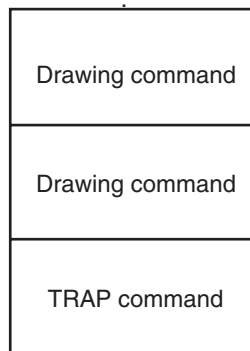
**(d) Example**

Display list area



Drawing starts

⋮



Drawing stops

TRA bit in status register (SR)  
is set to 1.If TRE = 1 at this time,  
an interrupt is generated externally.

## 23.3 Register Specifications

The CPU writing to the registers, excluding the system control registers, is prohibited after rendering has started and until the TRAP command is executed, except for the drawing halted period specified by the INT command. However, if a CPU write to the interrupt enable register (IER) conflicts with a WPR command write, the CPU write is given priority. Hereafter, “reset” refers to both a hardware reset and software reset unless specified otherwise. A hardware reset is a power-on reset.

**Table 23.5 Register Configuration**

Class	Register Name	Abbrev.	RW	WPR <sup>*1</sup>	Area P4 Address <sup>*2</sup>	Area 7 Address <sup>*2</sup>	Access Size
System control	System control	SCLR	R/W	N	H'FFEA 0000	H'1FEA 0000	32
	STatus	SR	R	N	H'FFEA 0004	H'1FEA 0004	32
	Status register clear	SRCR	W	N	H'FFEA 0008	H'1FEA 0008	32
	Interrupt enable	IER	R/W	Y	H'FFEA 000C	H'1FEA 000C	32
	Interrupt command ID	ICIDR	R	N	H'FFEA 0010	H'1FEA 0010	32
Memory control	Return address 0	RTN0R	R	Y	H'FFEA 0040	H'1FEA 0040	32
	Return address 1	RTN1R	R	Y	H'FFEA 0044	H'1FEA 0044	32
	Display list start address	DLSAR	R/W	N	H'FFEA 0048	H'1FEA 0048	32
	2-dimensional source area start address	SSAR	R/W	Y	H'FFEA 004C	H'1FEA 004C	32
	Rendering start address	RSAR	R/W	Y	H'FFEA 0050	H'1FEA 0050	32
	Work area start address	WSAR	R/W	Y	H'FFEA 0054	H'1FEA 0054	32
	Source stride	SSTRR	R/W	Y	H'FFEA 0058	H'1FEA 0058	32
	Destination stride	DSTRR	R/W	Y	H'FFEA 005C	H'1FEA 005C	32
	Endian conversion control	ENDCVR	R/W	N	H'FFEA 0060	H'1FEA 0060	32
Color control	Source transparent color	STCR	R/W	Y	H'FFEA 0080	H'1FEA 0080	32

Class	Register Name	Abbrev.	RW	WPR*1	Area P4 Address*2	Area 7 Address*2	Access Size
Color control	Destination transparent color	DTCR	R/W	Y	H'FFEA 0084	H'1FEA 0084	32
	Alpha value	ALPHR	R/W	Y	H'FFEA 0088	H'1FEA 0088	32
	Color offset	COFSR	R/W	Y	H'FFEA 008C	H'1FEA 008C	32
Rendering control	Rendering control	RCLR	R/W	Y	H'FFEA 00C0	H'1FEA 00C0	32
	Command status	CSTR	R	N	H'FFEA 00C4	H'1FEA 00C4	32
	Current pointer	CURR	R	N	H'FFEA 00C8	H'1FEA 00C8	32
	Local offset	LCOR	R	N	H'FFEA 00CC	H'1FEA 00CC	32
	System clipping area MAX	SCLMAR	R	Y	H'FFEA 00D0	H'1FEA 00D0	32
	User clipping area MIN	UCLMIR	R	Y	H'FFEA 00D4	H'1FEA 00D4	32
	User clipping area MAX	UCLMAR	R	Y	H'FFEA 00D8	H'1FEA 00D8	32
	Relative user clipping area MIN	RUCLMIR	R	Y	H'FFEA 00DC	H'1FEA 00DC	32
	Relative user clipping area MAX	RUCLMA R	R	Y	H'FFEA 00E0	H'1FEA 00E0	32
	Rendering control 2	RCL2R	R/W	Y	H'FFEA 00F0	H'1FEA 00F0	32
	Pattern offset	POFSR	R/W	Y	H'FFEA 00F8	H'1FEA 00F8	32
Coordinate transformation control	Coordinate transformation control	GTRCR	R/W	Y	H'FFEA 0100	H'1FEA 0100	32
	Matrix parameter A	MTRAR	R/W	Y	H'FFEA 0104	H'1FEA 0104	32
	Matrix parameter B	MTRBR	R/W	Y	H'FFEA 0108	H'1FEA 0108	32
	Matrix parameter C	MTRCR	R/W	Y	H'FFEA 010C	H'1FEA 010C	32
	Matrix parameter D	MTRDR	R/W	Y	H'FFEA 0110	H'1FEA 0110	32
	Matrix parameter E	MTRER	R/W	Y	H'FFEA 0114	H'1FEA 0114	32

Class	Register Name	Abbrev.	RW	WPR*1	Area P4 Address*2	Area 7 Address*2	Access Size
Coordinate transformation control	Matrix parameter F	MTRFR	R/W	Y	H'FFEA 0118	H'1FEA 0118	32
	Matrix parameter G	MTRGR	R/W	Y	H'FFEA 011C	H'1FEA 011C	32
	Matrix parameter H	MTRHR	R/W	Y	H'FFEA 0120	H'1FEA 0120	32
	Matrix parameter I	MTRIR	R/W	Y	H'FFEA 0124	H'1FEA 0124	32
	Coordinate transformation offset X	GTROFSX R	R/W	Y	H'FFEA 0128	H'1FEA 0128	32
	Coordinate transformation offset Y	GTROFSY R	R/W	Y	H'FFEA 012C	H'1FEA 012C	32
	Z clipping area MIN	ZCLPMIN R	R/W	Y	H'FFEA 0130	H'1FEA 0130	32
	Z clipping area MAX	ZCLPMAX R	R/W	Y	H'FFEA 0134	H'1FEA 0134	32
	Z saturation value MIN	ZSATVMI NR	R/W	Y	H'FFEA 0138	H'1FEA 0138	32

Notes: \*1 Y: WPR command setting is enable. N: WPR command setting is not disable.

\*2 The area P4 address is an address when accessing through area P4 in a virtual address space. The area 7 address is an address when accessing through area 7 in a physical space using the TLB.

Writing to the undefined address space is prohibited. If writing to such an address space is done, the G2D operation is not guaranteed.

Table 23.6 Register Bit Configuration

Class	Abbrev.	Data																																					
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
System control	SCLR																																						
	SRES																																						
	RS																																						
	SR																																						
	VER																																						
	MTRER																																						
	CER																																						
	INT																																						
	TRA																																						
	SRCR																																						
	MTCL																																						
	CECL																																						
	INCL																																						
	TRCL																																						
	IER																																						
	MTE																																						
	CEE																																						
	INE																																						
	TRE																																						
	ICIDR																																						
Memory control	RTN0R																																	0	0				
	RTN1R																																	0	0				
	DLSAR																																0	0	0	0			
	SSAR																																	0	0	0	0		
	RSAR																																	0	0	0	0		
	WSAR																																		0	0	0	0	
	SSTRR																																		0	0	0	0	
	DSTRR																																			0	0	0	0
	ENDCVR																																						
	LWSWAP																																						
	WSWAP																																						
	BYTESWAP																																						
	BITSWAP																																						
Color control	STCR																																						
	STC1																																						
	STC8																																						
	STC16																																						
	DTCR																																						
	DTC8																																						
	DTC16																																						
	ALPHR																																						
	COFSR																																						
	RGB: 565	COR																																					

[illegible]



TYPE	Register Abbrev.	Data																															
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Coordinate transformation control	MTRFR																																
	MTRGR																																
	MTRHR																																
	MTRIR																																
	GTROFSXR																																
	GTROFSYR																																
	ZCLPMINR																																
	ZCLPMAXR																																
	ZSATVMINR																																

Table 23.7 Initial Register Values at Hardware Reset and Software Reset

	Register abbrev.	Hardware reset	Software reset	Data																																
				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
System control	SCLR	Y		1																														0		
			Y	-																														0		
	SR	Y	Y	1	0	0	0								0																	0	0	0		
	SRCR	Y	Y												0																	0	0	0		
	IER	Y	Y												0																	0	0	0		
	ICIDR	N	N																										*	*	*	*	*	*	*	
Memory control	RTN0R	N	N			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	RTN1R	N	N			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	DLSAR	N	N			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	SSAR	N	N			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	DSAR	N	N			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	WSAR	N	N			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	SSTRR	N	N																								*	*	*	*	*	*	*	*	*	
	DSTRR	N	N																								*	*	*	*	*	*	*	*	*	
	ENDCVR	Y	Y																												0	0	0	0	0	
	Color control	STCR	N	N								*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
DTCR		N	N								*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
ALPHR		N	N																									*	*	*	*	*	*	*	*	*
COFSR		N	N			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
Rendering control	RCLR	Y	Y						0	0			0	0		0	0	0	0															0	0	
	CSTR	N	N			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	CURR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	LCOR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	SCLMAR	N	N																							*	*	*	*	*	*	*	*	*		
	UCLMIR	N	N													*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	UCLMAR	N	N													*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	RUCLMIR	N	N													*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	RUCLMAR	N	N													*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	RCL2R	Y	Y	0	0						0	0	0	0	0	0	0	0	0	1	0	0		0	0	0	0	0	0	0	0	0	1	0	0	
POF3R	Y	Y				0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0		

	Register abbrev.	Hardware reset	Software reset	Data																																
				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Coordinate transformation control	GTRCR	Y	Y	0																														0		
	MTRAR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRBR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRCR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRDR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRER	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRFR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRGR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRHR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRIR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	GTROFSXR	N	N																	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		
	GTROFSYR	N	N																	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	ZCLPMIN	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	ZCLPMAX	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	ZSATVMINR	N	N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

[Legend]

- \* : Undefined value. Value is retained at a hardware reset and software reset.
- 0 : Initialized to 0 at a hardware reset and software reset.
- 1 : Initialized to 1 at a hardware reset.
- : Reserved. This bit is always read as 0. The write value should always be 0.
- \* : Reserved. Value is retained at a hardware reset and software reset. The read value is undefined. The write value should always be 0.
- 0 : Reserved. Initialized to 0 at a hardware reset and software reset. The write value should always be 0.
- 1 : Reserved. Initialized to 1 at a hardware reset and software reset. The write value should always be 1.

## 23.3.1 System Control Registers

### (1) System Control Register (SCLR)

Offset: H'000

Initial Value: H'80000000

The system control register (SCLR) is a 32-bit readable/writable register that specifies system operation.

SCLR is initialized as follows at a hardware reset:

- Bit SRES is set to 1.
- Bit RS is cleared to 0.

Setting both the SRES and RS bits to 1 simultaneously is prohibited.

**Bit 31—Software Reset (SRES):** Resets the G2D.

Bit 31: SRES	Description
--------------	-------------

0	Command processing execution is enabled.
1	<p>This bit is set to 1 when a hardware reset is performed. (Initial value)</p> <p>Clear this bit to 0 in initialization.</p> <p>When this bit is set to 1 by software, a reset is performed for drawing operations only. The G2D registers are also initialized.</p> <p>While this bit is set to 1, this is the only register that can be written to.</p> <p>Note: For the software reset to be correctly reflected in this LSI, a method for reflecting and confirming write access is necessary, similar to that for memory access. Accordingly, after a software reset starts, execute the following processing before the software reset is canceled.</p> <ol style="list-style-type: none"> <li>1. When the G2D priority is equal to the CPU priority, execute dummy read three times for a random SDRAM area.</li> <li>2. When the G2D priority is level 2 whereas the CPU priority is level 3, execute dummy read once for a random SDRAM area.</li> <li>3. When the G2D priority is level 3 whereas the CPU priority is level 2, finish all SDRAM accesses by modules of level 2 or level 3, excluding the G2D.</li> </ol>

**Bits 30 to 1—Reserved:** The write value should always be 0. These bits are always read as 0.

**Bit 0—Rendering Start (RS):** Specifies the start of rendering. During the drawing period (from rendering start to TRAP command execution), writing 1 to this bit is prohibited.

Bit 0: RS	Description
-----------	-------------

0	Rendering is not started. (Initial value)
1	Rendering is started. This bit is cleared to 0 after rendering starts.

## (2) Status Register (SR)

Offset: H'004

Initial Value: H'80000000

The status register (SR) is a 32-bit read-only register used to read the internal status of the G2D from outside.

SR is initialized as follows at a hardware reset:

- The VER flag is set to 1000.
- All other flags are cleared to 0.

**Bits 31 to 28—Version Flag (VER):** This flag is read as 1000.

**Bit 18—Matrix Operation Error Flag (MTRER):** Flag that indicates that a coordinate transformation matrix operation result TX, TY, or W has exceeded the allowable range and saturation processing was executed.

Note: The MTRER bit is not masked by the coordinate transformation enable bit (GTE) in the coordinate transformation control register (GTRCR) or the rendering attribute MTRE bit. Thus, when GTE = 0 or when GTE = 1 with MTRE = 0, the MTRER bit may be set to 1 even when coordinate transformation is not performed. Therefore, do not use the MTRER bit unless both the GTE and MTRE bits are set to 1 throughout the period from rendering start to TRAP command issuance.

**Bit 18:**

<b>MTRER</b>	<b>Description</b>
0	Normal state. After MTRER flag clearing by the SRES bit in SCLR or the MTCL bit in SRCR, the coordinate transformation matrix operation result TX, TY, or W has not exceeded the allowable range (saturation processing not performed). (Initial value)
1	The coordinate transformation matrix operation result TX, TY, or W has exceeded the allowable range, and saturation processing was performed. Drawing operation is not halted.  The MTRER flag retains its state until cleared by a reset or by SRCR.

---

**Bit 2—Command Error Flag (CER):** Flag that indicates that an illegal command has been fetched.

Bit 2: CER	Description
0	<p>Normal state. An illegal command has not been fetched since CER flag clearing by the SRES bit in SCLR or the CECL bit in SRCR.</p> <p>An illegal command is one in which the upper eight bits of the command code are undefined. The G2D does not check the legality of the rendering attributes in the lower 16 bits. (Initial value)</p>
1	<p>Drawing operation halt state. Drawing operation remains halted because an illegal command was fetched after CER flag clearing by the SRES bit in SCLR or the CECL bit in SRCR. To resume drawing operation, after executing a software reset, make the bit setting for rendering start.</p> <p>The CER flag retains its state until cleared by a reset or by SRCR.</p>

**Bit 1—Interrupt Flag (INT):** Flag that indicates that the NOP/INT command has been fetched (only when the rendering attribute INT bit is 1).

Bit 1: INT	Description
0	<p>The NOP/INT command has not been fetched since INT flag clearing by the SRES bit in SCLR or the INCL bit in SRCR. (Initial value)</p>
1	<p>Drawing operation halt state. Drawing operation remains halted because the NOP/INT command was fetched after INT flag clearing by the SRES bit in SCLR or the INCL bit in SRCR (only when the rendering attribute INT bit is 1).</p> <p>Clearing the INT flag by the INCL bit in SRCR resumes drawing operation from the next command.</p> <p>The INT flag retains its state until cleared by a reset or by SRCR.</p> <p>Note: Do not rewrite the display list when drawing operation is halted by the INT command.</p>

**Bit 0—Trap Flag (TRA):** Flag that indicates the end of command execution.

Bit 0: TRA	Description
0	<p>The TRAP command has not been fetched since TRA flag clearing by the SRES bit in SCLR or the TRCL bit in SRCR. (Initial value)</p>
1	<p>Command execution has ended, or the current command is not being executed.</p> <p>The TRA flag retains its state until cleared by a reset or by SRCR.</p>

**Bits 27 to 19 and 17 to 3—Reserved:** These bits are always read as 0.

### (3) Status Register Clear Register (SRCR)

Offset: H'008

Initial Value: H'00000000

The status register clear register (SRCR) is a 32-bit write-only register that clears the corresponding flags in the status register (SR). When SR clearing is completed, all of the values in SRCR are cleared to 0 internally (the bits are read as 0).

Bit	Bit Name	Abbreviation	Description
18	Matrix operation error flag clear	MTCL	Writing 1 to the MTCL bit clears the MTRER flag in SR to 0.
2	Command error flag clear	CECL	Writing 1 to the CECL bit clears the CER flag in SR to 0.
1	Interrupt flag clear	INCL	Writing 1 to the INCL bit clears the INT flag in SR to 0.
0	Trap flag clear	TRCL	Writing 1 to the TRCL bit clears the TRA flag in SR to 0.
31 to 19 and 17 to 3	Reserved	—	The write value should always be 0.

### (4) Interrupt Enable Register (IER)

Offset: H'00C

Initial Value: H'00000000

The interrupt enable register (IER) is a 32-bit readable/writable register that enables or disables interrupts by the corresponding flags in the status register (SR). When a bit in SR is set to 1 and the bit at the corresponding bit position in IER is also 1, an interrupt request is sent to the CPU.

The interrupt generation condition is as follows.

$$\text{Interrupt generation condition} = a + b + c + d$$

a = MTRER. MTE

b = CER. CEE

c = INT. INE  
d = TRA. TRE

**Bit 18—Matrix Operation Error Flag Enable (MTE):** Enables or disables interrupts initiated by the MTRER flag in SR.

Note: The MTRER bit is not masked by the coordinate transformation enable bit (GTE) in the coordinate transformation control register (GTRCR) or the rendering attribute MTRE bit. Thus, when GTE = 0 or when GTE = 1 with MTRE = 0, the MTRER bit may be set to 1 even when coordinate transformation is not performed. Therefore, do not use the MTRER bit unless both the GTE and MTRE bits are set to 1 throughout the period from rendering start to TRAP command issuance.

Bit 18: MTE	Description
-------------	-------------

0	Interrupts initiated by the MTRER flag in SR are disabled. (Initial value)
1	Interrupts initiated by the MTRER flag in SR are enabled.

**Bit 2—Command Error Flag Enable (CEE):** Enables or disables interrupts initiated by the CER flag in SR.

Bit 2: CEE	Description
------------	-------------

0	Interrupts initiated by the CER flag in SR are disabled. (Initial value)
1	Interrupts initiated by the CER flag in SR are enabled.

**Bit 1—Interrupt Flag Enable (INE):** Enables or disables interrupts initiated by the INT flag in SR.

Bit 1: INE	Description
------------	-------------

0	Interrupts initiated by the INT flag in SR are disabled. (Initial value)
1	Interrupts initiated by the INT flag in SR are enabled.

**Bit 0—Trap Flag Enable (TRE):** Enables or disables interrupts initiated by the TRA flag in SR.

Bit 0: TRE	Description
------------	-------------

0	Interrupts initiated by the TRA flag in SR are disabled. (Initial value)
1	Interrupts initiated by the TRA flag in SR are enabled.

**Bits 31 to 19 and 17 to 3—Reserved:** The write value should always be 0.

### (5) Interrupt Command ID Register (ICIDR)

Offset: H'010

Initial Value: Undefined

The interrupt command ID register (ICIDR) is a 32-bit read-only register used to store the ID specified by the rendering attribute if the rendering attribute INT bit is set to 1 when the NOP/INT command is fetched. The unused bits are always read as 0. ICIDR retains its value at a reset.

#### Bits 7 to 0—Interrupt Command ID

**Bits 31 to 8—Reserved:** These bits are always read as 0.

## 23.3.2 Memory Control Registers

### (1) Return Address Register 0 (RTN0R)

Offset: H'040

Initial Value: Undefined

The return address register 0 (RTN0R) is a 32-bit read-only register which stores the return address when the rendering attribute No bit is 0 in the GOSUB command. The address indicated by RTN0R is a longword address (bits A28 to A2). RTN0R retains its value at a reset. (The unused bits are always read as undefined values.)

### (2) Return Address Register 1 (RTN1R)

Offset: H'044

Initial Value: Undefined

The return address register 1 (RTN1R) is a 32-bit read-only register which stores the return address when the rendering attribute No bit is 1 in the GOSUB command. The address indicated by RTN1R is a longword address (bits A28 to A2). RTN1R retains its value at a reset. (The unused bits are always read as undefined values.)

### (3) Display List Start Address Register (DLSAR)

Offset: H'048

Initial Value: Undefined



The display list start address register (DLSAR) is a 32-bit readable/writable register which specifies the memory area to be used as the display list. The start physical address (bits A28 to A0) of the display list is set in 16-byte units. Even in 32-bit addressing mode, write the lower 29 bits of the specified 32-bit address to bits 28 to 0. Write 0 to the lower four bits. Write 0 to the unused bits (these unused bits are always read as undefined values.) DLSAR retains its value at a reset. Do not map the display list to a tile addressing area (for details, see section 11, Memory Controller Unit (MCU)).

#### **(4) 2-Dimensional Source Area Start Address Register (SSAR)**

Offset: H'04C

Initial Value: Undefined

The 2-dimensional source area start address register (SSAR) is a 32-bit readable/writable register which specifies the memory area to be used as the 2-dimensional source area. The physical address set in this register becomes the physical address for the origin of the 2-dimensional source coordinates. The start physical address (bits A28 to A0) of the 2-dimensional source area is set in 16-byte units. Even in 32-bit addressing mode, write the lower 29 bits of the specified 32-bit address to bits 28 to 0. Write 0 to the lower four bits. Write 0 to the unused bits (these unused bits are always read as undefined values.) SSAR retains its value at a reset. When mapping the 2-dimensional source area to a tile addressing area (for details, see section 11, Memory Controller Unit (MCU)), write 0 to the lower nine bits (512-byte units).

#### **(5) Rendering Start Address Register (RSAR)**

Offset: H'050

Initial Value: Undefined

The rendering start address register (RSAR) is a 32-bit readable/writable register which specifies the memory area to be used as the rendering area. The physical address set in this register becomes the physical address for the rendering coordinate origin. The start physical address (bits A28 to A0) of the rendering area is set in 16-byte units. Even in 32-bit addressing mode, write the lower 29 bits of the specified 32-bit address to bits 28 to 0. Write 0 to the lower four bits. Write 0 to the unused bits (these unused bits are always read as undefined values.) RSAR retains its value at a reset. When mapping the rendering area to a tile addressing area (for details, see section 11, Memory Controller Unit (MCU)), write 0 to the lower nine bits (512-byte units).

Set the rendering start address so that the rendering area does not overlap with the work area.

## (6) Work Area Start Address Register (WSAR)

Offset: H'054

Initial Value: Undefined

The work area start address register (WSAR) is a 32-bit readable/writable register which specifies the memory area to be used as the work area. The physical address set in this register becomes the physical address for the work coordinate origin. The start physical address (bits A28 to A0) of the work area is set in 16-byte units. Even in 32-bit addressing mode, write the lower 29 bits of the specified 32-bit address to bits 28 to 0. Write 0 to the lower four bits. Write 0 to the unused bits (these unused bits are always read as undefined values.) WSAR retains its value at a reset. Do not map the work area to a tile addressing area (for details, see section 11, Memory Controller Unit (MCU)).

Use only the work drawing commands for drawing in the work area. When writing to the work area by the CPU, avoid the drawing period (from rendering start to TRAP command execution (including the drawing halt period specified by the NOP/INT command)). Do not use a figure drawn by a work drawing command as the source figure.

## (7) Source Stride Register (SSTRR)

Offset: H'058

Initial Value: Undefined

The source stride register (SSTRR) is a 32-bit readable/writable register which specifies the stride of the 2-dimensional source area. SSTRR retains its value at a reset.

**Bits 12 to 0—Source Stride (SSTRIDE):** These bits specify the stride of the 2-dimensional source area in pixel units. Set the value in the range of  $16 \leq \text{SSTRIDE} \leq 4096$ . Write 0 to the lower four bits (16-pixel units).

When the 2-dimensional source area to be used is mapped to a tile addressing area (for details, see section 11, Memory Controller Unit (MCU)), only a value of 512, 1024, 2048, or 4096 can be set.

**Bits 31 to 13—Reserved:** The write value should always be 0. These bits are always read as 0.

## (8) Destination Stride Register (DSTRR)

Offset: H'05C

Initial Value: Undefined

The destination stride register (DSTRR) is a 32-bit readable/writable register which specifies the stride of the destination area. DSTRR retains its value at a reset.

**Bits 12 to 0—Destination Stride (DSTRIDE):** These bits specify the stride of the destination area in pixel units. Set the value in the range of  $256 \leq \text{DSTRIDE} \leq 4096$ . Write 0 to the lower four bits (16-pixel units).

When the destination area to be used is mapped to a tile addressing area (for details, see section 11, Memory Controller Unit (MCU)), only a value of 512, 1024, 2048, or 4096 can be set.

**Bits 31 to 13—Reserved:** The write value should always be 0. These bits are always read as 0.

### (9) Endian Conversion Control Register (ENDCVR)

Offset: H'060

Initial Value: H'00000000

The endian conversion control register (ENDCVR) is a 32-bit readable/writable register which specifies the endian conversion mode.

**Bits 31 to 4—Reserved:** The write value should always be 0. These bits are always read as 0.

**Bit 3—Longword Swap (LWSWAP):** Swaps data in longword (32-bit) units.

**Bit 3:**

LWSWAP	Description
0	Data is not swapped. (Initial value)
1	Data is swapped in longword (32-bit) units.

**Bit 2—Word Swap (WSWAP):** Swaps data in word (16-bit) units.

**Bit 2:**

WSWAP	Description
0	Data is not swapped. (Initial value)
1	Data is swapped in word (16-bit) units.

**Bit 1—Byte Swap (BYTESWAP):** Swaps data in byte (8-bit) units.

**Bit 1:**

<b>BYTESWAP</b>	<b>Description</b>
-----------------	--------------------

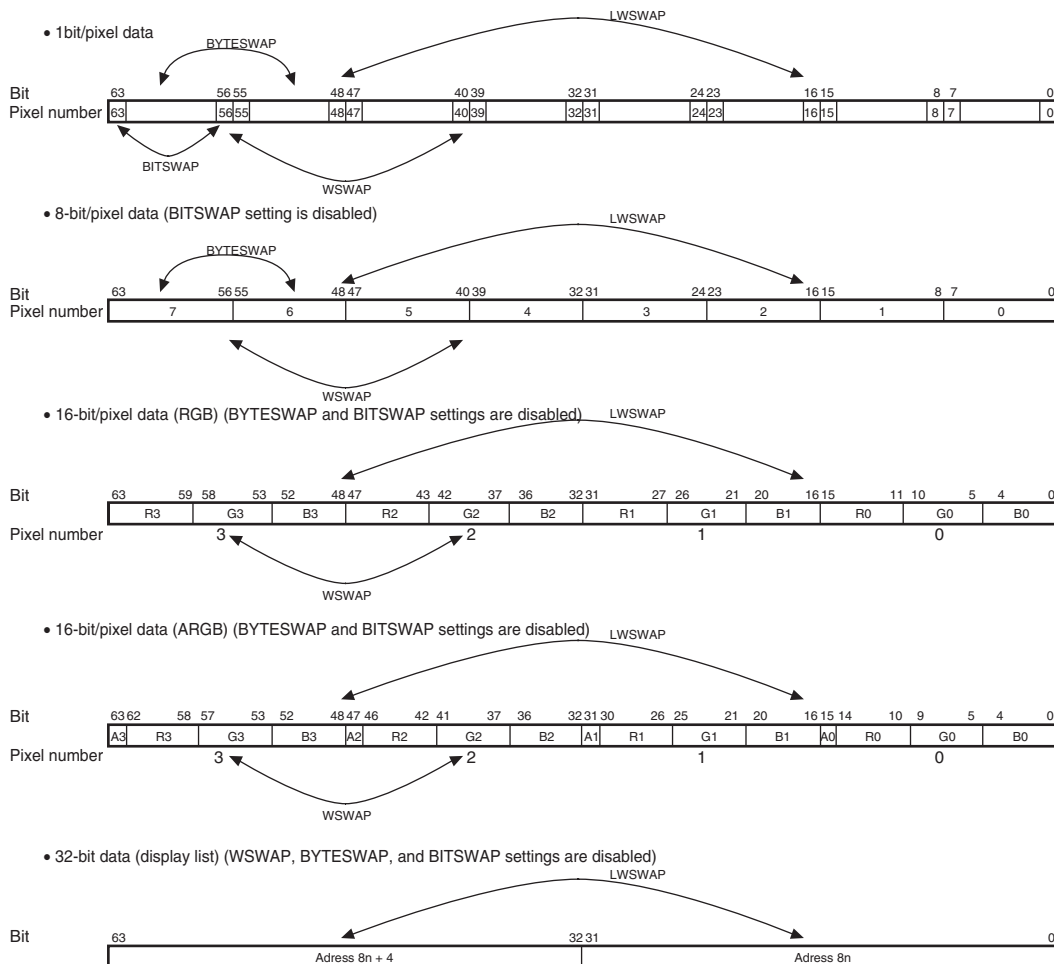
0	Data is not swapped. (Initial value)
1	Data is swapped in byte (8-bit) units.

**Bit 0—Bit Swap (BITSWAP):** Swaps data in bit units.

**Bit 0:**

<b>BITSWAP</b>	<b>Description</b>
----------------	--------------------

0	Data is not swapped. (Initial value)
1	Data is swapped in bit units.



### 23.3.3 Color Control Registers

#### (1) Source Transparent Color Register (STCR)

Offset: H'080

Initial Value: Undefined

The source transparent color register (STCR) is a 32-bit readable/writable register which compares the source data with the color set in this register when the rendering attribute STRANS bit is set to 1. The source color becomes transparent and drawing not performed when the source data matches

the color set in this register if the source transparent color polarity bit (STP) in the rendering control register (RCLR) is 0, and when the source data does not match the color set in this register if the STP bit in RCLR is 1. STCR retains its value at a reset.

Bit	Bit Name	Description
24	STC1	Transparent color for 1-bit/pixel source
23 to 16	STC8	Transparent color for 8-bit/pixel source
15 to 0	STC16	Transparent color for 16-bit/pixel source

For 16-bit/pixel source data, use the same format specified by the SPF bit in the rendering control register (RCLR). When SPF = 1 (ARGB = 1555), the A value is not compared.

**Bits 31 to 25—Reserved:** The write value should always be 0. These bits are always read as 0.

## (2) Destination Transparent Color Register (DTCR)

Offset: H'084

Initial Value: Undefined

The destination transparent color register (DTCR) is a 32-bit readable/writable register which compares the destination data with the color set in this register when the rendering attribute DTRANS bit is set to 1. The destination color becomes transparent and drawing not performed when the destination data matches the color set in this register if the destination transparent color polarity bit (DTP) in the rendering control register (RCLR) is 0, and when the destination data does not match the color set in this register if the DTP bit in RCLR is 1. DTCR retains its value at a reset.

Bit	Bit Name	Description
23 to 16	DTC8	Transparent color for 8-bit/pixel destination
15 to 0	DTC16	Transparent color for 16-bit/pixel destination

For 16-bit/pixel destination data, use the same format specified by the DPF bit in the rendering control register (RCLR). When DPF = 1 (ARGB = 1555), the A value is not compared.

**Bits 31 to 24—Reserved:** The write value should always be 0. These bits are always read as 0.

### (3) Alpha Value Register (ALPHR)

Offset: H'088

Initial Value: Undefined

The alpha value register (ALPHR) is a 32-bit readable/writable register which specifies the alpha blending value when the rendering attribute  $\alpha E$  bit is set to 1. ALPHR retains its value at a reset. For blending of the blue and red components, the upper five bits of the alpha value are valid. For blending of the green component, the upper six bits are valid when the destination pixel format is RGB and the upper five bits are valid when it is ARGB.

Bit	Bit Name	Description
7 to 0	ALPH	These bits set the alpha value.

Destination  $\approx$  source  $\times$  ALPH/255 + destination (1 – ALPH/255)  
(approximate expression when ALPH is an 8-bit value)

**Bits 31 to 8—Reserved:** The write value should always be 0. These bits are always read as 0.

### (4) Color Offset Register (COFSR)

Offset: H'08C

Initial Value: Undefined

The color offset register (COFSR) is a 32-bit readable/writable register. In 16-bit/pixel drawing, if the rendering attribute COOF bit is set to 1, the result of adding the value in COFSR to the value of the source data (color expanded data for a binary source and the specified color for the monochrome specification) is drawn. The operation is performed by saturation processing. In 8-bit/pixel drawing, the rendering attribute COOF bit must be cleared to 0. The offset components are treated as signed integers. Negative numbers are expressed as two's complement. COFSR retains its value at a reset.

- Source pixel format is RGB = 565 (SPF = 0)

Bit	Name	Description
23 to 19	COR (Color offset R)	Color offset red component
15 to 10	COG (Color offset G)	Color offset green component
7 to 3	COB (Color offset B)	Color offset blue component

Bits 18 to 16, 9, 8, and 2 to 0 are discarded. These bits are always read as 0.

- Source pixel format is ARGB = 1555 (SPF = 1)

Bit	Name	Description
23 to 19	COR (Color offset R)	Color offset red component
15 to 11	COG (Color offset G)	Color offset green component
7 to 3	COB (Color offset B)	Color offset blue component

Bits 18 to 16, 10 to 8, and 2 to 0 are discarded. These bits are always read as 0.

**Bits 31 to 24—Reserved:** The write value should always be 0. These bits are always read as 0.

### 23.3.4 Rendering Control Registers

#### (1) Rendering Control Register (RCLR)

Offset: H'0C0

Initial Value: H'00000000

The rendering control register (RCLR) is a 32-bit readable/writable register which specifies the rendering attributes.

**Bit 25—Source Transparent Color Polarity (STP):** Selects whether source transparency occurs when the source data and the value set in the source transparent color register (STCR) match or do not match.

Bit 25: STP	Description
0	Source transparency at a match (Initial value)
1	Source transparency at a mismatch



**Bit 24—Destination Transparent Color Polarity (DTP):** Selects whether destination transparency occurs when the destination data and the value set in the destination transparent color register (DTCR) match or do not match.

Bit 24: DTP	Description
-------------	-------------

0	Destination transparency at a match (Initial value)
1	Destination transparency at a mismatch

**Bit 21—Source Pixel Format (SPF):** Specifies the pixel format for the multi-valued source. This setting is valid only for a multi-valued 16-bit/pixel source. This bit should be cleared to 0 for an 8-bit-pixel source. Set this bit to match the destination pixel format.

Bit 21: SPF	Description
-------------	-------------

0	RGB = 565 format (Initial value)
1	ARGB = 1555 format

**Bit 20—Destination Pixel Format (DPF):** Specifies the pixel format for the destination. This setting is valid only for a 16-bit/pixel destination. This bit should be cleared to 0 for an 8-bit-pixel destination. Set this bit to match the multi-valued source pixel format.

Bit 20: DPF	Description
-------------	-------------

0	RGB = 565 format (Initial value)
1	ARGB = 1555 format

**Bit 18—Graphic Bit Mode (GBM):** Specifies the graphic bit mode for the multi-valued source and destination.

Bit 18: GBM	Description
-------------	-------------

0	8-bit/pixel (Initial value)
1	16-bit/pixel

**Bit 17—Source A Value Use (SAU):** When the pixel format of the source and destination is the ARGB format, drawing is performed while referencing the source A value as the destination A value.

**Bit 16—A Value (AVALUE):** When the pixel format of the source and destination is the ARGB format, drawing is performed with the destination A value as 0 or 1.

## Relationship between Source/Destination Pixel Format (SPF/DPF) and SAU and AVALUE Bits

SPF	DPF	SAU	AVALUE	Description
0	0	*	*	Source = RGB (565) and destination = RGB (565). The SAU and AVALUE bit settings are invalid.
0	1	*	*	Setting prohibited.
1	0	*	*	Setting prohibited.
1	1	0	0	Source = ARGB (1555) and destination = ARGB (1555). The destination A value is drawn as 0.
		0	1	Source = ARGB (1555) and destination = ARGB (1555). The destination A value is drawn as 1.
		1	*	Source = ARGB (1555) and destination = ARGB (1555). The destination A value is drawn referencing the source A value. The AVALUE bit setting is invalid.

Note: \* Don't care

When SAU = 1, the A value of the command parameter Color0 or Color1 is referenced in a binary source reference command and the A value of the command parameter Color is referenced in a monochrome specification command.

In the LINED command, the A value of the ground (destination) is written back, regardless of the settings of the SAU and AVALUE bits.

**Bit 1—Line Pre-Clipping Enable (LPCE):** This bit setting is valid in a LINE, RLINE, LINEW, or RLINEW command. When this bit is set to 1, pre-clipping is performed in line-segment units in the 2-dimensional clipping areas (system clipping, user clipping, and relative user clipping areas). If a line segment in the middle is pre-clipped, the pattern continuity is broken (the pattern starts from the final point of the line segment previously drawn).

Bit 1: LPCE	Description
0	Pre-clipping is not performed (Initial value)
1	Pre-clipping is performed in line-segment units in the 2-dimensional clipping areas

**Bit 0—Connection Drawing Mask (COM):** Selects whether the linkage parts of bold lines are drawn or not.

Bit 0: COM	Description
0	Linkage parts of bold lines are drawn (Initial value)
1	Linkage parts of bold lines are not drawn

**Bits 31 to 26, 23, 22, 19, and 15 to 2—Reserved:** The write value should always be 0. These bits are always read as 0.

## (2) Command Status Register (CSTR)

Offset: H'0C4

Initial Value: Undefined

The command status register (CSTR) is a 32-bit read-only register which stores the address of the fetched command word (op code word).

The address indicated by CSTR is a longword address (bits A28 to A2). The unused bits are always read as 0. CSTR retains its value at a reset.

## (3) Current Pointer Register (CURR)

Offset: H'0C8

Initial Value: Undefined

The current pointer register (CURR) is a 32-bit read-only register which indicates the current pointer coordinates.

The upper word indicates the X coordinate (XC) of the pointer and the lower word indicates the Y coordinate (YC) of the pointer.

CURR retains its value at a reset.

**(4) Local Offset Register (LCOR)**

Offset: H'0CC

Initial Value: Undefined

The local offset register (LCOR) is a 32-bit read-only register which indicates the offset coordinates.

The upper word indicates the X coordinate (XO) of the offset and the lower word indicates the Y coordinate (YO) of the offset.

LCOR retains its value at a reset.

**(5) System Clipping Area MAX Register (SCLMAR)**

Offset: H'0D0

Initial Value: Undefined

The system clipping area MAX register (SCLMAR) is a 32-bit read-only register which indicates the maximum values of the system clipping coordinates. The upper word indicates the maximum value of the system clipping X coordinate (SXMAX) and the lower word indicates the maximum value of the system clipping Y coordinate (SYMAX). The unused bits are always read as 0. When setting this register by the WPR command, set the maximum values of the drawing range (Max. 4095.  $SXMAX < DSTRR$ ).

SCLMAR retains its value at a reset.

**(6) User Clipping Area MIN Register (UCLMIR)**

Offset: H'0D4

Initial Value: Undefined

The user clipping area MIN register (UCLMIR) is a 32-bit read-only register which indicates the minimum values of the user clipping coordinates. The upper word indicates the minimum value of the user clipping X coordinate (UXMIN) and the lower word indicates the minimum value of the user clipping Y coordinate (UYMIN). The unused bits are always read as 0. When setting this register by the WPR command, set UXMIN and UYMIN in the following ranges:  $0 \leq UXMIN \leq UXMAX \leq SXMAX \leq 4095$ ,  $0 \leq UYMIN \leq UYMAX \leq SYMAX \leq 4095$ .

UCLMIR retains its value at a reset.

**(7) User Clipping Area MAX Register (UCLMAR)**

Offset: H'0D8

Initial Value: Undefined

The user clipping area MAX register (UCLMAR) is a 32-bit read-only register which indicates the maximum values of the user clipping coordinates. The upper word indicates the maximum value of the user clipping X coordinate (UXMAX) and the lower word indicates the maximum value of the user clipping Y coordinate (UYMAX). The unused bits are always read as 0. When setting this register by the WPR command, set UXMAX and UYMAX in the following ranges:  $0 \leq UXMIN \leq UXMAX \leq SXMAX \leq 4095$ ,  $0 \leq UYMIN \leq UYMAX \leq SYMAX \leq 4095$ .

UCLMAR retains its value at a reset.

**(8) Relative User Clipping Area MIN Register (RUCLMIR)**

Offset: H'0DC

Initial Value: Undefined

The relative user clipping area MIN register (RUCLMIR) is a 32-bit read-only register which indicates the minimum values of the relative user clipping coordinates (offset values added to the local offset). When setting this register by the WPR command, set the relative coordinates from the local offset. The upper word indicates the minimum value of the relative user clipping X coordinate (RUXMIN) and the lower word indicates the minimum value of the relative user clipping Y coordinate (RUYMIN). The unused bits are always read as 0. When setting this register by the WPR command, set RUXMIN and RUYMIN in the following ranges:  $0 \leq RUXMIN \leq RUXMAX \leq SXMAX \leq 4095$ ,  $0 \leq RUYMIN \leq RUYMAX \leq SYMAX \leq 4095$ . For details on the setting ranges, see (5) Relative Clipping Specification (RCLIP), in section 23.1.5, Rendering Attributes.

RUCLMIR retains its value at a reset.

**(9) Relative User Clipping Area MAX Register (RUCLMAR)**

Offset: H'0E0

Initial Value: Undefined

The relative user clipping area MAX register (RUCLMAR) is a 32-bit read-only register which indicates the maximum values of the relative user clipping coordinates (offset values added to the local offset). When setting this register by the WPR command, set the relative coordinates from

the local offset. The upper word indicates the maximum value of the relative user clipping X coordinate (RUXMAX) and the lower word indicates the maximum value of the relative user clipping Y coordinate (RUYMAX). The unused bits are always read as 0. When setting this register by the WPR command, set RUXMAX and RUYMAX in the following ranges:  $0 \leq \text{RUXMIN} \leq \text{RUXMAX} \leq \text{SXMAX} \leq 4095$ ,  $0 \leq \text{RUYMIN} \leq \text{RUYMAX} \leq \text{SYMAX} \leq 4095$ . For details on the setting ranges, see (5) Relative Clipping Specification (RCLIP), in section 23.1.5, Rendering Attributes.

RUCLMAR retains its value at a reset.

## (10) Rendering Control 2 Register (RCL2R)

Offset: H'0F0

Initial Value: H'00004004

The rendering control 2 register (RCL2R) is a 32-bit readable/writable register which specifies the rendering attributes.

**Bit 21—Destination Alpha Enable (DAE):** This bit is used in combination with the alpha blend enable ( $\alpha$ E) bit. With the ARGB = 1555 format, only the pixels whose destination (ground) A value is 1 are alpha blended. Pixels whose destination (ground) A value is 0 are not drawn.

Bit 21: DAE	Description
0	Alpha blending is performed regardless of the destination (ground) A value (Initial value)
1	Alpha blending is performed for only the pixels whose destination (ground) A value is 1

Notes: 1. Clear this bit to 0 for the RGB = 565 format or at 8-bit-pixel drawing.  
 2. Clear this bit to 0 for commands other than the POLYGON4 type commands.  
 3. Clear this bit to 0 when the alpha blend enable bit ( $\alpha$ E) is 0.  
 4. This bit is not decoded by a command so it must be set or cleared in each relevant command.

**Bit 20—Pattern Style Enable (PSTYLE):** This bit is used in combination with the source style specification (STYLE). The source pattern is created repeatedly in the pattern size based on the destination coordinates.

**Bit 20:****PSTYLE****Description**

0	Pattern style disabled (Initial value)
1	Source pattern is created based on the destination coordinates

- Notes:
1. Set the source offset TXOFS and TYOFS to 0.
  2. Clear this bit to 0 when the source style specification bit (STYLE) is 0.
  3. Clear the source address specification bit (SS) to 0.
  4. Clear this bit to 0 for commands other than the POLYGON4A and POLYGON4B commands.
  5. This bit is not decoded by a command so it must be set or cleared in each relevant command.

**Bits 19 and 18—Pattern X Size (PXSIZE):** These bits specify the pattern X size when the pattern style enable (PSTYLE) bit is 1.

**Bit 19:****PXSIZE[1]****Bit 18: PXSIZE[0]****Description**

0	0	Pattern X size = 8 pixels
	1	Pattern X size = 16 pixels
1	0	Pattern X size = 32 pixels
	1	Pattern X size = 64 pixels

Note: Set the specified pattern X size (8, 16, 32, or 64) in the source size TDX.

**Bits 17 and 16—Pattern Y Size (PYSIZE):** These bits specify the pattern Y size when the pattern style enable (PSTYLE) bit is 1.

**Bit 17:****PYSIZE[1]****Bit 16: PYSIZE[0]****Description**

0	0	Pattern Y size = 8 pixels
	1	Pattern Y size = 16 pixels
1	0	Pattern Y size = 32 pixels
	1	Pattern Y size = 64 pixels

Note: Set the specified pattern Y size (8, 16, 32, or 64) in the source size TDY.

**Bits 31 to 22 and 15 to 0—Reserved:** The write value should be the initial value.

### (11) Pattern Offset Register (POFSR)

Offset: H'0F8

Initial Value: H'00000000

The pattern offset register (POFSR) is a 32-bit readable/writable register which specifies the offset value when the pattern style enable (PSTYLE) bit is 1. The setting of this bit is referenced only when the pattern style enable (PSTYLE) bit is 1.

**Bits 31 to 16—Pattern Offset X (POFSX):** These bits specify the pattern offset value in the X direction as a 16-bit integer. A negative number is expressed as two's complement.

**Bits 15 to 0—Pattern Offset Y (POFSY):** These bits specify the pattern offset value in the Y direction as a 16-bit integer. A negative number is expressed as two's complement.

## 23.3.5 Coordinate Transformation Control Registers

### (1) Coordinate Transformation Control Register (GTRCR)

Offset: H'100

Initial Value: H'00000000

The coordinate transformation control register (GTRCR) is a 32-bit readable/writable register which sets the enable bits for enabling or disabling coordinate transformation.

**Bit 31—Coordinate Transformation Enable (GTE):** Performs coordinate transformation.

Bit 31: GTE	Description
0	Coordinate transformation is not performed. The rendering attribute MTRE bit is disabled. (Initial value)
1	The rendering attribute MTRE bit is enabled.



**Bit 0—Affine Transformation Enable (AFE):** Does not perform W division and offset addition at coordinate transformation. This bit is enabled when both the rendering attribute MTRE bit and GTE bit are set to 1.

Bit 0: AFE	Description
0	<p>The vertex coordinates X', Y' are obtained by dividing the matrix operation result coordinates TX, TY by WC, and then adding the offset values.</p> $X' = TX/WC + GTROFSX$ $Y' = TY/WC + GTROFSY$ <p>GTROFSX and GTROFSY are set in the coordinate transformation offset X register (GTROFSX) and coordinate transformation offset Y register (GTROFSY), respectively. (Initial value)</p>
1	<p>The vertex coordinates X', Y' are the matrix operation result coordinates TX, TY.</p> $X' = TX$ $Y' = TY$

**Bits 30 to 1—Reserved:** The write value should always be 0. These bits are always read as 0.

## (2) Matrix Parameter A Register (MTRAR)

Offset: H'104

Initial Value: Undefined

The matrix parameter A register (MTRAR) is a 32-bit readable/writable register which specifies a matrix parameter at coordinate change in the single-precision floating-point format defined by the IEEE 754 standard. However, since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), MTRAR should be set within the range of  $-2^{15} \leq \text{MTRAR} < 2^{15}$ .

MTRAR retains its value at a reset.

**Note:** For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

### (3) Matrix Parameter B Register (MTRBR)

Offset: H'108

Initial Value: Undefined

The matrix parameter B register (MTRBR) is a 32-bit readable/writable register which specifies a matrix parameter at coordinate change in the single-precision floating-point format defined by the IEEE 754 standard. However, since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), MTRBR should be set within the range of  $-2^{15} \leq \text{MTRBR} < 2^{15}$ .

MTRBR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

### (4) Matrix Parameter C Register (MTRCR)

Offset: H'10C

Initial Value: Undefined

The matrix parameter C register (MTRCR) is a 32-bit readable/writable register which specifies a matrix parameter at coordinate change in the single-precision floating-point format defined by the IEEE 754 standard. However, since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), MTRCR should be set within the range of  $-2^{15} \leq \text{MTRCR} < 2^{15}$ .

MTRCR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

### (5) Matrix Parameter D Register (MTRDR)

Offset: H'110

Initial Value: Undefined

The matrix parameter D register (MTRDR) is a 32-bit readable/writable register which specifies a matrix parameter at coordinate change in the single-precision floating-point format defined by the IEEE 754 standard. However, since internal computation is carried out with 32-bit fixed-point

operations (16-bit integer portion and 16-bit fractional portion), MTRDR should be set within the range of  $-2^{15} \leq \text{MTRDR} < 2^{15}$ .

MTRDR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

#### (6) Matrix Parameter E Register (MTRER)

Offset: H'114

Initial Value: Undefined

The matrix parameter E register (MTRER) is a 32-bit readable/writable register which specifies a matrix parameter at coordinate change in the single-precision floating-point format defined by the IEEE 754 standard. However, since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), MTRER should be set within the range of  $-2^{15} \leq \text{MTRER} < 2^{15}$ .

MTRER retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

#### (7) Matrix Parameter F Register (MTRFR)

Offset: H'118

Initial Value: Undefined

The matrix parameter F register (MTRFR) is a 32-bit readable/writable register which specifies a matrix parameter at coordinate change in the single-precision floating-point format defined by the IEEE 754 standard. However, since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), MTRFR should be set within the range of  $-2^{15} \leq \text{MTRFR} < 2^{15}$ .

MTRFR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

**(8) Matrix Parameter G Register (MTRGR)**

Offset: H'11C

Initial Value: Undefined

The matrix parameter G register (MTRGR) is a 32-bit readable/writable register which specifies a matrix parameter at coordinate change in the single-precision floating-point format defined by the IEEE 754 standard. However, since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), MTRGR should be set within the range of  $-2^{15} \leq \text{MTRGR} < 2^{15}$ .

MTRGR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

**(9) Matrix Parameter H Register (MTRHR)**

Offset: H'120

Initial Value: Undefined

The matrix parameter H register (MTRHR) is a 32-bit readable/writable register which specifies a matrix parameter at coordinate change in the single-precision floating-point format defined by the IEEE 754 standard. However, since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), MTRHR should be set within the range of  $-2^{15} \leq \text{MTRHR} < 2^{15}$ .

MTRHR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

**(10) Matrix Parameter I Register (MTRIR)**

Offset: H'124

Initial Value: Undefined

The matrix parameter I register (MTRIR) is a 32-bit readable/writable register which specifies a matrix parameter at coordinate change in the single-precision floating-point format defined by the IEEE 754 standard. However, since internal computation is carried out with 32-bit fixed-point

operations (16-bit integer portion and 16-bit fractional portion), MTRIR should be set within the range of  $-2^{15} \leq \text{MTRIR} < 2^{15}$ .

MTRIR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

### **(11) Coordinate Transformation Offset X Register (GTROFSXR)**

Offset: H'128

Initial Value: Undefined

The coordinate transformation offset X register (GTROFSXR) is a 32-bit readable/writable register which specifies the X offset value at coordinate change as a 16-bit integer. A negative number is expressed as two's complement.

GTROFSXR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

### **(12) Coordinate Transformation Offset Y Register (GTROFSYR)**

Offset: H'12C

Initial Value: Undefined

The coordinate transformation offset Y register (GTROFSYR) is a 32-bit readable/writable register which specifies the Y offset value at coordinate change as a 16-bit integer. A negative number is expressed as two's complement.

GTROFSYR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

### **(13) Z Clipping Area MIN Register (ZCLPMINR)**

Offset: H'130

Initial Value: Undefined

The Z clipping area MIN register (ZCLPMINR) is a 32-bit readable/writable register which specifies the minimum value of the Z clipping area in the single-precision floating-point format defined by the IEEE 754 standard. Since the setting is compared with the W value, set a value corresponding to W in ZCLPMINR.

Since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), ZCLPMINR should be set within the range of  $2^{-16} \leq \text{ZCLPMINR} \leq \text{ZCLPMAXR} < 2^{15}$ .

ZCLPMINR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

#### **(14) Z Clipping Area MAX Register (ZCLPMAXR)**

Offset: H'134

Initial Value: Undefined

The Z clipping area MAX register (ZCLPMAXR) is a 32-bit readable/writable register which specifies the maximum value of the Z clipping area in the single-precision floating-point format defined by the IEEE 754 standard. Since the setting is compared with the W value, set a value corresponding to W in ZCLPMAXR.

Since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), ZCLPMAXR should be set within the range of  $2^{-16} \leq \text{ZCLPMINR} \leq \text{ZCLPMAXR} < 2^{15}$ .

ZCLPMAXR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.

#### **(15) Z Saturation Value MIN Register (ZSATVMINR)**

Offset: H'138

Initial Value: Undefined

The Z saturation value MIN register (ZSATVMINR) is a 32-bit readable/writable register which specifies the minimum Z saturation value in the single-precision floating-point format defined by

the IEEE 754 standard. Since the setting is compared with the W value, set a value corresponding to W in ZSATVMINR.

Since internal computation is carried out with 32-bit fixed-point operations (16-bit integer portion and 16-bit fractional portion), ZSATVMINR should be set within the range of  $2^{-16} \leq \text{ZSATVMINR} \leq \text{ZCLPMINR} \leq \text{ZCLPMAXR} < 2^{15}$ .

ZSATVMINR retains its value at a reset.

Note: For details on the setting range, see (2)  $4 \times 4$  Matrix Operation, to (5) Coordinate Transformation Flow and Saturation Processing, in section 23.1.2, Basic Functions.





## Section 24 Video Display Controller (VDC2)

### 24.1 Overview

The video display controller (VDC2) provides functions for reading four planes of graphic images (layers 1 to 4) stored in the external memory and overlaying them. It outputs 18-bit RGB video (each color is represented by six bits) and digital video data conforming to BTA T-1004.

### 24.2 Features

Item	Function
Operating frequency	T-1004 display clock: 54 MHz RGB666 display clock: 6.0 MHz to 36.0 MHz (depends on the display panel size)
Input image format	16-bit RGB565 progressive (SDRAM)
Display size	<ul style="list-style-type: none"> <li>18-bit progressive RGB output 720 × 480 (NTSC) 720 × 576 (PAL) 320 × 240 (QVGA) 640 × 480 (VGA) 800 × 480 (WVGA)</li> <li>8-bit digital output conforming to BTA T-1004 (parallel interface in the 8:4:4 bit format) (the RGB data output timing can be set to the rising or falling edge of the clock through the SYNCNT register setting) 720 × 480 (NTSC)</li> </ul>
Display planes	Up to four planes (layers 1 to 4)
$\alpha$ blending	Mixes layers 1 to 4 according to the transparency ( $\alpha$ value).
Chroma-keying	Applies chroma-key processing to the specified RGB color (transparency can be specified as the $\alpha$ value)

Item	Function
Output video format	RGB666 progressive video output (each of RGB colors is represented by 6 bits: 18 bits in total)  8-bit digital video output conforming to BTA T-1004 (parallel interface in the 8:4:4 bit format) (the RGB data output timing can be set to the rising or falling edge of the clock through the SYNCNT register setting)
Sync signal output	Either a combination of Vsync, Hsync, data enable, and COM/CDE signals, or a combination of SPL, CLS, SPS, data enable, and COM/CDE signals can be selected (each signal output timing can be set to the rising or falling edge of the clock and the polarity can be selected through the SYNCNT register setting).
External sync mode	The VDC2 can operate with external sync signals (EX-VSYNC and EX-HSYNC) and the panel clock (the external sync signal timing can be set to the rising or falling edge of the clock and the polarity can be selected through the SYNCNT register setting).  Note that only RGB666 video data can be output in this mode.
Chroma enable signal output	Outputs a chroma data enable (CDE) signal for the specified color in the video image.

## 24.3 Input/Output Pins

**Table 24.1 Pin Configuration**

Symbol	I/O	Pin Name	Function
DR [5:0]	Output	Digital red data	Video data output pins.
DG [5:0]	Output	Digital green data	Video data output pins.
DB [5:0]	Output	Digital blue data	Video data output pins.
VSYNC/SPS	Output	Vertical sync signal/gate start signal	Vertical sync signal/gate start signal.
HSYNC/SPL	Output	Horizontal sync signal/sampling start signal	Horizontal sync signal/sampling start signal.
DE_V/CLS	Output	Vertical data enable signal/gate clock signal	Vertical data enable signal/gate clock signal.
DE_H/DE_C	Output	Horizontal data enable signal/display enable signal	Horizontal data enable signal/display enable signal.
COM/CDE	Output	Gate control signal/chroma data enable signal	Gate control signal/display enable signal (asserted when the data matches the chroma-key color specified in the register).
BT_DATA[7:0]	Output	BTA-T1004 display data	BTA-T1004 display data output pins.
BT_VSYNC	Output	BTA-T1004 vertical sync	BTA-T1004 vertical sync signal.
BT_HSYNC	Output	BTA-T1004 horizontal sync	BTA-T1004 horizontal sync signal.
BT_DE_C	Output	BTA-T1004 display enable	BTA-T1004 display enable signal.
EX_VSYNC	Input	VSYNC input	VSYNC input pin used in external sync mode.
EX_HSYNC	Input	HSYNC input	HSYNC input pin used in external sync mode.
DCLKIN	Input	Panel source clock input	Display source clock input pin. Input an appropriate frequency depending on the display panel.
DCLKOUT	Output	Panel clock output	Panel clock output pin.

## 24.4 VDC2 Configuration

The VDC2 consists of seven functional blocks listed in table 24.2. Figure 24.1 shows the entire block diagram of the VDC2.

**Table 24.2 Functional Blocks in VDC2**

Block Name	Overview of Functions
Graphics block 1 (layer 1)	Reads a graphic image (RGB565: layer 1) stored in the external memory through the pixel bus and outputs it to graphics block 2.
Graphics block 2 (layer 2)	Reads a graphic image (RGB565: layer 2) stored in the external memory through the pixel bus, overlays it on the output from graphics block 1, and outputs the result to graphics block 3.
Graphics block 3 (layer 3)	Reads a graphic image (RGB565: layer 3) stored in the external memory through the pixel bus, overlays it on the output from graphics block 2, and outputs the result to graphics block 4.
Graphics block 4 (layer 4)	Reads a graphic image (RGB565: layer 4) stored in the external memory through the pixel bus, overlays it on the output from graphics block 3, and outputs the resultant image data.
Display control block	Converts the output (RGB) from graphics block 4 into the YCbCr(4:2:2) format and outputs the data in the 8:4:4 parallel format conforming to the BTA T-1004 standard. It also outputs the control signals for the TFT-LCD panel.
Input timing control block	Selects the timing of the external sync signal input with respect to the clock rising or falling edge and selects the sync signal polarity.
Output timing control block	Controls the timing of the sync signal output with respect to the clock rising or falling edge and controls the sync signal polarity. It also controls the timing of the RGB666 video output signals with respect to the clock rising or falling edge.

**Note:** Layers 1 to 4 have the same configuration except that the bottom layer (layer 1) receives no image from another layer as the target of  $\alpha$  blending.

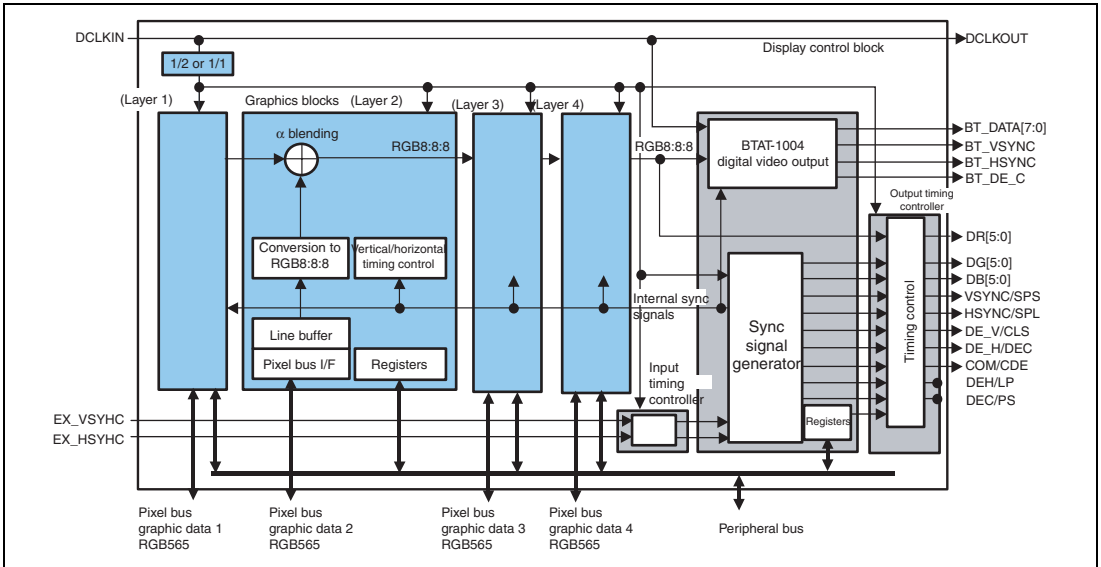
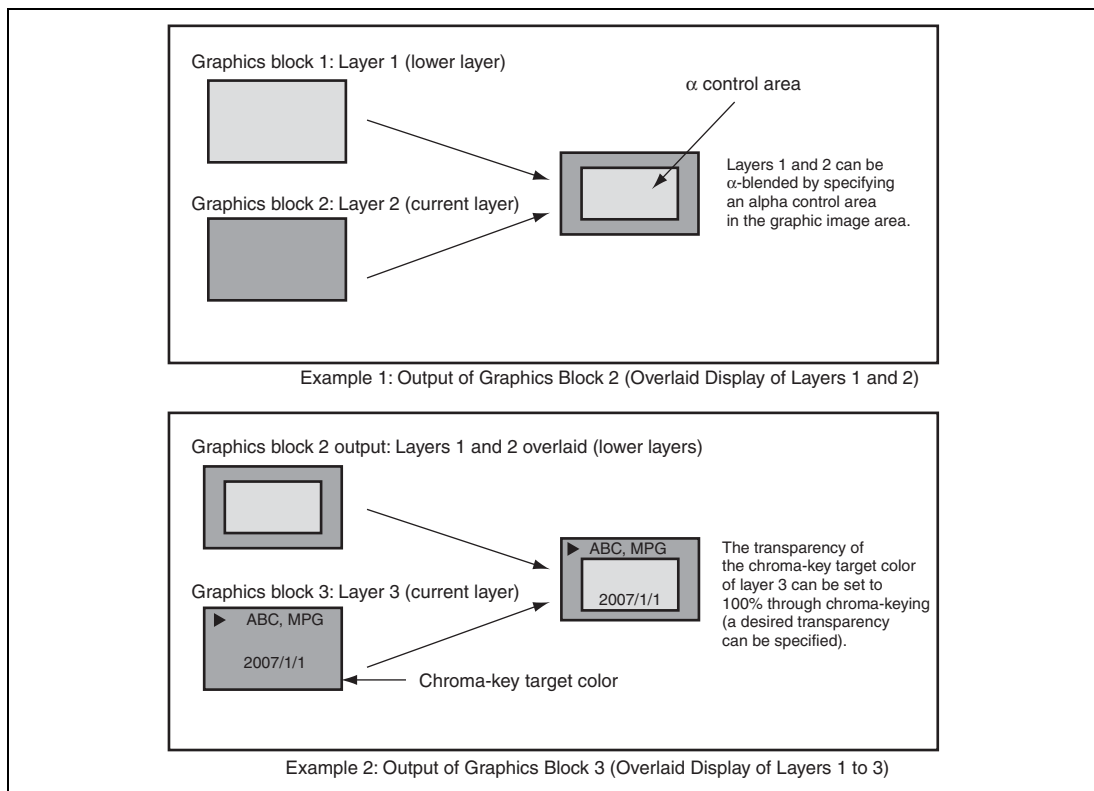


Figure 24.1 Block Diagram of VDC2

## 24.5 Functional Descriptions

### 24.5.1 Graphics (Layers 1 to 4)

The graphics blocks display in the RGB565 (16-bit) format the image data stored in the memory area. The graphics blocks control display by using the external input sync signals or internally generated sync signals. A single plane of an image can be displayed, and two to four planes of images can also be displayed through overlay processing. In overlaid display, the lower-layer images can be displayed through the current image (current layer) by specifying the  $\alpha$  control area for the current layer (transparent processing). The transparency can be specified in  $1/256 \times 100\%$  units. Transparent processing for the lower-layer images is also available through chroma-keying, which specifies the transparency of the specified target color. Figure 24.2 shows examples of overlaid display.

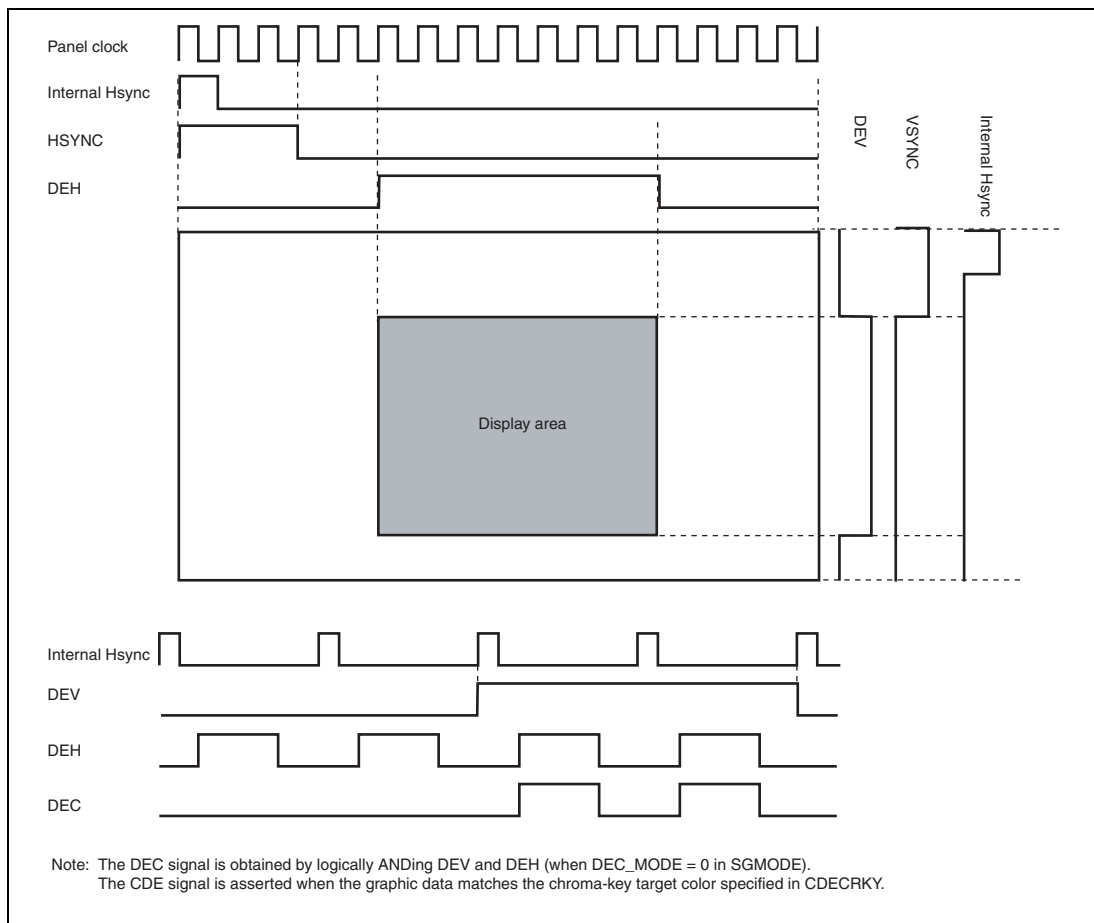


**Figure 24.2 Examples of Overlaid Display**

### 24.5.2 Sync Signal Generation

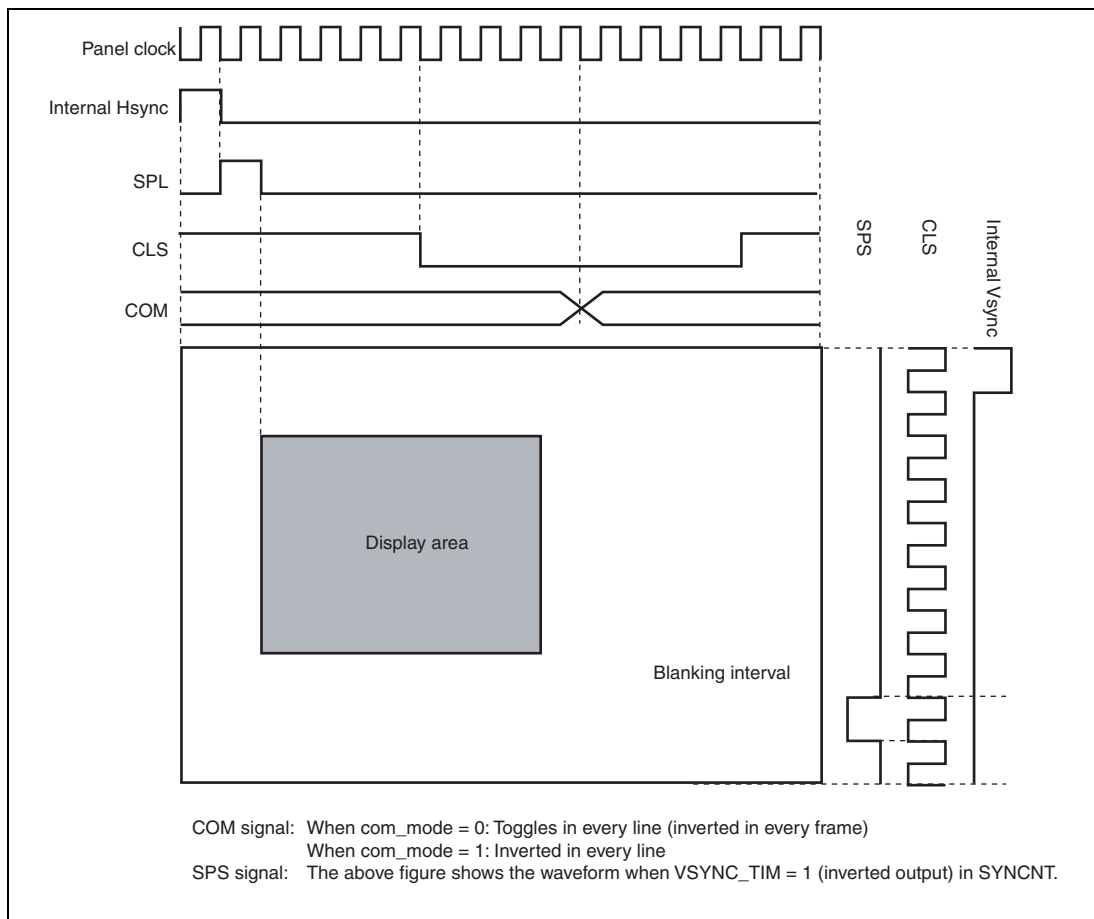
Figures 24.3 and 24.4 show examples of sync signal formats that can be generated. The VDC2 generates and outputs Vsync, Hsync, DEV, and DEH/DEC.

VSYNC:	Vertical sync signal	SPS:	Gate start signal
HSYNC:	Horizontal sync signal	SPL:	Sampling start signal
DEV:	Vertical data enable signal	CLS:	Gate clock signal
DEH:	Horizontal data enable signal	DEC:	Data enable signal (composite)
CDE:	Chroma data enable signal	COM:	Gate control signal



**Figure 24.3 Format 1 (Vsync, Hsync, DEV, DEH, DEC, and CDE Output)**





**Figure 24.4 Format 2 (SPS, SPL, CLS, and COM Output)**

### 24.5.3 External Sync Mode

External sync mode outputs the graphic images with synchronizing the vertical or the horizontal sync signal that are from the external sync signal generating circuit such as TV or video. Inputs the vertical sync signal, the horizontal sync signal or the clock to corresponding pins, EX\_VSYNC, EX\_HSYNC or DCLKIN.

Set up the registers related to the sync signals as follows.

#### (1) Setting up External Sync Mode

Set the SYNC\_SEL bit to 1 in SGMODE register in order to make the external sync mode. If the electrode of input vertical/horizontal signal is negative, reverse the input data with setting the EX\_V\_TYPE bit to 1 and the EX\_H\_TYPE bit to 1 in SYNCNT register.

#### (2) Setting up Output for COM/CDE Pins

Set the COM\_CDE\_SEL bit to 1 and the CDE\_EXE bit to 1 in SGMODEA register, and output CDE signal to the COM/CDE pins. Assert the CDE signal only if the signal corresponds to the target color of chroma-keying setting in the CDECRKY register. The COM\_TYPE bit in SYNCNT controls the electrode of CDE signal.

#### (3) Setting up Timing for Input and Output

Set up the sampling timing for vertical/horizontal signal which is input data and the output timing for RGB data and CDE signal which are output data with DCLKIN rising or falling according to the specifications of the display as an external sync signal generating circuit. (refer to the SYNCNT register)

VDC2 wait the EX\_VSYNC signal staying in the vertical blanking interval until the signal input. (Vsync does not perform self-processing.) In the same way, when the EX\_HSYNC signal is input to VDC2, VDC2 performs horizontal indicating completion and transfer to the next processing. VDC2 wait the EX\_HSYNC signal staying in the horizontal blanking interval until the signal input. (Hsync does not perform self-processing.)

### 24.5.4 Digital Video Output

For the BTA T-1004 digital video output, the VDC2 generates an 8-bit luminance signal (Y), chrominance signals (CB and CR), EAV, and SAV conforming to the BTA T-1004 (8:4:4-format parallel bit interface) standard.

### 24.5.5 Conversion from RGB565 to YC444

The VDC2 converts the RGB565 format into the YC444 format in accordance with the ITU-R BT601 standard. First, RGB565 is converted to RGB888 through equation (1) shown below to control the R, G, and B values within the range from 0 to 255. Conversion from RGB to YC uses colorimetry conversion equation (2) prescribed in the BT601 standard.

$$R_1 = \frac{255}{31} R$$

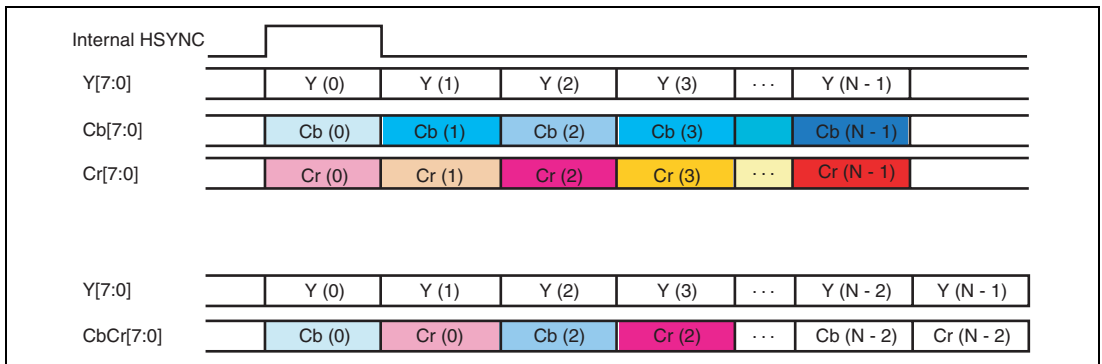
$$G_1 = \frac{255}{63} G \quad \dots\dots (1)$$

$$B_1 = \frac{255}{31} B$$

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R_1 \\ G_1 \\ B_1 \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad \dots\dots (2)$$

### 24.5.6 Conversion from YC444 to YC422

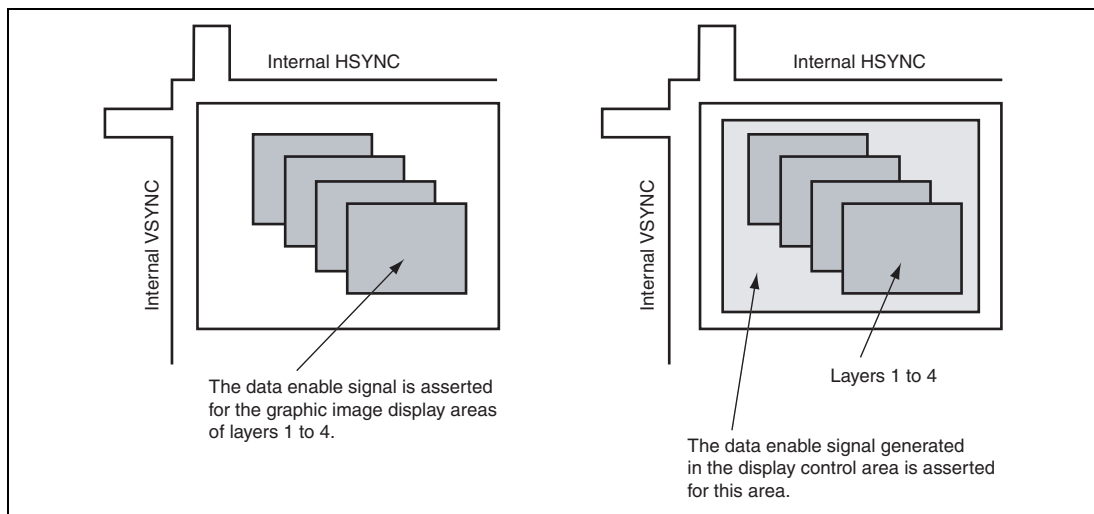
The VDC2 converts the YC444 format to the YCbCr422 format. Figure 24.5 shows the timing of this conversion. The chrominance conversion from YC444 to YC422 uses the holding method.



**Figure 24.5 Timing of Conversion from YC444 to YC422**

### 24.5.7 Data Enable Signal (Composite)

Either the data enable signal generated in the graphics blocks (obtained by logically ORing signals for layers 1 to 4) or the data enable signal (rectangle) generated in the display control block can be selected through the DEC\_MODE bit in SGMODE.



**Figure 24.6 Data Enable Signals**

## 24.6 Register Descriptions

The following registers are allocated to the SH register map space.

### Legends for register description:

Initial value: Register value after 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.

—/W: Write-only; an undefined value is read.

**Table 24.3 Register Configuration in Graphics Block 1**

Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size
Graphics block control register	GRCMEN1	R/W	H'FFEC 0000	H'1FEC 0000	32, 16, or 8
Bus control register	GRCBUSCNT1	R/W	H'FFEC 0004	H'1FEC 0004	32, 16, or 8
Reserved	—	R	H'FFEC 0008	H'1FEC 0008	32, 16, or 8
Reserved	—	R	H'FFEC 000C	H'1FEC 000C	32, 16, or 8
Reserved	—	R	H'FFEC 0300	H'1FEC 0300	32, 16, or 8
Reserved	—	R	H'FFEC 0304	H'1FEC 0304	32, 16, or 8
Graphic image base address register	GROPSADR1	R/W	H'FFEC 0308	H'1FEC 0308	32, 16, or 8
Graphic image area register	GROPSWH1	R/W	H'FFEC 030C	H'1FEC 030C	32, 16, or 8
Graphic image line offset register	GROPSOFST1	R/W	H'FFEC 0310	H'1FEC 0310	32, 16, or 8
Graphic image start position register	GROPDPHV1	R/W	H'FFEC 0314	H'1FEC 0314	32, 16, or 8
Reserved	—	R	H'FFEC 0318	H'1FEC 0318	32, 16, or 8
Reserved	—	R	H'FFEC 031C	H'1FEC 031C	32, 16, or 8
Reserved	—	R	H'FFEC 0320	H'1FEC 0320	32, 16, or 8
Reserved	—	R	H'FFEC 0324	H'1FEC 0324	32, 16, or 8
Reserved	—	R	H'FFEC 0328	H'1FEC 0328	32, 16, or 8
Color register for outside of graphic image area	GROPBASERGB1	R/W	H'FFEC 032C	H'1FEC 032C	32, 16, or 8

Note: \* Use a P4 area address to access a register in the P4 area in the virtual address space. Use an area 7 address to access a register from area 7 in the physical address space through the TLB.

**Table 24.4 Register Configuration in Graphics Block 2**

Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size
Graphics block control register	GRCMEN2	R/W	H'FFED 0000	H'1FED 0000	32, 16, or 8
Bus control register	GRCBUSCNT2	R/W	H'FFED 0004	H'1FED 0004	32, 16, or 8
Reserved	—	R	H'FFED 0008	H'1FED 0008	32, 16, or 8
Reserved	—	R	H'FFED 000C	H'1FED 000C	32, 16, or 8
Reserved	—	R	H'FFED 0300	H'1FED 0300	32, 16, or 8
Reserved	—	R	H'FFED 0304	H'1FED 0304	32, 16, or 8
Graphic image base address register	GROPSADR2	R/W	H'FFED 0308	H'1FED 0308	32, 16, or 8
Graphic image area register	GROPSWH2	R/W	H'FFED 030C	H'1FED 030C	32, 16, or 8
Graphic image line offset register	GROPSOFST2	R/W	H'FFED 0310	H'1FED 0310	32, 16, or 8
Graphic image start position register	GROPDPHV2	R/W	H'FFED 0314	H'1FED 0314	32, 16, or 8
$\alpha$ control area register	GROPEWH2	R/W	H'FFED 0318	H'1FED 0318	32, 16, or 8
$\alpha$ control area start position register	GROPEDPHV2	R/W	H'FFED 031C	H'1FED 031C	32, 16, or 8
$\alpha$ control register	GROPEDPA2	R/W	H'FFED 0320	H'1FED 0320	32, 16, or 8
Chroma-key control register	GROPCRKY0_2	R/W	H'FFED 0324	H'1FED 0324	32, 16, or 8
Chroma-key color register	GROPCRKY1_2	R/W	H'FFED 0328	H'1FED 0328	32, 16, or 8
Color register for outside of graphic image area	GROPBASERGB2	R/W	H'FFED 032C	H'1FED 032C	32, 16, or 8

Note: \* Use a P4 area address to access a register in the P4 area in the virtual address space. Use an area 7 address to access a register from area 7 in the physical address space through the TLB.

**Table 24.5 Register Configuration in Graphics Block 3**

Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size
Graphics block control register	GRCMEN3	R/W	H'FFEE 0000	H'1FEE 0000	32, 16, or 8
Bus control register	GRCBUSCNT3	R/W	H'FFEE 0004	H'1FEE 0004	32, 16, or 8
Reserved	—	R	H'FFEE 0008	H'1FEE 0008	32, 16, or 8
Reserved	—	R	H'FFEE 000C	H'1FEE 000C	32, 16, or 8
Reserved	—	R	H'FFEE 0300	H'1FEE 0300	32, 16, or 8
Reserved	—	R	H'FFEE 0304	H'1FEE 0304	32, 16, or 8
Graphic image base address register	GROPSADR3	R/W	H'FFEE 0308	H'1FEE 0308	32, 16, or 8
Graphic image area register	GROPSWH3	R/W	H'FFEE 030C	H'1FEE 030C	32, 16, or 8
Graphic image line offset register	GROPSOFST3	R/W	H'FFEE 0310	H'1FEE 0310	32, 16, or 8
Graphic image start position register	GROPDPHV3	R/W	H'FFEE 0314	H'1FEE 0314	32, 16, or 8
$\alpha$ control area register	GROPEWH3	R/W	H'FFEE 0318	H'1FEE 0318	32, 16, or 8
$\alpha$ control area start position register	GROPEDPHV3	R/W	H'FFEE 031C	H'1FEE 031C	32, 16, or 8
$\alpha$ control register	GROPEDPA3	R/W	H'FFEE 0320	H'1FEE 0320	32, 16, or 8
Chroma-key control register	GROPCRKY0_3	R/W	H'FFEE 0324	H'1FEE 0324	32, 16, or 8
Chroma-key color register	GROPCRKY1_3	R/W	H'FFEE 0328	H'1FEE 0328	32, 16, or 8
Color register for outside of graphic image area	GROPBASERGB3	R/W	H'FFEE 032C	H'1FEE 032C	32, 16, or 8

Note: \* Use a P4 area address to access a register in the P4 area in the virtual address space. Use an area 7 address to access a register from area 7 in the physical address space through the TLB.



**Table 24.6 Register Configuration in Graphics Block 4**

Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size
Graphics block control register	GRCMEN4	R/W	H'FFEF 0000	H'1FEF 0000	32, 16, or 8
Bus control register	GRCBUSCNT4	R/W	H'FFEF 0004	H'1FEF 0004	32, 16, or 8
Reserved	—	R	H'FFEF 0008	H'1FEF 0008	32, 16, or 8
Reserved	—	R	H'FFEF 000C	H'1FEF 000C	32, 16, or 8
Reserved	—	R	H'FFEF 0300	H'1FEF 0300	32, 16, or 8
Reserved	—	R	H'FFEF 0304	H'1FEF 0304	32, 16, or 8
Graphic image base address register	GROPSADR4	R/W	H'FFEF 0308	H'1FEF 0308	32, 16, or 8
Graphic image area register	GROPSWH4	R/W	H'FFEF 030C	H'1FEF 030C	32, 16, or 8
Graphic image line offset register	GROPSOFST4	R/W	H'FFEF 0310	H'1FEF 0310	32, 16, or 8
Graphic image start position register	GROPDPHV4	R/W	H'FFEF 0314	H'1FEF 0314	32, 16, or 8
$\alpha$ control area register	GROPEWH4	R/W	H'FFEF 0318	H'1FEF 0318	32, 16, or 8
$\alpha$ control area start position register	GROPEDPHV4	R/W	H'FFEF 031C	H'1FEF 031C	32, 16, or 8
$\alpha$ control register	GROPEDPA4	R/W	H'FFEF 0320	H'1FEF 0320	32, 16, or 8
Chroma-key control register	GROPCRKY0_4	R/W	H'FFEF 0324	H'1FEF 0324	32, 16, or 8
Chroma-key color register	GROPCRKY1_4	R/W	H'FFEF 0328	H'1FEF 0328	32, 16, or 8
Color register for outside of graphic image area	GROPBASERGB4	R/W	H'FFEF 032C	H'1FEF 032C	32, 16, or 8

Note: \* Use a P4 area address to access a register in the P4 area in the virtual address space. Use an area 7 address to access a register from area 7 in the physical address space through the TLB.

**Table 24.7 Register Configuration in Display Control Block**

Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size
SG mode register	SGMODE	R/W	H'FFEB 0000	H'1FEB 0000	32, 16, or 8
Interrupt output control register	SGINTCNT	R/W	H'FFEB 0004	H'1FEB 0004	32, 16, or 8
Sync signal control register	SYNCNT	R/W	H'FFEB 0008	H'1FEB 0008	32, 16, or 8
External sync signal input timing control register	EXTSYNCNT	R/W	H'FFEB 000C	H'1FEB 000C	32, 16, or 8
Reserved	—	R	H'FFEB 0100	H'1FEB 0100	32, 16, or 8
Sync signal size register	SYNSIZE	R/W	H'FFEB 0104	H'1FEB 0104	32, 16, or 8
Vertical sync signal timing control register	VSYNCTIM	R/W	H'FFEB 0108	H'1FEB 0108	32, 16, or 8
Horizontal sync signal timing control register	HSYNCTIM	R/W	H'FFEB 010C	H'1FEB 010C	32, 16, or 8
Gate clock signal timing control register	CLSTIM	R/W	H'FFEB 0110	H'1FEB 0110	32, 16, or 8
Sampling start signal timing control register	SPLTIM	R/W	H'FFEB 0118	H'1FEB 0118	32, 16, or 8
Gate control signal timing control register	COMTIM	R/W	H'FFEB 011C	H'1FEB 011C	32, 16, or 8
SGDE area start position register	SGDESTART	R/W	H'FFEB 0120	H'1FEB 0120	32, 16, or 8
SGDE area size register	SGDESIZE	R/W	H'FFEB 0124	H'1FEB 0124	32, 16, or 8
CDE chroma-key color register	CDECRKY	R/W	H'FFEB 0128	H'1FEB 0128	32, 16, or 8
Reserved	—	R	H'FFEB 0148	H'1FEB 0148	32, 16, or 8
T-1004 control register	T1004CNT	R/W	H'FFEB 0200	H'1FEB 0200	32, 16, or 8
T-1004 video start position register	T1004OFFSET	R/W	H'FFEB 0204	H'1FEB 0204	32, 16, or 8
Reserved	—	R	H'FFEB 0208	H'1FEB 0208	32, 16, or 8
Reserved	—	R	H'FFEB 020C	H'1FEB 020C	32, 16, or 8

Note: \* Use a P4 area address to access a register in the P4 area in the virtual address space. Use an area 7 address to access a register from area 7 in the physical address space through the TLB.

## 24.6.1 Graphics Block Control Registers (GRCMEN1 to GRCMEN4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEN	VEN
Initial value:	0	0	0	0	0	0	0	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	WE	0	R/W	Enables register value transfer. Writing 1 to this bit transfers the register values (registers at H'000 to H'31C and H'32C) in synchronization with Vsync. After register transfer is completed, this bit is cleared to 0.
30 to 2	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
1	DEN	0	R/W	Enables graphics display. 0: Disables display operation 1: Enables display operation
0	VEN	0	R/W	Enables lower-layer graphics display. 0: Disables display operation 1: Enables display operation

**Table 24.8 Functions of Display Enable Bits**

DEN	VEN	Operation	Output	Control
0	0	Does not read image data from memory or process lower-layer graphics display.	Outputs the color specified in GROPBASERGB over the entire screen (negates the enable signal output).	
0	1	Does not read image data from memory but processes lower-layer graphics display.	Outputs only the lower-layer graphics (outputs the lower-layer graphics enable signal)	Displays only the lower-layer graphics.
1	0	Reads image data from memory but does not process lower-layer graphics display.	Outputs only the current graphics (outputs the current graphics enable signal).	Displays only the current graphics.
1	1	Reads image data from memory and processes lower-layer graphics display.	Performs the specified processing for the current graphics and lower-layer graphics and displays them (logically ORs the current graphics and lower-layer graphics enable signals and outputs the result).	Displays the current and lower-layer graphics.

- Notes: 1. When the  $\alpha$  control area (specified by GROPEW and GROPEDPHV) is larger than the graphic image area (specified by GROPSWH and GRODPHV), only the lower-layer graphic images are displayed.
2. These bits should be set for each layer. When (DEN, VEN) = (0, 0) in the upper layer, the graphics in the lower layers are not output.

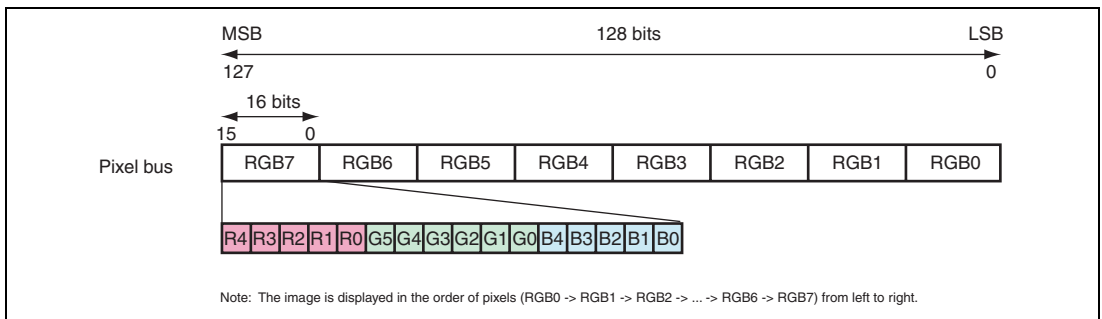
## 24.6.2 Bus Control Registers (GRCBUSCNT1 to GRCBUSCNT4)

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ENDIAN
Initial value:	0	0	0	0	0	0	0	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	ENDIAN	0	R/W	Specifies the endian for the pixel bus. 0: Little endian 1: Big endian



**Figure 24.7 Pixel Bus Endian (ENDIAN = 0)**

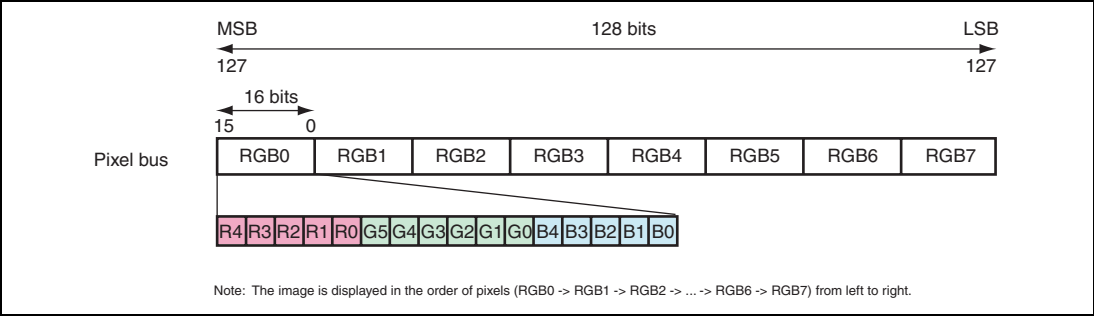


Figure 24.8 Pixel Bus Endian (ENDIAN = 1)

24.6.3 Graphic Image Base Address Registers (GROPSADR1 to GROPSADR4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	GROPSADR[28:16]												
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/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
	GROPSADR[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 29	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.
28 to 0	GROPSADR [28:0]	H'0000000	R/W	These bits specify the address from which a graphic image is to be read. The lowest bit should always be 0.

Note: The VDC2 processes 16-bit RGB data; it cannot handle data located beyond a 2-byte alignment boundary.

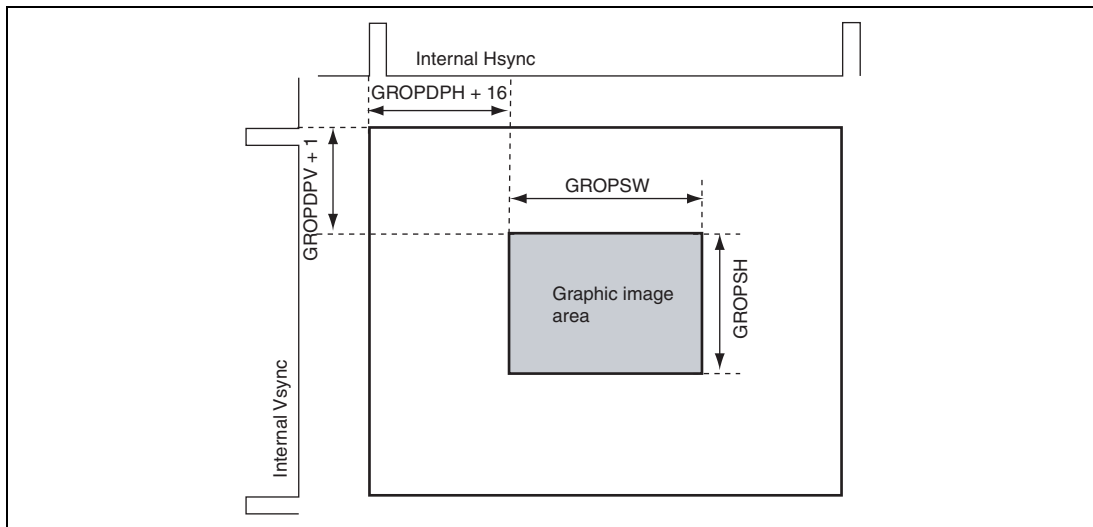
## 24.6.4 Graphic Image Area Registers (GROPSWH1 to GROPSWH4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GROPSH[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	GROPSH [9:0]	H'000	R/W	These bits specify the height of the graphic image area in number of lines.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GROPSW [9:0]	H'000	R/W	These bits specify the width of the graphic image area in number of panel clock cycles.



**Figure 24.9 Graphic Image Area Settings (Reading from Memory)**

A graphic image area should be specified within the following range; otherwise, correct operation is not guaranteed.

(Panel clock cycles for 1H) > GROPSW (width) + GROPDPH (horizontal display start position) + (16 panel clock cycles)

(Lines for 1 frame) > GROPSH (height) + GROPDPV (vertical display start position) + (1 line)



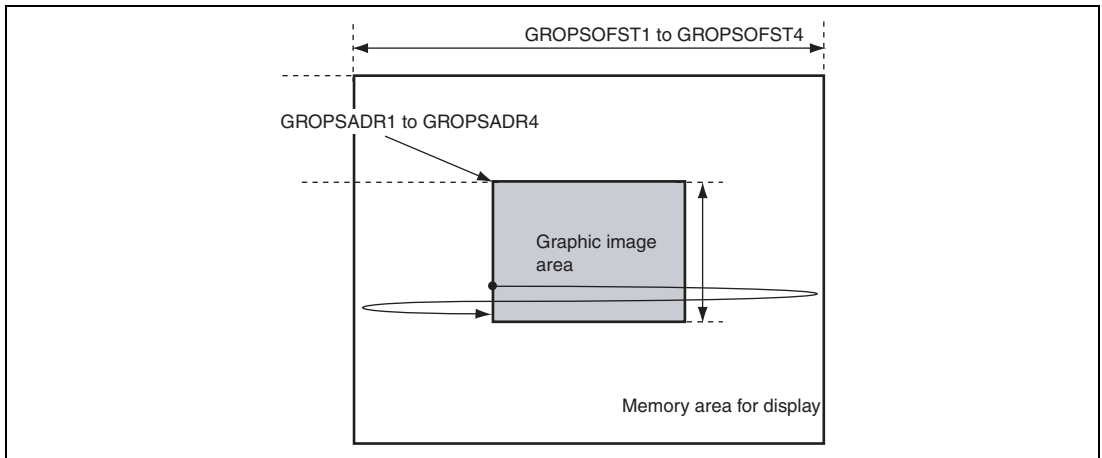
### 24.6.5 Graphic Image Line Offset Registers (GROPSOFST1 to GROPSOFST4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	GROPSOFST[28:16]												
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/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
	GROPSOFST[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 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 0	GROPSOFST [28:0]	H'0000000	R/W	These bits specify the line offset for the graphic image. The lower four bits should always be 0000.



**Figure 24.10 Graphic Image Memory Area Settings**

The start (left side) address of line  $n$  is obtained by adding the base address register value (GROPSADR1 to GROPSADR4) and the line offset (GROPSOFST1 to GROPSOFST4)  $\times n$ .

## 24.6.6 Graphic Image Start Position Registers (GROPDPHV1 to GROPDPHV4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GROPDPV[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	GROPDPV [9:0]	H'000	R/W	These bits specify the vertical display start position of the graphic image area in number of lines.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GROPDPH [9:0]	H'000	R/W	These bits specify the horizontal display start position of the graphic image area in number of panel clock cycles.

Note: The display start address is offset as follows (see figure 24.9).

Vertical offset: (GROPDPV value) + 1 line

Horizontal offset: (GROPDPH value) + 16 panel clock cycles

## 24.6.7 α Control Area Registers (GROPEWH2 to GROPEWH4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GROPEH[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	GROPEH [9:0]	H'000	R/W	These bits specify the height of the $\alpha$ control area in number of lines.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GROPEW [9:0]	H'000	R/W	These bits specify the width of the $\alpha$ control area in number of panel clock cycles.

Note: Layer 1 is the bottom image which has no  $\alpha$  control target, so the above settings are prohibited for layer 1.

Each register specifies the size of the  $\alpha$  control area (rectangle). See figure 24.11.

#### 24.6.8 $\alpha$ Control Area Start Position Registers (GROPEDPHV2 to GROPEDPHV4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GROPEDPV[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	GROPEDPV [9:0]	H'000	R/W	These bits specify the vertical start position of the $\alpha$ control area in number of lines.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GROPEDPH [9:0]	H'000	R/W	These bits specify the horizontal start position of the $\alpha$ control area in number of panel clock cycles.

Note: Layer 1 is the bottom image which has no  $\alpha$  control target, so the above settings are prohibited for layer 1.

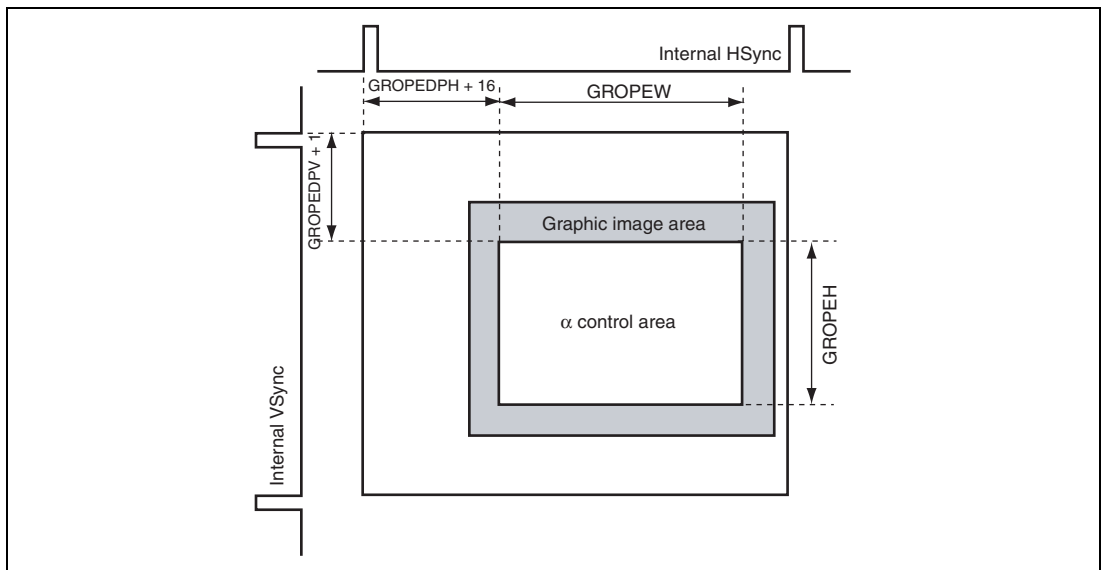


Figure 24.11  $\alpha$  Control Area Settings

## 24.6.9 $\alpha$ Control Registers (GROPEDPA2 to GROPEDPA4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEFA[7:0]								ACOE[7:0]							
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARATE[7:0]								WE	—	—	AST	—	AMOD[1:0]		AEN
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/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DEFA[7:0]	H'FF	R/W	These bits specify the initial $\alpha$ value.
23 to 16	ACOE[7:0]	H'00	R/W	These bits specify a coefficient for $\alpha$ value calculation. This value is added to or subtracted from the DEFA value.
15 to 8	ARATE[7:0]	H'00	R/W	These bits specify the frame rate of $\alpha$ control. (480p Vsync is used as the unit of counting.)
7	WE	0	R/W	Enables transfer of the $\alpha$ control register values. Writing 1 to this bit transfers the register values (registers at H'320 to H'328) in synchronization with Vsync. After register transfer is completed, this bit is cleared to 0.  0: Disables transfer 1: Enables transfer
6, 5	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
4	AST	0	R	$\alpha$ blending status flag.  0: Addition or subtraction has been completed 1: Addition or subtraction is in progress
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, 1	AMOD[1:0]	00	R/W	These bits specify the $\alpha$ processing mode. 00: Initial $\alpha$ value (does not change the value) 01: $\alpha$ value addition 10: $\alpha$ value subtraction 11: Setting prohibited
0	AEN	0	R/W	Enables or disables $\alpha$ control. 0: Disables $\alpha$ control (same as $\alpha$ value = 1) 1: enables $\alpha$ control

Note: Layer 1 is the bottom image which has no  $\alpha$  control target, so the above settings are prohibited for layer 1.

When AEN = 1 and WE = 1, the  $\alpha$  value is loaded in the internal circuits in synchronization with Vsync.

If AMOD[1:0] = [0 0], the  $\alpha$  value specified in DEFA is applied to the video area.

If AMOD[1:0] = [0 1], the ACOEF value is added to the DEFA value according to the field rate and the result is applied to the video area as the  $\alpha$  value. When the  $\alpha$  value becomes 255 or larger, processing stops (fade-out).

If AMOD[1:0] = [1 0], the ACOEF value is subtracted from the DEFA value according to the field rate and the result is applied to the video area as the  $\alpha$  value. When the  $\alpha$  value becomes 0 or smaller, processing stops (fade-in).

**Table 24.9  $\alpha$  Value and Blending Ratio**

$\alpha$ Value (Decimal)	Graphics	Lower-Layer Graphics
255	256/256	0/256
254	254/256	1/256
253	253/256	2/256
252	252/256	3/256
	:	:
	2/256	253/256
1	1/256	254/256
0	0/256	256/256

### 24.6.10 Chroma-Key Control Registers (GROPCRKY0\_2 to GROPCRKY0\_4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CKEN
Initial value:	0	0	0	0	0	0	0	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
	CROMAKR[4:0]					CROMAKG[5:0]					CROMARKB[4:0]					
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 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	CKEN	0	R/W	Enables or disables chroma-key processing. 0: Disables chroma-key processing 1: Enables chroma-key processing
15 to 11	CROMAKR [4:0]	00000	R/W	These bits specify chroma-key target color R.
10 to 5	CROMAKG [5:0]	000000	R/W	These bits specify chroma-key target color G.
4 to 0	CROMAKB [4:0]	00000	R/W	These bits specify chroma-key target color B.

Note: Layer 1 is the bottom image which has no  $\alpha$  control target, so the above settings are prohibited for layer 1.

When WE =1 in GROPEDPA, the register setting is loaded in the internal circuits in synchronization with Vsync.

While the chroma-key processing is enabled, if the graphics data values (RGB16 format) of a pixel all match the CROMAKR[4:0], CROMAKG[5:0], and CROMAKB[4:0] settings, the pixel color is replaced with the color (RGB16 format) specified in the chroma-key color register (GROPCRKY1) and  $\alpha$  processing specified through the ALPHA[7:0] bits is applied.

Chroma-keying thus enables characters or a cursor to be displayed on lower-layer graphics.

## 24.6.11 Chroma-Key Color Registers (GROPCRKY1\_2 to GROPCRKY1\_4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ALPHA[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R[4:0]					G[5:0]					B[4:0]					
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	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	ALPHA[7:0]	H'00	R/W	These bits specify the $\alpha$ value after replacement.
15 to 11	R[4:0]	00000	R/W	These bits specify the R value after replacement.
10 to 5	G[5:0]	000000	R/W	These bits specify the G value after replacement.
4 to 0	B[4:0]	00000	R/W	These bits specify the B value after replacement.

Note: Layer 1 is the bottom image which has no  $\alpha$  control target, so the above settings are prohibited for layer 1.

Each register specifies a set of color information to replace the color that matches the chroma-key target RGB values.

$\alpha$  calculation is done as follows.

Output R = R (current graphic image)  $\times \alpha$  + R (lower-layer graphic image)  $\times (1 - \alpha)$

Output G = G (current graphic image)  $\times \alpha$  + G (lower-layer graphic image)  $\times (1 - \alpha)$

Output B = B (current graphic image)  $\times \alpha$  + B (lower-layer graphic image)  $\times (1 - \alpha)$



## 24.6.12 Color Registers for Outside of Graphic Image Area (GROPBASERGB1 to GROPBASERGB4)

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
	BASE_R[4:0]					BASE_G[5:0]					BASE_B[4:0]					
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 11	BASE_R [4:0]	00000	R/W	These bits specify the R value for outside of the graphic image area.
10 to 5	BASE_G [5:0]	000000	R/W	These bits specify the G value for outside of the graphic image area.
4 to 0	BASE_B [4:0]	00000	R/W	These bits specify the B value for outside of the graphic image area.

Note: This setting is valid only when VEN = 0 in GRCMEN.

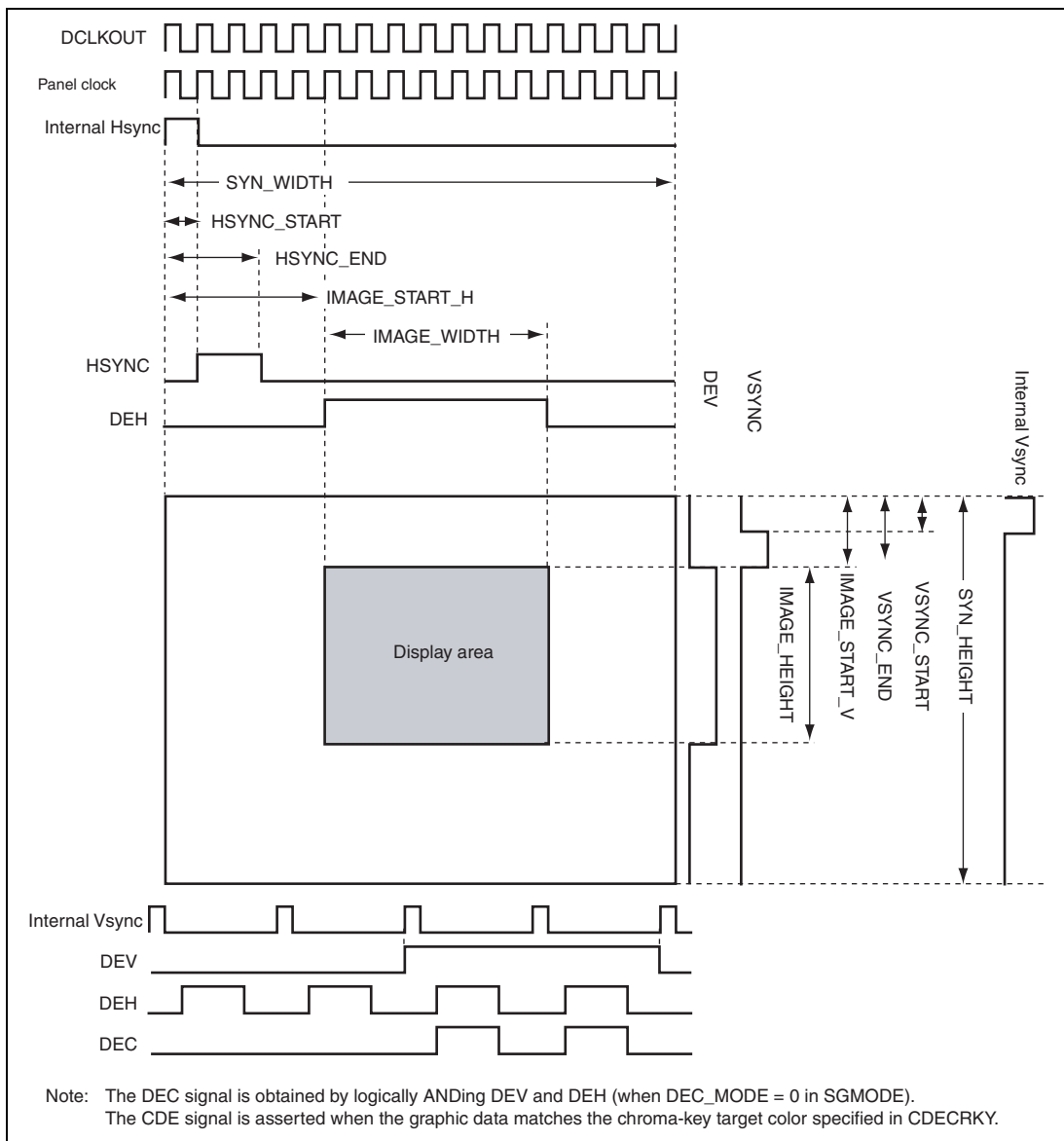


Figure 24.12 Screen Format

### 24.6.13 SG Mode Register (SGMODE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
	—	—	—	—	—	—	COM_CDE_SEL	CDE_EXE	—	—	DE_SEL	DEC_MODE	—	—	SYNC_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/W	R/W	R	R	R/W	R/W	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31	WE	0	R/W	Enables register value transfer* <sup>1</sup> . Writing 1 to this bit transfers the register values (registers at H'FFEB_0000 to H'FFEB_0208).
30 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	COM_CDE_SEL	0	R/W	Selects the COM or CDE signal output. 0: Outputs COM 1: Outputs CDE
8	CDE_EXE	0	R/W	Enables CDE operation. This setting becomes effective in synchronization with the internal Vsync timing. 0: Disables CDE operation (0 is always output through CDE when the COM_TYPE bit is 0 in SYNCNT) 1: Enables CDE operation
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	DE_SEL	0	R/W	Selects the DEH or DEC signal output. 0: Outputs DEH (horizontal data enable) 1: Outputs DEC (horizontal and vertical composite data enable)

Bit	Bit Name	Initial Value	R/W	Description
4	DEC_MODE	0	R/W	<p>Selects the enable mode.</p> <p>0: Outputs the data enable signal selected through the SGDESTART and SGDESIZE settings</p> <p>1: Outputs the data enable signal generated in the graphics blocks (composite signal obtained by logically ORing the data enable signals of the layers)*<sup>2</sup></p>
3, 2	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
1	SYNC_SEL	0	R/W	<p>Selects the sync signals.</p> <p>0: Uses the internal sync signals</p> <p>1: Uses the external sync signals (delayed by five panel clock cycles in the VDC2)</p>
0	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>

Notes: 1. Clear the WE bit to 0 before modifying the values of the registers located at H'000 to H'208; set the WE bit to 1 after modifying the registers.

2. When setting the DEC\_MODE bit to 1, also set the DE\_SEL bit to 1.

## 24.6.14 Interrupt Output Control Register (SGINTCNT)

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
	—	—	—	—	—	—	—	—	—	—	—	VSYNC_MASK	—	—	—	VSYNC_STATUS
Initial value:	0	0	0	0	0	0	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/WC0

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	VSYNC_MASK	0	R/W	Masks the VSYNC interrupt* <sup>1</sup> . 0: Enables interrupts 1: Masks interrupts
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	VSYNC_STATUS	1	R/WC0	Indicates the VSYNC interrupt status* <sup>2</sup> . 0: An interrupt has occurred 1: No interrupts have occurred

Notes: 1. Writing 1 to the interrupt mask bit clears the interrupt status.  
2. Writing 0 to the interrupt status bit clears the interrupt status.

The VSYNC\_STATUS value is output through the db\_n\_int\_n signal terminal (low-active signal) of the VDC block.

## 24.6.15 Sync Signal Control Register (SYNCNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	RGB_TIM	—	—	EX_V_TIM	EX_H_TIM	—	—	VSYNC_TIM	HSYNC_TIM	DEV_TIM	DEH_TIM	DEC_TIM	COM_TIM
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/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
	—	—	—	—	—	—	EX_V_TYPE	EX_H_TYPE	—	—	VSYNC_TYPE	HSYNC_TYPE	DEV_TYPE	DEH_TYPE	DEC_TYPE	COM_TYPE
Initial value:	0	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/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	RGB_TIM	0	R/W	Specifies the RGB data output timing. 0: Outputs data at the rising edge of the panel clock 1: Outputs data at the falling edge of the panel clock
27, 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	EX_V_TIM	0	R/W	Specifies the external VSYNC input timing. 0: Latches the external VSYNC at the rising edge of the panel clock 1: Latches the external VSYNC at the falling edge of the panel clock
24	EX_H_TIM	0	R/W	Specifies the external HSYNC input timing. 0: Latches the external HSYNC at the rising edge of the panel clock 1: Latches the external HSYNC at the falling edge of the panel clock
23 to 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	VSYNC_TIM	0	R/W	<p>Specifies the VSYNC/SPS output timing.</p> <p>0: Outputs VSYNC/SPS at the rising edge of the panel clock</p> <p>1: Outputs VSYNC/SPS at the falling edge of the panel clock</p>
20	HSYNC_TIM	0	R/W	<p>Specifies the HSYNC/SPL output timing.</p> <p>0: Outputs HSYNC/SPL at the rising edge of the panel clock</p> <p>1: Outputs HSYNC/SPL at the falling edge of the panel clock</p>
19	DEV_TIM	0	R/W	<p>Specifies the DEV/CLS output timing.</p> <p>0: Outputs DEV/CLS at the rising edge of the panel clock</p> <p>1: Outputs DEV/CLS at the falling edge of the panel clock</p>
18	DEH_TIM	0	R/W	<p>Specifies the DEH/LP output timing.</p> <p>0: Outputs DEH/LP at the rising edge of the panel clock</p> <p>1: Outputs DEH/LP at the falling edge of the panel clock</p>
17	DEC_TIM	0	R/W	<p>Specifies the DEC/PS output timing.</p> <p>0: Outputs DEC/PS at the rising edge of the panel clock</p> <p>1: Outputs DEC/PS at the falling edge of the panel clock</p>
16	COM_TIM	0	R/W	<p>Specifies the COM/CDE output timing.</p> <p>0: Outputs COM/CDE at the rising edge of the panel clock</p> <p>1: Outputs COM/CDE at the falling edge of the panel clock</p>
15 to 10	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
9	EX_V_TYPE	0	R/W	Controls whether to invert the external VSYNC input. 0: Does not invert the external VSYNC input 1: Inverts the external VSYNC input
8	EX_H_TYPE	0	R/W	Controls whether to invert the external HSYNC input. 0: Does not invert the external HSYNC input 1: Inverts the external HSYNC input
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	VSYNC_TYPE	0	R/W	Controls whether to invert the VSYNC/SPS output. 0: Does not invert the VSYNC/SPS output. 1: Inverts the VSYNC/SPS output.
4	HSYNC_TYPE	0	R/W	Controls whether to invert the HSYNC/SPL output. 0: Does not invert the HSYNC/SPL output 1: Inverts the HSYNC/SPL output
3	DEV_TYPE	0	R/W	Controls whether to invert the DEV/CLS output. 0: Does not invert the DEV/CLS output 1: Inverts the DEV/CLS output
2	DEH_TYPE	0	R/W	Controls whether to invert the DEH/LP output. 0: Does not invert the DEH/LP output 1: Inverts the DEH/LP output
1	DEC_TYPE	0	R/W	Controls whether to invert the DEC/PS output. 0: Does not invert the DEC/PS output 1: Inverts the DEC/PS output
0	COM_TYPE	0	R/W	Controls whether to invert the COM/CDE output. 0: Does not invert the COM/CDE output 1: Inverts the COM/CDE output



## 24.6.16 External Sync Signal Input Timing Control Register (EXTSYNCNT)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	EX_STATUS	0	R	Indicates the status of the external VSYNC and HSYNC phases. 0: VSYNC and HSYNC are in phase 1: VSYNC and HSYNC are out of phase
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	EX_V_DLY [1:0]	H'0	R/W	These bits delay the external VSYNC input (dot clock cycles). 00: No delay 01: Delays by one dot clock cycle 10: Delays by two dot clock cycles 11: Delays by three dot clock cycles
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	EX_H_DLY [1:0]	H'0	R/W	These bits delay the external HSYNC input (dot clock cycles). 00: No delay 01: Delays by one dot clock cycle 10: Delays by two dot clock cycles 11: Delays by three dot clock cycles

The VDC2 assumes that the external sync signals are input with the horizontal and vertical sync signal timing for the LCD panel conforming to the VESA standard. This register adjusts the phases of the external input sync signals when they are sampled in the VDC2.

### 24.6.17 Sync Signal Size Register (SYNSIZE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SYN_HEIGHT[9:0]									
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1
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
	—	—	—	—	—	SYN_WIDTH[10:0]										
Initial value:	0	0	0	0	0	0	1	1	0	1	0	1	1	0	1	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
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	SYN_HEIGHT [9:0]	H'20D	R/W	These bits specify the height including the vertical blanking interval in number of lines. Initial value: H'20D = 525 lines
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SYN_WIDTH [10:0]	H'35A	R/W	These bits specify the width including the horizontal blanking interval in number of panel clock cycles. Initial value: H'35A = 858 dots

## 24.6.18 Vertical Sync Signal Timing Control Register (VSYNC\_TIM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	VSYNC_START[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	VSYNC_END[9:0]									
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/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 always read as 0. The write value should always be 0.
25 to 16	VSYNC_START [9:0]	H'000	R/W	These bits specify in number of lines the interval between the internal vertical sync signal and the point where the vertical sync signal (VSYNC) for the screen is set to 1.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	VSYNC_END [9:0]	H'001	R/W	These bits specify in number of lines the interval between the internal vertical sync signal and the point where the vertical sync signal (VSYNC) for the screen is cleared to 0.

Note: Be sure to satisfy  $VSYNC\_START \neq VSYNC\_END$ ; otherwise, correct operation is not guaranteed.

## 24.6.19 Horizontal Sync Signal Timing Control Register (HSYNTIM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	HSYNC_START[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	HSYNC_END[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	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
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	HSYNC_START [10:0]	H'000	R/W	These bits should always be set to H'000. These bits specify in number of panel clock cycles the interval between the internal horizontal sync signal and the point where the horizontal sync signal (HSYNC) for the screen is set to 1.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	HSYNC_END [10:0]	H'00A	R/W	These bits specify in number of panel clock cycles the interval between the internal horizontal sync signal and the point where the horizontal sync signal (HSYNC) for the screen is cleared to 0.

Note: Be sure to satisfy  $HSYNC\_START \neq HSYNC\_END$ ; otherwise, correct operation is not guaranteed.

## 24.6.20 Gate Clock Signal Timing Control Register (CLSTIM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	CLS_START[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CLS_END[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	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 to 16	CLS_START [10:0]	H'000	R/W	These bits specify in number of panel clock cycles the interval between the internal horizontal sync signal and the point where the gate clock signal (CLS) is set to 1.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	CLS_END [10:0]	H'000	R/W	These bits specify in number of panel clock cycles the interval between the internal horizontal sync signal and the point where the gate clock signal (CLS) is cleared to 0.

Note: Be sure to satisfy  $CLS\_START \neq CLS\_END$ ; otherwise, correct operation is not guaranteed.

## 24.6.21 Sampling Start Signal Timing Control Register (SPLTIM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SPL_START[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SPL_END[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	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 to 16	SPL_START [10:0]	H'000	R/W	These bits specify in number of panel clock cycles the interval between the internal horizontal sync signal and the point where the sampling start signal (SPL) is set to 1.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SPL_END [10:0]	H'000	R/W	These bits specify in number of panel clock cycles the interval between the internal horizontal sync signal and the point where the sampling start signal (SPL) is cleared to 0.

Note: Be sure to satisfy  $SPL\_START \neq SPL\_END$ ; otherwise, correct operation is not guaranteed.

## 24.6.22 Gate Control Signal Timing Control Register (COMTIM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COM_MODE	—	—	—	—	—	COMTIM_V[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	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
	—	—	—	—	—	COMTIM_H[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	R/W	Description
31	COM_MODE	0	R/W	<p>Selects the gate control signal (COM) toggle mode.</p> <p>0: Toggles the signal output in every line in an alternating sequence of high and low and inverts the phase in every frame (when the sequence in frame n is high -&gt; low -&gt; high ..., it is inverted to low -&gt; high -&gt; low ... in frame n + 1).</p> <p>1: Toggles the signal output in every frame.</p>
30 to 26	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
25 to 16	COMTIM_V [9:0]	H'000	R/W	<p>These bits specify in number of lines the interval between the internal vertical sync signal and the frame start position of the gate control signal (COM). A value of 0 specifies that a frame starts in the first line, and a value of 1 specifies that a frame starts in the second line.</p>
15 to 11	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
10 to 0	COMTIM_H [10:0]	H'000	R/W	<p>These bits specify in number of panel clock cycles the horizontal interval between the internal horizontal sync signal and the position where the gate control signal (COM) toggles.</p>

**Note:** Be sure to satisfy  $COMTIM\_V < SYN\_HEIGHT$ ; otherwise, correct operation is not guaranteed.

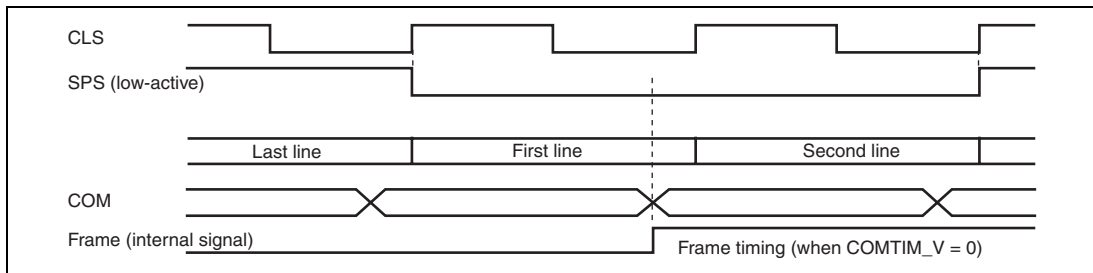


Figure 24.13 COM Signal Timing

### 24.6.23 SGDE Area Start Position Register (SGDESTART)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SGDE_START_V[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

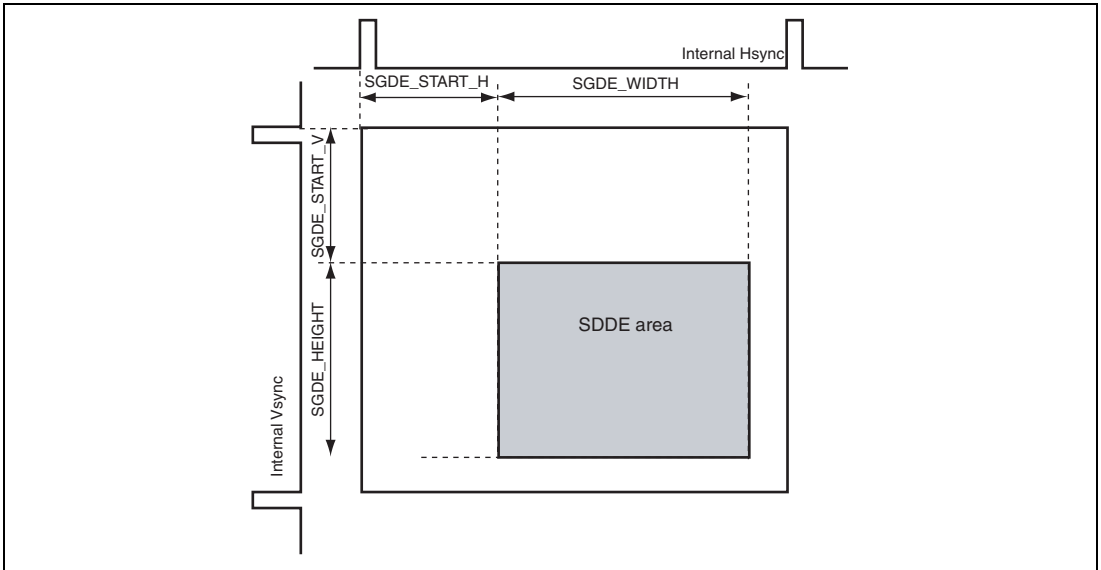
  

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SGDE_START_H[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	SGDE_START_V[9:0]	H'000	R/W	These bits specify in number of lines the vertical interval between the internal vertical sync signal and the start of the data enable (DE) signal output. Setting to 0 is prohibited.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SGDE_START_H[10:0]	H'000	R/W	These bits specify in number of panel clock cycles the horizontal interval between the internal horizontal sync signal and the start of the data enable (DE) signal output.

- Notes:
1. Be sure to satisfy  $SYN\_HEIGHT > SGDE\_HEIGHT + SGDE\_START\_V$ ; otherwise, correct operation is not guaranteed.
  2. Be sure to satisfy  $SYN\_WIDTH > SGDE\_WIDTH + SGDE\_START\_H$ ; otherwise, correct operation is not guaranteed.





**Figure 24.14 Settings of DE Area Generated in SG Block**

**24.6.24 SGDE Area Size Register (SGDESIZE)**

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SGDE_HEIGHT[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SGDE_WIDTH[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	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	SGDE_HEIGHT [9:0]	H'000	R/W	These bits specify the vertical length (height) of the data enable (DE) signal area in number of lines.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SGDE_WIDTH [10:0]	H'000	R/W	These bits specify the horizontal length (width) of the data enable (DE) signal area in number of panel clock cycles.

## 24.6.25 CDE Chroma-Key Color Register (CDECRKY)

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
	CDE_R[4:0]					CDE_G[5:0]					CDE_B[4:0]					
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 11	CDE_R[4:0]	00000	R/W	These bits specify the R value as the target of chroma-keying to output the CDE signal.
10 to 5	CDE_G[5:0]	000000	R/W	These bits specify the G value as the target of chroma-keying to output the CDE signal.
4 to 0	CDE_B[4:0]	00000	R/W	These bits specify the B value as the target of chroma-keying to output the CDE signal.

**Note:** After the overlay processing (layer 1 + layer 2 + layer 3 + layer 4) is applied to a graphic image, the resultant image data is compared with the above specified color and the VDC2 outputs the CDE signal when they match (chroma-keying is not applied for each layer).

**24.6.26 T1004 Control Register (T1004CNT)**

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	VSYNC_ _TYPE	HSYNC_ _TYPE	DEC_ _TYPE
Initial value:	0	0	0	0	0	0	0	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	VSYNC_ TYPE	0	R/W	Selects the polarity of VSYNC in T-1004 format. 0: Positive 1: Negative
1	HSYNC_ TYPE	0	R/W	Selects the polarity of HSYNC in T-1004 format. 0: Positive 1: Negative
0	DEC_TYPE	0	R/W	Selects the polarity of DEC (data enable) in T-1004 format. 0: Positive 1: Negative

### 24.6.27 T1004 Video Start Position Register (T1004OFFSET)

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

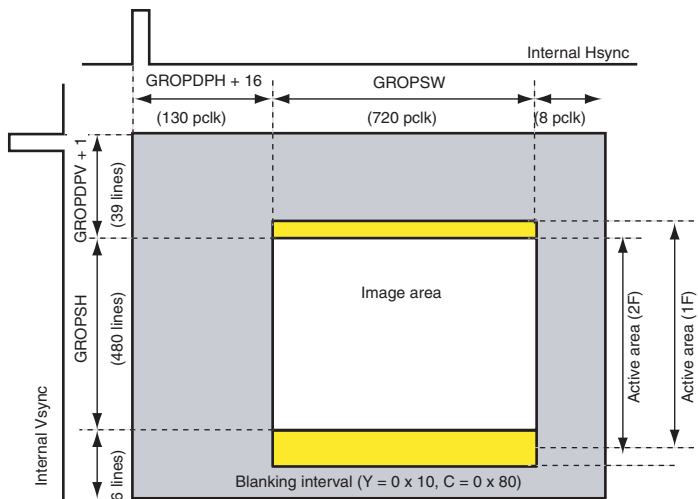
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	T1004OFFSET_H[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	T1004OFFSET_H[10:0]	0	R/W	These bits adjust the horizontal phase of the video signal and blanking interval in 2-pixel units. Specifying a larger value shifts the video display position left. The lowest two bits (bits 1 and 0) should always be 0.

Table 24.10 shows an example of register settings to display the video at the top-left corner of the active area.

**Table 24.10 Example of Register Settings for T-1004 Output**

	Register	Setting	Description
Graphics blocks	GROPDPHV1 to GROPDPHV4	H'0026_0072	Starts video output from line 40 with respect to the internal Vsync and panel clock cycle 131 with respect to the internal Hsync.
Display control block	SYNSIZE	H'020D_35A	Specifies 525 lines for the vertical sync signal period and 858 panel clock cycles for the horizontal sync signal.
	T1004OFFSET	H'0000_0010	Adjusts the horizontal phase of the video signal and blanking interval. Increasing the register value by H'4 shifts the display position left by two pixels; decreasing the value by H'4 shifts the display position right by two pixels.



**Figure 24.15 T-1004 Video Output Position**

## 24.7 Operating Procedures

### 24.7.1 Display Control Block

1. Disabling register value transfer  
Clear the WE bit to 0 in the SG mode register.
2. Setting the registers in the display control block  
Make appropriate settings in the registers shown in table 24.9.  
Specify the polarity of the external pins first.
3. Enabling register value transfer  
Set the WE bit to 1 in the SG mode register.

### 24.7.2 Graphics Blocks

1. Disabling register value transfer  
Clear the WE bit to 0 in the graphics block control registers.  
Clear the WE bit to 0 in the  $\alpha$  control registers.
2. Setting the registers in the graphics blocks  
Make appropriate settings in the registers shown in tables 24.5 through 24.8.
3. Enabling register value transfer  
Set the WE bit to 1 in the graphics block control registers.  
Set the WE bit to 1 in the  $\alpha$  control registers.  
The display operation specified in the registers starts from the next frame.





## Section 25 NAND Flash Memory Controller (FLCTL)

The NAND flash memory controller (FLCTL) provides interfaces for an external NAND-type flash memory. To take measures for errors specific to flash memory, the FLCTL supports the ECC-code generation function and error detection function.

Note: The flash memory using Multi Level Cell (MLC) technology is not supported by this LSI.

### 25.1 Features

#### NAND-Type Flash Memory Interface:

- Interface directly connectable to NAND-type flash memory
- Read or write in sector units (512 + 16 bytes) and ECC processing executed  
An access unit of 2048 + 64 bytes, referred to as a page, is used in some datasheets for NAND-type flash memory. In this manual, an access unit of 512 + 16 bytes, referred to as a sector, is always used.
- Read or write in byte units

**Access Modes:** The FLCTL can select one of the following two access modes.

- Command access mode: Performs an access by specifying a command to be issued from the FLCTL to flash memory, address, and data size to be input or output. Read, write, or erasure of data without ECC processing can be achieved.
- Sector access mode: Performs a read or write in physical sector units by specifying a physical sector and controls ECC-code generation and check. By specifying the number of sectors, the continuous physical sectors can be read or written.

#### Sectors and Control Codes:

- A sector is comprised of 512-byte data and 16-byte control code. The 16-byte control code includes 8-byte ECC.
- The position of the ECC in the control code can be specified in 4-byte units.
- User information can be written to the control code other than the ECC.

**ECC:**

- 8-byte ECC code is generated and error check is performed for a sector (512-byte data + 16-byte control code). (Note that the ECC code generation in the 16-byte control code and the number of bytes to be checked differ depending on the specifications.)
- Error correction capability is up to three errors.
- In a write operation, an ECC code is generated for data and control code prior to the ECC. The control code following the ECC is not considered.
- In a read operation, an ECC error is checked for data and control code prior to the ECC. An ECC on the control code in the FIFO is replaced with the check result by the ECC circuit, not an ECC code read from flash memory.
- An error correction is not performed even when an ECC error occurs. Error corrections must be performed by software.

**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 a read error occurs, an ECC in the control code is other than 0. This read error is reflected on the ECC error source flag.
- When an ECC error occurs, perform an error correction, specify another sector to be replaced, and copy the contents of the block to another sector as required.

**Data Transfer FIFO and Data Register:**

- The 224-byte FLDTFIFO is incorporated for data transfer of flash memory.
- The 32-byte FLECFIFO is incorporated for data transfer of control code.
- The overrun/underrun detection flag is provided for the access from the CPU and DMA.

**DMA Transfer:**

- By individually specifying the destinations of data and control code of flash memory to the DMA controller, data and control code can be sent to different areas.

**Access Size:**

- Registers can be accessed in 32 bits or 8 bits. Registers must be accessed in the specified access size.

**Access Time:**

- The operating frequency of the FLCTL pins can be specified by the FCKSEL bit and the QTSEL bit in the common control register (FLCMNCR), regardless of the operating frequency of the peripheral bus.
- Before changing the CPG specification, the FLCTL must be placed in a module stop state.
- In NAND-type flash memory, the  $\overline{\text{FRE}}$  and  $\overline{\text{FWE}}$  pins operate with the frequency on the pins which CPG designated. To ensure the setup time, these operating frequencies must be specified within the maximum operating frequency of memory to be connected.
- The operating clock FCLK on the pins for the NAND-type flash memory is generated by dividing the peripheral bus operating clock Pck.

REJ09B0360-0100



## 25.2 Input/Output Pins

The pin configuration of the FLCTL is listed in table 25.1.

**Table 25.1 Pin Configuration**

Pin Name	Function	I/O	Corresponding Flash Memory Pin	Description
$\overline{\text{FCE}}$	Chip enable	Output	$\overline{\text{CE}}$	Enables flash memory connected to this LSI.
FD7 to FD0	Data I/O pins	I/O	I/O7 to I/O0	I/O pins for command, address, and data.
FCLE	Command data enable	Output	CLE	Command Latch Enable (CLE) Asserted when a command is output.
FALE	Address latch enable	Output	ALE	Address Latch Enable (ALE) Asserted when an address is output and negated when data is input or output.
$\overline{\text{FRE}}$	Read enable	Output	$\overline{\text{RE}}$	Read Enable ( $\overline{\text{RE}}$ ) Reads data at the falling edge of $\overline{\text{RE}}$ .
$\overline{\text{FWE}}$	Write enable	Output	$\overline{\text{WE}}$	Write Enable Flash memory latches a command, address, and data at the rising edge of $\overline{\text{WE}}$ .
FR/ $\overline{\text{B}}$	Ready/busy	Input	R/ $\overline{\text{B}}$	Ready/Busy Indicates ready state at high level; indicates busy state at low level.

## 25.3 Register Descriptions

Table 25.2 shows the FLCTL register configuration. Table 25.3 shows the register state in each processing mode.

**Table 25.2 Register Configuration of FLCTL**

Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
Common control register	FLCMNCR	R/W	H'FFE9 0000	H'1FE9 0000	32
Command control register	FLCMDCR	R/W	H'FFE9 0004	H'1FE9 0004	32
Command code register	FLCMCDR	R/W	H'FFE9 0008	H'1FE9 0008	32
Address register	FLADR	R/W	H'FFE9 000C	H'1FE9 000C	32
Address register 2	FLADR2	R/W	H'FFE9 003C	H'1FE9 003C	32
Data register	FLDATAR	R/W	H'FFE9 0010	H'1FE9 0010	32
Data counter register	FLDTCNTR	R/W	H'FFE9 0014	H'1FE9 0014	32
Interrupt DMA control register	FLINTDMACR	R/W	H'FFE9 0018	H'1FE9 0018	32
Ready busy timeout setting register	FLBSYTMR	R/W	H'FFE9 001C	H'1FE9 001C	32
Ready busy timeout counter	FLBSYCNT	R	H'FFE9 0020	H'1FE9 0020	32
Data FIFO register	FLDTFIFO	R/W	H'FFE9 0024/ H'FFE9 0050	H'1FE9 0024/ H'1FE9 0050	32
Control code FIFO register	FLECFIFO	R/W	H'FFE9 0028/ H'FFE9 0060	H'1FE9 0028/ H'1FE9 0060	32
Transfer control register	FLTRCR	R/W	H'FFE9 002C	H'1FE9 002C	8

**Table 25.3 Register State of FLCTL in Each Processing Mode**

<b>Register Abbreviation</b>	<b>Power-On Reset</b>	<b>Standby</b>	<b>Module Standby</b>	<b>Sleep</b>
FLCMNCR	Initialized	Retained	Retained	Retained
FLCMDCR	Initialized	Retained	Retained	Retained
FLCMCDR	Initialized	Retained	Retained	Retained
FLADR	Initialized	Retained	Retained	Retained
FLADR2	Initialized	Retained	Retained	Retained
FLDATAR	Initialized	Retained	Retained	Retained
FLDTCNTR	Initialized	Retained	Retained	Retained
FLINTDMACR	Initialized	Retained	Retained	Retained
FLBSYTMR	Initialized	Retained	Retained	Retained
FLBSYCNT	Initialized	Retained	Retained	Retained
FLDTFIFO	Initialized	Retained	Retained	Retained
FLECFIFO	Initialized	Retained	Retained	Retained
FLTRCR	Initialized	Retained	Retained	Retained

### 25.3.1 Common Control Register (FLCMNCR)

FLCMNCR is a 32-bit readable/writable register that specifies the type (NAND) of flash memory, access mode, and  $\overline{\text{FCE}}$  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
	FCK SEL	—	ECCPOS[1:0]	ACM[1:0]	NAND WF	—	—	—	—	—	—	CE0	—	—	TYPE SEL	—
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 1-Gbit or larger NAND flash memory with the page configuration of 2048 + 64 bytes. 0: When flash memory with the page configuration of 512 + 16 bytes is used. 1: When NAND flash memory with the page configuration of 2048 + 64 bytes is used. Note: When TYPESEL = 0, this bit should not be set to 1.



Bit	Bit Name	Initial Value	R/W	Description
17	QTSEL	0	R/W	<p>Select Dividing Rates for Flash Clock</p> <p>Selects the dividing rate of clock FCLK in the flash memory. This bit is used together with FCKSEL.</p> <ul style="list-style-type: none"> <li>QTSEL = 0, FCKSEL = 0: Divides a clock (Pck) provided from the CPG by two and uses it as FCLK.</li> <li>QTSEL = 0, FCKSEL = 1: Uses a clock (Pck) provided from the CPG as FCLK.</li> <li>QTSEL = 1, FCKSEL = 0: Divides a clock (Pck) provided from the CPG by four and uses it as FCLK.</li> <li>QTSEL = 1, FCKSEL = 1: Setting prohibited</li> </ul>
16	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
15	FCKSEL	0	R/W	<p>Flash Clock Select</p> <p>Selects the dividing rate of clock FCLK in the flash memory. This bit is used together with QTSEL. Refer to the description of QTSEL.</p>
14	—	0	R	<p>Reserved</p> <p>This bit is always read as 0. The write value should always be 0.</p>
13, 12	ECCPOS [1:0]	00	R/W	<p>ECC Position Specification 1 and 0</p> <p>Specify the position (0/4th/8th byte) to place the ECC in the control code area.</p> <p>00: Places the ECC at the 0 to 7th byte of control code area</p> <p>01: Places the ECC at the 4th to 11th byte of control code area</p> <p>10: Places the ECC at the 8th to 15th byte of control code area</p> <p>11: Setting prohibited</p>

Bit	Bit Name	Initial Value	R/W	Description
11, 10	ACM[1:0]	00	R/W	Access Mode Specification 1 and 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: Performs address or data input/output in one FCLK cycle 1: Performs address or data input/output in two FCLK cycles
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: Disables the chip (Outputs high level to the $\overline{\text{FCE}}$ pin) 1: Enables the chip (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 1: NAND-type flash memory is selected Note: Set this bit to 1 when using FLCTL.

### 25.3.2 Command Control Register (FLCMD CR)

FLCMD CR is a 32-bit readable/writable register that issues a command in command access mode, specifies address issue, and specifies source or destination of data transfer. In sector access mode, FLCMD CR 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]					ADR MD	CDS RC	DOSR	—	—	SEL RW	DOA DR	ADRCNT[1:0]	DOC MD2	DOC MD1
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	<p>Address Issue Byte Count Specification</p> <p>Specifies the number of bytes for the address data to be issued in address stage. This bit is used together with ADRCNT[1:0].</p> <p>0: Issue the address of byte count, specified by ADRCNT[1:0].</p> <p>1: Issue 5-byte address. ADRCNT[1:0] should be set to 00.</p>
30 to 27	SCTCNT [19:16]	All 0	R/W	<p>Sector Transfer Count Specification [19:16]</p> <p>These bits are extended bits of the sector transfer count specification [15:0], SCTCNT[15:0].</p> <p>SCTCNT[19:16] and SCTCNT[15:0] are used together to operate as SCTCNT[19:0], the 20-bit counter.</p>

Bit	Bit Name	Initial Value	R/W	Description
26	ADRMD	0	R/W	<p>Sector Access Address Specification</p> <p>This bit is invalid in command access mode. This bit is valid only in sector access mode.</p> <p>0: The value of the address register is handled as a physical sector number. Use this value usually in sector access.</p> <p>1: The value of the address register is output as the address of flash memory.</p> <p>Note: Clear this bit to 0 in continuous sector access.</p>
25	CDSRC	0	R/W	<p>Data Buffer Specification</p> <p>Specifies the data buffer to be read from or written to in the data stage in command access mode.</p> <p>0: Specifies FLDATAR as the data buffer.</p> <p>1: Specifies FLDTFIFO as the data buffer.</p>
24	DOSR	0	R/W	<p>Status Read Check</p> <p>Specifies whether or not the status read is performed after the second command has been issued in command access mode.</p> <p>0: Performs no status read</p> <p>1: Performs status read</p>
23, 22	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
21	SELRW	0	R/W	<p>Data Read/Write Specification</p> <p>Specifies the direction of read or write in data stage.</p> <p>0: Read</p> <p>1: Write</p>
20	DOADR	0	R/W	<p>Address Stage Execution Specification</p> <p>Specifies whether or not the address stage is executed in command access mode.</p> <p>0: Performs no address stage</p> <p>1: Performs address stage</p>

Bit	Bit Name	Initial Value	R/W	Description
19, 18	ADRCNT [1:0]	00	R/W	<p>Address Issue Byte Count Specification</p> <p>Specify the number of bytes for the address data to be issued in address stage.</p> <p>00: Issue 1-byte address</p> <p>01: Issue 2-byte address</p> <p>10: Issue 3-byte address</p> <p>11: Issue 4-byte address</p>
17	DOCMD2	0	R/W	<p>Second Command Stage Execution Specification</p> <p>Specifies whether or not the second command stage is executed in command access mode.</p> <p>0: Does not execute the second command stage</p> <p>1: Executes the second command stage</p>
16	DOCMD1	0	R/W	<p>First Command Stage Execution Specification</p> <p>Specifies whether or not the first command stage is executed in command access mode.</p> <p>0: Does not execute the first command stage</p> <p>1: Executes the first command stage</p>
15 to 0	SCTCNT [15:0]	H'0000	R/W	<p>Sector Transfer Count Specification [15:0]</p> <p>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.</p> <p>These bits are used together with SCTCNT[19:16].</p> <p>In command access mode, these bits are H'0 0001.</p>

### 25.3.3 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.

### 25.3.4 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 4th 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 3rd 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 2nd 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 1st 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 always read as 0. The write value should always be 0.
25 to 0	ADR[25:0]	H'000 0000	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 = 1, the ADR[25:0] bits are valid. When the ADRCNT2 bit in FLCMDCR = 0, the ADR[17:0] bits are valid.



### 25.3.5 Address Register 2 (FLADR2)

FLADR2 is a 32-bit readable/writable register, and is valid when the ADRCNT2 bit in FLCMDCR is set to 1. FLADR2 specifies an address to be output in command access 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 5th data to be output to flash memory as an address in command access mode.

### 25.3.6 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	<p>FLECFIFO Access Count</p> <p>Specify the number of longwords in FLECFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLECFIFO.</p> <p>In FLECFIFO read, these bits specify the number of longwords of the data that can be read from FLECFIFO.</p> <p>In FLECFIFO write, these bits specify the number of longwords of unoccupied area that can be written in FLECFIFO.</p>
23 to 16	DTFLW[7:0]	H'00	R	<p>FLDTFIFO Access Count</p> <p>Specify the number of longwords in FLDTFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLDTFIFO.</p> <p>In FLDTFIFO read, these bits specify the number of longwords of the data that can be read from FLDTFIFO.</p> <p>In FLDTFIFO write, these bits specify the number of longwords of unoccupied area that can be written in FLDTFIFO.</p>
15 to 12	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
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.)

### 25.3.7 Data Register (FLDATAR)

FLDATAR is a 32-bit readable/writable register. It stores input/output data used when 0 is written to the CDSRC bit in FLCMDCR 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 write: Specify write data In read: 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 write: Specify write data In read: Store read data

Bit	Bit Name	Initial Value	R/W	Description
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 write: Specify write data In read: 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 write: Specify write data In read: Store read data

### 25.3.8 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 been 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	ST ERB	BTO ERB	TRR EQF1	TRR EQF0	STER INTE	RBERR 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 an interrupt when an ECC error occurs 1: Enables an interrupt when an ECC error occurs

Bit	Bit Name	Initial Value	R/W	Description
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	FIFOTRG [1:0]	00	R/W	FIFO Trigger Setting Change the condition for the FIFO transfer request. In flash-memory read: 00: Issue an interrupt or a DMA transfer request to the CPU when FLDTFIFO stores 4 bytes of data. 01: Issue an interrupt or a DMA transfer request to the CPU when FLDTFIFO stores 16 bytes of data. 10: Issue an interrupt or a DMA transfer request to the CPU when FLDTFIFO stores 128 bytes of data. 11: Issue an interrupt to the CPU when FLDTFIFO stores 128 bytes of data, or issue a DMA transfer request to the CPU when FLDTFIFO stores 16 bytes of data. In flash-memory programming: 00: Issue an interrupt to the CPU when FLDTFIFO has empty area of 4 bytes or more (do not set DMA transfer). 01: Issue an interrupt or a DMA transfer request to the CPU when FLDTFIFO has empty area of 16 bytes or more. 10: Issue an interrupt to the CPU when FLDTFIFO has empty area of 128 bytes or more (do not set DMA transfer). 11: Issue an interrupt to the CPU when FLDTFIFO has empty area of 128 bytes or more, or issue a DMA transfer request to the CPU when FLDTFIFO has empty area of 16 bytes or more.
19	AC1CLR	0	R/W	FLECFIFO Clear Clears FLECFIFO. 0: Retains the FLECFIFO value. In flash-memory access, this bit should be cleared to 0. 1: Clears FLECFIFO. After FLECFIFO has been cleared, this bit should be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
18	AC0CLR	0	R/W	<p>FLDTFIFO Clear</p> <p>Clears FLDTFIFO.</p> <p>0: Retains the FLDTFIFO value. In flash-memory access, this bit should be cleared to 0.</p> <p>1: Clears FLDTFIFO. After FLDTFIFO has been cleared, this bit should be cleared to 0.</p>
17	DREQ1EN	0	R/W	<p>FLECFIFODMA Request Enable</p> <p>Enables or disables the DMA transfer request issued from FLECFIFO.</p> <p>0: Disables the DMA transfer request issued from FLECFIFO</p> <p>1: Enables the DMA transfer request issued from FLECFIFO</p>
16	DREQ0EN	0	R/W	<p>FLDTFIFODMA Request Enable</p> <p>Enables or disables the DMA transfer request issued from FLDTFIFO.</p> <p>0: Disables the DMA transfer request issued from the FLDTFIFO</p> <p>1: Enables the DMA transfer request issued from the FLDTFIFO</p>
15 to 10	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
9	ECERB	0	R/W	<p>ECC Error</p> <p>Indicates the result of ECC error detection. This bit is set to 1 if an ECC error occurs while flash memory is read in sector access mode.</p> <p>No interrupt occurs even if this bit is set to 1.</p> <p>This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag.</p> <p>0: Indicates that no ECC error occurs (Latched ECC is all 0.)</p> <p>1: Indicates that an ECC error occurs</p>

Bit	Bit Name	Initial Value	R/W	Description
8	STERB	0	R/W	<p>Status Error</p> <p>Indicates the result of status read. This bit is set to 1 if the specific bit in the bits STAT[7:0] in FLBSYCNTR is set to 1 in status read.</p> <p>This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag.</p> <p>0: Indicates that no status error occurs (the specific bit in the bits STAT[7:0] in FLBSYCNTR is 0.)</p> <p>1: Indicates that a status error occurs</p> <p>For details on the specific bit in STAT[7:0] bits, see section 25.4.6, Status Read.</p>
7	BTOERB	0	R/W	<p>Timeout Error</p> <p>This bit is set to 1 if a timeout error occurs (the bits RBTIMCNT[19:0] in FLBSYCNTR are decremented to 0).</p> <p>This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag.</p> <p>0: Indicates that no timeout error occurs</p> <p>1: Indicates that a timeout error occurs</p>
6	TRREQF1	0	R/W	<p>FLECFIFO Transfer Request Flag</p> <p>Indicates that a transfer request is issued from FLECFIFO.</p> <p>This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag.</p> <p>0: Indicates that no transfer request is issued from FLECFIFO</p> <p>1: Indicates that a transfer request is issued from FLECFIFO</p>
5	TRREQF0	0	R/W	<p>FLDTFIFO Transfer Request Flag</p> <p>Indicates that a transfer request is issued from FLDTFIFO.</p> <p>This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag.</p> <p>0: Indicates that no transfer request is issued from FLDTFIFO</p> <p>1: Indicates that a transfer request is issued from FLDTFIFO</p>

Bit	Bit Name	Initial Value	R/W	Description
4	STERINTE	0	R/W	<p>Interrupt Enable at Status Error</p> <p>Enables or disables an interrupt request to the CPU when a status error has occurred.</p> <p>0: Disables the interrupt request to the CPU by a status error</p> <p>1: Enables the interrupt request to the CPU by a status error</p>
3	RBERINTE	0	RW	<p>Interrupt Enable at Timeout Error</p> <p>Enables or disables an interrupt request to the CPU when a timeout error has occurred.</p> <p>0: Disables the interrupt request to the CPU by a timeout error</p> <p>1: Enables the interrupt request to the CPU by a timeout error</p>
2	TEINTE	0	R/W	<p>Transfer End Interrupt Enable</p> <p>Enables or disables an interrupt request to the CPU when a transfer has been ended (TREND bit in FLTRCR).</p> <p>0: Disables the transfer end interrupt request to the CPU</p> <p>1: Enables the transfer end interrupt request to the CPU</p>
1	TRINTE1	0	R/W	<p>FLECFIFO Transfer Request Enable to CPU</p> <p>Enables or disables an interrupt request to the CPU by a transfer request issued from FLECFIFO.</p> <p>0: Disables an interrupt request to the CPU by a transfer request from FLECFIFO.</p> <p>1: Enables an interrupt request to the CPU by a transfer request from FLECFIFO.</p> <p>When the DMA transfer is enabled, this bit should be cleared to 0.</p>



Bit	Bit Name	Initial Value	R/W	Description
0	TRINTE0	0	R/W	<p>FLDTFIFO Transfer Request Enable to CPU</p> <p>Enables or disables an interrupt request to the CPU by a transfer request issued from FLDTFIFO.</p> <p>0: Disables an interrupt request to the CPU by a transfer request from FLDTFIFO</p> <p>1: Enables an interrupt request to the CPU by a transfer request from FLDTFIFO</p> <p>When the DMA transfer is enabled, this bit should be cleared to 0.</p>

### 25.3.9 Ready Busy Timeout Setting Register (FLBSYTMR)

FLBSYTMR is a 32-bit readable/writable register that specifies the timeout time when the  $\overline{\text{FR}}/\overline{\text{B}}$  pin is busy.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RBTMOUT[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
	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 20	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
19 to 0	RBTMOUT[19:0]	H'00000	R/W	<p>Ready Busy Timeout</p> <p>Specify timeout time (the number of Pck clocks) in busy state. When these bits are set to 0, timeout is not generated.</p>

### 25.3.10 Ready Busy Timeout Counter (FLBSYCNT)

FLBSYCNT is a 32-bit read-only register.

The status of flash memory obtained by the status read is stored in the bits STAT[7:0].

The timeout time set in the bits RBTMOUT[19:0] in FLBSYTMR is copied to the bits RBTIMCNT[19:0] and counting down is started when the FR/ $\overline{B}$  pin is placed in a busy state. When values in the RBTIMCNT[19:0] become 0, 1 is set to the BTOERB bit in FLINTDMACR, thus notifying that a timeout error has occurred. In this case, an FLSTE interrupt request 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[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	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 flash memory status obtained by the status read.
23 to 20	—	All 0	R	Reserved These bits are always read as 0.
19 to 0	RBTIMCNT[19:0]	H'00000	R	Ready Busy Timeout Counter When the FR/ $\overline{B}$ pin is placed in a busy state, the values of the bits RBTMOUT[19:0] in FLBSYTMR are copied to these bits. These bits are counted down while the FR/ $\overline{B}$ pin is busy. A timeout error occurs when these bits are decremented to 0.

### 25.3.11 Data FIFO Register (FLDTFIFO)

FLDTFIFO is used to read or write the data FIFO area.

In DMA transfer, data in this register must be specified as the destination (source).

Note that the direction of read or write specified by the SELRW bit in FLCMDCR must match that specified in this register.

When transferring 16-byte DMA, access FLDTFIFO from the address on the 16-byte address boundary.

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:	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
	DTFO[15:8]								DTFO[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	DTFO [31:24]	H'00	R/W	First Data Specify 1st data to be input or output via the FD7 to FD0 pins. In write: Specify write data In read: Store read data
23 to 16	DTFO [23:16]	H'00	R/W	Second Data Specify 2nd data to be input or output via the FD7 to FD0 pins. In write: Specify write data In read: Store read data
15 to 8	DTFO[15:8]	H'00	R/W	Third Data Specify 3rd data to be input or output via the FD7 to FD0 pins. In write: Specify write data In read: Store read data

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DTFO[7:0]	H'00	R/W	Fourth Data Specify 4th data to be input or output via the FD7 to FD0 pins. In write: Specify write data In read: Store read data

### 25.3.12 Control Code FIFO Register (FLECFIFO)

FLECFIFO is used to read or write the control code FIFO area.

In DMA transfer, data in this register must be specified as the destination (source).

Note that the direction of read or write specified by the SELRW bit in FLCMDCR must match that specified in this register.

When transferring 16-byte DMA, access FLECFIFO from the address on the 16-byte address boundary.

Before accessing the FLECFIFO, clear the FIFO data by setting the AC1CLR bit in the FINTDMACR to 1.

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:	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
	ECFO[15:8]								ECFO[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	ECFO [31:24]	H'00	R/W	First Data Specify 1st data to be input or output via the FD7 to FD0 pins. In write: Specify write data In read: Store read data

Bit	Bit Name	Initial Value	R/W	Description
23 to 16	ECFO [23:16]	H'00	R/W	Second Data Specify 2nd data to be input or output via the FD7 to FD0 pins. In write: Specify write data In read: Store read data
15 to 8	ECFO[15:8]	H'00	R/W	Third Data Specify 3rd data to be input or output via the FD7 to FD0 pins. In write: Specify write data In read: Store read data
7 to 0	ECFO[7:0]	H'00	R/W	Fourth Data Specify 4th data to be input or output via the FD7 to FD0 pins. In write: Specify write data In read: Store read data

### 25.3.13 Transfer Control Register (FLTRCR)

Setting the TRSTRT bit to 1 initiates access to flash memory. Access completion can be checked by the TREND bit. During the transfer (from when the TRSTRT bit is set to 1 until the TREND bit is set to 1), the processing should not be forcibly ended (by setting the TRSTRT bit to 0).

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TR END	TR STRT
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.

Bit	Bit Name	Initial Value	R/W	Description
1	TREND	0	R/W	Processing End Flag Bit 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 By setting this bit from 0 to 1 when the TREND bit is 0, processing in the access mode specified by the access mode specification bits ACM[1:0] is initiated. 0: Stops transfer 1: Starts transfer

## 25.4 Operation

### 25.4.1 Operating Modes

Two operating modes are supported.

- Command access mode
- Sector access mode

The ECC generation and error check are performed in sector access mode.

### 25.4.2 Register Setting Procedure

Figure 25.2 shows the register setting flow required for accessing to the flash memory.

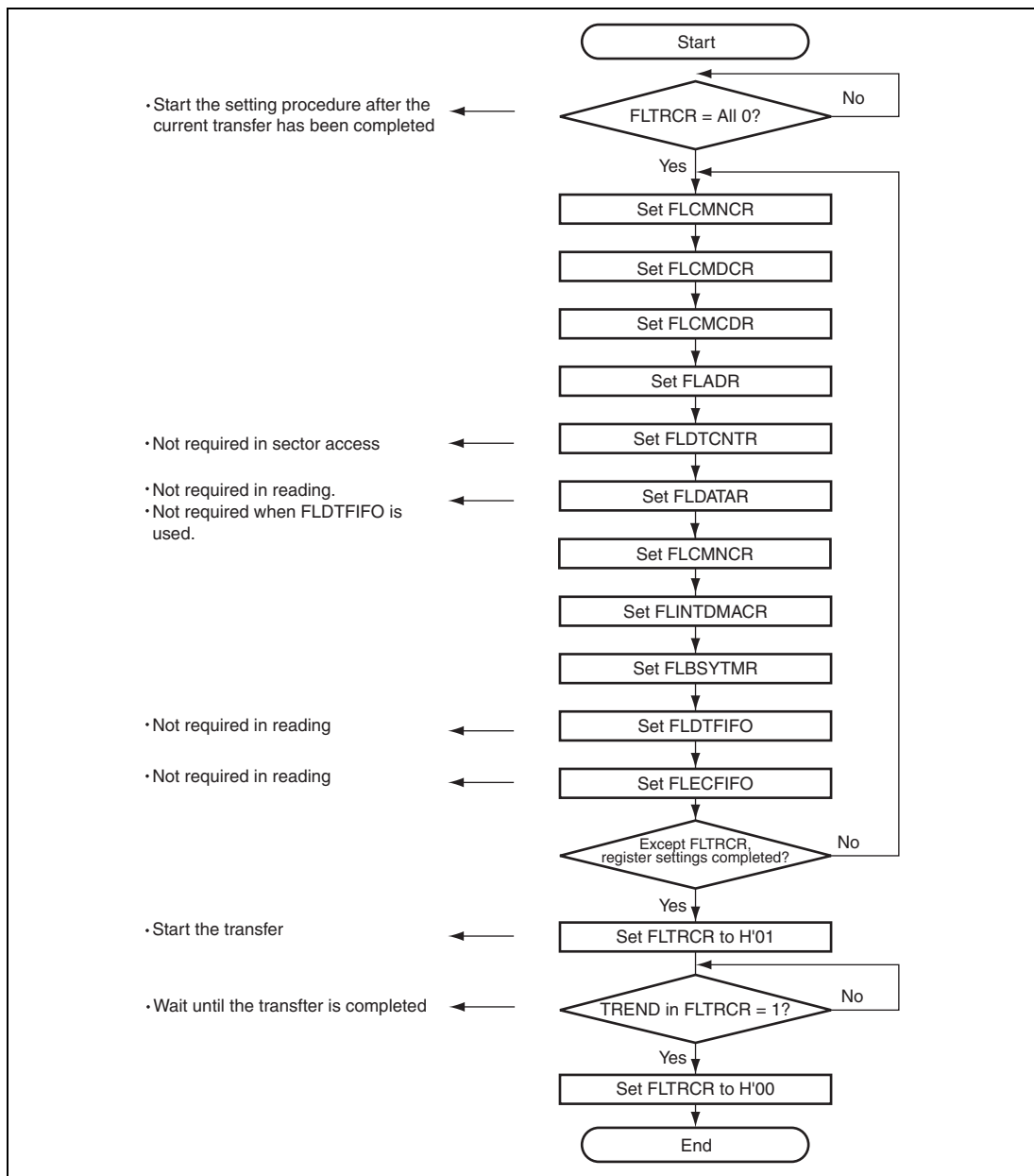
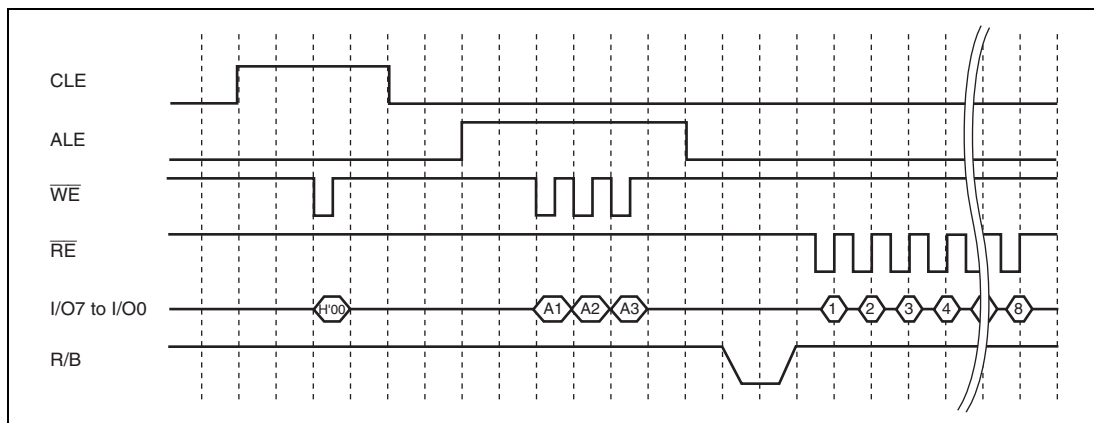


Figure 25.2 Register Setting Flow

### 25.4.3 Command Access Mode

Command access mode accesses flash memory by specifying a command to be issued to flash memory, address, data, read or write direction, and number of times to the registers. In this mode, I/O data can be transferred by the DMA via FLDTFIFO.

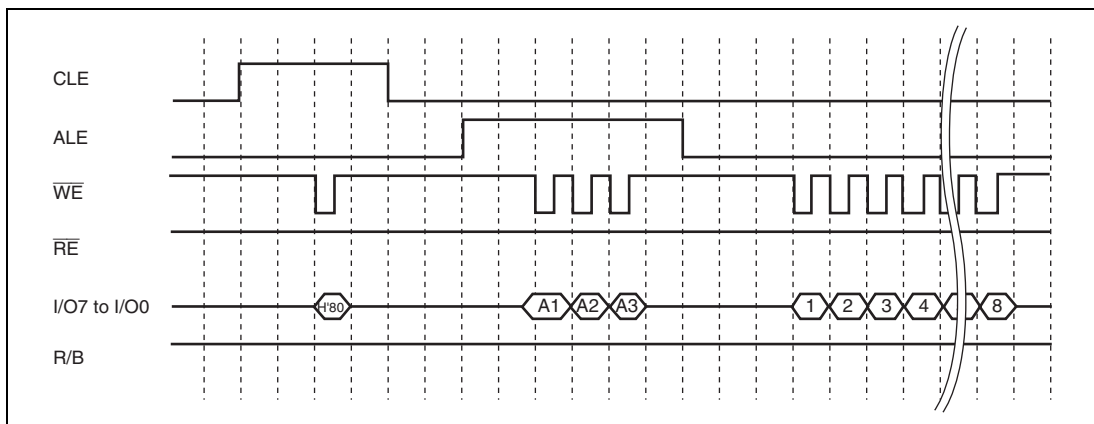
**NAND-Type Flash Memory Access:** Figure 25.3 shows an example of read operation for NAND-type flash memory. In this example, the first command is specified as H'00, address data length is specified as 3 bytes, and the number of read bytes is specified as 8 bytes in the data counter.



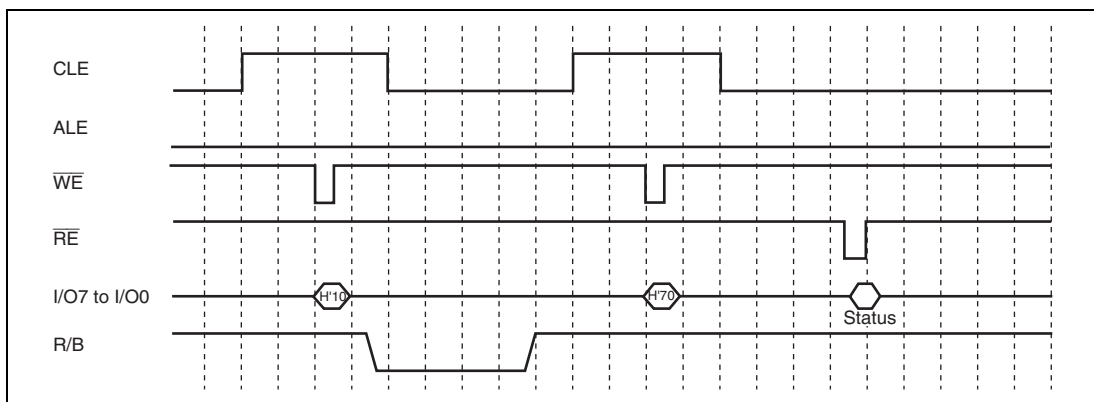
**Figure 25.3 Read Operation Timing for NAND-Type Flash Memory (1)**

Figures 25.4 and 25.5 show examples of programming operation for NAND-type flash memory.



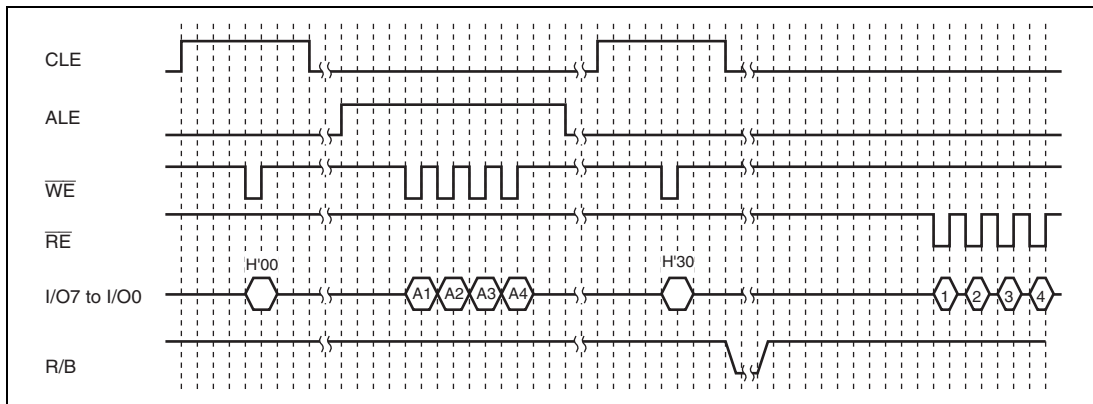


**Figure 25.4 Programming Operation Timing for NAND-Type Flash Memory (1)**



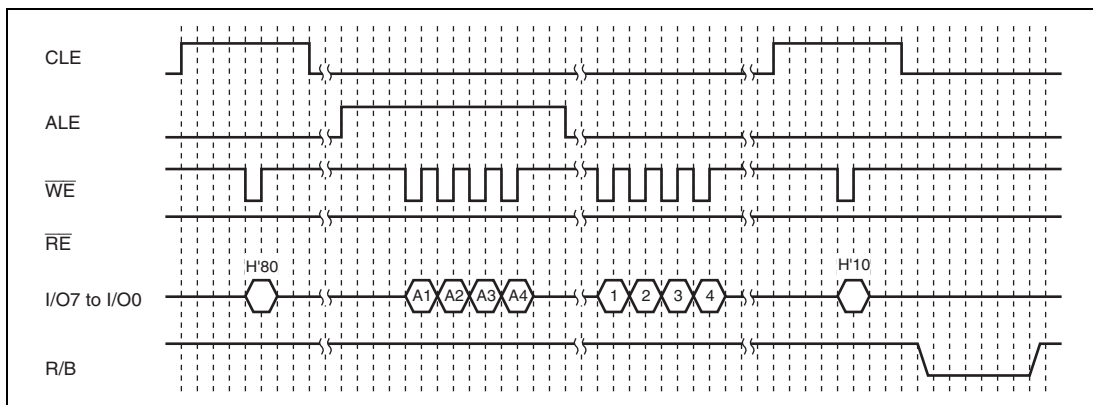
**Figure 25.5 Programming Operation Timing for NAND-Type Flash Memory (2)**

**NAND-Type Flash Memory (2048 + 64 Bytes) Access:** Figure 25.6 shows an example of read operation for NAND-type flash memory (2048 + 64 bytes). In this example, the first command is specified as H'00, the second command is specified as H'30, and address data length is specified as 4 bytes. The number of read bytes is specified as 4 bytes in the data counter.

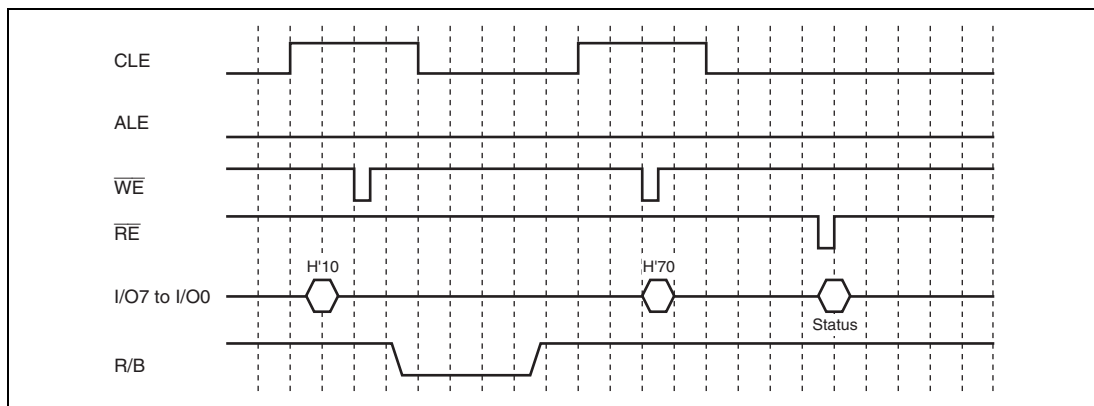


**Figure 25.6 Read Operation Timing for NAND-Type Flash Memory**

Figures 26.7 and 26.8 show examples of programming operation for NAND-type flash memory (2048 + 64 bytes).



**Figure 25.7 Programming Operation Timing for NAND-Type Flash Memory (1)**



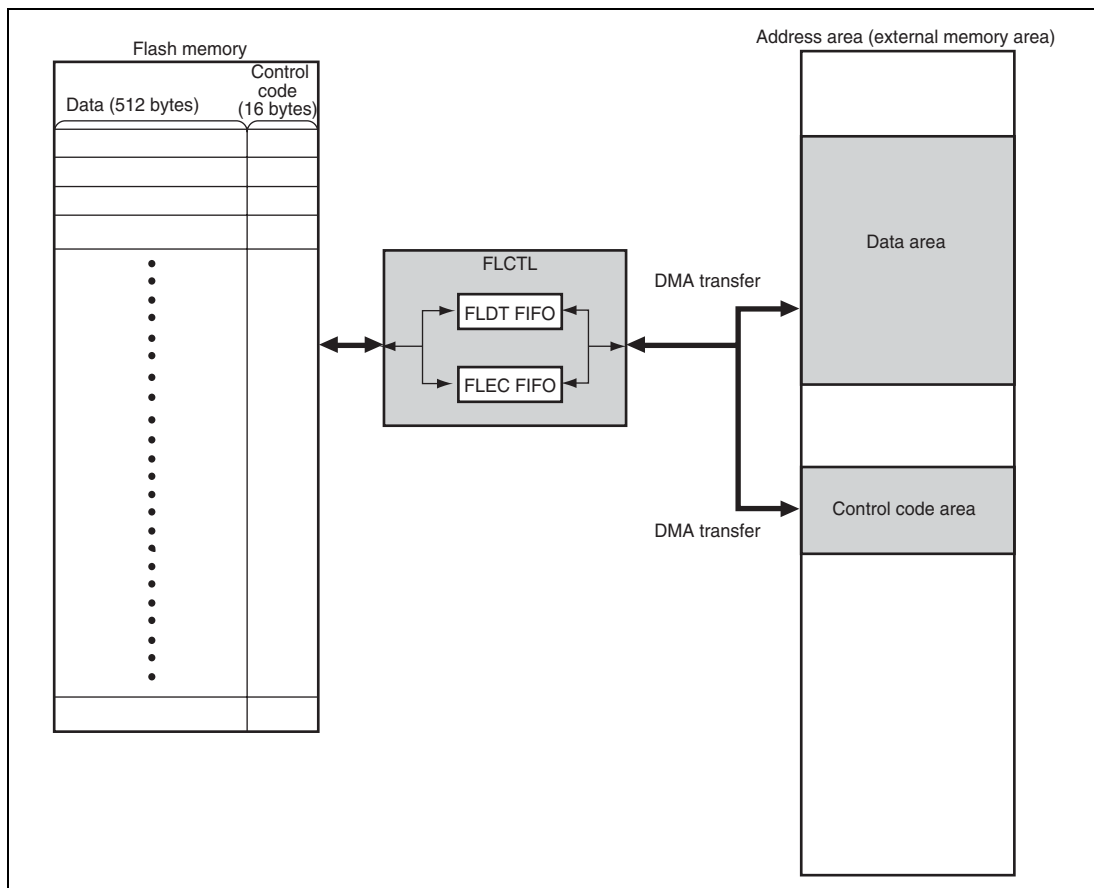
**Figure 25.8 Programming Operation Timing for NAND-Type Flash Memory (2)**

#### 25.4.4 Sector Access Mode

In sector access mode, flash memory can be read or programmed in sector units by specifying the number of physical sectors to be accessed. In programming, an ECC is added. In read, an ECC error check (detection) is performed.

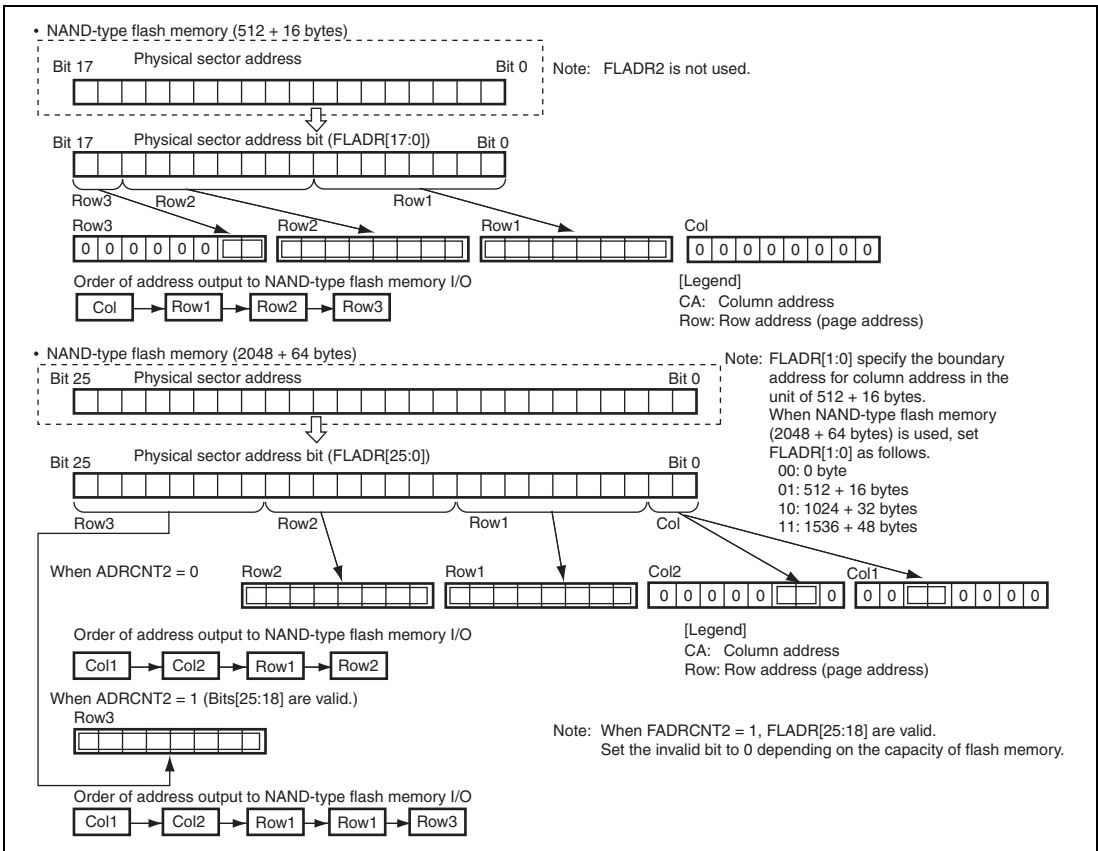
Since 512-byte data is stored in FLDTFIFO and 16-byte control code is stored in FLECFIFO, the DREQ1EN and DREQ0EN bits in FLINTDMACR can be set to transfer by the DMA.

Figure 25.9 shows the relationship of DMA transfer between sectors in flash memory (data and control code) and memory on the address space.



**Figure 25.9 Relationship between DMA Transfer and Sector (Data and Control Code), and Memory and DMA Transfer**

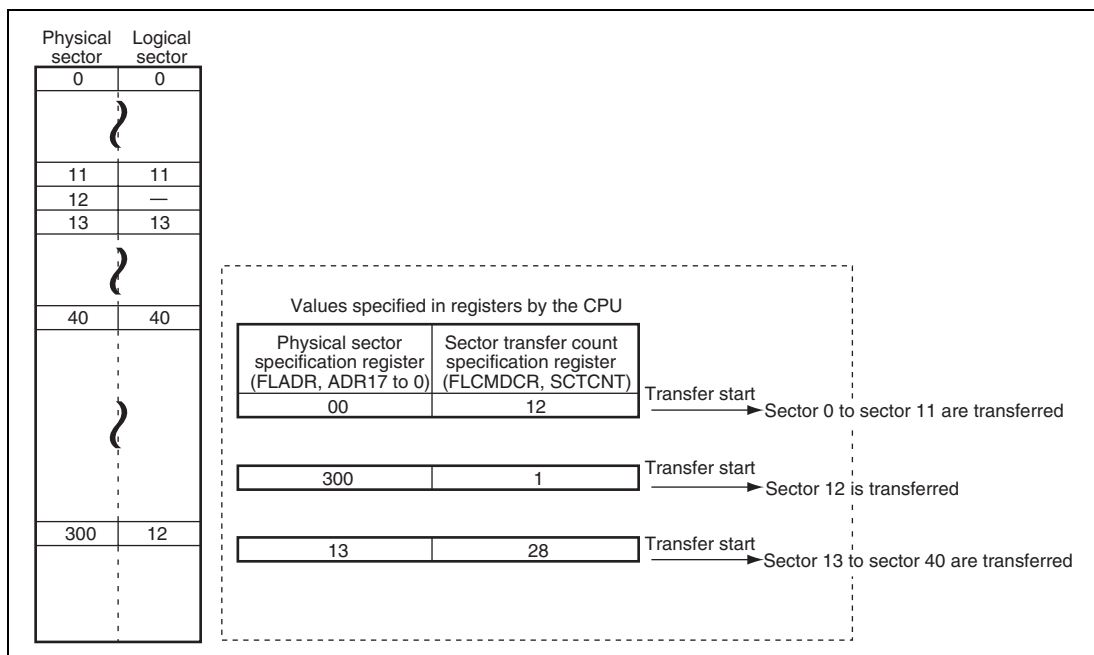
**Physical Sector:** Figure 25.10 shows the relationship between the physical sector address of NAND-type flash memory and the address of flash memory.



**Figure 25.10 Relationship between Sector Number and Address Expansion of NAND-Type Flash Memory**

**Continuous Sector Access:** Continuous physical sectors can be read or written by specifying the start physical sector of NAND-type flash memory and the number of sectors to be transferred.

Figure 25.11 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 25.11 Sector Access when Unusable Sector Exists in Continuous Sectors**

### 25.4.5 ECC Error Correction

The FLCTL generates and adds an ECC code during write operation in sector access mode and performs ECC error check during read operation in sector access mode. The FLCTL, however, does not perform error correction. Note that errors must be corrected by software.

### 25.4.6 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 I/O7 to I/O0 pins and stored in the bits STAT[7:0] in FLBSYCNT. The bits STAT[7:0] 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 STAT[7:0] 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.

The status register of NAND-type flash memory can be read by inputting command H'70 to NAND-type flash memory. If programming is executed in command access mode or sector access mode while the DOSR bit in FLCMDCR is set to 1, the FLCTL automatically inputs command H'70 to NAND-type flash memory and reads the status register of NAND-type flash memory. When the status register of NAND-type flash memory is read, the I/O7 to I/O0 pins indicate the following information as described in table 25.4.

**Table 25.4 Status Read of NAND-Type Flash Memory**

I/O	Status (definition)	Description
I/O7	Program protection	0: Cannot be programmed 1: Can be programmed
I/O6	Ready/busy	0: Busy state 1: Ready state
I/O5 to I/O1	Reserved	—
I/O0	Program/erase	0: Pass 1: Fail

## 25.5 Interrupt Sources

The FLCTL has six interrupt sources: Status error, ready/busy timeout error, ECC error, transfer end, FIFO0 transfer request, and FIFO1 transfer request. Each of the interrupt sources has its corresponding interrupt flag and the interrupt can be requested independently to the CPU if the interrupt is enabled by the interrupt enable bit. Note that the status error, ready/busy timeout error, and ECC error use the common FLSTE interrupt to the CPU.

**Table 25.5 FLCTL Interrupt Requests**

Interrupt Source	Interrupt Flag	Enable Bit	Description	Priority
FLSTE interrupt	STERB	STERINTE	Status error	Highest ↑ ↓ Lowest
	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	

Note: Flags for the FIFO0 overrun error/underrun error and FIFO1 overrun error/underrun error also exist. However, no interrupt is requested to the CPU.

## 25.6 DMA Transfer Specifications

The FLCTL can request DMA transfers separately to the data area FLDTFIFO and control code area FLECFIFO. Table 25.6 summarizes DMA transfer enable or disable states in each access mode.

**Table 25.6 DMA Transfer Specifications**

	Sector Access Mode	Command Access Mode
FLDTFIFO	DMA transfer enabled	DMA transfer enabled
FLECFIFO	DMA transfer enabled	DMA transfer disabled

For the setting of DMAC, see section 12, Direct Memory Access Controller (DMAC).



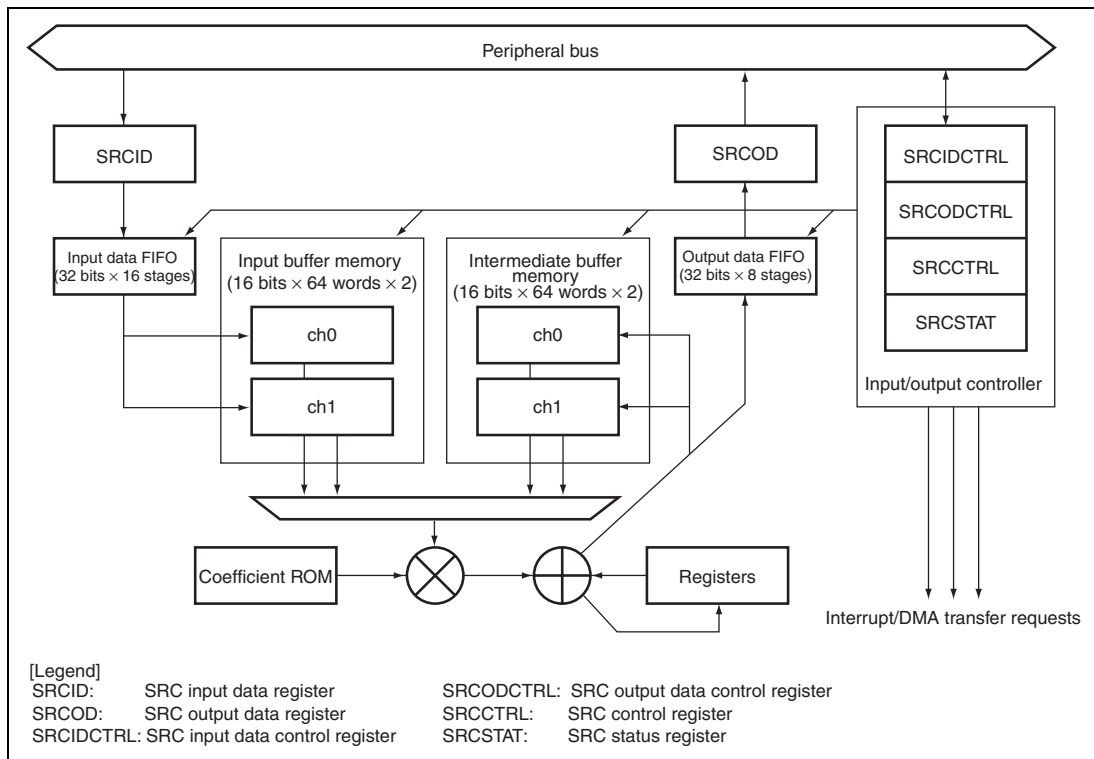
## Section 26 Sampling Rate Converter (SRC)

The sampling rate converter (SRC) converts the sampling rate for data produced by decoders such as WMA, MP3, or AAC.

### 26.1 Features

- Data size: 16 bits (stereo/monaural)
- Sampling rates  
Input: Either 8 kHz, 11.025 kHz, 12 kHz, 16 kHz, 22.05 kHz, 24 kHz, 32 kHz, 44.1 kHz, or 48 kHz is selectable.  
Output: Either 32 kHz, 44.1 kHz or 48 kHz is selectable.
- Processing capacity: A maximum of 10  $\mu$ s sample output interval ( $P_{ch} = 54$  MHz)
- SNR: 93 db or higher
- Three interrupt sources: Input data FIFO empty, output data FIFO full, and output data FIFO overwrite
- Two DMA transfer sources: Input data FIFO empty and output data FIFO full
- Module standby mode  
Power consumption can be reduced by stopping clock supply to the SRC when not used.

Figure 26.1 shows a block diagram of the SRC.



**Figure 26.1 Block Diagram of SRC**

## 26.2 Register Descriptions

The SRC has the following registers:

**Table 26.1 Register Configuration**

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
SRC input data register	SRCID	R/W	H'FFF2 0000	H'1FF2 0000	16, 32
SRC output data register	SRCOD	R	H'FFF2 0004	H'1FF2 0004	16, 32
SRC input data control register	SRCIDCTRL	R/W	H'FFF2 0008	H'1FF2 0008	16
SRC output data control register	SRCODCTRL	R/W	H'FFF2 000A	H'1FF2 000A	16
SRC control register	SRCCTRL	R/W	H'FFF2 000C	H'1FF2 000C	16
SRC status register	SRCSTAT	R/(W)*	H'FFF2 000E	H'1FF2 000E	16

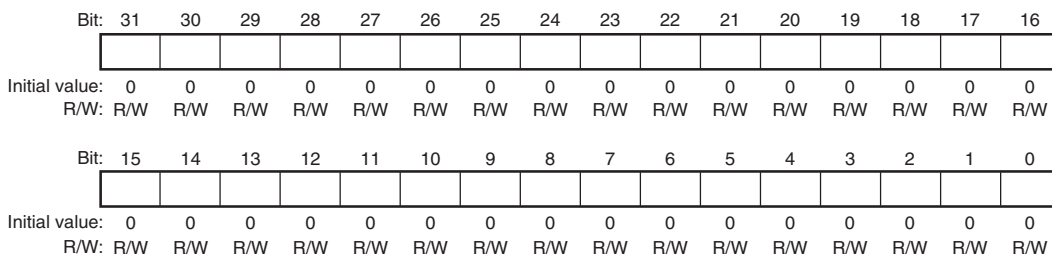
Note: \* Bits 15 to 3 are read-only. Only 0 can be written to bits 2 to 0 after having read as 1.

**Table 26.2 State of Registers in Each Operating Mode**

Register Name	Abbreviation	Power-on Reset	Sleep	Module Standby
SRC input data register	SRCID	H'0000 0000	Retained	Retained
SRC output data register	SRCOD	H'0000 0000	Retained	Retained
SRC input data control register	SRCIDCTRL	H'0000	Retained	Retained
SRC output data control register	SRCODCTRL	H'0000	Retained	Retained
SRC control register	SRCCTRL	H'0000	Retained	Retained
SRC status register	SRCSTAT	H'0002	Retained	Retained

### 26.2.1 SRC Input Data Register (SRCID)

SRCID is a 32-bit readable/writable register that is used to input the data before sampling rate conversion. All the bits are read as 0. The data input to SRCID is stored in the 16-stage input data FIFO. When the number of data in input data FIFO is 16, writing to SRCID is invalid. For stereo data, bits 31 to 16 are for ch 0 data, and bits 15 to 0 are for ch 1 data. For monaural data, data in bits 31 to 16 is valid, and data in bits 15 to 0 is invalid.



The data subject to sampling rate conversion is aligned differently depending on the IED bit setting in SRCIDCTRL. Table 26.3 shows the relationship between the IED bit setting and data alignment.

**Table 26.3 Alignment of Data before Sampling Rate Conversion**

IED	ch0[15:8]	ch0[7:0]	ch1[15:8]	ch1[7:0]
0	SRCID[31:24]	SRCID[23:16]	SRCID[15:8]	SRCID[7:0]
1	SRCID[23:16]	SRCID[31:24]	SRCID[7:0]	SRCID[15:8]

### 26.2.2 SRC Output Data Register (SRCOD)

SRCOD is a 32-bit read-only register used to output the data after sampling rate conversion. The data in 8-stage output data FIFO is read through SRCOD.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3
R/W:	R	R	R	R	R	R	R	R	R	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*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3	0*3
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The data in SRCOD is aligned differently depending on the OCH and OED bit setting in SRCODCTRL. Table 26.4 shows the correspondence between the OCH and OED bit setting and data alignment in SRCOD.

**Table 26.4 Alignment of Data in SRCOD**

OCH	OED	SRCOD[31:24]	SRCOD[23:16]	SRCOD[15:8]	SRCOD[7:0]
0	0	ch0[15:8]	ch0[7:0]	ch1[15:8]* <sup>2</sup>	ch1[7:0]* <sup>2</sup>
	1	ch0[7:0]	ch0[15:8]	ch1[7:0]* <sup>2</sup>	ch1[15:8]* <sup>2</sup>
1* <sup>1</sup>	0	ch1[15:8]	ch1[7:0]	ch0[15:8]	ch0[7:0]
	1	ch1[7:0]	ch1[15:8]	ch0[7:0]	ch0[15:8]

- Notes:
1. When processing monaural data, do not set the bit to 1.
  2. When processing monaural data, the data in these bits is invalid.
  3. If the CL bit in the SRCCTRL register is read after 1 is written to it, it is read as 0. If the CL bit is read before 1 is written to it, the read value cannot be guaranteed.

### 26.2.3 SRC Input Data Control Register (SRCIDCTRL)

SRCIDCTRL is a 16-bit readable/writable register that specifies the endian format of input data, enables/disables the interrupt requests, and specifies the triggering number of data units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	IED	IEN	-	-	-	-	-	-	-	IFTRG[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/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
9	IED	0	R/W	Input Data Endian  Specifies the endian format of the input data. 0: Big endian 1: Little endian
8	IEN	0	R/W	Input Data FIFO Empty Interrupt Enable  Enables/disables the input data FIFO empty interrupt request to be issued when the number of data units in the input FIFO becomes equal to or smaller than the triggering number specified by the IFTRG1 and IFTRG0 bits, thus resulting in the IINT bit in the SRC status register (SRCSTAT) being set to 1. 0: Input data FIFO empty interrupt is disabled. 1: Input data FIFO empty interrupt is enabled.
7 to 2	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	IFTRG[1:0]	00	R/W	<p>Input FIFO Data Triggering Number</p> <p>Specifies the condition in terms of the number on which the IINT bit in the SRC status register (SRCSTAT) is set to 1. When the number of data units in the input FIFO becomes equal to or smaller than the triggering number listed below, the IINT bit is set to 1.</p> <p>00: 0</p> <p>01: 4</p> <p>10: 8</p> <p>11: 12</p>

## 26.2.4 SRC Output Data Control Register (SRCODCTRL)

SRCODCTRL is a 16-bit readable/writable register that specifies whether to exchange the channels for the output data, specifies the endian format of output data, enables/disables the interrupt requests, and specifies the triggering number of data units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	OCH	OED	OEN	-	-	-	-	-	-	-	OFTRG[1:0]
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
10	OCH	0	R/W	<p>Output Data Channel Exchange</p> <p>Specifies whether to exchange the channels for the SRC output data register (SRCOD). When processing monaural data, do not set this bit to 1.</p> <p>0: Does not exchange the channels (the same order as data input)</p> <p>1: Exchanges the channels (the opposite order from data input)</p>

Bit	Bit Name	Initial Value	R/W	Description
9	OED	0	R/W	Output Data Endian Specifies the endian format of the output data. 0: Big endian 1: Little endian
8	OEN	0	R/W	Output Data FIFO Full Interrupt Enable Enables/disables the output data FIFO full interrupt request to be issued when the number of data units in the output FIFO becomes equal to or greater than the number specified by the OFTRG1 and OFTRG0 bits, thus resulting in the OINT bit in SRC status register (SRCSTAT) being set to 1. 0: Output data FIFO full interrupt is disabled. 1: Output data FIFO full interrupt is enabled.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	OFTRG[1:0]	00	R/W	Output FIFO Data Trigger Number Specifies the condition in terms of the number on which the OINT bit in the SRC status register (SRCSTAT) is set to 1. When the number of data units in the output FIFO becomes equal to or greater than the number listed below, the OINT bit is set to 1. 00: 1 01: 2 10: 4 11: 6



## 26.2.5 SRC Control Register (SRCCTRL)

SRCCTRL is a 16-bit readable/writable register that enables/disables the SRC module operation, enables/disables the interrupt requests, and specifies flush processing, clear processing of the internal work memory, and the input and output sampling rates.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SRCEN	-	EEN	FL	CL		IFS[3:0]			-	-	OFS[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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description												
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.												
12	SRCEN	0	R/W	SRC Module Enable Enables/disables the SRC module operation. 0: Disables the SRC module operation. 1: Enables the SRC module operation. Note: When the SRCEN bit is 1, do not modify the following bits: <table><tr><th>Register Name</th><th>Bit</th><th>Bit Name</th></tr><tr><td>SRCIDCTRL</td><td>9</td><td>IED</td></tr><tr><td>SRCODCTRL</td><td>9, 10</td><td>OCH, OED</td></tr><tr><td>SRCCTRL</td><td>7 to 4, 0</td><td>IFS[3:0], OFS</td></tr></table>	Register Name	Bit	Bit Name	SRCIDCTRL	9	IED	SRCODCTRL	9, 10	OCH, OED	SRCCTRL	7 to 4, 0	IFS[3:0], OFS
Register Name	Bit	Bit Name														
SRCIDCTRL	9	IED														
SRCODCTRL	9, 10	OCH, OED														
SRCCTRL	7 to 4, 0	IFS[3:0], OFS														
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	EEN	0	R/W	<p>Output Data FIFO Overwrite Interrupt Enable</p> <p>Enables/disables the output data FIFO overwrite interrupt request to be issued when the data in the output FIFO has been overwritten before being read thus setting the OVF bit in SRC status register (SRCSTAT) to 1.</p> <p>0: Output data FIFO overwrite interrupt is disabled. 1: Output data FIFO overwrite interrupt is enabled.</p>
9	FL	0	R/W	<p>Internal Work Memory Flush</p> <p>Writing 1 to this bit starts converting the sampling rate of all the data in the input FIFO, input buffer memory, and intermediate memory (i.e., flush processing). This bit is always read as 0. When SRCEN = 0, writing 1 to this bit does not trigger flush processing.</p> <p>If this bit is set to 1 while the number of data units in the input buffer memory is less than 64, the flash processing is not performed because valid output data cannot be obtained.</p>
8	CL	0	R/W	<p>Internal Work Memory Clear</p> <p>Writing 1 to this bit clears the input FIFO, output FIFO, input buffer memory, intermediate memory, and accumulator. This bit is always read as 0.</p> <p>Before operating the SRC, the SRC should be internally cleared by writing 1 to this bit. To perform the clearing processing correctly, wait 32 cycles of peripheral bus clock after writing 1 to this bit and then perform the next processing. In addition, when this bit is set to 1, IFS[3:0] and OFS should also be set.</p>

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IFS[3:0]	All 0	R/W	Input Sampling Rate Specifies the input sampling rate. 0000: 8.0 kHz 0001: 11.025 kHz 0010: 12.0 kHz 0011: Setting prohibited 0100: 16.0 kHz 0101: 22.05 kHz 0110: 24.0 kHz 0111: Setting prohibited 1000: 32.0 kHz 1001: 44.1 kHz 1010: 48.0 kHz 1011: Setting prohibited 1100: Setting prohibited 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
3 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1 to 0	OFS[1:0]	All 0	R/W	Output Sampling Rate Specifies the output sampling rate. 00: 44.1 kHz 01: 48.0 kHz 10: 32.0 kHz 01: Setting prohibited

The number of output data units obtained as conversion result can be calculated by using the following expression (A) or (B). Table 26.5 shows the relationship of setting value and applicable formula between IFS and OFS[1:0].

$$\text{Number of output data} = \text{Number of input data} \times \frac{\text{Output sampling rate}}{\text{Input sampling rate}} \quad \dots (A)$$

$$\text{Number of output data} = \text{Number of input data} \times \frac{\text{Output sampling rate}}{\text{Input sampling rate}} - 1 \quad \dots (B)$$

**Table 26.5 Relationship between Sampling Rate Setting and Number of Output Data**

OFS[1:0] Value (Output Sampling Rate (kHz))	IFS [3:0]Setting (Input Sampling Rate [kHz])								
	0000 (8.0)	0001 (11.025)	0010 (12.0)	0100 (16.0)	0101 (22.05)	0110 (24.0)	1000 (32.0)	1001 (44.1)	1010 (48.0)
00 (44.1) B	B	A	A	B	A	A	B	—	A
01 (48.0) B	B	B	A	B	B	A	B	B	—
10 (32.0) A	A	B	B	A	B	A	—	B	A

## 26.2.6 SRC Status Register (SRCSTAT)

SRCSTAT is a 16-bit readable/writable register that indicates the number of data units in the input and output data FIFOs, whether the various interrupt sources have been generated or not, and the flush processing status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OFDN[3:0]				IFDN[4:0]				-	-	FLF	-	OVF	IINT	OINT	
Initial value:	0	0	0	0	0	0	0	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)*

Note: \* Only 0 can be written after having read as 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	OFDN[3:0]	All 0	R	Output FIFO Data Count Indicates the number of data units in the output FIFO.

Bit	Bit Name	Initial Value	R/W	Description
11 to 7	IFDN[4:0]	All 0	R	Input FIFO Data Count Indicates the number of data units in the input FIFO.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FLF	0	R	Flush Processing Status Flag Indicates whether flush processing is in progress or not. [Clearing conditions] <ul style="list-style-type: none"> <li>When flush processing has been completed.</li> <li>When 1 has been written to the CL bit in SRCCTRL.</li> </ul> [Setting condition] <ul style="list-style-type: none"> <li>When 1 has been written to the FL bit in SRCCTRL.</li> </ul>
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	OVF	0	R/(W)*	Output Data FIFO Overwrite Interrupt Request Flag Indicates that the sampling rate conversion for the next data has been completed when there are eight units of data in the output FIFO. The sampling rate conversion stops until the output data FIFO becomes not full after the SRC output data register (SRCOD) has been read. [Clearing condition] <ul style="list-style-type: none"> <li>When 0 has been written to the OVF bit after reading OVF = 1.</li> <li>When 1 has been written to the CL bit in SRCCTRL.</li> </ul> [Setting condition] <ul style="list-style-type: none"> <li>When the sampling rate conversion for the next data has been completed when there are eight units of data in the output FIFO.</li> </ul>

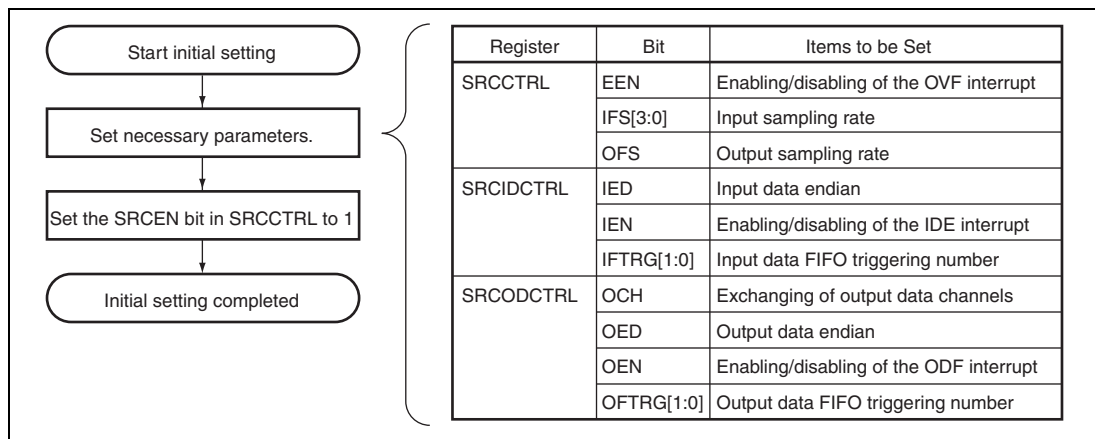
Bit	Bit Name	Initial Value	R/W	Description
1	IINT	1	R/(W)*	<p>Input Data FIFO Empty Interrupt Request Flag</p> <p>Indicates that the number of data units in the input FIFO has become equal to or smaller than the triggering number specified by the IFTRG1 and IFTRG0 bits in the SRC input data control register (SRCIDCTRL).</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When 0 has been written to the IINT bit after reading IINT = 1.</li> <li>When the DMAC has transferred data to the input FIFO resulting in the number of data units in the FIFO exceeding that of the specified triggering number.</li> </ul> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>When the number of data units in the input FIFO has become equal to or smaller than the specified triggering number.</li> <li>When 1 has been written to the CL bit in SRCCTRL.</li> </ul>
0	OINT	0	R/(W)*	<p>Output Data FIFO Full Interrupt Request Flag</p> <p>Indicates that the number of data units in the output FIFO has become equal to or greater than the triggering number specified by the OFTRG[1:0] bits in the SRC output data control register (SRCODCTRL).</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When 0 has been written to the OINT bit after reading OINT = 1.</li> <li>When the DMAC has transferred data from the output FIFO resulting in the number of data units in the FIFO being less than the specified triggering number.</li> </ul> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>When the number of data units in the output FIFO has become equal to or greater than the specified triggering number.</li> </ul>

Note: \* Only 0 can be written after having read as 1.

## 26.3 Operation

### 26.3.1 Initial Setting

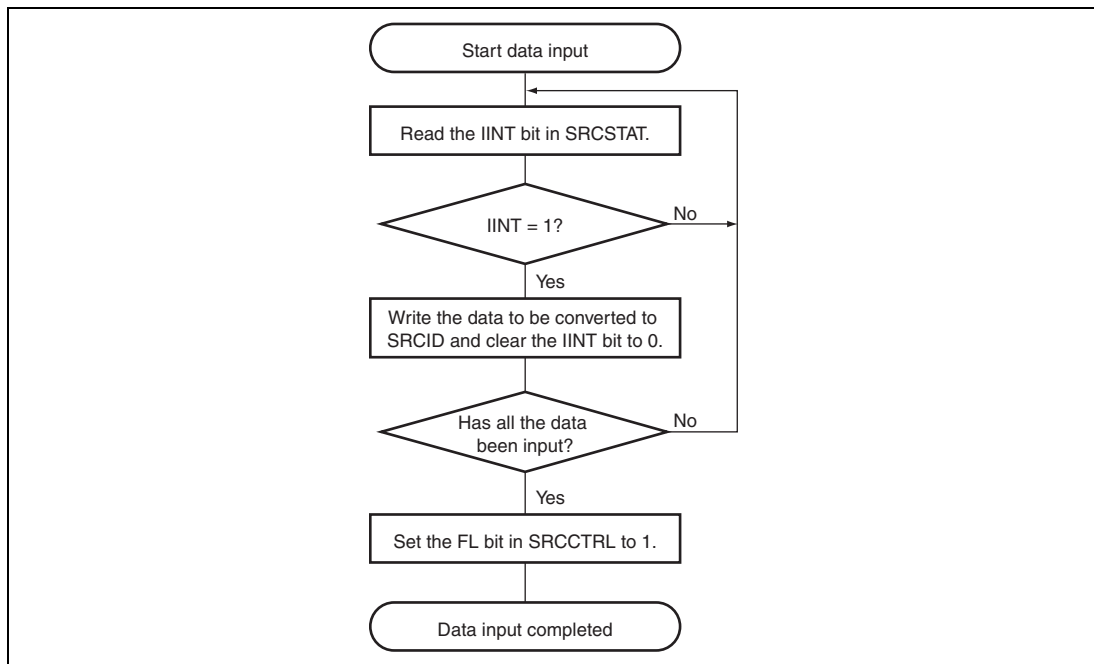
Figure 26.2 shows a sample flowchart for initial setting.



**Figure 26.2 Sample Flowchart for Initial Setting**

### 26.3.2 Data Input

Figure 26.3 is a sample flowchart for data input.



**Figure 26.3 Sample Flowchart for Data Input**

#### (1) When Interrupts are Issued to CPU

1. Set the IEN bit in SRCIDCTRL to 1.
2. Set the interrupt controller.
3. When the IINT bit in SRCSTAT is set to 1, the IDE interrupt request is issued. In the interrupt processing routine, read the IINT bit and confirm that it is 1, write data to SRCID, and write 0 to the IINT bit. Then return from the interrupt processing routine.
4. Repeat step 3 until all the data has been input, and write 1 to the FL bit in SRCCTRL.

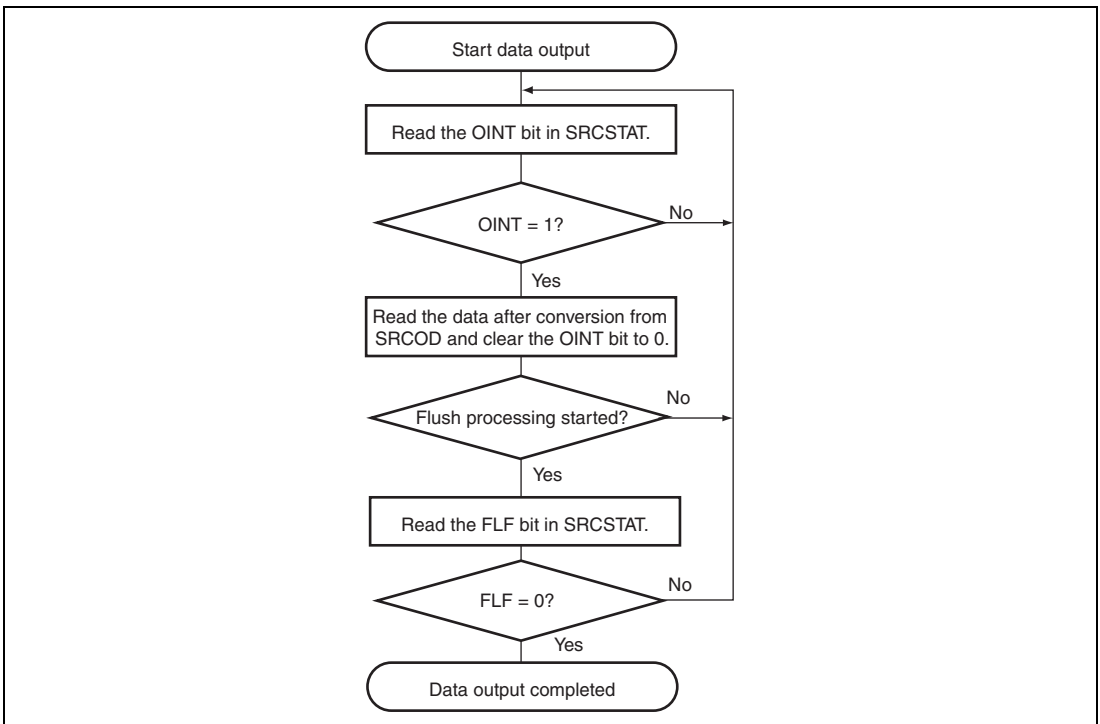


## (2) When Interrupts are Used to Activate DMAC

1. Assign IDEI of the SRC to one channel of the DMAC.
2. Set the IEN bit in SRCIDCTRL to 1.
3. When the IINT bit in SRCSTAT is set to 1, the IDE interrupt request is issued thus activating the DMAC. When the DMAC has written data to the SRCID thus resulting in the number of data units in the input data FIFO exceeding that of the triggering number specified by the IFTRG1 and IFTRG0 bits in SRCIDCTRL, the IINT bit is cleared to 0.
4. Repeat step 3 until all the data has been input, and write 1 to the FL bit in SRCCTRL.

### 26.3.3 Data Output

Figure 26.4 is a sample flowchart for data output.



**Figure 26.4 Sample Flowchart for Data Output**

### **(1) When Interrupts are Issued to CPU**

1. Set the OEN bit in SRCODCTRL to 1.
2. Set the interrupt controller.
3. When the OINT bit in SRCSTAT is set to 1, the ODF interrupt request is issued. In the interrupt processing routine, read the OINT bit and confirm that it is 1, read data from SRCOD, and write 0 to the OINT bit. Then return from the interrupt processing routine.
4. After flush processing starts, repeat step 3 until the FLF bit in SRCSTAT is read as 0.

### **(2) When Interrupts are Used to Activate DMAC**

1. Assign ODFI of the SRC to one channel of the DMAC.
2. Set the OEN bit in SRCODCTRL to 1.
3. When the OINT bit in SRCSTAT is set to 1, the ODF interrupt request is issued thus activating the DMAC. When the DMAC has read data from SRCOD thus resulting in the number of data units in the output data FIFO being less than the triggering number specified by the OFTRG1 and OFTRG0 bits in SRCODCTRL, the OINT bit is cleared to 0.
4. After flush processing starts, repeat step 3 until the FLF bit in SRCSTAT is read as 0.

## 26.4 Interrupts

The SRC has three interrupt sources: input data FIFO empty (IDEI), output data FIFO full (ODFI), and output data FIFO overwrite (OVF). Table 26.6 summarizes the interrupts.

**Table 26.6 Interrupt Requests and Generation Conditions**

Interrupt Request	Abbreviation	Interrupt Condition	DMAC Activation
Input data FIFO empty	IDEI	IINT = 1, IEN = 1, and SRCEN = 1	Possible
Output data FIFO full	ODFI	OINT = 1, OEN = 1, and SRCEN = 1	Possible
Output data FIFO overwrite	OVF	OVF = 1, EEN = 1, and SRCEN = 1	Not possible

When the interrupt condition is satisfied, the CPU executes the interrupt exception handling routine. The interrupt source flags should be cleared in the routine.

The IDEI and ODFI interrupts can activate the DMAC when the DMAC is set to allow this. When the DMAC has written data to SRCID resulting in the number of data units in the input data FIFO exceeding that of the specified triggering number, the IINT bit is cleared to 0. Similarly, when the DMAC has read data from SRCOD resulting in the number of data units in the output data FIFO being less than the specified triggering number, the OINT bit is cleared to 0.

## 26.5 Usage Note

### 26.5.1 Note on Access Register

After the FL bit in SRCCTRL is set to 1, it takes 3 cycles of peripheral bus clock until the FLF bit in SRCSTAT is set to 1. While the CPU executes the next instruction without waiting the register write completion. Accordingly, the FLF set status cannot be read by the instruction immediately. To check the execution status of flash processing, dummy read the SRCCTRL or SRCSTAT after following the SRCCTRL write instruction and wait until the FLF bit is set.

### 26.5.2 Note on Flash Processing

After set 1 to the FL bit in SRCCTRL, the SRC continues exchange processing with setting to 0 after following the destination of the data that has already input. Flash processing allowed to be executed only under the condition that the destination bit of audio data has input completely and no following data exists.

In a case that implement the exchange processing after the flash processing, clear the internal work memories with using either way listed as follows.

- Set 1 to the CL bit in SRCCTRL.
- Set 0 to the SRCEN bit in SRCCTRL and back to 1.

## Section 27 General Purpose I/O (GPIO)

### 27.1 Features

This LSI has ten general purpose I/O (GPIO) ports (A to J), which provide 77 input/output pins in total.

The port pins are multiplexed with on-chip peripheral module pins, and their functions (GPIO or on-chip peripheral module pins) can be selected.

The GPIO has the following features.

- Each port pin is a multiplexed pin, for which the pin function and pull-up MOS can be controlled individually through the corresponding port control register.
- Each port has a data register that stores data for the pins.
- GPIO interrupts are supported (for ports A and B).

Tables 27.1 and 27.2 list the multiplexed pins controlled through the GPIO registers.

**Table 27.1 Multiplexed Pins Controlled by Port Control Registers**

Port	Port Function (Related Module)	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	GPIO Interrupt
A	PA7 input/output (port)	STATUS1 output (SYSTEM)	$\overline{\text{RTS2}}$ input/output (SCIF)	—	—	PINT15 input (INTC)
	PA6 input/output (port)	STATUS0 output (SYSTEM)	$\overline{\text{CTS2}}$ input/output (SCIF)	—	—	PINT14 input (INTC)
	PA5 input/output (port)	$\overline{\text{FCE}}$ output (FLCTL)	—	—	—	PINT13 input (INTC)
	PA4 input/output (port)	$\overline{\text{FRE}}$ output (FLCTL)	—	—	—	PINT12 input (INTC)
	PA3 input/output (port)	$\overline{\text{FWE}}$ output (FLCTL)	—	—	—	PINT11 input (INTC)
	PA2 input/output (port)	TxD2 output (SCIF)	—	—	—	PINT10 input (INTC)
	PA1 input/output (port)	RxD2 input (SCIF)	—	—	—	PINT9 input (INTC)
	PA0 input/output (port)	SCK2 input/output (SCIF)	—	—	—	PINT8 input (INTC)
B	PB7 input/output (port)	A25 output (ADDRESS)	$\overline{\text{DREQ0}}$ input (DMAC)	$\overline{\text{RTS0}}$ input/output (SCIF)	—	PINT7 input (INTC)
	PB6 input/output (port)	A24 output (ADDRESS)	$\overline{\text{DACK0}}$ output (DMAC)	$\overline{\text{CTS0}}$ input/output (SCIF)	—	PINT6 input (INTC)
	PB5 input/output (port)	A23 output (ADDRESS)	$\overline{\text{DTEND0}}$ output (DMAC)	$\overline{\text{RTS1}}$ input/output (SCIF)	—	PINT5 input (INTC)
	PB4 input/output (port)	A22 output (ADDRESS)	$\overline{\text{CTS1}}$ input/output (SCIF)	—	—	PINT4 input (INTC)
	PB3 input/output (port)	A21 output (ADDRESS)	—	—	—	PINT3 input (INTC)
	PB2 input/output (port)	A20 output (ADDRESS)	—	—	—	PINT2 input (INTC)

Port	Port Function (Related Module)	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	GPIO Interrupt
B	PB1 input/output (port)	A19 output (ADDRESS)	—	—	—	PINT1 input (INTC)
	PB0 input/output (port)	A18 output (ADDRESS)	—	—	—	PINT0 input (INTC)
C	PC7 input/output (port)	AUDIO_CLK0 input (SSI)	—	—	—	—
	PC6 input/output (port)	AUDIO_CLK1 input (SSI)	—	—	—	—
	PC5 input/output (port)	AUDIO_CLK2 input (SSI)	—	—	—	—
	PC4 input/output (port)	SSIWS2 input/output (SSI)	—	—	—	—
	PC3 input/output (port)	SSISCK2 input/output (SSI)	—	—	—	—
	PC2 input/output (port)	SSIDATA2 input/output (SSI)	—	—	—	—
	PC1 input/output (port)	ASEBRKAK/ BRKACK input/output (AUD)	TCLK input (TMU)	—	—	—
	PC0 input/output (port)	FALE output (FLCTL)	—	—	—	—
D	PD7 input/output (port)	CRS input (EtherC)	IDEA1_M output (ATAPI)	—	—	—
	PD6 input/output (port)	TX_ER output (EtherC)	IDEIOWR_M output (ATAPI)	—	—	—
	PD5 input/output (port)	TX_CLK input (EtherC)	IDED15_M input/output (ATAPI)	—	—	—
	PD4 input/output (port)	TX_EN output (EtherC)	IDED0_M input/output (ATAPI)	—	—	—
	PD3 input/output (port)	MII_TXD0 output (EtherC)	SSISCK5 input/output (SSI)	IDEIORDY_M input (ATAPI)	—	—
	PD2 input/output (port)	MII_TXD1 output (EtherC)	SSIWS5 input/output (SSI)	IDEIORD_M output (ATAPI)	—	—

Port	Port Function (Related Module)	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	GPIO Interrupt
D	PD1 input/output (port)	MII_TXD2 output (EtherC)	AUDIO_CLK5 input (SSI)	IDEINT_M input (ATAPI)	—	—
	PD0 input/output (port)	MII_TXD3 output (EtherC)	SSIDATA5 input/output (SSI)	IODACK_M output (ATAPI)	—	—
E	PE7 input/output (port)	COL input (EtherC)	IDEA2_M output (ATAPI)	—	—	—
	PE6 input/output (port)	RX_ER input (EtherC)	IODREQ_M input (ATAPI)	—	—	—
	PE5 input/output (port)	RX_CLK input (EtherC)	IDED1_M input/output (ATAPI)	—	—	—
	PE4 input/output (port)	RX_DV input (EtherC)	IDED14_M input/output (ATAPI)	—	—	—
	PE3 input/output (port)	MII_RXD0 input (EtherC)	SSIWS4 input/output (SSI)	IDED2_M input/output (ATAPI)	—	—
	PE2 input/output (port)	MII_RXD1 input (EtherC)	SSISCK4 input/output (SSI)	IDED13_M input/output (ATAPI)	—	—
	PE1 input/output (port)	MII_RXD2 input (EtherC)	SSIDATA4 input/output (SSI)	IDED3_M input/output (ATAPI)	—	—
	PE0 input/output (port)	MII_RXD3 input (EtherC)	AUDIO_CLK4 input (SSI)	IDED12_M input/output (ATAPI)	—	—
F	PF7 input/output (port)	D32 input/output (DATA)	—	—	—	—
	PF6 input/output (port)	D33 input/output (DATA)	—	—	—	—
	PF5 input/output (port)	D34 input/output (DATA)	—	—	—	—
	PF4 input/output (port)	EXOUT output (EtherC)	IDECS1_M output (ATAPI)	—	—	—
	PF3 input/output (port)	LNKSTA input (EtherC)	IDECS0_M output (ATAPI)	—	—	—
	PF2 input/output (port)	WOL output (EtherC)	IDEA0_M output (ATAPI)	—	—	—



Port	Port Function (Related Module)	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	GPIO Interrupt
F	PF1 input/output (port)	MDIO input/output (EtherC)	IDED11_M input/output (ATAPI)	—	—	—
	PF0 input/output (port)	MDC output (EtherC)	IDED4_M input/output (ATAPI)	—	—	—
G	PG7 input/output (port)	LCD_DATA15 output (LCDC)	DR3 output (VDC2)	—	—	—
	PG6 input/output (port)	LCD_DATA14 output (LCDC)	DR2 output (VDC2)	—	—	—
	PG5 input/output (port)	LCD_DATA13 output (LCDC)	DR1 output (VDC2)	—	—	—
	PG4 input/output (port)	LCD_DATA12 output (LCDC)	DR0 output (VDC2)	—	—	—
	PG3 input/output (port)	LCD_DATA11 output (LCDC)	DG5 output (VDC2)	—	—	—
	PG2 input/output (port)	LCD_DATA10 output (LCDC)	DG4 output (VDC2)	—	—	—
	PG1 input/output (port)	LCD_DATA9 output (LCDC)	DG3 output (VDC2)	—	—	—
	PG0 input/output (port)	LCD_DATA8 output (LCDC)	DG2 output (VDC2)	—	—	—
H	PH7 input/output (port)	AUDIO_CLK3 input (SSI)	—	—	—	—
	PH6 input/output (port)	SSIWS3 input/output (SSI)	—	—	—	—
	PH5 input/output (port)	SSISCK3 input/output (SSI)	—	—	—	—
	PH4 input/output (port)	SSIDATA3 input/output (SSI)	—	—	—	—
	PH3 input/output (port)	LCD_CL2 output (LCDC)	DE_V output (VDC2)	—	—	—
	PH2 input/output (port)	LCD_DON output (LCDC)	DCLKOUT output (VDC2)	—	—	—

Port	Port Function (Related Module)	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	GPIO Interrupt
H	PH1 input/output (port)	LCD_VCP_WC output (LCDC)	DR4 output (VDC2)	—	—	—
	PH0 input/output (port)	LCD_VEP_WC output (LCDC)	DR5 output (VDC2)	—	—	—
I	PI4 input/output (port)	LCD_DATA7 output (LCDC)	DG1 output (VDC2)	BT_DATA7 output (VDC2)	—	—
	PI3 input/output (port)	LCD_DATA6 output (LCDC)	DG0 output (VDC2)	BT_DATA6 output (VDC2)	—	—
	PI2 input/output (port)	LCD_DATA5 output (LCDC)	DB5 output (VDC2)	BT_DATA5 output (VDC2)	—	—
	PI1 input/output (port)	LCD_DATA4 output (LCDC)	DB4 output (VDC2)	BT_DATA4 output (VDC2)	—	—
	PI0 input/output (port)	PI0 input/output (port)	COM/CDE output (VDC2)	—	—	—
J	PJ7 input/output (port)	PJ7 input/output (port)	—	—	IDED10_M input/output (ATAPI)	—
	PJ6 input/output (port)	PJ6 input/output (port)	—	—	IDED5_M input/output (ATAPI)	—
	PJ5 input/output (port)	PJ5 input/output (port)	—	—	IDED9_M input/output (ATAPI)	—
	PJ4 input/output (port)	PJ4 input/output (port)	—	—	IDED6_M input/output (ATAPI)	—
	PJ3 input/output (port)	PJ3 input/output (port)	—	—	IDED7_M input/output (ATAPI)	—
	PJ2 input/output (port)	PJ2 input/output (port)	—	—	IDED8_M input/output (ATAPI)	—
	PJ1 input/output (port)	PJ1 input/output (port)	—	—	IDERST_M output (ATAPI)	—
	PJ0 input/output (port)	PJ0 input/output (port)	—	—	DIRECTION_ M output (ATAPI)	—

Note: In the table, the pin functions in the shaded column can be used immediately after a reset.

**Table 27.2 Multiplexed Pins Controlled by Pin Select Registers**

Register	Bit	Function 1	Function 2	Function 3	Function 4
PTSEL_K	PTSEL_K7 [1:0]	WDTOVF output (SYSTEM)	IRQ1 input (INT)	AUDCK output (AUD)	DACK1 output (DMAC)
	PTSEL_K6 [1:0]	SCK0 input/output (SCIF)	AUDSYNC output (AUD)	FCLE output (FLCTL)	—
	PTSEL_K5	SCK1 input/output (SCIF)	FR/B input (FLCTL)	—	—
	PTSEL_K4 [1:0]	LCD_DATA0 output (LCDC)	DB0 output (VDC2)	BT_DATA0 output (VDC2)	—
	PTSEL_K3 [1:0]	LCD_CL1 output (LCDC)	HSYNC/SPL* input/output (VDC2)	BT_HSYNC output (VDC2)	—
	PTSEL_K2 [1:0]	LCD_CLK input (LCDC)	DCLKIN input (VDC2)	—	—
	PTSEL_K1 [1:0]	LCD_FLM output (LCDC)	VSYNC/SPS* input/output (VDC2)	BT_VSYNC output (VDC2)	—
	PTSEL_K0 [1:0]	LCD_M_DISP output (LCDC)	DE_H/DE_C output (VDC2)	BT_DE_C output (VDC2)	—
PTSEL_P	PTSEL_P11	RXD0 input (SCIF)	AUDATA0 output (AUD)	—	—
	PTSEL_P10	TXD0 output (SCIF)	AUDATA1 output (AUD)	—	—
	PTSEL_P9	RXD1 input (SCIF)	AUDATA2 output (AUD)	—	—
	PTSEL_P8	TXD1 output (SCIF)	AUDATA3 output (AUD)	—	—
PTSEL_R	PTSEL_R15	D63 input/output (DATA)	IDED1 input/output (ATAPI)	—	—
	PTSEL_R14	D62 input/output (DATA)	IDED0 input/output (ATAPI)	—	—
	PTSEL_R13	D61 input/output (DATA)	IDED3 input/output (ATAPI)	—	—
	PTSEL_R12	D60 input/output (DATA)	IDED2 input/output (ATAPI)	—	—
	PTSEL_R11	D59 input/output (DATA)	IDED5 input/output (ATAPI)	—	—

Register	Bit	Function 1	Function 2	Function 3	Function 4
PTSEL_R	PTSEL_R10	D58 input/output (DATA)	IDED4 input/output (ATAPI)	—	—
	PTSEL_R9	D57 input/output (DATA)	IDED7 input/output (ATAPI)	—	—
	PTSEL_R8	D56 input/output (DATA)	IDED6 input/output (ATAPI)	—	—
	PTSEL_R7	D55 input/output (DATA)	DIRECTION output (ATAPI)	—	—
	PTSEL_R6	D54 input/output (DATA)	IDERST $\overline{\text{ }}$ output (ATAPI)	—	—
	PTSEL_R5	D53 input/output (DATA)	IDED8 input/output (ATAPI)	—	—
	PTSEL_R4	D52 input/output (DATA)	IDED9 input/output (ATAPI)	—	—
	PTSEL_R3	D51 input/output (DATA)	IDED10 input/output (ATAPI)	—	—
	PTSEL_R2	D50 input/output (DATA)	IDED11 input/output (ATAPI)	—	—
	PTSEL_R1	D49 input/output (DATA)	IDED12 input/output (ATAPI)	—	—
	PTSEL_R0	D48 input/output (DATA)	IDED13 input/output (ATAPI)	—	—
PTSEL_S	PTSEL_S15	IRQ0 input (INT)	DTEND1 $\overline{\text{ }}$ output (DMAC)	—	—
	PTSEL_S14	IRQOUT $\overline{\text{ }}$ output (INT)	DREQ1 $\overline{\text{ }}$ input (DMAC)	—	—
	PTSEL_S13	D47 input/output (DATA)	IDECS0 $\overline{\text{ }}$ output (ATAPI)	—	—
	PTSEL_S12	D46 input/output (DATA)	IDECS1 $\overline{\text{ }}$ output (ATAPI)	—	—
	PTSEL_S11	D45 input/output (DATA)	IDODACK $\overline{\text{ }}$ output (ATAPI)	—	—
	PTSEL_S10	D44 input/output (DATA)	IDEINT input (ATAPI)	—	—
	PTSEL_S9	D43 input/output (DATA)	IDEIORDY input (ATAPI)	—	—
	PTSEL_S8	D42 input/output (DATA)	IDEIORD $\overline{\text{ }}$ output (ATAPI)	—	—

Register	Bit	Function 1	Function 2	Function 3	Function 4
PTSEL_S	PTSEL_S7	D41 input/output (DATA)	IODREQ input (ATAPI)	—	—
	PTSEL_S6	D40 input/output (DATA)	IDEIOWR output (ATAPI)	—	—
	PTSEL_S5	D39 input/output (DATA)	IDED14 input/output (ATAPI)	—	—
	PTSEL_S4	D38 input/output (DATA)	IDED15 input/output (ATAPI)	—	—
	PTSEL_S3	D37 input/output (DATA)	IDEA1 output (ATAPI)	—	—
	PTSEL_S2	D36 input/output (DATA)	IDEA2 output (ATAPI)	—	—
	PTSEL_S1	D35 input/output (DATA)	IDEA0 output (ATAPI)	—	—

Note: In the table, the pin functions in the shaded column can be used immediately after a reset.

\* This pin function switches over input/output functions in a special select register.

## 27.2 Register Descriptions

Table 27.3 shows the GPIO register configuration. Table 27.4 shows the register states in each operating mode.

**Table 27.3 Register Configuration**

Register Name	Abbreviation	R/W	P4 Area Address* <sup>1</sup>	Area 7 Address* <sup>1</sup>	Access Size* <sup>2</sup>
Port A control register	PTIO_A	R/W	H'FFF1 0000	H'1FF1 0000	16
Port B control register	PTIO_B	R/W	H'FFF1 0004	H'1FF1 0004	16
Port C control register	PTIO_C	R/W	H'FFF1 0008	H'1FF1 0008	16
Port D control register	PTIO_D	R/W	H'FFF1 000C	H'1FF1 000C	16
Port E control register	PTIO_E	R/W	H'FFF1 0010	H'1FF1 0010	16
Port F control register	PTIO_F	R/W	H'FFF1 0014	H'1FF1 0014	16
Port G control register	PTIO_G	R/W	H'FFF1 0018	H'1FF1 0018	16
Port H control register	PTIO_H	R/W	H'FFF1 001C	H'1FF1 001C	16
Port I control register	PTIO_I	R/W	H'FFF1 0020	H'1FF1 0020	16
Port J control register	PTIO_J	R/W	H'FFF1 0024	H'1FF1 0024	16
Port A data register	PTDAT_A	R/W	H'FFF1 0040	H'1FF1 0040	16
Port B data register	PTDAT_B	R/W	H'FFF1 0044	H'1FF1 0044	16
Port C data register	PTDAT_C	R/W	H'FFF1 0048	H'1FF1 0048	16
Port D data register	PTDAT_D	R/W	H'FFF1 004C	H'1FF1 004C	16
Port E data register	PTDAT_E	R/W	H'FFF1 0050	H'1FF1 0050	16
Port F data register	PTDAT_F	R/W	H'FFF1 0054	H'1FF1 0054	16
Port G data register	PTDAT_G	R/W	H'FFF1 0058	H'1FF1 0058	16
Port H data register	PTDAT_H	R/W	H'FFF1 005C	H'1FF1 005C	16
Port I data register	PTDAT_I	R/W	H'FFF1 0060	H'1FF1 0060	16
Port J data register	PTDAT_J	R/W	H'FFF1 0064	H'1FF1 0064	16
Input-pin pull-up control register	PTPUL_SPCL	R/W	H'FFF1 00E0	H'1FF1 00E0	16
Pin select register A	PTSEL_A	R/W	H'FFF1 0080	H'1FF1 0080	16
Pin select register B	PTSEL_B	R/W	H'FFF1 0084	H'1FF1 0084	16
Pin select register C	PTSEL_C	R/W	H'FFF1 0088	H'1FF1 0088	16

Register Name	Abbreviation	R/W	P4 Area Address* <sup>1</sup>	Area 7 Address* <sup>1</sup>	Access Size* <sup>2</sup>
Pin select register D	PTSEL_D	R/W	H'FFF1 008C	H'1FF1 008C	16
Pin select register E	PTSEL_E	R/W	H'FFF1 0090	H'1FF1 0090	16
Pin select register F	PTSEL_F	R/W	H'FFF1 0094	H'1FF1 0094	16
Pin select register G	PTSEL_G	R/W	H'FFF1 0098	H'1FF1 0098	16
Pin select register H	PTSEL_H	R/W	H'FFF1 009C	H'1FF1 009C	16
Pin select register I	PTSEL_I	R/W	H'FFF1 00A0	H'1FF1 00A0	16
Pin select register J	PTSEL_J	R/W	H'FFF1 00A4	H'1FF1 00A4	16
Pin select register K	PTSEL_K	R/W	H'FFF1 00A8	H'1FF1 00A8	16
Pin select register P	PTSEL_P	R/W	H'FFF1 00AC	H'1FF1 00AC	16
Pin select register R	PTSEL_R	R/W	H'FFF1 00B0	H'1FF1 00B0	16
Pin select register S	PTSEL_S	R/W	H'FFF1 00B4	H'1FF1 00B4	16
Hi-Z register A	PTHIZ_A	R/W	H'FFF1 00E8	H'1FF1 00E8	16
Hi-Z register B	PTHIZ_B	R/W	H'FFF1 00EC	H'1FF1 00EC	16
Special select register	PTSEL_SPCL	R/W	H'FFF1 00F0	H'1FF1 00F0	16

Notes: 1. Use a P4 area address to access a register in the P4 area in the virtual address space.  
 Use an area 7 address to access a register from area 7 in the physical address space through the TLB.

2. The registers should always be read or written to in 16 bits.

**Table 27.4 Register States in Each Operating Mode**

<b>Register name</b>	<b>Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
Port A control register	PTIO_A	H'0000	Retained	Retained
Port B control register	PTIO_B	H'0000	Retained	Retained
Port C control register	PTIO_C	H'0000	Retained	Retained
Port D control register	PTIO_D	H'0000	Retained	Retained
Port E control register	PTIO_E	H'0000	Retained	Retained
Port F control register	PTIO_F	H'0000	Retained	Retained
Port G control register	PTIO_G	H'0000	Retained	Retained
Port H control register	PTIO_H	H'0000	Retained	Retained
Port I control register	PTIO_I	H'0002	Retained	Retained
Port J control register	PTIO_J	H'AAAA	Retained	Retained
Port A data register	PTDAT_A	H'0000	Retained	Retained
Port B data register	PTDAT_B	H'0000	Retained	Retained
Port C data register	PTDAT_C	H'0000	Retained	Retained
Port D data register	PTDAT_D	H'0000	Retained	Retained
Port E data register	PTDAT_E	H'0000	Retained	Retained
Port F data register	PTDAT_F	H'0000	Retained	Retained
Port G data register	PTDAT_G	H'0000	Retained	Retained
Port H data register	PTDAT_H	H'0000	Retained	Retained
Port I data register	PTDAT_I	H'0000	Retained	Retained
Port J data register	PTDAT_J	H'0000	Retained	Retained
Input-pin pull-up control register	PTPUL_SPCL	H'0000	Retained	Retained
Pin select register A	PTSEL_A	H'0000	Retained	Retained
Pin select register B	PTSEL_B	H'0000	Retained	Retained
Pin select register C	PTSEL_C	H'0000	Retained	Retained
Pin select register D	PTSEL_D	H'0000	Retained	Retained
Pin select register E	PTSEL_E	H'0000	Retained	Retained
Pin select register F	PTSEL_F	H'0000	Retained	Retained
Pin select register G	PTSEL_G	H'0000	Retained	Retained
Pin select register H	PTSEL_H	H'0000	Retained	Retained



Register name	Abbreviation	Power-on Reset	Sleep	Standby
Pin select register I	PTSEL_I	H'0000	Retained	Retained
Pin select register J	PTSEL_J	H'0000	Retained	Retained
Pin select register K	PTSEL_K	H'0000	Retained	Retained
Pin select register P	PTSEL_P	H'0000	Retained	Retained
Pin select register R	PTSEL_R	H'0000	Retained	Retained
Pin select register S	PTSEL_S	H'0000	Retained	Retained
Hi-Z register A	PTHIZ_A	H'0000	Retained	Retained
Hi-Z register B	PTHIZ_B	H'0000	Retained	Retained
Special select register	PTSEL_SPCL	H'0000	Retained	Retained

### 27.2.1 Port A Control Register (PTIO\_A)

PTIO\_A is a 16-bit readable/writable register that controls each pin function and input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_A7[1:0]		PTIO_A6[1:0]		PTIO_A5[1:0]		PTIO_A4[1:0]		PTIO_A3[1:0]		PTIO_A2[1:0]		PTIO_A1[1:0]		PTIO_A0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTIO_A7[1:0]	00	R/W	PTA7 Mode 00: Other functions (STATUS1 and $\overline{\text{RTS2}}$ ) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
13, 12	PTIO_A6[1:0]	00	R/W	PTA6 Mode 00: Other functions (STATUS0 and $\overline{\text{CTS2}}$ ) 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
11, 10	PTIO_A5[1:0]	00	R/W	PTA5 Mode 00: Other functions ( $\overline{FCE}$ ) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
9, 8	PTIO_A4[1:0]	00	R/W	PTA4 Mode 00: Other functions ( $\overline{FRE}$ ) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
7, 6	PTIO_A3[1:0]	00	R/W	PTA3 Mode 00: Other functions ( $\overline{FWE}$ ) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
5, 4	PTIO_A2[1:0]	00	R/W	PTA2 Mode 00: Other functions (TxD2) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
3, 2	PTIO_A1[1:0]	00	R/W	PTA1 Mode 00: Other functions (RxD2) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
1, 0	PTIO_A0[1:0]	00	R/W	PTA0 Mode 00: Other functions (SCK2) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)

## 27.2.2 Port B Control Register (PTIO\_B)

PTIO\_B is a 16-bit readable/writable register that controls each pin function and input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_B7[1:0]		PTIO_B6[1:0]		PTIO_B5[1:0]		PTIO_B4[1:0]		PTIO_B3[1:0]		PTIO_B2[1:0]		PTIO_B1[1:0]		PTIO_B0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTIO_B7[1:0]	00	R/W	PTB7 Mode 00: Other functions (A25, $\overline{\text{DREQ0}}$ , and $\overline{\text{RTS0}}$ ) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
13, 12	PTIO_B6[1:0]	00	R/W	PTB6 Mode 00: Other functions (A24, $\overline{\text{DACK0}}$ , and $\overline{\text{CTS0}}$ ) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
11, 10	PTIO_B5[1:0]	00	R/W	PTB5 Mode 00: Other functions (A23, $\overline{\text{DTEND0}}$ , and $\overline{\text{RTS1}}$ ) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
9, 8	PTIO_B4[1:0]	00	R/W	PTB4 Mode 00: Other functions (A22 and $\overline{\text{CTS1}}$ ) 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, 6	PTIO_B3[1:0]	00	R/W	PTB3 Mode 00: Other functions (A21) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
5, 4	PTIO_B2[1:0]	00	R/W	PTB2 Mode 00: Other functions (A20) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
3, 2	PTIO_B1[1:0]	00	R/W	PTB1 Mode 00: Other functions (A19) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
1, 0	PTIO_B0[1:0]	00	R/W	PTB0 Mode 00: Other functions (A18) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)

### 27.2.3 Port C Control Register (PTIO\_C)

PTIO\_C is a 16-bit readable/writable register that controls each pin function and input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_C7[1:0]		PTIO_C6[1:0]		PTIO_C5[1:0]		PTIO_C4[1:0]		PTIO_C3[1:0]		PTIO_C2[1:0]		PTIO_C1[1:0]		PTIO_C0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTIO_C7[1:0]	00	R/W	PTC7 Mode 00: Other functions (AUDIO_CLK0) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
13, 12	PTIO_C6[1:0]	00	R/W	PTC6 Mode 00: Other functions (AUDIO_CLK1) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
11, 10	PTIO_C5[1:0]	00	R/W	PTC5 Mode 00: Other functions (AUDIO_CLK2) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
9, 8	PTIO_C4[1:0]	00	R/W	PTC4 Mode 00: Other functions (SSIWS2) 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, 6	PTIO_C3[1:0]	00	R/W	PTC3 Mode 00: Other functions (SSISCK2) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
5, 4	PTIO_C2[1:0]	00	R/W	PTC2 Mode 00: Other functions (SSIDATA2) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
3, 2	PTIO_C1[1:0]	00	R/W	PTC1 Mode 00: Other functions ( $\overline{\text{ASEBRKAK}}$ /BRKACK and TCLK) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
1, 0	PTIO_C0[1:0]	00	R/W	PTC0 Mode 00: Other functions (FALE) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)

### 27.2.4 Port D Control Register (PTIO\_D)

PTIO\_D is a 16-bit readable/writable register that controls each pin function and input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_D7[1:0]		PTIO_D6[1:0]		PTIO_D5[1:0]		PTIO_D4[1:0]		PTIO_D3[1:0]		PTIO_D2[1:0]		PTIO_D1[1:0]		PTIO_D0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTIO_D7[1:0]	00	R/W	PTD7 Mode 00: Other functions (CRS and IDEA1_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
13, 12	PTIO_D6[1:0]	00	R/W	PTD6 Mode 00: Other functions (TX_ER and IDEIOWR_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
11, 10	PTIO_D5[1:0]	00	R/W	PTD5 Mode 00: Other functions (TX_CLK and IDDED15_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
9, 8	PTIO_D4[1:0]	00	R/W	PTD4 Mode 00: Other functions (TX_EN and IDEDO_M) 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, 6	PTIO_D3[1:0]	00	R/W	PTD3 Mode 00: Other functions (MII_TXD0, SSISCK5, and IDEIORDY_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
5, 4	PTIO_D2[1:0]	00	R/W	PTD2 Mode 00: Other functions (MII_TXD1, SSIWS5, and IDEIORD_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
3, 2	PTIO_D1[1:0]	00	R/W	PTD1 Mode 00: Other functions (MII_TXD2, AUDIO_CLK5, and IDEINT_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
1, 0	PTIO_D0[1:0]	00	R/W	PTD0 Mode 00: Other functions (MII_TXD3, SSIDATA5, and IODACK_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)



### 27.2.5 Port E Control Register (PTIO\_E)

PTIO\_E is a 16-bit readable/writable register that controls each pin function and input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_E7[1:0]		PTIO_E6[1:0]		PTIO_E5[1:0]		PTIO_E4[1:0]		PTIO_E3[1:0]		PTIO_E2[1:0]		PTIO_E1[1:0]		PTIO_E0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTIO_E7[1:0]	00	R/W	PTE7 Mode 00: Other functions (COL and IDEA2_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
13, 12	PTIO_E6[1:0]	00	R/W	PTE6 Mode 00: Other functions (RX_ER and IODREQ_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
11, 10	PTIO_E5[1:0]	00	R/W	PTE5 Mode 00: Other functions (RX_CLK and IDDED1_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
9, 8	PTIO_E4[1:0]	00	R/W	PTE4 Mode 00: Other functions (RX_DV and IDDED14_M) 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, 6	PTIO_E3[1:0]	00	R/W	PTE3 Mode 00: Other functions (MII_RXD0, SSIWS4, and IDED2_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
5, 4	PTIO_E2[1:0]	00	R/W	PTE2 Mode 00: Other functions (MII_RXD1, SSISCK4, and IDED13_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
3, 2	PTIO_E1[1:0]	00	R/W	PTE1 Mode 00: Other functions (MII_RXD2, SSIDATA4, and IDED3_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
1, 0	PTIO_E0[1:0]	00	R/W	PTE0 Mode 00: Other functions (MII_RXD3, AUDIO_CLK4, and IDED12_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)

### 27.2.6 Port F Control Register (PTIO\_F)

PTIO\_F is a 16-bit readable/writable register that controls each pin function and input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_F7[1:0]		PTIO_F6[1:0]		PTIO_F5[1:0]		PTIO_F4[1:0]		PTIO_F3[1:0]		PTIO_F2[1:0]		PTIO_F1[1:0]		PTIO_F0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTIO_F7[1:0]	00	R/W	PTF7 Mode 00: Other functions (D32) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
13, 12	PTIO_F6[1:0]	00	R/W	PTF6 Mode 00: Other functions (D33) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
11, 10	PTIO_F5[1:0]	00	R/W	PTF5 Mode 00: Other functions (D34) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
9, 8	PTIO_F4[1:0]	00	R/W	PTF4 Mode 00: Other functions (EXOUT and IDECS1_M) 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, 6	PTIO_F3[1:0]	00	R/W	PTF3 Mode 00: Other functions (LNKSTA and IDECS0_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
5, 4	PTIO_F2[1:0]	00	R/W	PTF2 Mode 00: Other functions (WOL and IDEA0_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
3, 2	PTIO_F1[1:0]	00	R/W	PTF1 Mode 00: Other functions (MDIO and IDED11_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
1, 0	PTIO_F0[1:0]	00	R/W	PTF0 Mode 00: Other functions (MDC and IDED4_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)

## 27.2.7 Port G Control Register (PTIO\_G)

PTIO\_G is a 16-bit readable/writable register that controls each pin function and input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_G7[1:0]		PTIO_G6[1:0]		PTIO_G5[1:0]		PTIO_G4[1:0]		PTIO_G3[1:0]		PTIO_G2[1:0]		PTIO_G1[1:0]		PTIO_G0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTIO_G7[1:0]	00	R/W	PTG7 Mode 00: Other functions (LCD_DATA15 and DR3) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
13, 12	PTIO_G6[1:0]	00	R/W	PTG6 Mode 00: Other functions (LCD_DATA14 and DR2) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
11, 10	PTIO_G5[1:0]	00	R/W	PTG5 Mode 00: Other functions (LCD_DATA13 and DR1) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
9, 8	PTIO_G4[1:0]	00	R/W	PTG4 Mode 00: Other functions (LCD_DATA12 and DR0) 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, 6	PTIO_G3[1:0]	00	R/W	PTG3 Mode 00: Other functions (LCD_DATA11 and DG5) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
5, 4	PTIO_G2[1:0]	00	R/W	PTG2 Mode 00: Other functions (LCD_DATA10 and DG4) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
3, 2	PTIO_G1[1:0]	00	R/W	PTG1 Mode 00: Other functions (LCD_DATA9 and DG3) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
1, 0	PTIO_G0[1:0]	00	R/W	PTG0 Mode 00: Other functions (LCD_DATA8 and DG2) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)

### 27.2.8 Port H Control Register (PTIO\_H)

PTIO\_H is a 16-bit readable/writable register that controls each pin function and input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_H7[1:0]		PTIO_H6[1:0]		PTIO_H5[1:0]		PTIO_H4[1:0]		PTIO_H3[1:0]		PTIO_H2[1:0]		PTIO_H1[1:0]		PTIO_H0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTIO_H7[1:0]	00	R/W	PTH7 Mode 00: Other functions (AUDIO_CLK3) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
13, 12	PTIO_H6[1:0]	00	R/W	PTH6 Mode 00: Other functions (SSIWS3) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
11, 10	PTIO_H5[1:0]	00	R/W	PTH5 Mode 00: Other functions (SSISCK3) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
9, 8	PTIO_H4[1:0]	00	R/W	PTH4 Mode 00: Other functions (SSIDATA3) 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, 6	PTIO_H3[1:0]	00	R/W	PTH3 Mode 00: Other functions (LCD_CL2 and DE_V) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
5, 4	PTIO_H2[1:0]	00	R/W	PTH2 Mode 00: Other functions (LCD_DON and DCLKOUT) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
3, 2	PTIO_H1[1:0]	00	R/W	PTH1 Mode 00: Other functions (LCD_VCP_WC and DR4) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
1, 0	PTIO_H0[1:0]	00	R/W	PTH0 Mode 00: Other functions (LCD_VEP_WC and DR5) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)



### 27.2.9 Port I Control Register (PTIO\_I)

PTIO\_I is a 16-bit readable/writable register that controls each pin function and input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PTIO_I4[1:0]		PTIO_I3[1:0]		PTIO_I2[1:0]		PTIO_I1[1:0]		PTIO_I0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
9, 8	PTIO_I4[1:0]	00	R/W	PTI4 Mode  00: Other functions (LCD_DATA7, DG1, and BT_DATA7) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
7, 6	PTIO_I3[1:0]	00	R/W	PTI3 Mode  00: Other functions (LCD_DATA6, DG0, and BT_DATA6) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
5, 4	PTIO_I2[1:0]	00	R/W	PTI2 Mode  00: Other functions (LCD_DATA5, DB5, and BT_DATA5) 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
3, 2	PTIO_I1[1:0]	00	R/W	PTI1 Mode 00: Other functions (LCD_DATA4, DB4, and BT_DATA4) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
1, 0	PTIO_I0[1:0]	10	R/W	PTI0 Mode 00: Other functions (COM/CDE) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)

### 27.2.10 Port J Control Register (PTIO\_J)

PTIO\_J is a 16-bit readable/writable register that controls each pin function and input pull-up MOS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_J7[1:0]		PTIO_J6[1:0]		PTIO_J5[1:0]		PTIO_J4[1:0]		PTIO_J3[1:0]		PTIO_J2[1:0]		PTIO_J1[1:0]		PTIO_J0[1:0]	
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTIO_J7[1:0]	10	R/W	PTJ7 Mode 00: Other functions (IDED10_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
13, 12	PTIO_J6[1:0]	10	R/W	PTJ6 Mode 00: Other functions (IDED5_M) 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
11, 10	PTIO_J5[1:0]	10	R/W	PTJ5 Mode 00: Other functions (IDED9_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
9, 8	PTIO_J4[1:0]	10	R/W	PTJ4 Mode 00: Other functions (IDED6_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
7, 6	PTIO_J3[1:0]	10	R/W	PTJ3 Mode 00: Other functions (IDED7_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
5, 4	PTIO_J2[1:0]	10	R/W	PTJ2 Mode 00: Other functions (IDED8_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
3, 2	PTIO_J1[1:0]	10	R/W	PTJ1 Mode 00: Other functions ( $\overline{\text{IDERST\_M}}$ ) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)
1, 0	PTIO_J0[1:0]	10	R/W	PTJ0 Mode 00: Other functions (DIRECTION_M) 01: Port output 10: Port input (Pull-up MOS: Off) 11: Port input (Pull-up MOS: On)

### 27.2.11 Port A Data Register (PTDAT\_A)

PTDAT\_A is a 16-bit readable/writable register that stores port A data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_A7	PTDAT_A6	PTDAT_A5	PTDAT_A4	PTDAT_A3	PTDAT_A2	PTDAT_A1	PTDAT_A0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  The lower 8-bit value is always read from these bits. The write value should always be 0.
7	PTDAT_A7	0	R/W	Each bit stores output data of a pin used as a general output port.  When the pin is used as a general output port, if the port is read, the value of the corresponding bit in this register will be read.  When the pin is used as a general input port, if this register is read, the status of the corresponding pin will be read.
6	PTDAT_A6	0	R/W	
5	PTDAT_A5	0	R/W	
4	PTDAT_A4	0	R/W	
3	PTDAT_A3	0	R/W	
2	PTDAT_A2	0	R/W	
1	PTDAT_A1	0	R/W	
0	PTDAT_A0	0	R/W	

## 27.2.12 Port B Data Register (PTDAT\_B)

PTDAT\_B is a 16-bit readable/writable register that stores port B data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_B7	PTDAT_B6	PTDAT_B5	PTDAT_B4	PTDAT_B3	PTDAT_B2	PTDAT_B1	PTDAT_B0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  The lower 8-bit value is always read from these bits. The write value should always be 0.
7	PTDAT_B7	0	R/W	Each bit stores output data of a pin used as a general output port.
6	PTDAT_B6	0	R/W	
5	PTDAT_B5	0	R/W	When the pin is used as a general output port, if the port is read, the value of the
4	PTDAT_B4	0	R/W	corresponding bit in this register will be read.
3	PTDAT_B3	0	R/W	When the pin is used as a general input port, if this register is read, the status of the
2	PTDAT_B2	0	R/W	corresponding pin will be read.
1	PTDAT_B1	0	R/W	When the pin is set to a function other than the general port, if it is used as an input pin,
0	PTDAT_B0	0	R/W	the pin status will be read from the corresponding bit in this register and writing to the bit is ignored; for an output pin, an undefined value will be read from the bit and writing to the bit is ignored.

### 27.2.13 Port C Data Register (PTDAT\_C)

PTDAT\_C is a 16-bit readable/writable register that stores port C data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_C7	PTDAT_C6	PTDAT_C5	PTDAT_C4	PTDAT_C3	PTDAT_C2	PTDAT_C1	PTDAT_C0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  The lower 8-bit value is always read from these bits. The write value should always be 0.
7	PTDAT_C7	0	R/W	Each bit stores output data of a pin used as a general output port.
6	PTDAT_C6	0	R/W	When the pin is used as a general output port, if the port is read, the value of the corresponding bit in this register will be read.
5	PTDAT_C5	0	R/W	When the pin is used as a general input port, if this register is read, the status of the corresponding pin will be read.
4	PTDAT_C4	0	R/W	When the pin is set to a function other than the general port, if it is used as an input pin, the pin status will be read from the corresponding bit in this register and writing to the bit is ignored; for an output pin, an undefined value will be read from the bit and writing to the bit is ignored.
3	PTDAT_C3	0	R/W	
2	PTDAT_C2	0	R/W	
1	PTDAT_C1	0	R/W	
0	PTDAT_C0	0	R/W	

### 27.2.14 Port D Data Register (PTDAT\_D)

PTDAT\_D is a 16-bit readable/writable register that stores port D data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_D7	PTDAT_D6	PTDAT_D5	PTDAT_D4	PTDAT_D3	PTDAT_D2	PTDAT_D1	PTDAT_D0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  The lower 8-bit value is always read from these bits. The write value should always be 0.
7	PTDAT_D7	0	R/W	Each bit stores output data of a pin used as a general output port.
6	PTDAT_D6	0	R/W	
5	PTDAT_D5	0	R/W	When the pin is used as a general output port, if the port is read, the value of the
4	PTDAT_D4	0	R/W	corresponding bit in this register will be read.
3	PTDAT_D3	0	R/W	When the pin is used as a general input port, if this register is read, the status of the
2	PTDAT_D2	0	R/W	corresponding pin will be read.
1	PTDAT_D1	0	R/W	When the pin is set to a function other than the general port, if it is used as an input pin,
0	PTDAT_D0	0	R/W	the pin status will be read from the corresponding bit in this register and writing to the bit is ignored; for an output pin, an undefined value will be read from the bit and writing to the bit is ignored.

### 27.2.15 Port E Data Register (PTDAT\_E)

PTDAT\_E is a 16-bit readable/writable register that stores port E data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_E7	PTDAT_E6	PTDAT_E5	PTDAT_E4	PTDAT_E3	PTDAT_E2	PTDAT_E1	PTDAT_E0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  The lower 8-bit value is always read from these bits. The write value should always be 0.
7	PTDAT_E7	0	R/W	Each bit stores output data of a pin used as a general output port.
6	PTDAT_E6	0	R/W	When the pin is used as a general output port, if the port is read, the value of the corresponding bit in this register will be read.
5	PTDAT_E5	0	R/W	When the pin is used as a general input port, if this register is read, the status of the corresponding pin will be read.
4	PTDAT_E4	0	R/W	When the pin is set to a function other than the general port, if it is used as an input pin, the pin status will be read from the corresponding bit in this register and writing to the bit is ignored; for an output pin, an undefined value will be read from the bit and writing to the bit is ignored.
3	PTDAT_E3	0	R/W	
2	PTDAT_E2	0	R/W	
1	PTDAT_E1	0	R/W	
0	PTDAT_E0	0	R/W	



## 27.2.16 Port F Data Register (PTDAT\_F)

PTDAT\_F is a 16-bit readable/writable register that stores port F data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_F7	PTDAT_F6	PTDAT_F5	PTDAT_F4	PTDAT_F3	PTDAT_F2	PTDAT_F1	PTDAT_F0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  The lower 8-bit value is always read from these bits. The write value should always be 0.
7	PTDAT_F7	0	R/W	Each bit stores output data of a pin used as a general output port.
6	PTDAT_F6	0	R/W	
5	PTDAT_F5	0	R/W	When the pin is used as a general output port, if the port is read, the value of the
4	PTDAT_F4	0	R/W	corresponding bit in this register will be read.
3	PTDAT_F3	0	R/W	When the pin is used as a general input port, if this register is read, the status of the
2	PTDAT_F2	0	R/W	corresponding pin will be read.
1	PTDAT_F1	0	R/W	
0	PTDAT_F0	0	R/W	When the pin is set to a function other than the general port, if it is used as an input pin, the pin status will be read from the corresponding bit in this register and writing to the bit is ignored; for an output pin, an undefined value will be read from the bit and writing to the bit is ignored.

### 27.2.17 Port G Data Register (PTDAT\_G)

PTDAT\_G is a 16-bit readable/writable register that stores port G data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_G7	PTDAT_G6	PTDAT_G5	PTDAT_G4	PTDAT_G3	PTDAT_G2	PTDAT_G1	PTDAT_G0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  The lower 8-bit value is always read from these bits. The write value should always be 0.
7	PTDAT_G7	0	R/W	Each bit stores output data of a pin used as a general output port.
6	PTDAT_G6	0	R/W	When the pin is used as a general output port, if the port is read, the value of the corresponding bit in this register will be read.
5	PTDAT_G5	0	R/W	When the pin is used as a general input port, if this register is read, the status of the corresponding pin will be read.
4	PTDAT_G4	0	R/W	When the pin is set to a function other than the general port, if it is used as an input pin, the pin status will be read from the corresponding bit in this register and writing to the bit is ignored; for an output pin, an undefined value will be read from the bit and writing to the bit is ignored.
3	PTDAT_G3	0	R/W	
2	PTDAT_G2	0	R/W	
1	PTDAT_G1	0	R/W	
0	PTDAT_G0	0	R/W	

### 27.2.18 Port H Data Register (PTDAT\_H)

PTDAT\_H is a 16-bit readable/writable register that stores port H data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_H7	PTDAT_H6	PTDAT_H5	PTDAT_H4	PTDAT_H3	PTDAT_H2	PTDAT_H1	PTDAT_H0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  The lower 8-bit value is always read from these bits. The write value should always be 0.
7	PTDAT_H7	0	R/W	Each bit stores output data of a pin used as a general output port.
6	PTDAT_H6	0	R/W	
5	PTDAT_H5	0	R/W	When the pin is used as a general output port, if the port is read, the value of the
4	PTDAT_H4	0	R/W	corresponding bit in this register will be read.
3	PTDAT_H3	0	R/W	When the pin is used as a general input port,
2	PTDAT_H2	0	R/W	if this register is read, the status of the
1	PTDAT_H1	0	R/W	corresponding pin will be read.
0	PTDAT_H0	0	R/W	When the pin is set to a function other than the general port, if it is used as an input pin, the pin status will be read from the corresponding bit in this register and writing to the bit is ignored; for an output pin, an undefined value will be read from the bit and writing to the bit is ignored.

### 27.2.19 Port I Data Register (PTDAT\_I)

PTDAT\_I is a 16-bit readable/writable register that stores port I data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PTDAT_I4	PTDAT_I3	PTDAT_I2	PTDAT_I1	PTDAT_I0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  The lower 8-bit value is always read from these bits. The write value should always be 0.
7 to 5	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
4	PTDAT_I4	0	R/W	Each bit stores output data of a pin used as a general output port.  When the pin is used as a general output port, if the port is read, the value of the corresponding bit in this register will be read.  When the pin is used as a general input port, if this register is read, the status of the corresponding pin will be read.  When the pin is set to a function other than the general port, if it is used as an input pin, the pin status will be read from the corresponding bit in this register and writing to the bit is ignored; for an output pin, an undefined value will be read from the bit and writing to the bit is ignored.
3	PTDAT_I3	0	R/W	
2	PTDAT_I2	0	R/W	
1	PTDAT_I1	0	R/W	
0	PTDAT_I0	0	R/W	

### 27.2.20 Port J Data Register (PTDAT\_J)

PTDAT\_J is a 16-bit readable/writable register that stores port J data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_J7	PTDAT_J6	PTDAT_J5	PTDAT_J4	PTDAT_J3	PTDAT_J2	PTDAT_J1	PTDAT_J0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved  The lower 8-bit value is always read from these bits. The write value should always be 0.
7	PTDAT_J7	0	R/W	Each bit stores output data of a pin used as a general output port.
6	PTDAT_J6	0	R/W	When the pin is used as a general output port, if the port is read, the value of the corresponding bit in this register will be read.
5	PTDAT_J5	0	R/W	When the pin is used as a general input port, if this register is read, the status of the corresponding pin will be read.
4	PTDAT_J4	0	R/W	When the pin is set to a function other than the general port, if it is used as an input pin, the pin status will be read from the corresponding bit in this register and writing to the bit is ignored; for an output pin, an undefined value will be read from the bit and writing to the bit is ignored.
3	PTDAT_J3	0	R/W	
2	PTDAT_J2	0	R/W	
1	PTDAT_J1	0	R/W	
0	PTDAT_J0	0	R/W	

### 27.2.21 Input-Pin Pull-Up Control Register (PTPUL\_SPCL)

PTPUL\_SPCL is a 16-bit readable/writable register that individually controls the pull-up for the pin corresponding to each bit.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTPUL_	PTPUL_	—	—	—	—	—	—	—	—	—	—	—	—	—
		IRQ1	IRQ0													
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	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved  This bit is always read as 0. The write value should always be 0.
14	PTPUL_IRQ1	0	R/W	Controls pull-up of the IRQ1 pin. 0: IRQ1 pin pull-up off 1: IRQ1 pin pull-up on
13	PTPUL_IRQ0	0	R/W	Controls pull-up of the IRQ0 pin. 0: IRQ0 pin pull-up off 1: IRQ0 pin pull-up on
12 to 0	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.

### 27.2.22 Pin Select Register 0 (PTSEL\_A)

PTSEL\_A is a 16-bit readable/writable register that selects the functions for the port A (PA) pins that multiplex two or more functions other than the port function. To use one of these multiplexed functions for a pin, the port control register should be set to select the functions other than the port function after setting the corresponding bit in PTSEL\_A.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_A7[1:0]		PTSEL_A6[1:0]		—	PTSEL_A5	—	PTSEL_A4	—	PTSEL_A3	—	PTSEL_A2	—	PTSEL_A1	—	PTSEL_A0
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/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTSEL_A7[1:0]	00	R/W	Port A (PA7) Function Select 00: STATUS1 01: $\overline{\text{RTS2}}$ function 10: PA7 function 11: Setting prohibited
13, 12	PTSEL_A6[1:0]	00	R/W	Port A (PA6) Function Select 00: STATUS0 01: $\overline{\text{CTS2}}$ function 10: PA6 function 11: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PTSEL_A5	0	R/W	Port A (PA5) Function Select 0: $\overline{\text{FCE}}$ function 1: PA5 function
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PTSEL_A4	0	R/W	Port A (PA4) Function Select 0: $\overline{\text{FRE}}$ function 1: PA4 function

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PTSEL_A3	0	R/W	Port A (PA3) Function Select 0: $\overline{\text{FWE}}$ function 1: PA3 function
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PTSEL_A2	0	R/W	Port A (PA2) Function Select 0: TXD2 function 1: PA2 function
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PTSEL_A1	0	R/W	Port A (PA1) Function Select 0: RXD2 function 1: PA1 function
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PTSEL_A0	0	R/W	Port A (PA0) Function Select 0: SCK2 function 1: PA0 function



### 27.2.23 Pin Select Register 1 (PTSEL\_B)

PTSEL\_B is a 16-bit readable/writable register that selects the functions for the port B (PB) pins that multiplex two or more functions other than the port function. To use one of these multiplexed functions for a pin, the port control register should be set to select the functions other than the port function after setting the corresponding bit in PTSEL\_B.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_B7[1:0]		PTSEL_B6[1:0]		PTSEL_B5[1:0]		PTSEL_B4[1:0]		—	PTSEL_B3	—	PTSEL_B2	—	PTSEL_B1	—	PTSEL_B0
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/W	R	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTSEL_B7[1:0]	00	R	Port B (PB7) Function Select 00: A25 function 01: PB7 function 10: $\overline{\text{DREQ0}}$ function 11: $\overline{\text{RTS0}}$ function
13, 12	PTSEL_B6[1:0]	00	R/W	Port B (PB6) Function Select 00: A24 function 01: PB6 function 10: $\overline{\text{DACK0}}$ function 11: $\overline{\text{CTS0}}$ function
11, 10	PTSEL_B5[1:0]	00	R/W	Port B (PB5) Function Select 00: A23 function 01: PB5 function 10: $\overline{\text{DTEND0}}$ function 11: $\overline{\text{RTS1}}$ function
9, 8	PTSEL_B4[1:0]	00	R/W	Port B (PB4) Function Select 00: A22 function 01: PB4 function 10: $\overline{\text{CTS1}}$ function 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PTSEL_B3	0	R/W	Port B (PB3) Function Select 0: A21 function 1: PB3 function
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PTSEL_B2	All 0	R/W	Port B (PB2) Function Select 0: A20 function 1: PB2 function
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PTSEL_B1	All 0	R/W	Port B (PB1) Function Select 0: A19 function 1: PB1 function
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PTSEL_B0	All 0	R/W	Port B (PB0) Function Select 0: A18 function 1: PB0 function

### 27.2.24 Pin Select Register 2 (PTSEL\_C)

PTSEL\_C is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTSEL_C7	—	PTSEL_C6	—	PTSEL_C5	—	PTSEL_C4	—	PTSEL_C3	—	PTSEL_C2	PTSEL_C1[1:0]		—	PTSEL_C0
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/W	R	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PTSEL_C7	0	R/W	Port C (PC7) Function Select 0: AUDIO_CLK0 function 1: PC7 function
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PTSEL_C6	0	R/W	Port C (PC6) Function Select 0: AUDIO_CLK1 function 1: PC6 function
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PTSEL_C5	0	R/W	Port C (PC5) Function Select 0: AUDIO_CLK2 function 1: PC5 function
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PTSEL_C4	0	R/W	Port C (PC4) Function Select 0: SSIWS2 function 1: PC4 function

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PTSEL_C3	0	R/W	Port C (PC3) Function Select 0: SSISCK2 function 1: PC3 function
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PTSEL_C2	0	R/W	Port C (PC2) Function Select 0: SSIDATA2 function 1: PC2 function
3, 2	PTSEL_C1[1:0]	00	R/W	Port C (PC1) Function Select 00: ASEBRKAK/BRKACK function 01: TCLK function 10: PC1 function 11: Setting prohibited
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PTSEL_C0	0	R/W	Port C (PC0) Function Select 0: FALE function 1: PC0 function

### 27.2.25 Pin Select Register 3 (PTSEL\_D)

PTSEL\_D is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_D7[1:0]		PTSEL_D6[1:0]		PTSEL_D5[1:0]		PTSEL_D4[1:0]		PTSEL_D3[1:0]		PTSEL_D2[1:0]		PTSEL_D1[1:0]		PTSEL_D0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTSEL_D7[1:0]	00	R/W	Port D (PD7) Function Select 00: CRS function 01: PD7 function 10: IDEA1_M function 11: Setting prohibited
13, 12	PTSEL_D6[1:0]	00	R/W	Port D (PD6) Function Select 00: TX_ER function 01: PD6 function 10: IDEIOWR_M function 11: Setting prohibited
11, 10	PTSEL_D5[1:0]	00	R/W	Port D (PD5) Function Select 00: TX_CLK function 01: PD5 function 10: IDDED15_M function 11: Setting prohibited
9, 8	PTSEL_D4[1:0]	00	R/W	Port D (PD4) Function Select 00: TX_EN function 01: PD4 function 10: IDDED0_M function 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PTSEL_D3[1:0]	00	R/W	Port D (PD3) Function Select 00: MII_TXD0 function 01: SSISCK5 function 10: IDEIORDY_M function 11: PD3 function
5, 4	PTSEL_D2[1:0]	00	R/W	Port D (PD2) Function Select 00: MII_TXD1 function 01: SSIWS5 function 10: IDEIORD_M function 11: PD2 function
3, 2	PTSEL_D1[1:0]	00	R/W	Port D (PD1) Function Select 00: MII_TXD2 function 01: AUDIO_CLK5 function 10: IDEINT_M function 11: PD1 function
1, 0	PTSEL_D0[1:0]	00	R/W	Port D (PD0) Function Select 00: MII_TXD3 function 01: SSIDATA5 function 10: IODACK_M function 11: PD0 function

### 27.2.26 Pin Select Register 4 (PTSEL\_E)

PTSEL\_E is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_E7[1:0]		PTSEL_E6[1:0]		PTSEL_E5[1:0]		PTSEL_E4[1:0]		PTSEL_E3[1:0]		PTSEL_E2[1:0]		PTSEL_E1[1:0]		PTSEL_E0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTSEL_E7[1:0]	00	R/W	Port E (PE7) Function Select 00: COL function 01: PE7 function 10: IDEA2_M function 11: Setting prohibited
13, 12	PTSEL_E6[1:0]	00	R/W	Port E (PE6) Function Select 00: RX_ER function 01: PE6 function 10: IODREQ_M function 11: Setting prohibited
11, 10	PTSEL_E5[1:0]	00	R/W	Port E (PE5) Function Select 00: RX_CLK function 01: PE5 function 10: IDDED1_M function 11: Setting prohibited
9, 8	PTSEL_E4[1:0]	00	R/W	Port E (PE4) Function Select 00: RX_DV function 01: PE4 function 10: IDDED14_M function 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PTSEL_E3[1:0]	00	R/W	Port E (PE3) Function Select 00: MII_RXD0 function 01: SSIWS4 function 10: IDDED2_M function 11: PE3 function
5, 4	PTSEL_E2[1:0]	00	R/W	Port E (PE2) Function Select 00: MII_RXD1 function 01: SSISCK4 function 10: IDDED13_M function 11: PE2 function
3, 2	PTSEL_E1[1:0]	00	R/W	Port E (PE1) Function Select 00: MII_RXD2 function 01: SSIDATA4 function 10: IDDED3_M function 11: PE1 function
1, 0	PTSEL_E0[1:0]	00	R/W	Port E (PE0) Function Select 00: MII_RXD3 function 01: AUDIO_CLK4 function 10: IDDED12_M function 11: PE0 function



### 27.2.27 Pin Select Register 5 (PTSEL\_F)

PTSEL\_F is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTSEL_F7	—	PTSEL_F6	PTSEL_F5[1:0]	PTSEL_F4[1:0]	PTSEL_F3[1:0]	PTSEL_F2[1:0]	PTSEL_F1[1:0]	PTSEL_F0[1:0]						
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/W	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. The write value should always be 0.
14	PTSEL_F7	0	R/W	Port F (PF7) Function Select 0: D32 function 1: PF7 function
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PTSEL_F6	0	R/W	Port F (PF6) Function Select 0: D33 function 1: PF6 function
11, 10	PTSEL_F5[1:0]	00	R/W	Port F (PF5) Function Select 00: D34 function 01: PF5 function 10: Setting prohibited 11: Setting prohibited
9, 8	PTSEL_F4[1:0]	00	R/W	Port F (PF4) Function Select 00: EXOUT function 01: PF4 function 10: IDECS1_M function 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PTSEL_F3[1:0]	00	R/W	Port F (PF3) Function Select 00: LNKSTA function 01: PF3 function 10: IDECS0_M function 11: Setting prohibited
5, 4	PTSEL_F2[1:0]	00	R/W	Port F (PF2) Function Select 00: WOL function 01: PF2 function 10: IDEA0_M function 11: Setting prohibited
3, 2	PTSEL_F1[1:0]	00	R/W	Port F (PF1) Function Select 00: MDIO function 01: PF1 function 10: IDDED11_M function 11: Setting prohibited
1, 0	PTSEL_F0[1:0]	00	R/W	Port F (PF0) Function Select 00: MDC function 01: PF0 function 10: IDDED4_M function 11: Setting prohibited

### 27.2.28 Pin Select Register 6 (PTSEL\_G)

PTSEL\_G is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_G7[1:0]		PTSEL_G6[1:0]		PTSEL_G5[1:0]		PTSEL_G4[1:0]		PTSEL_G3[1:0]		PTSEL_G2[1:0]		PTSEL_G1[1:0]		PTSEL_G0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTSEL_G7[1:0]	00	R/W	Port G (PG7) Function Select 00: LCD_DATA15 function 01: DR3 function 10: PG7 function 11: Setting prohibited
13, 12	PTSEL_G6[1:0]	00	R/W	Port G (PG6) Function Select 00: LCD_DATA14 function 01: DR2 function 10: PG6 function 11: Setting prohibited
11, 10	PTSEL_G5[1:0]	00	R/W	Port G (PG5) Function Select 00: LCD_DATA13 function 01: DR1 function 10: PG5 function 11: Setting prohibited
9, 8	PTSEL_G4[1:0]	00	R/W	Port G (PG4) Function Select 00: LCD_DATA12 function 01: DR0 function 10: PG4 function 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PTSEL_G3[1:0]	00	R/W	Port G (PG3) Function Select 00: LCD_DATA11 function 01: DG5 function 10: PG3 function 11: Setting prohibited
5, 4	PTSEL_G2[1:0]	00	R/W	Port G (PG2) Function Select 00: LCD_DATA10 function 01: DG4 function 10: PG2 function 11: Setting prohibited
3, 2	PTSEL_G1[1:0]	00	R/W	Port G (PG1) Function Select 00: LCD_DATA9 function 01: DG3 function 10: PG1 function 11: Setting prohibited
1, 0	PTSEL_G0[1:0]	00	R/W	Port G (PG0) Function Select 00: LCD_DATA8 function 01: DG2 function 10: PG0 function 11: Setting prohibited

### 27.2.29 Pin Select Register 7 (PTSEL\_H)

PTSEL\_H is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTSEL_H7	—	PTSEL_H6	—	PTSEL_H5	—	PTSEL_H4	PTSEL_H3[1:0]	PTSEL_H2[1:0]	PTSEL_H1[1:0]	PTSEL_H0[1:0]				
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/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PTSEL_H7	0	R/W	Port H (PH7) Function Select 0: AUDIO_CLK3 function 1: PH7 function
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PTSEL_H6	0	R/W	Port H (PH6) Function Select 0: SSIWS3 function 1: PH6 function
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PTSEL_H5	0	R/W	Port H (PH5) Function Select 0: SSISCK3 function 1: PH5 function
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PTSEL_H4	0	R/W	Port H (PH4) Function Select 0: SSIDATA3 function 1: PH4 function

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PTSEL_H3[1:0]	00	R/W	Port H (PH3) Function Select 00: LCD_CL2 function 01: DE_V function 10: PH3 function 11: Setting prohibited
5, 4	PTSEL_H2[1:0]	00	R/W	Port H (PH2) Function Select 00: LCD_DON function 01: DCLKOUT function 10: Setting prohibited 11: PH2 function
3, 2	PTSEL_H1[1:0]	00	R/W	Port H (PH1) Function Select 00: LCD_VCP_WC function 01: DR4 function 10: PH1 function 11: Setting prohibited
1, 0	PTSEL_H0[1:0]	00	R/W	Port H (PH0) Function Select 00: LCD_VEP_WC function 01: DR5 function 10: PH0 function 11: Setting prohibited

### 27.2.30 Pin Select Register 8 (PTSEL\_I)

PTSEL\_I is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_I7[1:0]		PTSEL_I6[1:0]		PTSEL_I5[1:0]		PTSEL_I4[1:0]		PTSEL_I3[1:0]		PTSEL_I2[1:0]		PTSEL_I1[1:0]		PTSEL_I0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTSEL_I7[1:0]	00	R/W	Port I (PI7) Function Select 00: LCD_DATA3 function 01: DB3 function 10: BT_DATA3 function 11: Setting prohibited
13, 12	PTSEL_I6[1:0]	00	R/W	Port I (PI6) Function Select 00: LCD_DATA2 function 01: DB2 function 10: BT_DATA2 function 11: Setting prohibited
11, 10	PTSEL_I5[1:0]	00	R/W	Port I (PI5) Function Select 00: LCD_DATA1 function 01: DB1 function 10: BT_DATA1 function 11: Setting prohibited
9, 8	PTSEL_I4[1:0]	00	R/W	Port I (PI4) Function Select 00: LCD_DATA7 function 01: DG1 function 10: BT_DATA7 function 11: PI4 function

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PTSEL_I3[1:0]	00	R/W	Port I (PI3) Function Select 00: LCD_DATA6 function 01: DG0 function 10: BT_DATA6 function 11: PI3 function
5, 4	PTSEL_I2[1:0]	00	R/W	Port I (PI2) Function Select 00: LCD_DATA5 function 01: DB5 function 10: BT_DATA5 function 11: PI2 function
3, 2	PTSEL_I1[1:0]	00	R/W	Port I (PI1) Function Select 00: LCD_DATA4 function 01: DB4 function 10: BT_DATA4 function 11: PI1 function
1, 0	PTSEL_I0[1:0]	00	R/W	Port I (PI0) Function Select 00: PI0 function 01: COM/CDE function 10: Setting prohibited 11: Setting prohibited



### 27.2.31 Pin Select Register 9 (PTSEL\_J)

PTSEL\_J is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_J7[1:0]		PTSEL_J6[1:0]		PTSEL_J5[1:0]		PTSEL_J4[1:0]		PTSEL_J3[1:0]		PTSEL_J2[1:0]		PTSEL_J1[1:0]		PTSEL_J0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTSEL_J7[1:0]	00	R/W	Port J (PJ7) Function Select 00: PJ7 function 01: Setting prohibited 10: Setting prohibited 11: IDDED10_M function
13, 12	PTSEL_J6[1:0]	00	R/W	Port J (PJ6) Function Select 00: PJ6 function 01: Setting prohibited 10: Setting prohibited 11: IDDED5_M function
11, 10	PTSEL_J5[1:0]	00	R/W	Port J (PJ5) Function Select 00: PJ5 function 01: Setting prohibited 10: Setting prohibited 11: IDDED9_M function
9, 8	PTSEL_J4[1:0]	00	R/W	Port J (PJ4) Function Select 00: PJ4 function 01: Setting prohibited 10: Setting prohibited 11: IDDED6_M function

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PTSEL_J3[1:0]	00	R/W	Port J (PJ3) Function Select 00: PJ3 function 01: Setting prohibited 10: Setting prohibited 11: IDED7_M function
5, 4	PTSEL_J2[1:0]	00	R/W	Port J (PJ2) Function Select 00: PJ2 function 01: Setting prohibited 10: Setting prohibited 11: IDED8_M function
3, 2	PTSEL_J1[1:0]	00	R/W	Port J (PJ1) Function Select 00: PJ1 function 01: Setting prohibited 10: Setting prohibited 11: IDERST_M function
1, 0	PTSEL_J0[1:0]	00	R/W	Port J (PJ0) Function Select 00: PJ0 function 01: Setting prohibited 10: Setting prohibited 11: DIRECTION_M function

### 27.2.32 Pin Select Register 10 (PTSEL\_K)

PTSEL\_K is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_K7[1:0]		PTSEL_K6[1:0]		—	PTSEL_K5	PTSEL_K4[1:0]		PTSEL_K3[1:0]		PTSEL_K2[1:0]		PTSEL_K1[1:0]		PTSEL_K0[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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PTSEL_K7[1:0]	00	R/W	Port K (PK7) Function Select 00: $\overline{\text{WDT OV F}}$ function 01: IRQ1 function 10: AUDCK function 11: $\overline{\text{DACK1}}$ function
13, 12	PTSEL_K6[1:0]	00	R/W	Port K (PK6) Function Select 00: SCK0 function 01: AUDSYNC function 10: FCLE function 11: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PTSEL_K5	0	R/W	Port K (PK5) Function Select 0: SCK1 function 1: $\text{FR}/\overline{\text{B}}$ function
9, 8	PTSEL_K4[1:0]	00	R/W	Port K (PK4) Function Select 00: LCD_DATA0 function 01: DB0 function 10: BT_DATA0 function 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PTSEL_K3[1:0]	00	R/W	Port K (PK3) Function Select 00: LCD_CL1 function 01: HSYNC/SPL function 10: BT_HSYNC function 11: Setting prohibited
5, 4	PTSEL_K2[1:0]	00	R/W	Port K (PK2) Function Select 00: LCD_CLK function 01: DCLKIN function 10: Setting prohibited 11: Setting prohibited
3, 2	PTSEL_K1[1:0]	00	R/W	Port K (PK1) Function Select 00: LCD_FLM function 01: VSYNC/SPS function* 10: BT_VSYNC function 11: Setting prohibited
1, 0	PTSEL_K0[1:0]	00	R/W	Port K (PK0) Function Select 00: LCD_M_DISP function 01: DE_C/DE_H function 10: BT_DE_C function 11: Setting prohibited

**Note:** \* This pin function switches over input/output functions in a special select register.

### 27.2.33 Pin Select Register 11 (PTSEL\_P)

PTSEL\_P is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PTSEL_P11	PTSEL_P10	PTSEL_P9	PTSEL_P8	—	—	—	—	—	—	—	—
Initial 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	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PTSEL_P11	0	R/W	Port P (PP11) Function Select 0: RXD0 function 1: AUDATA0 function
10	PTSEL_P10	0	R/W	Port P (PP10) Function Select 0: TXD0 function 1: AUDATA1 function
9	PTSEL_P9	0	R/W	Port P (PP9) Function Select 0: RXD1 function 1: AUDATA2 function
8	PTSEL_P8	0	R/W	Port P (PP8) Function Select 0: TXD1 function 1: AUDATA3 function
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

### 27.2.34 Pin Select Register 12 (PTSEL\_R)

PTSEL\_R is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_R15	PTSEL_R14	PTSEL_R13	PTSEL_R12	PTSEL_R11	PTSEL_R10	PTSEL_R9	PTSEL_R8	PTSEL_R7	PTSEL_R6	PTSEL_R5	PTSEL_R4	PTSEL_R3	PTSEL_R2	PTSEL_R1	PTSEL_R0
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	PTSEL_R15	0	R/W	Port P (PR15) Function Select 0: D63 function 1: IDDED1 function
14	PTSEL_R14	0	R/W	Port P (PR14) Function Select 0: D62 function 1: IDDED0 function
13	PTSEL_R13	0	R/W	Port P (PR13) Function Select 0: D61 function 1: IDDED3 function
12	PTSEL_R12	0	R/W	Port P (PR12) Function Select 0: D60 function 1: IDDED2 function
11	PTSEL_R11	0	R/W	Port P (PR11) Function Select 0: D59 function 1: IDDED5 function
10	PTSEL_R10	0	R/W	Port P (PR10) Function Select 0: D58 function 1: IDDED4 function
9	PTSEL_R9	0	R/W	Port P (PR9) Function Select 0: D57 function 1: IDDED7 function
8	PTSEL_R8	0	R/W	Port P (PR8) Function Select 0: D56 function 1: IDDED6 function

Bit	Bit Name	Initial Value	R/W	Description
7	PTSEL_R7	0	R/W	Port P (PR7) Function Select 0: D55 function 1: DIRECTION function
6	PTSEL_R6	0	R/W	Port P (PR6) Function Select 0: D54 function 1: IDERST function
5	PTSEL_R5	0	R/W	Port P (PR5) Function Select 0: D53 function 1: IDED8 function
4	PTSEL_R4	0	R/W	Port P (PR4) Function Select 0: D52 function 1: IDED9 function
3	PTSEL_R3	0	R/W	Port P (PR3) Function Select 0: D51 function 1: IDED10 function
2	PTSEL_R2	0	R/W	Port P (PR2) Function Select 0: D50 function 1: IDED11 function
1	PTSEL_R1	0	R/W	Port P (PR1) Function Select 0: D49 function 1: IDED12 function
0	PTSEL_R0	0	R/W	Port P (PR0) Function Select 0: D48 function 1: IDED13 function

## 27.2.35 Pin Select Register 13 (PTSEL\_S)

PTSEL\_S is a 16-bit readable/writable register that selects the functions for the pins that multiplex two or more functions other than the port function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_S15	PTSEL_S14	PTSEL_S13	PTSEL_S12	PTSEL_S11	PTSEL_S10	PTSEL_S9	PTSEL_S8	PTSEL_S7	PTSEL_S6	PTSEL_S5	PTSEL_S4	PTSEL_S3	PTSEL_S2	PTSEL_S1	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15	PTSEL_S15	0	R/W	Port S (PS15) Function Select 0: IRQ0 function 1: $\overline{DTEND1}$ function
14	PTSEL_S14	0	R/W	Port S (PS14) Function Select 0: $\overline{IRQOUT}$ function 1: $\overline{DREQ1}$ function
13	PTSEL_S13	0	R/W	Port S (PS13) Function Select 0: D47 function 1: $\overline{IDECSS0}$ function
12	PTSEL_S12	0	R/W	Port S (PS12) Function Select 0: D46 function 1: $\overline{IDECSS1}$ function
11	PTSEL_S11	0	R/W	Port S (PS11) Function Select 0: D45 function 1: $\overline{IODACK}$ function
10	PTSEL_S10	0	R/W	Port S (PS10) Function Select 0: D44 function 1: IODINT function
9	PTSEL_S9	0	R/W	Port S (PS9) Function Select 0: D43 function 1: IDEIORDY function
8	PTSEL_S8	0	R/W	Port S (PS8) Function Select 0: D42 function 1: $\overline{IDEIORD}$ function



Bit	Bit Name	Initial Value	R/W	Description
7	PTSEL_S7	0	R/W	Port S (PS7) Function Select 0: D41 function 1: IODREQ function
6	PTSEL_S6	0	R/W	Port S (PS6) Function Select 0: D40 function 1: IDEIOWR function
5	PTSEL_S5	0	R/W	Port S (PS5) Function Select 0: D39 function 1: IDED14 function
4	PTSEL_S4	0	R/W	Port S (PS4) Function Select 0: D38 function 1: IDED15 function
3	PTSEL_S3	0	R/W	Port S (PS3) Function Select 0: D37 function 1: IDEA1 function
2	PTSEL_S2	0	R/W	Port S (PS2) Function Select 0: D36 function 1: IDEA2 function
1	PTSEL_S1	0	R/W	Port S (PS1) Function Select 0: D35 function 1: IDEA0 function
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

### 27.2.36 HI-Z Register A (PTHIZ\_A)

PTHIZ\_A is a 16-bit readable/writable register that controls the high-impedance state of the on-chip module pins.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTHIZ_ATA	PTHIZ_TMU	PTHIZ_LCD	PTHIZ_IIC	PTHIZ_FLCTL	PTHIZ_DMAC	PTHIZ_SCI0	PTHIZ_SCI1	PTHIZ_SCI2	PTHIZ_ETH	PTHIZ_VDC2	PTHIZ_USB	—	—	—	—
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
15	PTHIZ_ATA	0	R/W	High-Impedance Control for ATAPI Pins 0: Normal state 1: High-impedance state
14	PTHIZ_TMU	0	R/W	High-Impedance Control for TMU Pins 0: Normal state 1: High-impedance state
13	PTHIZ_LCD	0	R/W	High-Impedance Control for LCD Pins 0: Normal state 1: High-impedance state
12	PTHIZ_IIC	0	R/W	High-Impedance Control for IIC Pins 0: Normal state 1: High-impedance state
11	PTHIZ_FLCTL	0	R/W	High-Impedance Control for FLCTL Pins 0: Normal state 1: High-impedance state
10	PTHIZ_DMAC	0	R/W	High-Impedance Control for DMAC Pins 0: Normal state 1: High-impedance state
9	PTHIZ_SCI0	0	R/W	High-Impedance Control for SCIF Channel 0 Pins 0: Normal state 1: High-impedance state

Bit	Bit Name	Initial Value	R/W	Description
8	PTHIZ_SCI1	0	R/W	High-Impedance Control for SCIF Channel 1 Pins 0: Normal state 1: High-impedance state
7	PTHIZ_SCI2	0	R/W	High-Impedance Control for SCIF Channel 2 Pins 0: Normal state 1: High-impedance state
6	PTHIZ_ETH	0	R/W	High-Impedance Control for EtherC Pins 0: Normal state 1: High-impedance state
5	PTHIZ_VDC2	0	R/W	High-Impedance Control for VDC2 Pins 0: Normal state 1: High-impedance state
4	PTHIZ_USB	0	R/W	High-Impedance Control for USB Pins 0: Normal state 1: High-impedance state
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

### 27.2.37 HI-Z Register B (PTHIZ\_B)

PTHIZ\_B is a 16-bit readable/writable register that controls the high-impedance state of the on-chip module pins.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTHIZ_SSI0	PTHIZ_SSI1	PTHIZ_SSI2	PTHIZ_SSI3	PTHIZ_SSI4	PTHIZ_SSI5	—	—	—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PTHIZ_SSI0	0	R/W	High-Impedance Control for SSI Channel 0 Pins 0: Normal state 1: High-impedance state
14	PTHIZ_SSI1	0	R/W	High-Impedance Control for SSI Channel 1 Pins 0: Normal state 1: High-impedance state
13	PTHIZ_SSI2	0	R/W	High-Impedance Control for SSI Channel 2 Pins 0: Normal state 1: High-impedance state
12	PTHIZ_SSI3	0	R/W	High-Impedance Control for SSI Channel 3 Pins 0: Normal state 1: High-impedance state
11	PTHIZ_SSI4	0	R/W	High-Impedance Control for SSI Channel 4 Pins 0: Normal state 1: High-impedance state
10	PTHIZ_SSI5	0	R/W	High-Impedance Control for SSI Channel 5 Pins 0: Normal state 1: High-impedance state
9 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

### 27.2.38 Special Select Register (PTSEL\_SPCL)

PTSEL\_SPCL is a 16-bit readable/writable register that selects the functions input/output for HSYNC and VSYNC other than the port function.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved  These bits are always read as 0. The write value should always be 0.
1	PTSEL_VSYNC	0	R/W	VSYNC Function Select 0: VSYNC/SPS function 1: EX_VSYNC function
0	PTSEL_HSYNC	0	R/W	HSYNC Function Select 0: HSYNC/SPL function 1: EX_HSYNC function

## 27.3 Usage Examples

The following describes examples of GPIO setting procedures.

### 27.3.1 Port Function Select

Before selecting the port function, be sure to select the port output or input function as described later. Then, select the port function through the corresponding pin select register (PTSEL\_A to PTSEL\_J).

Note that an error such as a signal conflict will occur if the port function is selected through the pin select register while the input/output setting for the port is wrong.

### 27.3.2 Port Output Function

To select the port output function for a pin, write B'01 to the corresponding two bits in the port control register (PTIO\_A to PTIO\_J); the data in the corresponding bit in the port data register (PTDAT\_A to PTDAT\_J) is output from the port.

When the port output function is selected for a pin, the setting in the pull-up control register (PTPUL\_AB to PTUPL\_IJ) for the pin is ignored.

### 27.3.3 Port Input Function

To select the port input function for a pin, write B'10 to the corresponding two bits in the port control register (PTIO\_A to PTIO\_J) when turning off the pull-up MOS or B'11 when turning on the pull-up MOS; the data input through the pin can be read from the corresponding bit in the port data register (PTDAT\_A to PTDAT\_J).

### 27.3.4 On-Chip Module Function

To select the on-chip module function, first select the on-chip module to be used through the pin select register (PTSEL\_A to PTSEL\_J).

Then, write B'00 to the corresponding two bits in the port control register (PTIO\_A to PTIO\_J).

## Section 28 Power-Down Mode

In power-down modes, operations of the CPU and some of the on-chip peripheral modules are stopped to reduce power consumption.

### 28.1 Features

- Supports refresh standby mode
- Supports sleep mode and module standby mode

#### 28.1.1 Types of Power-Down Modes

The types and functions of power-down modes are as shown below.

- Sleep mode
- Refresh standby mode
- Module standby mode

Table 28.1 lists the states of the CPU and on-chip peripheral modules in each mode.

**Table 28.1 States in Power-Down Modes**

Power-Down Mode	Transition Condition	State								
		CPG	CPU	On-Chip Memory	On-Chip Peripheral Module	Pin	DDR-SDRAM	Cancellation	S1 <sup>*2</sup>	S0 <sup>*2</sup>
Sleep	SLEEP instruction executed with STBY = 0 in STBCR	Run	Halt (register contents retained)	Retained	Run	Held	AR or SR <sup>*1</sup>	- Interrupt - Power-on reset	1	0
Refresh standby	SLEEP instruction executed with STBY = 1 in STBCR	Halt	Halt (register contents retained)	Halt (contents retained)	Halt	Held (only CLKOUT operates)	SR <sup>*1</sup>	- NMI or IRQ - Power-on reset	0	1
Module standby	Corresponding bit in MSTPCR0/MSTPCR1 set to 1	Run	Run	Run	Selected modules halt	Held	AR or SR <sup>*1</sup>	Clear corresponding bit in MSTPCR0/MSTPCR1 to 0	0	0
Power-on reset	PRESET pin driven low	Initial state	Initial state	Initial state	Initial state	Initial state	Initial state	—	1	1
Normal operation		Run	Run	Run	Run	Run	Run	—	0	0

Notes: 1. AF: auto-refresh; SF: self-refresh

2. S1 and S0 are the output states on the STATUS1 and STATUS0 pins, respectively.



## 28.2 Input/Output Pins

Table 28.2 lists the pin configuration related to power-down modes.

**Table 28.2 Pin Configuration**

Pin Name	Function	I/O	Description
STATUS1	Processing state 1	Output	These pins indicate the operating state of this LSI.  STATUS[1:0]    Operating state H, H:            Power-on reset H, L:            Sleep mode L, H:            Refresh standby mode L, L:            Normal operation
STATUS0	Processing state 0	Output	

## 28.3 Register Descriptions

Table 28.3 shows the register configuration for power-down modes. Table 28.4 shows the register states in each operating mode.

**Table 28.3 Register Configuration**

Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
Standby control register	STBCR	R/W	H'FFC8 0020	H'1FC8 0020	32
Module stop register 0	MSTPCR0	R/W	H'FFC8 0030	H'1FC8 0030	32
Module stop register 1	MSTPCR1	R/W	H'FFC8 0038	H'1FC8 0038	32

**Table 28.4 Register States in Each Operating Mode**

Register Name	Abbreviation	Power-On Reset	Sleep	Standby
Standby control register	STBCR	H'0000 0000	Retained	Retained
Module stop register 0	MSTPCR0	H'0000 0000	Retained	Retained
Module stop register 1	MSTPCR1	H'0000 0000	Retained	Retained

### 28.3.1 Standby Control Register (STBCR)

STBCR is a 32-bit readable/writable register that selects a power-down mode to be entered after a SLEEP instruction is executed.

STBCR can be accessed only in longwords.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	STBY	0	R/W	Standby Selects whether to enter sleep mode or refresh standby mode after a SLEEP instruction is executed. 0: Sleep mode 1: Refresh standby mode Clear this bit to 0 when returning from the refresh standby mode by an interrupt.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

### 28.3.2 Module Stop Register 0 (MSTPCR0)

MSTPCR0 is a 32-bit readable/writable register that can individually start or stop the module assigned to each bit.

MSTPCR0 can be accessed only in longwords.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCDC	—	TMU	FLCTL	—	SCIF2	SCIF1	SCIF0	ETHER	IIC	ATAPI	G2D	—	VDC2	—	USB
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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W

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	INTC	0	R/W	INTC Module Stop Bit When set to 1, the clock supply to the INTC module is halted. 0: INTC operates 1: Clock supply to INTC is halted
21	DMAC	0	R/W	DMAC Module Stop Bit When set to 1, the clock supply to the DMAC module is halted. 0: DMAC operates 1: Clock supply to DMAC is halted
20	—	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
19	H-UDI	0	R/W	<p>H-UDI Module Stop Bit</p> <p>When set to 1, the clock supply to the H-UDI module is halted.</p> <p>0: H-UDI operates</p> <p>1: Clock supply to H-UDI is halted</p>
18	—	0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
17	UBC	0	R/W	<p>UBC Module Stop Bit</p> <p>When set to 1, the clock supply to the UBC module is halted.</p> <p>0: UBC operates</p> <p>1: Clock supply to UBC is halted</p>
16	—	0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
15	LCDC	0	R/W	<p>LCDC Module Stop Bit</p> <p>When set to 1, the clock supply to the LCDC module is halted.</p> <p>0: LCDC operates</p> <p>1: Clock supply to LCDC is halted</p>
14	—	0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
13	TMU	0	R/W	<p>TMU Module Stop Bit</p> <p>When set to 1, the clock supply to the TMU module is halted.</p> <p>0: TMU operates</p> <p>1: Clock supply to TMU is halted</p>
12	FLCTL	0	R/W	<p>FLCTL Module Stop Bit</p> <p>When set to 1, the clock supply to the FLCTL module is halted.</p> <p>0: FLCTL operates</p> <p>1: Clock supply to FLCTL is halted</p>

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	SCIF2	0	R/W	SCIF2 Module Stop Bit When set to 1, the clock supply to the SCIF2 module is halted. 0: SCIF2 operates 1: Clock supply to SCIF2 is halted
9	SCIF1	0	R/W	SCIF1 Module Stop Bit When set to 1, the clock supply to the SCIF1 module is halted. 0: SCIF1 operates 1: Clock supply to SCIF1 is halted
8	SCIF0	0	R/W	SCIF0 Module Stop Bit When set to 1, the clock supply to the SCIF0 module is halted. 0: SCIF0 operates 1: Clock supply to SCIF0 is halted
7	ETHER	0	R/W	ETHER Module Stop Bit When set to 1, the clock supply to the ETHER module is halted. 0: ETHER operates 1: Clock supply to ETHER is halted
6	IIC	0	R/W	IIC Module Stop Bit When set to 1, the clock supply to the IIC module is halted. 0: IIC operates 1: Clock supply to IIC is halted
5	ATAPI	0	R/W	ATAPI Module Stop Bit When set to 1, the clock supply to the ATAPI module is halted. 0: ATAPI operates 1: Clock supply to ATAPI is halted

Bit	Bit Name	Initial Value	R/W	Description
4	G2D	0	R/W	<p>G2D Module Stop Bit</p> <p>When set to 1, the clock supply to the G2D module is halted.</p> <p>0: G2D operates</p> <p>1: Clock supply to G2D is halted</p>
3	—	0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
2	VDC2	0	R/W	<p>VDC2 Module Stop Bit</p> <p>When set to 1, the clock supply to the VDC2 module is halted.</p> <p>0: VDC2 operates</p> <p>1: Clock supply to VDC2 is halted</p>
1	—	0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
0	USB	0	R/W	<p>USB Module Stop Bit</p> <p>When set to 1, the clock supply to the USB module is halted.</p> <p>0: USB operates</p> <p>1: Clock supply to USB is halted</p>

### 28.3.3 Module Stop Register 1 (MSTPCR1)

MSTPCR1 is a 32-bit readable/writable register that can individually start or stop the module assigned to each bit.

MSTPCR1 can be accessed only in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRC	—	—	—	—	—	SSI_B	SSI_A	—	—	—	—	—	—	—	—
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/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	SRC	0	R/W	SRC Module Stop Bit When set to 1, the clock supply to the SRC module is halted. 0: SRC operates 1: Clock supply to SRC is halted
30 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	SSI_B	0	R/W	SSI_B Module Stop Bit When set to 1, the clock supply to the SSI_B module is halted. 0: SSI_B operates 1: Clock supply to SSI_B is halted
24	SSI_A	0	R/W	SSI_A Module Stop Bit When set to 1, the clock supply to the SSI_A module is halted. 0: SSI_A operates 1: Clock supply to SSI_A is halted
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

## 28.4 Sleep Mode

### 28.4.1 Transition to Sleep Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of the CPU registers remain unchanged. On-chip peripheral modules continue to operate, and the clock output on the CLKOUT pin also continues. In sleep mode, a high level is output to the STATUS1 pin and a low level to the STATUS0 pin.

### 28.4.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ1, IRQ0 or on-chip peripheral module interrupt) or a reset.

Interrupts are accepted in sleep mode even when the BL bit in SR is 1. If necessary, save SPC and SSR to the stack before executing the SLEEP instruction.

#### (1) Canceling with Interrupt

When an NMI, IRQ1, IRQ0 or on-chip peripheral module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. A code indicating the interrupt source is set in INTEVT.

#### (2) Canceling with Reset

Sleep mode is canceled by a power-on reset caused by the  $\overline{\text{PRESET}}$  pin or watchdog timer overflow.



## 28.5 Refresh Standby Mode

### 28.5.1 Transition to Refresh Standby Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 1 causes a transition from the program execution state to refresh standby mode. In refresh standby mode, not only the CPU but also the clock and on-chip peripheral modules halt. However, the clock output from the CLKOUT pin continues.

The contents of the CPU and cache registers remain unchanged. Some registers of the on-chip peripheral modules are initialized.

The procedure for a transition to software standby mode is as follows:

1. Set the STBY bit in STBCR to 1.
2. Execute the SLEEP instruction.
3. Software standby mode is entered and the clocks within the LSI are halted. The output on the STATUS0 pin goes high.

### 28.5.2 Canceling Refresh Standby Mode

Refresh standby mode is canceled by an interrupt (NMI or IRQ/IRL) or a reset.

#### (1) Canceling with Interrupt

When an NMI or IRQ occurs, refresh standby mode is canceled and the STATUS0 pin goes low. Thereafter, interrupt exception handling is executed and a code indicating the interrupt source is set in INTEVT. After branching to the interrupt service routine, clear the STBY bit in the STBCR register back to 0. Since interrupts are accepted in refresh standby mode even when the BL bit in SR is 1, save SPC and SSR to the stack before executing the SLEEP instruction if necessary.

Immediately after an interrupt is detected, the clock output on the CLKOUT pin may be unstable until software standby mode is canceled.

#### (2) Canceling with Reset

Refresh standby mode is cancelled by a power-on reset by the  $\overline{\text{PRESET}}$  pin.

## **28.6 Module Standby Mode**

### **28.6.1 Transition to Module Standby Mode**

Setting the bits in the module stop register to 1 halts the clock supply to the corresponding on-chip peripheral modules.

Modules in module standby mode keep the state immediately before the transition to the module standby mode. The registers retain their contents before the module is halted, and the external pins also hold their states before halted. At waking up from the module standby state, operation starts from the condition immediately before the module was halted.

### **28.6.2 Canceling Module Standby Mode**

The module standby mode can be canceled by clearing the respective bit in the module stop register to 0 or by a power-on reset.

## 28.7 STATUS Pin Signal Change Timing

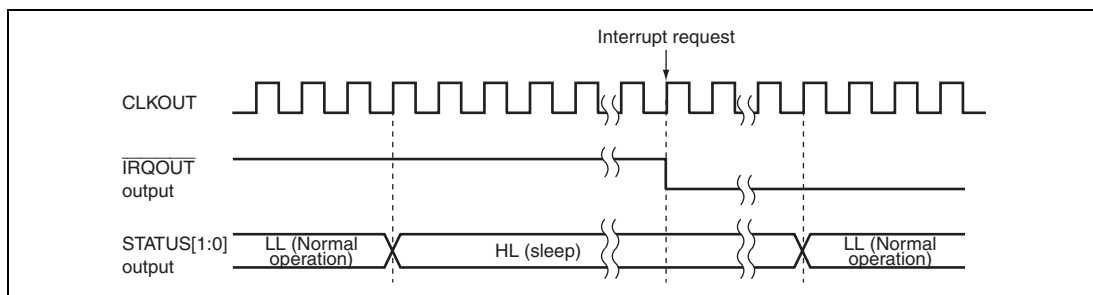
### 28.7.1 Timing at Reset

Refer to section 29.5, Status Pin Change Timing during Reset.

### 28.7.2 Timing at Sleep Mode Cancellation

#### (1) When an Interrupt Occurs in Sleep Mode

Figure 28.1 shows the timing of signal changes on the STATUS pins.



**Figure 28.1 STATUS Output when an Interrupt Occurs in Sleep Mode**



## Section 29 Watchdog Timer and Reset

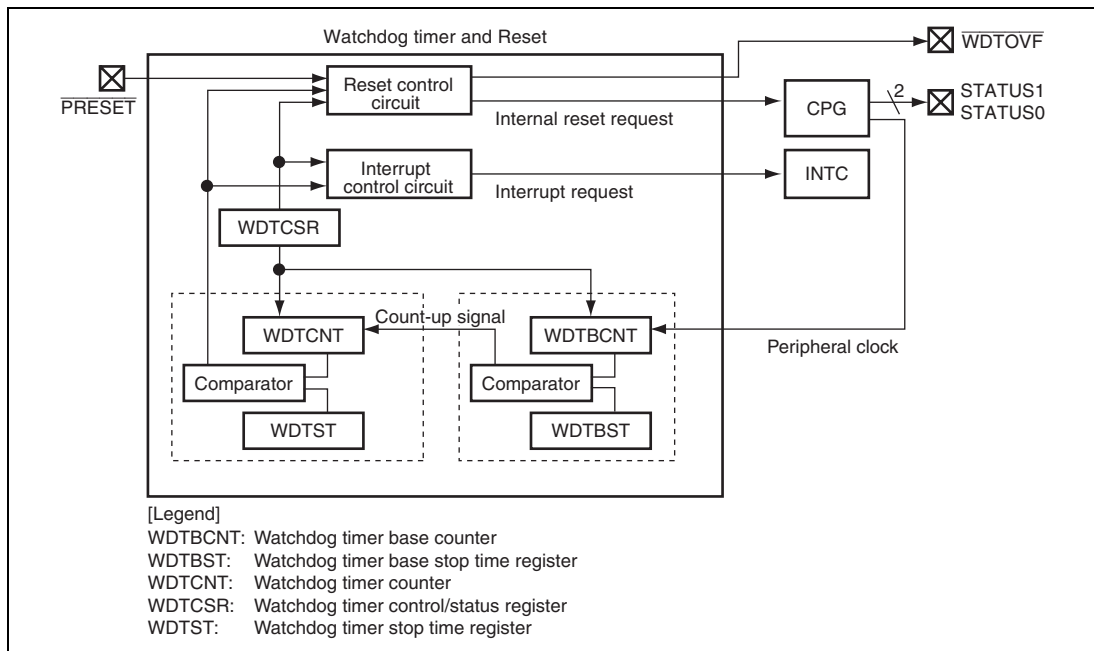
The reset and watchdog timer (WDT) control circuit comprises the reset control unit and WDT control unit which control the power-on reset sequence and a reset for on-chip peripheral modules and external devices.

The WDT is a one-channel timer which can be used as the watchdog timer or interval timer.

### 29.1 Features

- WDT monitors a system crash using a timer counting at specified intervals.
- WDT supports the watchdog timer mode and the interval timer mode.
- WDT generates an internal reset and output the WDTOVF signal when a WDT counter overflow occurs in watchdog timer mode.
- WDT generates the interval timer interrupt when a WDT counter overflow occurs in interval timer mode.
- The maximum time until the watchdog timer overflows is approximately 21 seconds (when the peripheral clock Pck is 50 MHz).
- Writing to WDT-related registers is not normally allowed. A specified code in the upper bits of write data enables writing to the registers.

Figure 29.1 shows a block diagram of the WDT.



**Figure 29.1 Block Diagram**

## 29.2 Input/Output Pins

Table 29.1 shows the pin configuration of the WDT.

**Table 29.1 Pin Configuration**

Pin name	Function	I/O	Description														
PRESET	Reset	Input	Power-on reset														
STATUS1	Processing state 1	Output	Indicate the processor's operating status														
STATUS0	Processing state 0		<table><tr><td>STATUS1</td><td>STATUS0</td><td>Operating Status</td></tr><tr><td>High</td><td>High</td><td>Reset</td></tr><tr><td>High</td><td>Low</td><td>Sleep mode</td></tr><tr><td>Low</td><td>High</td><td>Refresh standby mode</td></tr><tr><td>Low</td><td>Low</td><td>Normal operation</td></tr></table>	STATUS1	STATUS0	Operating Status	High	High	Reset	High	Low	Sleep mode	Low	High	Refresh standby mode	Low	Low
STATUS1	STATUS0	Operating Status															
High	High	Reset															
High	Low	Sleep mode															
Low	High	Refresh standby mode															
Low	Low	Normal operation															
			Pins for STATUS1 and STATUS0 are multiplexed to each other function independently.														
WDTOVF	Watchdog timer overflow	Output	Counter overflow signal output in the watchdog timer mode.														

## 29.3 Register Descriptions

Table 29.2 shows the registers of the WDT. Table 29.3 shows the register states in each processing mode.

**Table 29.2 Register Configuration**

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
Watchdog timer stop time register	WDTST	R/W	H'FFCC 0000	H'1FCC 0000	32
Watchdog timer control/status register	WDTCSR	R/W	H'FFCC 0004	H'1FCC 0004	32
Watchdog timer base stop time register	WDTBST	R/W	H'FFCC 0008	H'1FCC 0008	32
Watchdog timer counter	WDTCNT	R	H'FFCC 0010	H'1FCC 0010	32
Watchdog timer base counter	WDTBCNT	R	H'FFCC 0018	H'1FCC 0018	32

**Table 29.3 Register States in Each Processing Mode**

Register Name	Abbreviation	Power-on Reset by PRESET Pin	Power-on Reset by WDT/H-UDI	Sleep	Standby
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	Retained	Retained	Retained
Watchdog timer base counter	WDTBCNT	H'0000 0000	Retained	Retained	Retained



### 29.3.1 Watchdog Timer Stop Time Register (WDTST)

WDTST is a readable/writable 32-bit register that specifies the time until a watchdog timer overflows. The time until WDCNT overflows becomes the minimum value when set H'001 to the bits 11 to 0, and the maximum value when set H'000 to the bits 11 to 0. Use a longword access to write to the WDTST, with H'5A in the bits 31 to 24. The reading value of bits 31 to 24 is always H'00.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	(Given code)								—	—	—	—	—	—	—	—
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	(Given code)	H'00	R/W	Reserved (Given code for writing) These bits are always read as H'00. To write to this register, the write value 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	Counter value

### 29.3.2 Watchdog Timer Control/Status Register (WDTCSR)

WDTCSR is a readable/writable 32-bit register that comprises the timer mode-selecting bit and overflow flags. Use a longword access to write to the WDTCSR, with H'A5 in the bits 31 to 24. The reading value of bits 31 to 24 is always H'00.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	(Given code)								—	—	—	—	—	—	—	—
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/IT	—	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	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Given code)	H'00	R/W	Reserved (Given code for writing) These bits are always read as H'00. To write to this register, the write value must be H'A5.
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 Specifies starting and stopping of timer operation. 0: Stops counting up 1: Starts counting up
6	WT/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	—	0	—	Reserved This bit is always read as 0. The write value should always be 0
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: An overflow has not occurred 1: An overflow on WDTCNT has occurred
3	IOVF	0	R/W	Interval Timer Overflow Flag Indicates that WDTCNT has overflowed in interval timer mode. This flag is not set in watchdog timer mode. 0: An overflow has not occurred 1: An overflow on WDTCNT has occurred
2 to 0	—	R	All 0	Reserved These bits are always read as 0. The write value should always be 0.

### 29.3.3 Watchdog timer Base Stop Time Register (WDTBST)

WDTBST is a readable/writable 32-bit register that clears WDTBCNT. Use a longword write access to clear the WDTBCNT, with H'55 in the bits 31 to 24. The reading value of this register is always H'00.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	(Given code)								—	—	—	—	—	—	—	—
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

### 29.3.4 Watchdog Timer Counter (WDTCNT)

WDTCNT is a 32-bit read-only register that comprises 12-bit watchdog timer counter and counts up on the WDTBCNT overflow signal. When WDTCNT overflows, a reset is generated in watchdog timer mode, or an interrupt is generated in interval timer mode. Writing to WDTCNT 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

### 29.3.5 Watchdog Timer Base Counter (WDTBCNT)

WDTBCNT is a 32-bit read-only register that comprises 18-bit counter and counts up on the peripheral clock (Pck). When WDTBCNT overflows, WDTCNT is counted up and WDTBCNT is cleared to 0. Writing to WDTBCNT 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

## 29.4 Operation

### 29.4.1 Reset request

Power-on reset is available.

#### (1) Power-on reset

##### 1. Reset sources

- Input low level via PRESET pin.
- The WDT CNT overflows when the WT/IT bit in the WDTCSR is 1.
- The H-UDI reset occurs (For details, see section 31, User Debugging Interface (H-UDI)).

##### 2. Branch destination address: H'A000 0000

##### 3. Operation in branch

Exception code H'000 is set in the EXPEVT register. The VBR and SR registers are initialized, and the program branches to PC = H'A000 0000. By initialization, the VBR register is set to H'0000 0000. In the SR register, the MD, RB, and BL bits are set to 1, the FD bit is cleared to 0, and the IMASK3 to IMASK0 bits (interrupt mask level) are set to B'1111.

The CPU and the peripheral modules are also initialized. For details, see the register descriptions in each section.

When the power is turned on, be sure to input a low level to the PRESET pin. The TRST pin should also be brought low level 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;
}
```

### 29.4.2 Using watchdog timer mode

1. Set the WDTCNT overflow interval value in WDTST.
2. Set the WT/IT bit in WDTCSR to 1.
3. When the TME bit in WDTCSR is set to 1, the WDT count starts.
4. During operation in watchdog timer mode, clear to the WDTCNT or WDTBCNT periodically so that WDTCNT does not overflow. See section 29.4.5, Clearing WDT Counter for WDT counter clear method.
5. When the WDTCNT overflows, the WDT sets the WOVF flag in WDTCSR to 1, and generates a power-on reset.

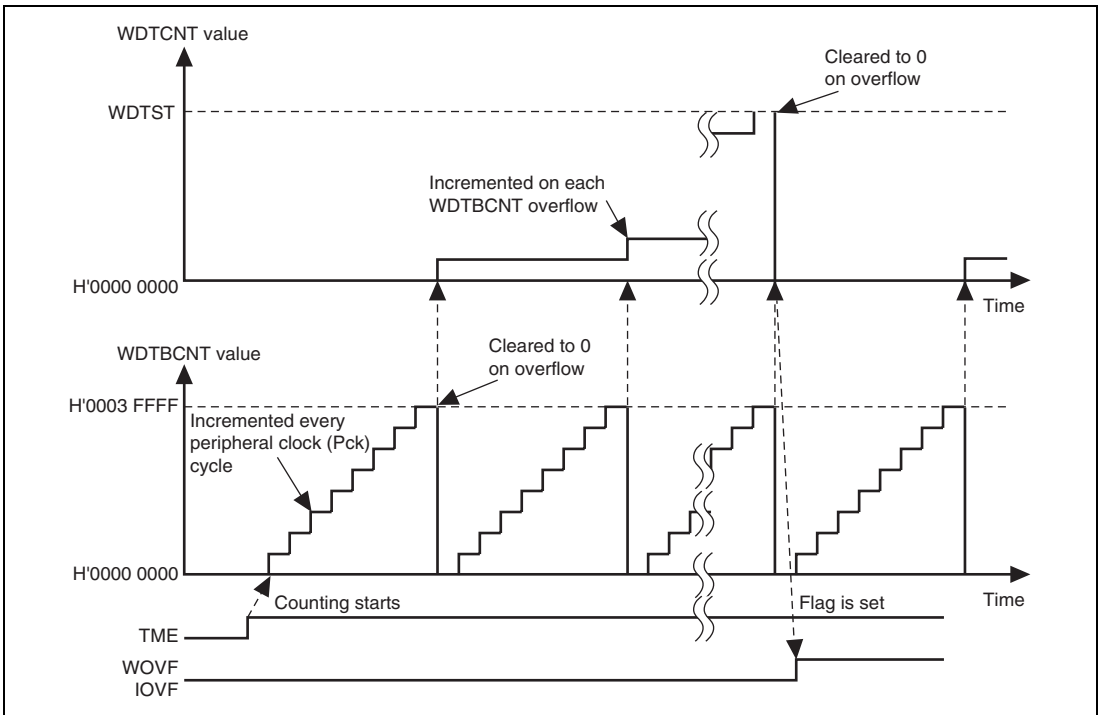
### 29.4.3 Using Interval timer mode

When the WDT is operating in interval timer mode, an interval timer interrupt is generated each time the counter overflows. This enables interrupts to be generated at fixed intervals.

1. Set the WDTCNT overflow time in WDTST.
2. Clear the WT/IT bit in WDTCSR to 0.
3. When the TME bit in WDTCSR is set to 1, the WDT count starts.
4. When the WDTCNT overflows, the WDT sets the IOVF flag in WDTCSR to 1, and sends an interval timer interrupt (ITI) request to INTC. The counter continues counting.

### 29.4.4 Time for WDT Overflow

The relationship between WDTCNT and WDTBCNT is shown in figure 29.2. 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 29.2 WDT Counting Operations (Example in Interval Timer Mode)**

WDTBCNT is a 18-bit up-counter operated on the peripheral clock (Pck). WDTBCNT is cleared when H'55 is set to the bits 31 to 24 in WDTBST.

If the peripheral clock frequency is 50 MHz, the WDTBCNT overflow time is approximately 5.243 ms ( $= 2^{18} [\text{bit}] \times 1/50 [\text{MHz}]$ ).

WDTCNT is a 12-bit counter, starts count up operation when overflow occurs in WDTBCNT. The time until WDTCNT overflows becomes the maximum value when H'000 are set to WDTST.

Where the peripheral clock frequency is 50 MHz, the maximum overflow time is approximately 21.475 s ( $= 2^{12} [\text{bit}] \times 5.243 [\text{ms}]$ ).

And the time until WDTCNT overflows becomes the minimum value when H'001 is set to WDTST. The minimum overflow time is approximately 5.243 ms ( $= 2^1 [\text{bit}] \times 5.243 [\text{ms}]$ ).

### 29.4.5 Clearing WDT Counter

Writing H'55 to WDTBST with longword access clears WDTBCNT and writing the overflow setting value to WDTST clears WDTCNT.

## 29.5 Status Pin Change Timing during Reset

### 29.5.1 Power-On Reset by PRESET

A power-on reset is to initialize the on-chip PLL circuit when this LSI goes to the power-on reset state by the  $\overline{\text{PERSET}}$  pin low level input and then it is necessary to ensure the synchronization settling time of the PLL circuit. Therefore, do not input high level to the  $\overline{\text{PRESET}}$  pin during the synchronization settling time of the PLL. The PLL synchronization settling time is the total value of the PLL1 synchronization settling time and the PLL2 synchronization settling time.

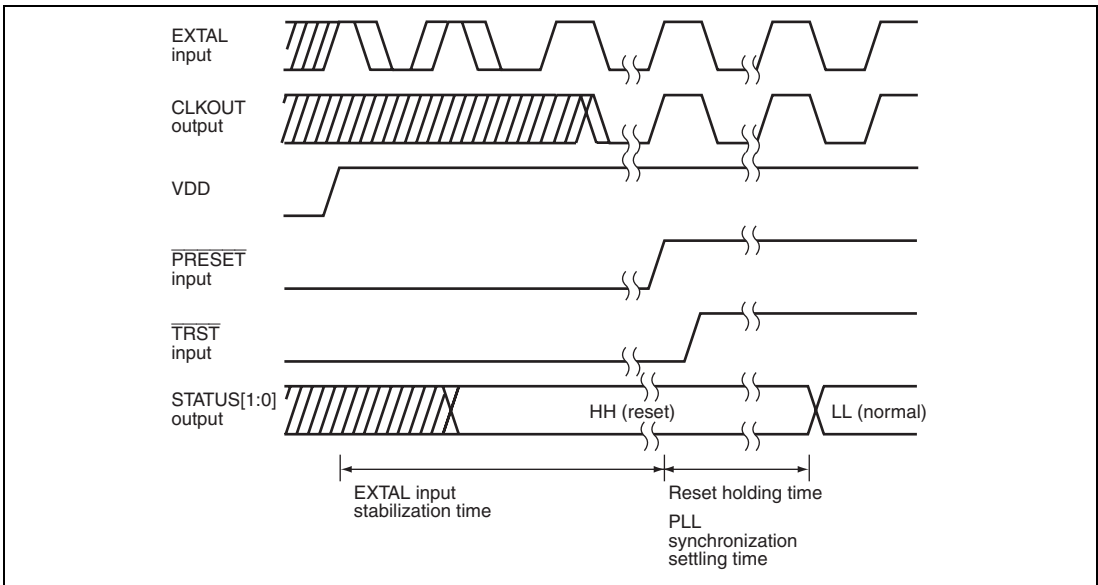
After the  $\overline{\text{PRESET}}$  pin input level is changed from low level to high level, the reset state is continued during the reset holding time in the LSI. The reset holding time is equal to or more than 10240 cycles of the input signal from EXTAL pin.

#### Turning On Power Supply

When turning on the power supply, the  $\overline{\text{PRESET}}$  pin input level should be low level. And the  $\overline{\text{TRST}}$  pin input level should be low level to initialize the H-UDI.

The STATUS [1:0] pins output timing that indicates the reset state is asynchronous, and that indicates a normal operation is synchronous with the peripheral clock (Pck) and asynchronous with both the EXTAL pin input clock and the CLKOUT pin output clock.



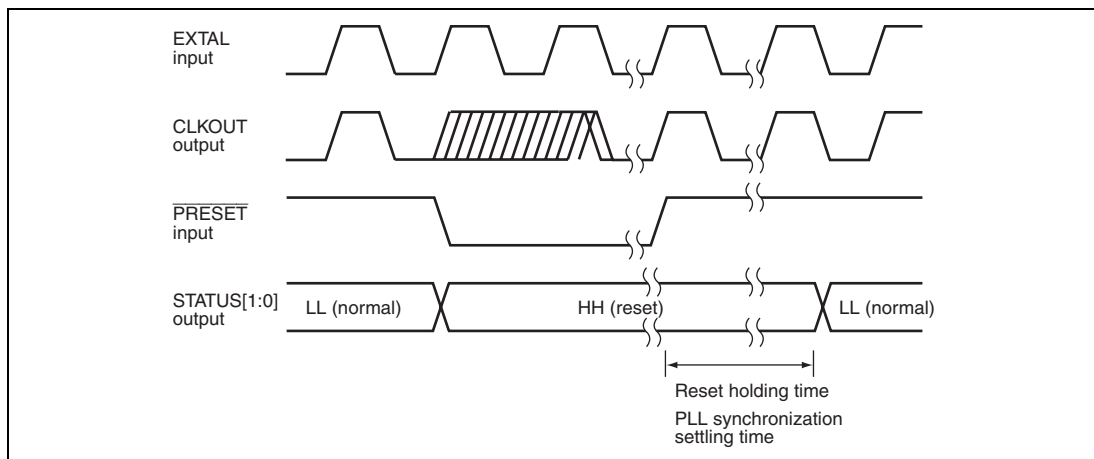


**Figure 29.3 STATUS Output during Power-on**

### **PRESET input during normal operation**

It is necessary to ensure the PLL synchronization settling time when the PRESET input during normal operation.

The STATUS [1:0] pins output timing that indicates the reset state is asynchronous, and that indicates a normal operation is synchronous with the peripheral clock (Pck) and asynchronous with both the EXTAL pin input clock and the CLKOUT pin output clock.

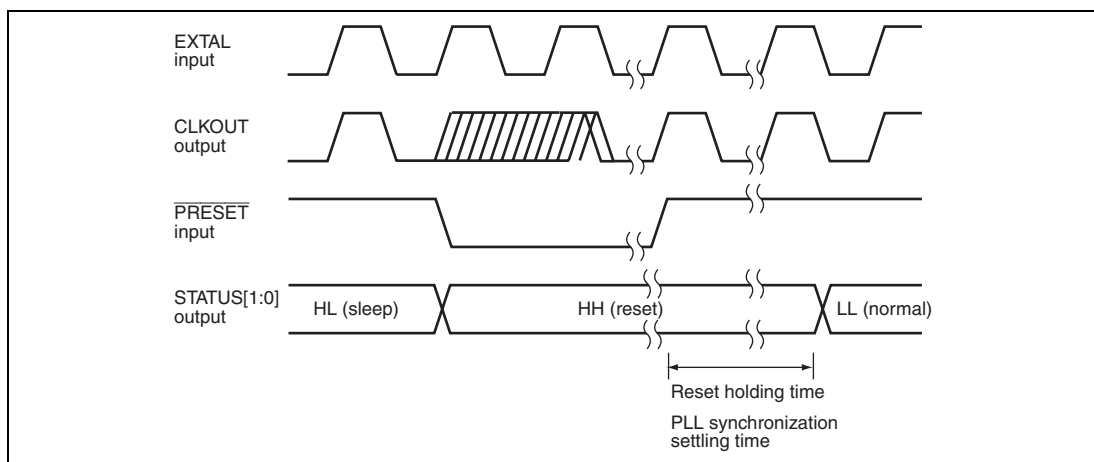


**Figure 29.4 STATUS Output by Reset input during Normal Operation**

### **PRESET input during Sleep Mode**

It is necessary to ensure the PLL oscillation time when power-on reset generates by the PRESET pin low level input during sleep mode.

The STATUS [1:0] pins output timing that indicates the reset state is asynchronous, and that indicates a normal operation is synchronous with the peripheral clock (Pck) and asynchronous with both the EXTAL pin input clock and the CLKOUT pin output clock.



**Figure 29.5 STATUS Output by Reset input during Sleep Mode**

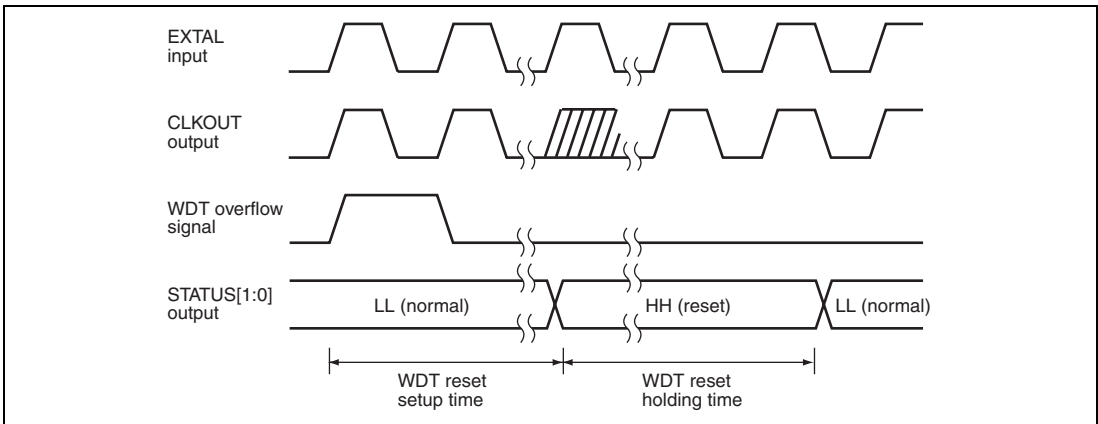
### 29.5.2 Power-On Reset by Watchdog Timer Overflow

The transition time from the watchdog timer overflowed to the power-on reset state (watchdog timer reset setup time) is 9 clock cycle of the EXTAL input clock and thereafter equal to or more than 18 clock cycles of the peripheral clock (Pck).

The power-on reset time (watchdog timer reset holding time) by the watchdog timer overflowed is equal to or more than one cycle of input signal from EXTAL pin and thereafter equal to or more than 5 clock cycles of the peripheral clock (Pck).

#### Power-On Reset by Watchdog timer Overflowed in Normal Operation

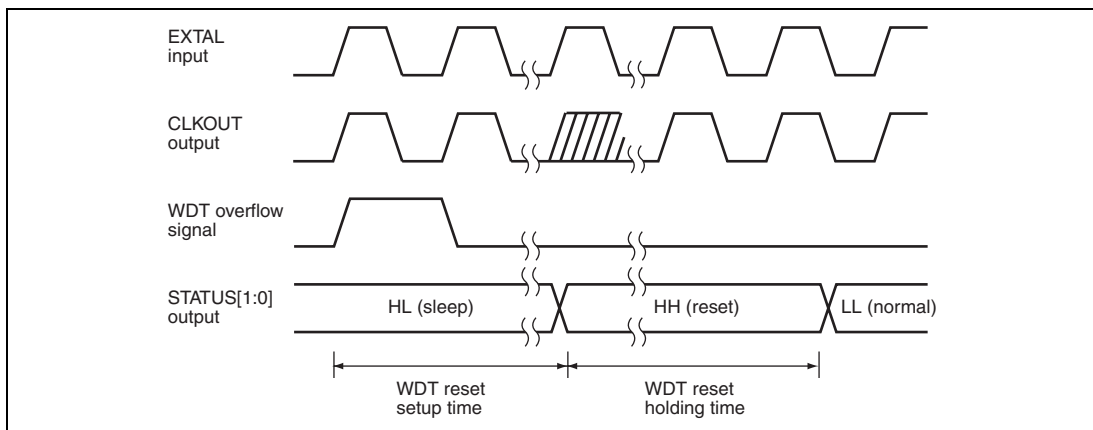
The STATUS [1:0] pins output timing that indicates the reset state or a normal operation is asynchronous with both the EXTAL pin input clock and the CLKOUT pin output clock because the STATUS [1:0] pins output timing is synchronous with the peripheral clock (Pck).



**Figure 29.6 STATUS Output by Watchdog timer overflow Power-On Reset during Normal Operation**

#### Power-On Reset by Watchdog timer Overflowed in Sleep Mode

The STATUS [1:0] pins output timing that indicates the reset state or a normal operation is asynchronous with both the EXTAL pin input clock and the CLKOUT pin output clock because the STATUS [1:0] pins output timing is synchronous with the peripheral clock (Pck).



**Figure 29.7 STATUS Output by Watchdog timer overflow Power-On Reset during Sleep Mode**

## Section 30 User Break Controller (UBC)

The user break controller (UBC) provides versatile functions to facilitate program debugging. These functions help to ease creation of a self-monitor/debugger, which allows easy program debugging using this LSI alone, without using the in-circuit emulator. Various break conditions can be set in the UBC: instruction fetch or read/write access of an operand, operand size, data contents, address value, and program stop timing for instruction fetch.

### 30.1 Features

1. The following break conditions can be set.

Break channels: Two (channels 0 and 1)

User break conditions can be set independently for channels 0 and 1, and can also be set as a single sequential condition for the two channels, that is, a sequential break. (Sequential break involves two cases such that the channel 0 break condition is satisfied in a certain bus cycle and then the channel 1 break condition is satisfied in a different bus cycle, and vice versa.)

- Address

When 40 bits containing ASID and 32-bit address are compared with the specified value, all the ASID bits can be compared or masked.

32-bit address can be masked bit by bit, allowing the user to mask the address in desired page sizes such as lower 12 bits (4-Kbyte page) and lower 10 bits (1-Kbyte page).

- Data

32 bits can be masked only for channel 1.

- Bus cycle

The program can break either for instruction fetch (PC break) or operand access.

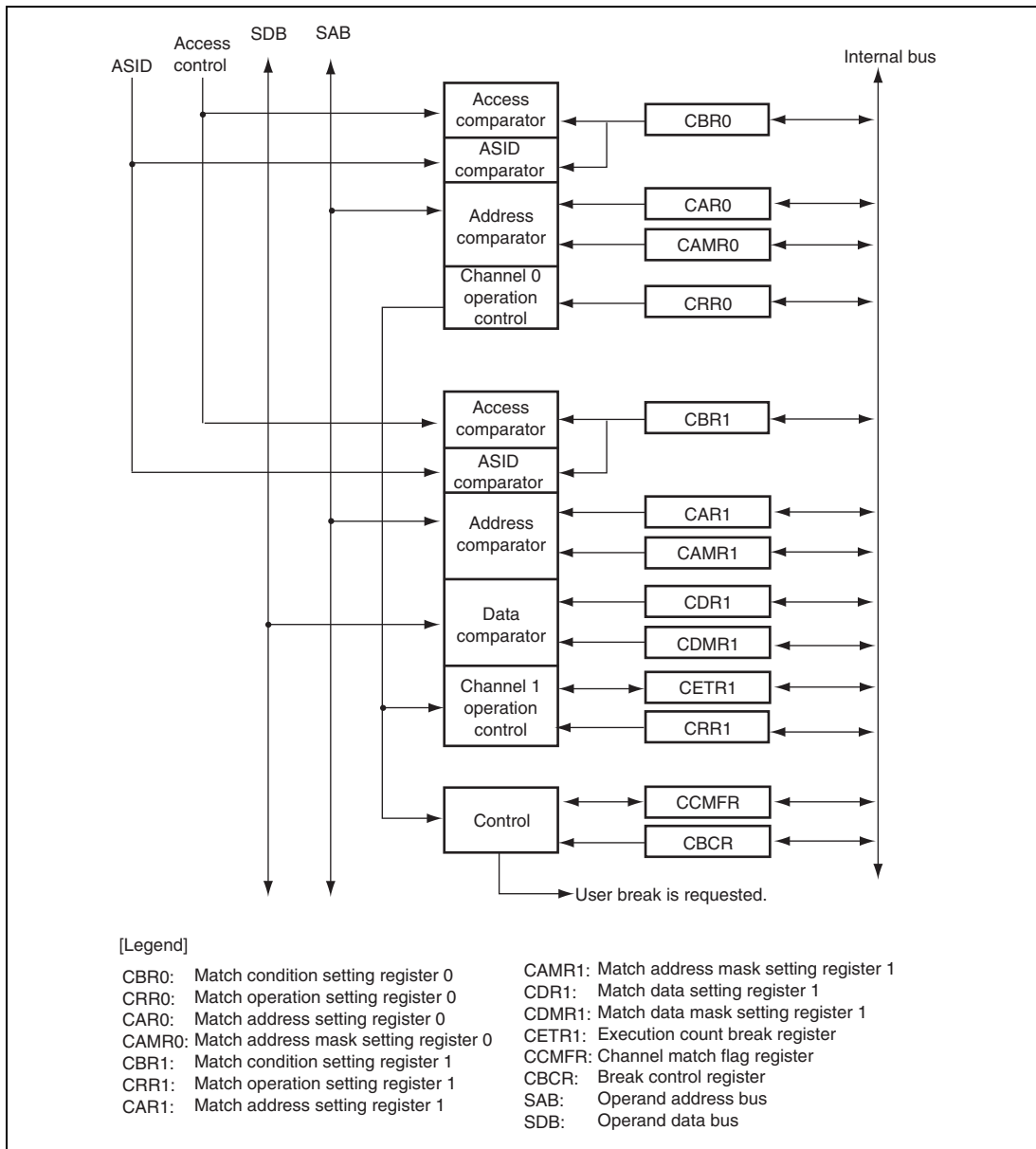
- Read or write access

- Operand sizes

Byte, word, longword, and quadword are supported.

2. The user-designated exception handling routine for the user break condition can be executed.
3. Pre-instruction-execution or post-instruction-execution can be selected as the PC break timing.
4. A maximum of  $2^{12} - 1$  repetition counts can be specified as the break condition (available only for channel 1).

Figure 30.1 shows the UBC block diagram.



**Figure 30.1 Block Diagram of UBC**

## 30.2 Register Descriptions

The UBC has the following registers.

**Table 30.1 Register Configuration**

Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
Match condition setting register 0	CBR0	R/W	H'FF200000	H'1F200000	32
Match operation setting register 0	CRR0	R/W	H'FF200004	H'1F200004	32
Match address setting register 0	CAR0	R/W	H'FF200008	H'1F200008	32
Match address mask setting register 0	CAMR0	R/W	H'FF20000C	H'1F20000C	32
Match condition setting register 1	CBR1	R/W	H'FF200020	H'1F200020	32
Match operation setting register 1	CRR1	R/W	H'FF200024	H'1F200024	32
Match address setting register 1	CAR1	R/W	H'FF200028	H'1F200028	32
Match address mask setting register 1	CAMR1	R/W	H'FF20002C	H'1F20002C	32
Match data setting register 1	CDR1	R/W	H'FF200030	H'1F200030	32
Match data mask setting register 1	CDMR1	R/W	H'FF200034	H'1F200034	32
Execution count break register 1	CETR1	R/W	H'FF200038	H'1F200038	32
Channel match flag register	CCMFR	R/W	H'FF200600	H'1F200600	32
Break control register	CBCR	R/W	H'FF200620	H'1F200620	32

Note: \* P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

**Table 30.2 Register Status in Each Processing State**

<b>Register Name</b>	<b>Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
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
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
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.



### 30.2.1 Match Condition Setting Registers 0 and 1 (CBR0 and CBR1)

CBR0 and CBR1 are readable/writable 32-bit registers which specify the break conditions for channels 0 and 1, respectively. The following break conditions can be set in the CBR0 and CBR1: (1) whether or not to include the match flag in the conditions, (2) whether or not to include the ASID, and the ASID value when included, (3) whether or not to include the data value, (4) operand size, (5) whether or not to include the execution count, (6) bus type, (7) instruction fetch cycle or operand access cycle, and (8) read or write access cycle.

- CBR0

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

Bit	Bit Name	Initial Value	R/W	Description
31	MFE	0	R/W	<b>Match Flag Enable</b>  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	<b>ASID Enable</b>  Specifies whether or not to include the ASID specified by the AIV bit of this register in the match conditions.  0: The ASID is not included in the match conditions; thus, not checked.  1: The ASID is included in the match conditions.

Bit	Bit Name	Initial Value	R/W	Description
29 to 24	MFI	100000	R/W	<p>Match Flag Specify</p> <p>Specifies the match flag to be included in the match conditions.</p> <p>000000: MF0 bit of the CCMFR register</p> <p>000001: MF1 bit of the CCMFR register</p> <p>Others: Reserved (setting prohibited)</p> <p>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.</p>
23 to 16	AIV	All 0	R/W	<p>ASID Specify</p> <p>Specifies the ASID value to be included in the match conditions.</p>
15	—	0	R	<p>Reserved</p> <p>For read/write in this bit, refer to General Precautions on Handling of Product.</p>
14 to 12	SZ	All 0	R/W	<p>Operand Size Select</p> <p>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.</p> <p>000: The operand size is not included in the match conditions; thus, not checked (any operand size specifies the match condition).<sup>*1</sup></p> <p>001: Byte access</p> <p>010: Word access</p> <p>011: Longword access</p> <p>100: Quadword access<sup>*2</sup></p> <p>Others: Reserved (setting prohibited)</p>
11 to 8	—	All 0	R	<p>Reserved</p> <p>For read/write in this bit, refer to General Precautions on Handling of Product.</p>

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD	All 0	R/W	<b>Bus Select</b> Specifies the bus to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 00: Operand bus for operand access Others: Reserved (setting prohibited)
5, 4	ID	All 0	R/W	<b>Instruction Fetch/Operand Access Select</b> 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	<b>Reserved</b> For read/write in this bit, refer to General Precautions on Handling of Product.
2, 1	RW	All 0	R/W	<b>Bus Command Select</b> 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	<b>Channel Enable</b> 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.

- CBR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFE	AIE	MFI						AIV							
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	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	<b>Match Flag Enable</b>  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	<b>ASID Enable</b>  Specifies whether or not to include the ASID specified by the AIV bit of this register in the match conditions.  0: The ASID is not included in the match conditions; thus, not checked.  1: The ASID is included in the match conditions.
29 to 24	MFI	100000	R/W	<b>Match Flag Specify</b>  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)  <b>Note:</b> 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 CCRM.FMF1 = 0.

Bit	Bit Name	Initial Value	R/W	Description
23 to 16	AIV	All 0	R/W	ASID Specify Specifies the ASID value to be included in the match conditions.
15	DBE	0	R/W	Data Value Enable* <sup>3</sup> Specifies whether or not to include the data value in the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 0: The data value is not included in the match conditions; thus, not checked. 1: The data value is included in the match conditions.
14 to 12	SZ	All 0	R/W	Operand Size Select Specifies the operand size to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 000: The operand size is not included in the match condition; thus, not checked (any operand size specifies the match condition). * <sup>1</sup> 001: Byte access 010: Word access 011: Longword access 100: Quadword access* <sup>2</sup> Others: Reserved (setting prohibited)
11	ETBE	0	R/W	Execution Count Value Enable Specifies whether or not to include the execution count value in the match conditions. If this bit is 1 and the match condition satisfaction count matches the value specified by the CETR1 register, the operation specified by the CRR1 register is performed. 0: The execution count value is not included in the match conditions; thus, not checked. 1: The execution count value is included in the match conditions.
10 to 8	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD	All 0	R/W	<b>Bus Select</b> Specifies the bus to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 00: Operand bus for operand access Others: Reserved (setting prohibited)
5, 4	ID	All 0	R/W	<b>Instruction Fetch/Operand Access Select</b> 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	<b>Reserved</b> For read/write in this bit, refer to General Precautions on Handling of Product.
2, 1	RW	All 0	R/W	<b>Bus Command Select</b> 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	<b>Channel Enable</b> Validates/invalidates the channel. If this bit is 0, all the other bits in this register are invalid. 0: Invalidates the channel. 1: Validates the channel.

- Notes:
1. If the data value is included in the match conditions, be sure to specify the operand size.
  2. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and the match data mask setting register.

3. The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions for these instructions.

### 30.2.2 Match Operation Setting Registers 0 and 1 (CRR0 and CRR1)

CRR0 and CRR1 are readable/writable 32-bit registers which specify the operation to be executed when channels 0 and 1 satisfy the match condition, respectively. The following operations can be set in the CRR0 and CRR1 registers: (1) breaking at a desired timing for the instruction fetch cycle and (2) requesting a break.

#### • CRR0

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
13	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
12 to 2	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
1	PCB	0	R/W	<b>PC Break Select</b> 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	<b>Break Enable</b> 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	<b>Reserved</b> For read/write in this bit, refer to General Precautions on Handling of Product.
13	—	1	R	<b>Reserved</b> This bit is always read as 1. The write value should always be 1.
12 to 2	—	All 0	R	<b>Reserved</b> For read/write in this bit, refer to General Precautions on Handling of Product.



Bit	Bit Name	Initial Value	R/W	Description
1	PCB	0	R/W	<b>PC Break Select</b> 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	<b>Break Enable</b> Specifies whether or not to request a break when the match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

### 30.2.3 Match Address Setting Registers 0 and 1 (CAR0 and CAR1)

CAR0 and CAR1 are readable/writable 32-bit registers specifying the virtual address to be included in the break conditions for channels 0 and 1, respectively.

- CAR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CA															
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
	CA															
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	CA	Undefined	R/W	<b>Compare Address</b> 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 :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CA															
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
	CA															
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	CA	Undefined	R/W	<p>Compare Address</p> <p>Specifies the address to be included in the break conditions.</p> <p>When the operand bus has been specified using the CBR1 register, specify the SAB address in CA[31:0].</p>

### 30.2.4 Match Address Mask Setting Registers 0 and 1 (CAMR0 and CAMR1)

CMAR0 and CMAR1 are readable/writable 32-bit registers which specify the bits to be masked among the address bits specified by using the match address setting register of the corresponding channel. (Set the bits to be masked to 1.)

- CAMR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CAM															
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
	CAM															
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	CAM	Undefined	R/W	<p>Compare Address Mask</p> <p>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.)</p> <p>0: Address bits CA[n] are included in the break condition.</p> <p>1: Address bits CA[n] are masked and not included in the break condition.</p> <p>[n] = any values from 31 to 0</p>

### • CAMR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CAM															
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
	CAM															
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	CAM	Undefined	R/W	<p>Compare Address Mask</p> <p>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.)</p> <p>0: Address bits CA[n] are included in the break condition.</p> <p>1: Address bits CA[n] are masked and not included in the break condition.</p> <p>[n] = any values from 31 to 0</p>

### 30.2.5 Match Data Setting Register 1 (CDR1)

CDR1 is a readable/writable 32-bit register which specifies the data value to be included in the break conditions for channel 1.

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CD	Undefined	R/W	Compare Data Value Specifies the data value to be included in the break conditions. When the operand bus has been specified using the CBR1 register, specify the SDB data value in CD[31:0].

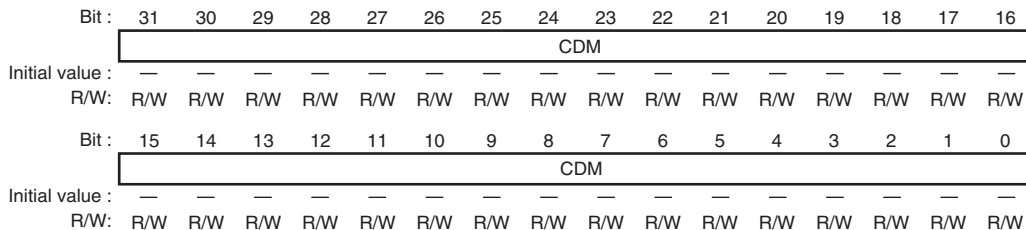
**Table 30.3 Settings for Match Data Setting Register**

Bus and Size Selected Using CBR1	CD[31:24]	CD[23:16]	CD[15:8]	CD[7:0]
Operand bus (byte)	Don't care	Don't care	Don't care	SDB7 to SDB0
Operand bus (word)	Don't care	Don't care	SDB15 to SDB8	SDB7 to SDB0
Operand bus (longword)	SDB31 to SDB24	SDB23 to SDB16	SDB15 to SDB8	SDB7 to SDB0

- Notes:
1. If the data value is included in the match conditions, be sure to specify the operand size.
  2. The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions for these instructions.
  3. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and match data mask setting register.

### 30.2.6 Match Data Mask Setting Register 1 (CDMR1)

CDMR1 is a readable/writable 32-bit register which specifies the bits to be masked among the data value bits specified using the match data setting register. (Set the bits to be masked to 1.)



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CDM	Undefined	R/W	<p>Compare Data Value Mask</p> <p>Specifies the bits to be masked among the data value bits specified using the CDR1 register. (Set the bits to be masked to 1.)</p> <p>0: Data value bits CD[n] are included in the break condition.</p> <p>1: Data value bits CD[n] are masked and not included in the break condition.</p> <p>[n] = any values from 31 to 0</p>

### 30.2.7 Execution Count Break Register 1 (CETR1)

CETR1 is a readable/writable 32-bit register which specifies the number of the channel hits before a break occurs. A maximum value of  $2^{12} - 1$  can be specified. When the execution count value is included in the match conditions by using the match condition setting register, the value of this register is decremented by one every time the channel is hit. When the channel is hit after the register value reaches H'001, a break occurs.

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
11 to 0	CET	Undefined	R/W	Execution Count Specifies the execution count to be included in the break conditions.

### 30.2.8 Channel Match Flag Register (CCMFR)

CCMFR is a readable/writable 32-bit register which indicates whether or not the match conditions have been satisfied for each channel. When a channel match condition has been satisfied, the corresponding flag bit is set to 1. To clear the flags, write the data containing value 0 for the bits to be cleared and value 1 for the other bits to this register. (The logical AND between the value which has been written and the current register value is actually written to the register.)

Sequential operation using multiple channels is available by using these match flags.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MF1	MF0
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
1	MF1	0	R/W	Channel 1 Condition Match Flag This flag is set to 1 when the channel 1 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 1 match condition has not been satisfied. 1: Channel 1 match condition has been satisfied.
0	MF0	0	R/W	Channel 0 Condition Match Flag This flag is set to 1 when the channel 0 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 0 match condition has not been satisfied. 1: Channel 0 match condition has been satisfied.

### 30.2.9 Break Control Register (CBCR)

CBCR is a readable/writable 32-bit register which specifies whether or not to use the user break debugging support function. For details on the user break debugging support function, refer to section 30.4, User Break Debugging Support Function.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved  For read/write in this bit, refer to General Precautions on Handling of Product.
0	UBDE	0	R/W	User Break Debugging Support Function Enable  Specifies whether or not to use the user break debugging support function.  0: Does not use the user break debugging support function.  1: Uses the user break debugging support function.



## 30.3 Operation Description

### 30.3.1 Definition of Words Related to Accesses

"Instruction fetch" refers to an access in which an instruction is fetched. For example, fetching the instruction located at the branch destination after executing a branch instruction is an instruction access. "Operand access" refers to any memory access accompanying execution of an instruction. For example, accessing an address ( $PC + disp \times 2 + 4$ ) in the instruction `MOV.W@(disp,PC),Rn` is an operand access. "Data" is used in contrast to "address".

All types of operand access are classified into read or write access. Special care must be taken in using the following instructions.

- `PREF`, `OCBP`, and `OCBWB`: Instructions for a read access
- `MOVCA.L` and `OCBI`: Instructions for a write access
- `TAS.B`: Instruction for a single read access or a single write access

The operand access accompanying the `PREF`, `OCBP`, `OCBWB`, and `OCBI` instructions is access without the data value; therefore, do not include the data value in the match conditions for these instructions.

The operand size should be defined for all types of operand access. Available operand sizes are byte, word, longword, and quadword. For operand access accompanying the `PREF`, `OCBP`, `OCBWB`, `MOVCA.L`, and `OCBI` instructions, the operand size is defined as longword.

### 30.3.2 User Break Operation Sequence

The following describes the sequence from when the break condition is set until the user break exception handling is initiated.

1. Specify the operand size, bus, instruction fetch/operand access, and read/write as the match conditions using the match condition setting register (`CBR0` or `CBR1`). Specify the break address using the match address setting register (`CAR0` or `CAR1`), and specify the address mask condition using the match address mask setting register (`CAMR0` or `CAMR1`). To include the `ASID` in the match conditions, set the `AIE` bit in the match condition setting register and specify the `ASID` value by the `AIV` bit in the same register. To include the data value in the match conditions, set the `DBE` bit in the match condition setting register; specify the break data using the match data setting register (`CDR1`); and specify the data mask condition using the match data mask setting register (`CDMR1`). To include the execution count in the match conditions, set the `ETBE` bit of the match condition setting register; and

specify the execution count using the execution count break register (CETR1). To use the sequential break, set the MFE bit of the match condition setting register; and specify the number of the first channel using the MFI bit.

2. Specify whether or not to request a break when the match condition is satisfied and the break timing when the match condition is satisfied as a result of fetching the instruction using the match operation setting register (CRR0 or CRR1). After having set all the bits in the match condition setting register except the CE bit and the other necessary registers, set the CE bit and read the match condition setting register again. This ensures that the set values in the control registers are valid for the subsequent instructions immediately after reading the register. Setting the CE bit of the match condition setting register in the initial state after reset via the control registers may cause an undesired break.
3. When the match condition has been satisfied, the corresponding condition match flag (MF1 or MF0) in the channel match flag register (CCMFR) is set. A break is also requested to the CPU according to the set values in the match operation setting register (CRR0 or CRR1). The CPU operates differently according to the BL bit value of the SR register: when the BL bit is 0, the CPU accepts the break request and executes the specified exception handling; and when the BL bit is 1, the CPU does not execute the exception handling.
4. The match flags (MF1 and MF0) can be used to confirm whether or not the corresponding match condition has been satisfied. Although the flag is set when the condition is satisfied, it is not cleared automatically; therefore, write 0 to the flag bit by issuing a memory store instruction to the channel match flag register (CCMFR) in order to use the flag again.
5. Breaks may occur virtually at the same time for channels 0 and 1. In this case, only one break request is sent to the CPU; however, the two condition match flags corresponding to these breaks may be set.
6. While the BL bit in the SR register is 1, no break requests are accepted. However, whether or not the condition has been satisfied is determined. When the condition is determined to be satisfied, the corresponding condition match flag is set.
7. If the sequential break conditions are set, the condition match flag is set every time the match conditions are satisfied for each channel. When the conditions have been satisfied for the first channel in the sequence but not for the second channel in the sequence, clear the condition match flag for the first channel in the sequence in order to release the first channel in the sequence from the match state.

### 30.3.3 Instruction Fetch Cycle Break

1. If the instruction fetch cycle is set in the match condition setting register (CBR0 or CBR1), the instruction fetch cycle is handled as a match condition. To request a break upon satisfying the match condition, set the BIE bit in the match operation setting register (CRR0 or CRR1) of the corresponding channel. Either before or after executing the instruction can be selected as the

break timing according to the PCB bit value. If the instruction fetch cycle is specified as a match condition, be sure to clear the LSB to 0 in the match address setting register (CAR0 or CAR1); otherwise, no break occurs.

2. If pre-instruction-execution break is specified for the instruction fetch cycle, the break is requested when the instruction is fetched and determined to be executed. Therefore, this function cannot be used for the instructions which are fetched through overrun (i.e., the instructions fetched during branching or making transition to the interrupt routine but not executed). For priorities of pre-instruction-execution break and the other exceptions, refer to section 5, Exception Handling. If pre-instruction-execution break is specified for the delayed slot of the delayed branch instruction, the break is requested before the delayed branch instruction is executed. However, do not specify pre-instruction-execution break for the delayed slot of the RTE instruction.
3. If post-instruction-execution break is specified for the instruction fetch cycle, the break is requested after the instruction which satisfied the match condition has been executed and before the next instruction is executed. Similar to pre-instruction-execution break, this function cannot be used for the instructions which are fetched through overrun. For priorities of post-instruction-execution break and the other exceptions, refer to section 5, Exception Handling. If post-instruction-execution break is specified for the delayed branch instruction and its delayed slot, the break does not occur until the first instruction at the branch destination.
4. If the instruction fetch cycle is specified as the channel 1 match condition, the DBE bit of match condition setting register CBR1 becomes invalid, the settings of match data setting register CDR1 and match data mask setting register CDMR1 are ignored. Therefore, the data value cannot be specified for the instruction fetch cycle break.

### 30.3.4 Operand Access Cycle Break

1. Table 30.4 shows the relation between the operand sizes specified using the match condition setting register (CBR0 or CBR1) and the address bits to be compared for the operand access cycle break.

**Table 30.4 Relation between Operand Sizes and Address Bits to be Compared**

Selected Operand Size	Address Bits to be Compared
Quadword	Address bits A31 to A3
Longword	Address bits A31 to A2
Word	Address bits A31 to A1
Byte	Address bits A31 to A0
Operand size is not included in the match conditions	Address bits A31 to A3 for quadword access
	Address bits A31 to A2 for longword access
	Address bits A31 to A1 for word access
	Address bits A31 to A0 for byte access

The above table means that if address H'00001003 is set in the match address setting register (CAR0 or CAR1), for example, the match condition is satisfied for the following access cycles (assuming that all the other conditions are satisfied):

- Longword access to address H'00001000
- Word access to address H'00001002
- Byte access to address H'00001003

2. When the data value is included in the channel 1 match conditions:

If the data value is included in the match conditions, be sure to select the quadword, longword, word, or byte as the operand size using the operand size select bit (SZ) of the match condition setting register (CBR1), and also set the match data setting register (CDR1) and the match data mask setting register (CDMR1). With these settings, the match condition is satisfied when both of the address and data conditions are satisfied. The data value and mask control for byte access, word access, and longword access should be set in bits 7 to 0, 15 to 0, and 31 to 0 in the bits CDR1 and CDMR1, respectively. For quadword access, 64-bit data is divided into the upper and lower 32-bit data units, and each unit is independently compared with the specified condition. When either the upper or lower 32-bit data unit satisfies the match condition, the match condition for the 64-bit data is determined to be satisfied.

3. The operand access accompanying the PREF, OCBP, OCBWB, and OCBI instructions are access without the data value; therefore, if the data value is included in the match conditions for these instructions, the match conditions will never be satisfied.
4. If the operand bus is selected, a break occurs after executing the instruction which has satisfied the conditions and immediately before executing the next instruction. However, if the data value is included in the match conditions, a break may occur after executing several instructions after the instruction which has satisfied the conditions; therefore, it is impossible to identify the instruction causing the break. If such a break has occurred for the delayed branch instruction or its delayed slot, the break does not occur until the first instruction at the branch destination.

However, do not specify the operand break for the delayed slot of the RTE instruction. And if the data value is included in the match conditions, it is not allowed to set the break for the preceding the RTE instruction by one to six instructions.

### 30.3.5 Sequential Break

1. Sequential break conditions can be specified by setting the MFE and MFI bits in the match condition setting registers (CBR0 and CBR1). (Sequential break involves two cases such that channel 0 break condition is satisfied then channel 1 break condition is satisfied, and vice versa.) To use the sequential break function, clear the MFE bit of the match condition setting register and the BIE bit of the match operation setting register of the first channel in the sequence, and set the MFE bit and specify the number of the second channel in the sequence using the MFI bit in the match condition setting register of the second channel in the sequence. If the sequential break condition is set, the condition match flag is set every time the match condition is satisfied for each channel. When the condition has been satisfied for the first channel in the sequence but not for the second channel in the sequence, clear the condition match flag for the first channel in the sequence in order to release the first channel in the sequence from the match state.
2. For channel 1, the execution count break condition can also be included in the sequential break conditions.
3. If the match conditions for the first and second channels in the sequence are satisfied within a significantly short time, sequential operation may not be guaranteed in some cases, as shown below.

- When the Match Condition is Satisfied at the Instruction Fetch Cycle for Both the First and Second Channels in the Sequence:

Instruction B is 0 instruction after instruction A	Equivalent to setting the same addresses; do not use this setting.
Instruction B is one instruction after instruction A	Sequential operation is not guaranteed.
Instruction B is two or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the instruction fetch cycle for the first channel in the sequence whereas the match condition is satisfied at the operand access cycle for the second channel in the sequence:

Instruction B is 0 or one instruction after instruction A	Sequential operation is not guaranteed.
Instruction B is two or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the operand access cycle for the first channel in the sequence whereas the match condition is satisfied at the instruction fetch cycle for the second channel in the sequence:

Instruction B is 0 to five instructions after instruction A	Sequential operation is not guaranteed.
Instruction B is six or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the operand access cycle for both the first and second channels in the sequence:

Instruction B is 0 to five instructions after instruction A	Sequential operation is not guaranteed.
Instruction B is six or more instructions after instruction A	Sequential operation is guaranteed.

### 30.3.6 Program Counter Value to be Saved

When a break has occurred, the address of the instruction to be executed when the program restarts is saved in the SPC then the exception handling state is initiated. A unique instruction causing a break can be identified unless the data value is included in the match conditions.

- When the instruction fetch cycle (before instruction execution) is specified as the match condition:

The address of the instruction which has satisfied the match conditions is saved in the SPC. The instruction which has satisfied the match conditions is not executed, but a break occurs instead. However, if the match conditions are satisfied for the delayed slot instruction, the address of the delayed branch instruction is saved in the SPC.

- When the instruction fetch cycle (after instruction execution) is specified as the match condition:

The address of the instruction immediately after the instruction which has satisfied the match conditions is saved in the SPC. The instruction which has satisfied the match conditions is executed, then a break occurs before the next instruction. If the match conditions are satisfied for the delayed branch instruction or its delayed slot, these instructions are executed and the address of the branch destination is saved in the SPC.

- When the operand access (address only) is specified as the match condition:

The address of the instruction immediately after the instruction which has satisfied the break conditions is saved in the SPC. The instruction which has satisfied the match conditions are executed, then a break occurs before the next instruction. However, if the conditions are satisfied for the delayed slot, the address of the branch destination is saved in the SPC.

- When the operand access (address and data) is specified as the match condition:

If the data value is added to the match conditions, the instruction which has satisfied the match conditions is executed. A user break occurs before executing an instruction that is one through six instructions after the instruction which has satisfied the match conditions. The address of the instruction is saved in the SPC; thus, it is impossible to identify exactly where a break will occur. If the conditions are satisfied for the delayed slot instruction, the address of the branch destination is saved in the SPC. If a branch instruction follows the instruction which has satisfied the match conditions, a break may occur after the delayed instruction and delayed slot are executed. In this case, the address of the branch destination is also saved in the SPC.

## 30.4 User Break Debugging Support Function

By using the user break debugging support function, the branch destination address can be modified when the CPU accepts the user break request. Specifically, setting the UBDE bit of break control register CBCR to 1 allows branching to the address indicated by DBR instead of branching to the address indicated by the [VBR + offset]. Figure 30.2 shows the flowchart of the user break debugging support function.



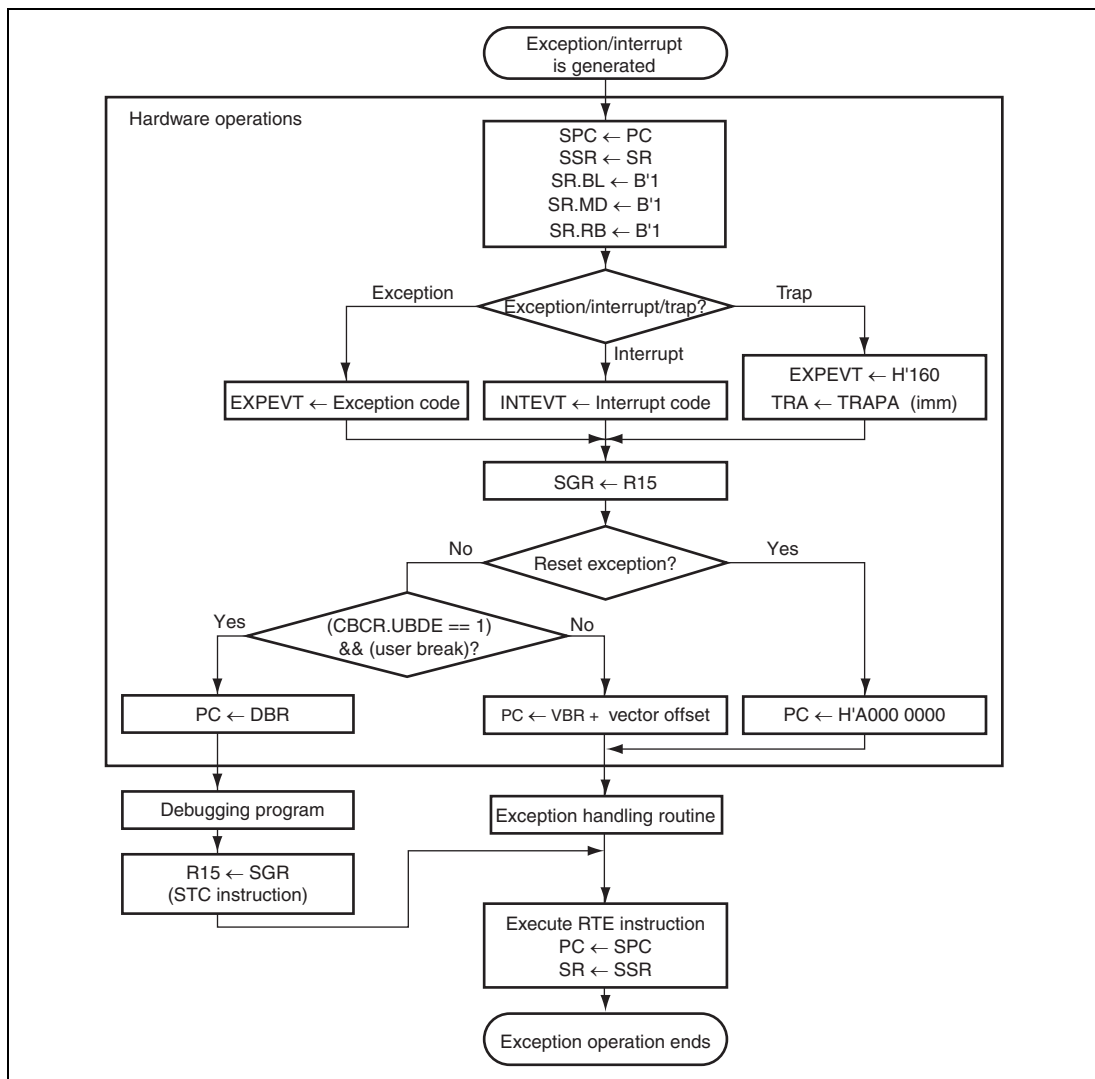


Figure 30.2 Flowchart of User Break Debugging Support Function

## 30.5 User Break Examples

### (1) Match Conditions are Specified for an Instruction Fetch Cycle

- Example 1-1

Register settings: CBR0 = H'00000013 / CRR0 = H'00002003 / CAR0 = H'00000404 /  
 CAMR0 = H'00000000 / CBR1 = H'00000013 / CRR1 = H'00002001 / CAR1 = H'00008010 /  
 CAMR1 = H'00000006 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 =  
 H'00000000 / CBCR = H'00000000

Specified conditions: Independent for channels 0 and 1

— Channel 0

Address: H'00000404 / Address mask: H'00000000

Bus cycle: Instruction fetch (after executing the instruction)

ASID is not included in the conditions.

— Channel 1:

Address: H'00008010 / Address mask: H'00000006

Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

Bus cycle: Instruction fetch (before executing instruction)

ASID, data values, and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00000404 or before executing the instruction at address H'00008010 to H'00008016.

- Example 1-2

Register settings: CBR0 = H'40800013 / CRR0 = H'00002000 / CAR0 = H'00037226 /  
 CAMR0 = H'00000000 / CBR1 = H'C0700013 / CRR1 = H'00002001 / CAR1 = H'0003722E /  
 CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 =  
 H'00000000 / CBCR = H'00000000

Specified conditions: Channel 0 → Channel1 sequential mode

— Channel 0

Address: H'00037226 / Address mask: H'00000000 / ASID: H'80

Bus cycle: Instruction fetch (before executing the instruction)

— Channel 1

Address: H'0003722E / Address mask: H'00000000 / ASID: H'70

Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00037226 where ASID is H'80 before executing the instruction at address H'0003722E where ASID is H'70.

- Example 1-3

Register settings: CBR0 = H'00000013 / CRR0 = H'00002001 / CAR0 = H'00027128 / CAMR0 = H'00000000 / CBR1 = H'00000013 / CRR1 = H'00002001 / CAR1 = H'00031415 / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

Specified conditions: Independent for channels 0 and 1

- Channel 0

Address: H'00027128 / Address mask: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

ASID is not included in the conditions.

- Channel 1

Address: H'00031415 / Address mask: H'00000000

Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

ASID, data values, and execution count are not included in the conditions.

With the above settings, the user break occurs for channel 0 before executing the instruction at address H'00027128. No user break occurs for channel 1 since the instruction fetch is executed only at even addresses.

- Example 1-4

Register settings: CBR0 = H'40800013 / CRR0 = H'00002000 / CAR0 = H'00037226 / CAMR0 = H'00000000 / CBR1 = H'C0700013 / CRR1 = H'00002001 / CAR1 = H'0003722E / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

Specified conditions: Channel 0 → Channel 1 sequential mode

- Channel 0

Address: H'00037226 / Address mask: H'00000000 / ASID: H'80

Bus cycle: Instruction fetch (before executing the instruction)

- Channel 1

Address: H'0003722E / Address mask: H'00000000 / ASID: H'70

Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00037226 where ASID is H'80 and before executing the instruction at address H'0003722E where ASID is H'70.

- Example 1-5

Register settings: CBR0 = H'00000013 / CRR0 = H'00002001 / CAR0 = H'00000500 / CAMR0 = H'00000000 / CBR1 = H'00000813 / CRR1 = H'00002001 / CAR1 = H'00001000 / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000005 / CBCR = H'00000000

Specified conditions: Independent for channels 0 and 1

- Channel 0

Address: H'00000500 / Address mask: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

ASID is not included in the conditions.

- Channel 1

Address: H'00001000 / Address mask: H'00000000

Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000005

Bus cycle: Instruction fetch (before executing the instruction)

Execution count: 5

ASID and data values are not included in the conditions.

With the above settings, the user break occurs for channel 0 before executing the instruction at address H'00000500. The user break occurs for channel 1 after executing the instruction at address H'00001000 four times; before executing the instruction five times.

- Example 1-6

Register settings: CBR0 = H'40800013 / CRR0 = H'00002003 / CAR0 = H'00008404 / CAMR0 = H'00000FFF / CBR1 = H'40700013 / CRR1 = H'00002001 / CAR1 = H'00008010 / CAMR1 = H'00000006 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

Specified conditions: Independent for channels 0 and 1

- Channel 0

Address: H'00008404 / Address mask: H'00000FFF / ASID: H'80

Bus cycle: Instruction fetch (after executing the instruction)

- Channel 1

Address: H'00008010 / Address mask: H'00000006 / ASID: H'70

Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00008000 to H'00008FFE where ASID is H'80 or before executing the instruction at address H'00008010 to H'00008016 where ASID is H'70.

## (2) Match Conditions are Specified for an Operand Access Cycle

- Example 2-1

Register settings: CBR0 = H'40800023 / CRR0 = H'00002001 / CAR0 = H'00123456 / CAMR0 = H'00000000 / CBR1 = H'4070A025 / CRR1 = H'00002001 / CAR1 = H'000ABCDE / CAMR1 = H'000000FF / CDR1 = H'0000A512 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

Specified conditions: Independent for channels 0 and 1

— Channel 0

Address: H'00123456 / Address mask: H'00000000 / ASID: H'80

Bus cycle: Operand bus, operand access, and read (operand size is not included in the conditions.)

— Channel 1

Address: H'000ABCDE / Address mask: H'000000FF / ASID: H'70

Data: H'0000A512 / Data mask: H'00000000 / Execution count: H'00000000

Bus cycle: Operand bus, operand access, write, and word size

Execution count is not included in the conditions.

With these settings, the user break occurs for channel 0 for the following accesses: longword read access to address H'000123454, word read access to address H'000123456, byte read access to address H'000123456 where ASID is H'80. The user break occurs for channel 1 when word H'A512 is written to address H'000ABC00 to H'000ABC0E where ASID is H'70.

## 30.6 Usage Notes

- A desired break may not occur between the time when the instruction for rewriting the UBC register is executed and the time when the written value is actually reflected on the register. After the UBC register is updated, execute one of the following three methods.
  - A. Read the updated UBC register, and execute a branch using the RTE instruction.  
(It is not necessary that a branch using the RTE instruction 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 IRMC.R1 before updating the UBC register and update with following sequence.
    - a. Write the UBC register.
    - b. Read the UBC register which is updated at 1.
    - c. Write the value which is read at 2 to the UBC register.

**Note:** When two or more UBC registers are updated, executing these methods at each updating the UBC registers is not necessary. At only last updating the UBC register, execute one of these methods.

- The PCB bit of the CRR0 and CRR1 registers is valid only when the instruction fetch is specified as the match condition.
- If the sequential break conditions are set, the sequential break conditions are satisfied when the conditions for the first and second channels in the sequence are satisfied in this order. Therefore, if the conditions are set so that the conditions for channels 0 and 1 should be satisfied simultaneously for the same bus cycle, the sequential break conditions will not be satisfied, causing no break.
- For the SLEEP instruction, do not allow the post-instruction-execution break where the instruction fetch cycle is the match condition. For the instructions preceding the SLEEP instruction by one to five instructions, do not allow the break where the operand access is the match condition.
- If the user break and other exceptions occur for the same instruction, they are determined according to the specified priority. For the priority, refer to section 5, Exception Handling. If the exception having the higher priority occurs, the user break does not occur.
  - The pre-instruction-execution break is accepted prior to any other exception.

- If the post-instruction-execution break and data access break have occurred simultaneously with the re-execution type exception (including the pre-instruction-execution break) having a higher priority, only the re-execution type exception is accepted, and no condition match flags are set. When the exception handling has finished thus clearing the exception source, and when the same instruction has been executed again, the break occurs setting the corresponding flag.
- If the post-instruction-execution break or operand access break has occurred simultaneously with the completion-type exception (TRAPA) having a higher priority, then no user break occurs; however, the condition match flag is set.
- When conditions have been satisfied simultaneously and independently for channels 0 and 1, resulting in identical SPC values for both of the breaks, the user break occurs only once. However, the condition match flags are set for both channels. For example,  
Instruction at address 110 (post-instruction-execution break for instruction fetch for channel 0)  
→ SPC = 112, CCMFR.MF0 = 1  
Instruction at address 112 (pre-instruction-execution break for instruction fetch for channel 1)  
→ SPC = 112, CCMFR.MF1 = 1
- It is not allowed to set the pre-instruction-execution break or the operand break in the delayed slot instruction of the RTE instruction. And if the data value is included in the match conditions of the operand break, do not set the break for the preceding the RTE instruction by one to six instructions.
- If the re-execution type exception and the post-instruction-execution break are in conflict for the instruction requiring two or more execution states, then the re-execution type exception occurs. Here, the CCMFR.MF0 (or CCMFR.MF1) bit may or may not be set to 1 when the break conditions have been satisfied.





## Section 31 User Debugging Interface (H-UDI)

The H-UDI is a serial interface which conforms to the JTAG (IEEE 1149.1: IEEE Standard Test Access Port and Boundary-Scan Architecture) standard. The H-UDI is also used for emulator connection.

### 31.1 Features

The H-UDI is a serial interface which conforms to the JTAG standard. The H-UDI is also used for emulator connection. When using an emulator, H-UDI functions should not be used. Refer to the appropriate emulator users manual for the method of connecting the emulator.

The H-UDI has six pins: TCK, TMS, TDI, TDO,  $\overline{\text{TRST}}$ , and  $\overline{\text{ASEBRKAK/BRKACK}}$ . The pin functions except  $\overline{\text{ASEBRKAK/BRKACK}}$  and serial communications protocol conform to the JTAG standard. This LSI has additional six pins for emulator connection: (AUDSYNC, AUDCK, and AUDATA3 to AUDATA0). These six pins for emulator are multiplexed with on-chip modules. And the H-UDI has one chip-mode setting pin: (MPMD).

The H-UDI has two TAP controller blocks; one is for the boundary-scan test and another is H-UDI function except the boundary-scan test. The H-UDI initial state is for the boundary scan after power-on or  $\overline{\text{TRST}}$  asserted. It is necessary to set H-UDI switchover command to use the H-UDI function. And the CPU cannot access the boundary scan TAP controller.

Figure 31.1 shows a block diagram of the H-UDI.

The H-UDI has the TAP (Test Access Port) controller 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 is directly accessed from the TDI and TDO pins.

The TAP controller, control registers and boundary scan TAP controller are initialized by driving the  $\overline{\text{TRST}}$  pin low or by applying the TCK signal for five or more clock cycles with the TMS pin set to 1. This initialization sequence is independent of the reset pin for this LSI. Other circuits are initialized by a normal reset.

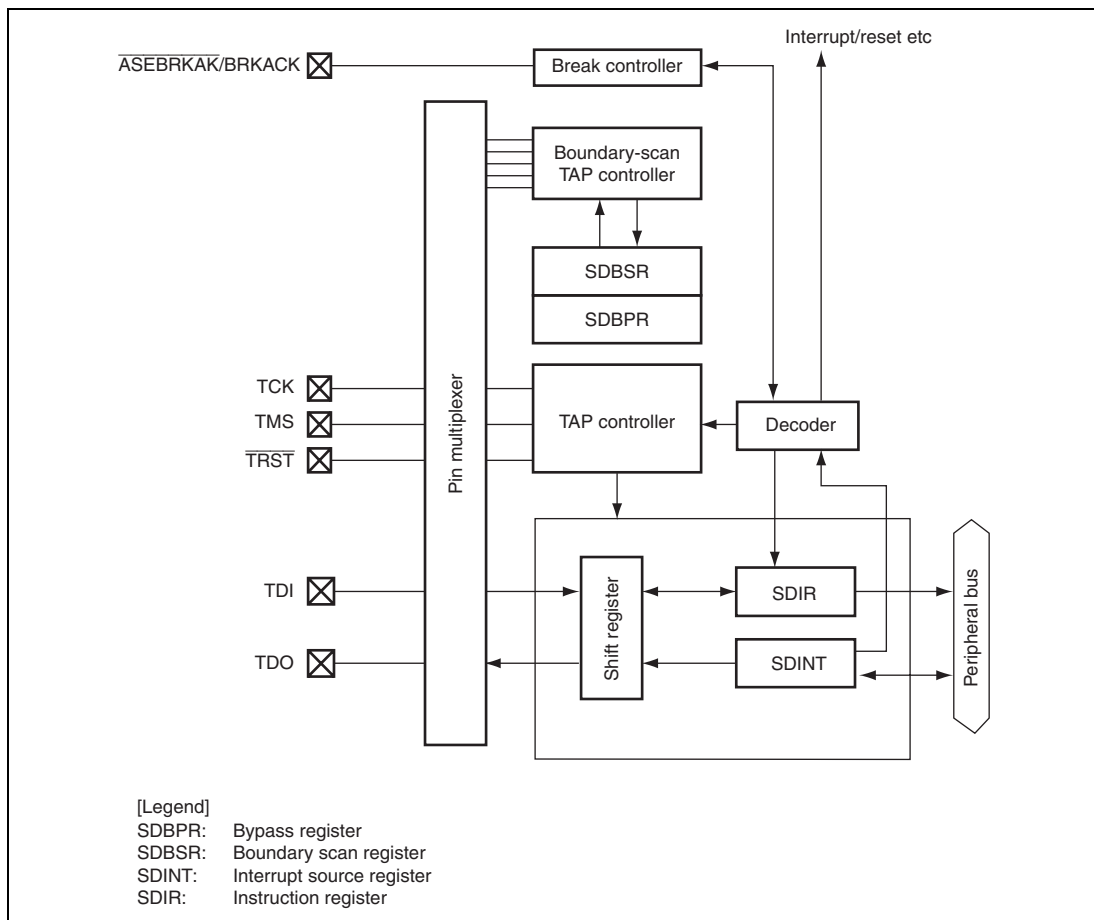


Figure 31.1 H-UDI Block Diagram

## 31.2 Input/Output Pins

Table 31.1 shows the pin configuration for the H-UDI.

**Table 31.1 Pin Configuration**

Pin Name	Function	I/O	Description	When Not in Use
TCK	Clock	Input	Functions as the serial clock input pin stipulated in the JTAG standard. Data input to the H-UDI via the TDI pin or data Output via the TDO pin is performed in synchronization with this signal.	Open <sup>*1</sup>
TMS	Mode	Input	Mode Select Input Changing this signal in synchronization with the TCK signal determines the significance of data input via the TDI pin. Its protocol conforms to the JTAG standard (IEEE standard 1149.1).	Open <sup>*1</sup>
$\overline{\text{TRST}}^{*2}$	Reset	Input	H-UDI Reset Input This signal is received asynchronously with a TCK signal. Asserting this signal resets the JTAG interface circuit. When a power is supplied, the $\overline{\text{TRST}}$ pin should be asserted for a given period regardless of whether or not the JTAG function is used, which differs from the JTAG standard.	Fixed to ground or connected to the PRESET pin <sup>*3</sup>
TDI	Data input	Input	Data Input Data is sent to the H-UDI by changing this signal in synchronization with the TCK signal.	Open <sup>*1</sup>
TDO	Data output	Output	Data Output Data is read from the H-UDI in synchronization with the TCK signal.	Open
ASEBRKAK/ BRKACK	Emulator	I/O	Pins for an emulator	Open <sup>*1</sup>
AUDSYNC, AUDCK, AUDATA3 to AUDATA0	Emulator	Output	Pins for an emulator	Open
MPMD	Chip-mode	Input	Selects the operation mode of this LSI, whether emulation support mode (Low level) or LSI operation mode (High level).	Open

- Notes:
1. This pin is pulled up in this LSI. When using interrupts or resets via the H-UDI or emulator, the use of external pull-up resistors will not cause any problem.
  2. When using interrupts or resets via the H-UDI or emulator, the  $\overline{\text{TRST}}$  pin should be designed so that it can be controlled independently and can be controlled to retain low level while the PRESET pin is asserted at a power-on reset.

3. This pin should be connected to ground, the  $\overline{\text{PRESET}}$ , or another pin which operates in the same manner as the  $\text{PRESET}$  pin. However, when connected to a ground pin, the following problem occurs. Since the  $\overline{\text{TRST}}$  pin is pulled up within this LSI, a weak current flows when the pin is externally connected to ground pin. The value of the current is determined by a resistance of the pull-up MOS for the port pin. Although this current does not affect the operation of this LSI, it consumes unnecessary power.

The TCK clock or the CPG of this LSI should be set to ensure that the frequency of the TCK clock is less than the peripheral-clock frequency of this LSI.

### 31.3 Boundary Scan TAP Controllers (IDCODE, EXTEST, SAMPLE/PRELOAD, BYPASS, CLAMP and HIGHZ)

The H-UDI contains two separate TAP controllers: one for controlling the boundary-scan function and another for controlling the H-UDI reset and interrupt functions. Assertion of  $\overline{\text{TRST}}$ , for example at power-on reset, activates the boundary-scan TAP controller and enables the boundary-scan function prescribed in the JTAG standards. Executing a switchover command to the H-UDI allows usage of the H-UDI reset and H-UDI interrupts. This LSI, however, has the following limitations:

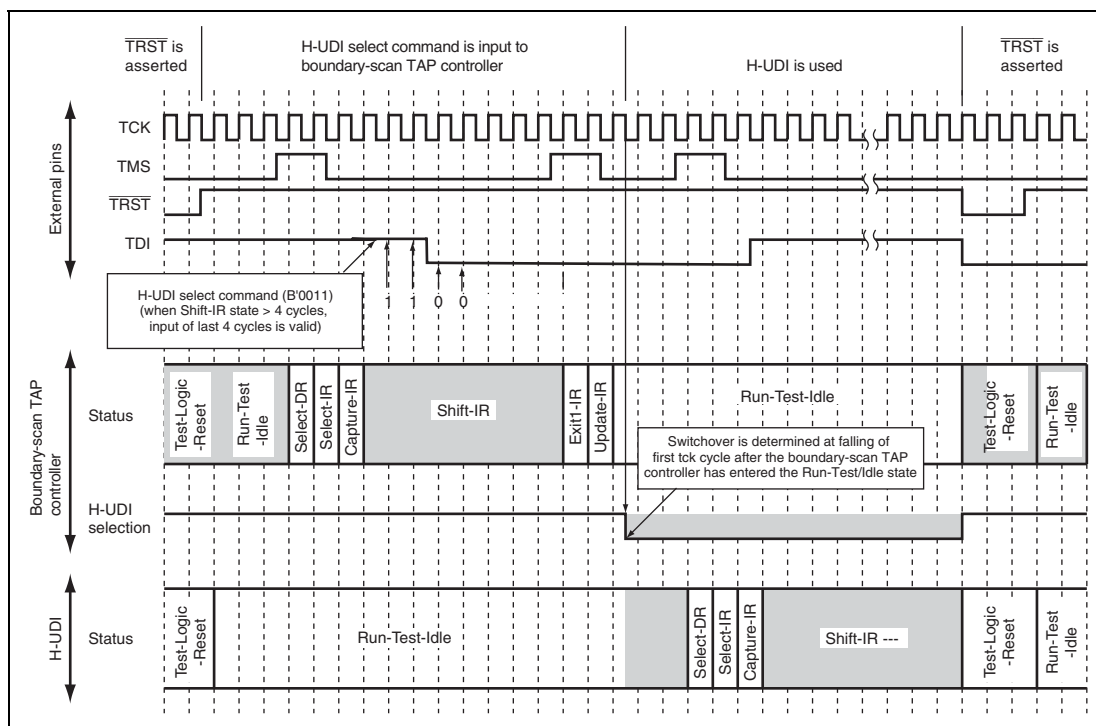
- Clock-related pins (EXTAL and XTAL) are out of the scope of the boundary-scan test.
- Reset-related pin ( $\overline{\text{PRESET}}$ ) is out of the scope of the boundary-scan test.
- H-UDI-related pins (TCK, TDI, TDO, TMS,  $\overline{\text{TRST}}$  and MPMD) are out of the scope of the boundary-scan test.
- USB-related pins (DM, DP, VBUS, and REFRIN) are out of the scope of the boundary-scan test
- During the boundary scan (IDCODE, EXTEST, SAMPLE/PRELOAD, BYPASS, CLAMP, HIGHZ, and H-UDI switchover command), the maximum TCK signal frequency is 2 MHz.
- The external controller has 4-bit access to the boundary-scan TAP controller via the H-UDI.

Note: During the boundary scan, the  $\overline{\text{PRESET}}$  pin should be fixed high-level.

Table 31.2 shows the commands supported by the boundary-scan TAP controller.

**Table 31.2 Commands Supported by Boundary-Scan TAP Controller**

Bit 3	Bit 2	Bit 1	Bit 0	Description
1	1	1	1	BYPASS
0	0	0	0	EXTEST
0	0	0	1	SAMPLE/PRELOAD
0	1	0	0	IDCODE
0	1	1	0	CLAMP
0	1	1	1	HIGHZ
0	0	1	1	H-UDI (select command)
Other than above				Setting prohibited

**Figure 31.2 Sequence for switching from Boundary-Scan TAP Controller to H-UDI**

## 31.4 Register Descriptions

The H-UDI has the following registers.

**Table 31.3 Register Configuration (1)**

Register Name	Abbrev.	R/W	CPU Side			
			Area P4 Address* <sup>1</sup>	Area 7 Address* <sup>1</sup>	Size	Initial Value* <sup>2</sup>
Instruction register	SDIR	R	H'FC11 0000	H'1C11 0000	16	H'0EFF
Interrupt source register	SDINT	R/W	H'FC11 0018	H'1C11 0018	16	H'0000
Boundary scan register	SDBSR	—	—	—	—	—
Bypass register	SDBPR	—	—	—	—	—

Notes: 1. The area P4 address is an address when accessing through area P4 in a virtual address space. The area 7 address is an address when accessing through area 7 in a physical space using the TLB.

2. The low level of the  $\overline{\text{TRST}}$  pin or the Test-Logic-Reset state of the TAP controller initializes to these values.

**Table 31.4 Register Configuration (2)**

Register Name	Abbrev.	R/W	H-UDI Side	
			Size	Initial Value* <sup>1</sup>
Instruction register	SDIR	R/W	32	H'FFFF FFFD (fixed value* <sup>2</sup> )
Interrupt source register	SDINT	W* <sup>3</sup>	32	H'0000 0000
Boundary scan register	SDBSR	—	—	—
Bypass register	SDBPR	R/W	1	Undefined

Note: 1. The low level of the  $\overline{\text{TRST}}$  pin or the Test-Logic-Reset state of the TAP controller initializes to these values.

2. When reading via the H-UDI, the value is always H'FFFF FFFD.

3. Only 1 can be written to the LSB by the H-UDI interrupt command.

**Table 31.5 Register Status in Each Processing State**

Register Name	Abbrev.	Power-On Reset	Sleep	Standby
Instruction register	SDIR	H'0EFF	Retained	Retained
Interrupt source register	SDINT	H'0000	Retained	Retained

### 31.4.1 Instruction Register (SDIR)

SDIR is a 16-bit read-only register that can be read from the CPU. Commands are set via the serial input (TDI). SDIR is initialized by  $\overline{\text{TRST}}$  or in the Test-Logic-Reset state and can be written by the H-UDI irrespective of the CPU mode. Operation is not guaranteed when a reserved command is set to this register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TI								—	—	—	—	—	—	—	—
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 Note: Though H-UDI reset asserted, CPG and WDT registers are not initialized.
7 to 0	—	All 1	R	Reserved These bits are always read as 1.

### 31.4.2 Interrupt Source Register (SDINT)

SDINT is a 16-bit register that can be read from or written to by the CPU. Specifying an H-UDI interrupt command in SDIR via H-UDI pin (Update-IR) sets the INTREQ bit to 1. While an H-UDI interrupt command is set in SDIR, SDINT which is connected between the TDI and TDO pins can be read as a 32-bit register. In this case, the upper 16 bits will be 0 and the lower 16 bits represent the SDINT value.

Only 0 can be written to the INTREQ bit by the CPU. While this bit is set to 1, an interrupt request will continue to be generated. This bit, therefore, should be cleared by the interrupt handling routine. It is initialized by  $\overline{\text{TRST}}$  or in the Test-Logic-Reset state.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved For reading from or writing to this bit, see General Precautions on Handling of Product.
0	INTREQ	0	R/W	Interrupt Request Indicates whether or not an interrupt by an H-UDI interrupt command has occurred. Clearing this bit to 0 by the CPU cancels an interrupt request. When writing 1 to this bit, the previous value is maintained.



### 31.4.3 Bypass Register (SDBPR)

SDBPR is a one-bit register that supports the J-TAG bypass mode. When the BYPASS command is set to the boundary scan TAP controller, the TDI and TDO are connected by way of SDBPR. This register cannot be accessed from the CPU regardless of the LSI mode. Though this register is not initialized by a power-on reset and the  $\overline{\text{TRST}}$  pin asserted, initialized to 0 in the Capture-DR state.

### 31.4.4 Boundary Scan Register (SDBSR)

SDBSR is a shift register, located on the PAD, for controlling the input/Output pins, which supports the boundary scan mode of the JTAG standard.

Using the EXTEST and SAMPLE/PRELOAD commands, a boundary-scan test complying with the JTAG standards (IEEE1149.1) can be carried out.

This register cannot be accessed from the CPU regardless of the LSI mode.

This register is not initialized by a power-on reset and the  $\overline{\text{TRST}}$  pin asserted.

**Table 31.6 SDBSR Configuration**

Number	Pin Name	I/O*
	From TDI	
603	SCK0/AUDSYNC/FCLE	CONTROL
602	SCK0/AUDSYNC/FCLE	OUTPUT
601	SCK0/AUDSYNC/FCLE	INPUT
600	LCD_VEP_WC/DR5/PH0	CONTROL
599	LCD_VEP_WC/DR5/PH0	OUTPUT
598	LCD_VEP_WC/DR5/PH0	INPUT
597	LCD_FLM/VSYNCS EX_VSYNCS/BT_VSYNCS	CONTROL
596	LCD_FLM/VSYNCS EX_VSYNCS/BT_VSYNCS	OUTPUT
595	LCD_FLM/VSYNCS EX_VSYNCS/BT_VSYNCS	INPUT
594	LCD_CL1/HSYNCS SPL_EX_HSYNCS/BT_HSYNCS	CONTROL
593	LCD_CL1/HSYNCS SPL_EX_HSYNCS/BT_HSYNCS	OUTPUT
592	LCD_CL1/HSYNCS SPL_EX_HSYNCS/BT_HSYNCS	INPUT
591	LCD_M_DISP/DE_C DE_H/BT_DE_C	CONTROL
590	LCD_M_DISP/DE_C DE_H/BT_DE_C	OUTPUT
589	LCD_M_DISP/DE_C DE_H/BT_DE_C	INPUT
588	LCD_VCP_WC/DR4/PH1	CONTROL
587	LCD_VCP_WC/DR4/PH1	OUTPUT
586	LCD_VCP_WC/DR4/PH1	INPUT
585	LCD_CLK/DCLKIN	CONTROL
584	LCD_CLK/DCLKIN	OUTPUT
583	LCD_CLK/DCLKIN	INPUT
582	LCD_DATA15/DR3/PG7	CONTROL
581	LCD_DATA15/DR3/PG7	OUTPUT
580	LCD_DATA15/DR3/PG7	INPUT
579	LCD_DATA14/DR2/PG6	CONTROL
578	LCD_DATA14/DR2/PG6	OUTPUT
577	LCD_DATA14/DR2/PG6	INPUT
576	LCD_DATA13/DR1/PG5	CONTROL
575	LCD_DATA13/DR1/PG5	OUTPUT

Number	Pin Name	I/O*
574	LCD_DATA13/DR1/PG5	INPUT
573	LCD_DATA12/DR0/PG4	CONTROL
572	LCD_DATA12/DR0/PG4	OUTPUT
571	LCD_DATA12/DR0/PG4	INPUT
570	LCD_DATA11/DG5/PG3	CONTROL
569	LCD_DATA11/DG5/PG3	OUTPUT
568	LCD_DATA11/DG5/PG3	INPUT
567	LCD_DATA10/DG4/PG2	CONTROL
566	LCD_DATA10/DG4/PG2	OUTPUT
565	LCD_DATA10/DG4/PG2	INPUT
564	LCD_DATA9/DG3/PG1	CONTROL
563	LCD_DATA9/DG3/PG1	OUTPUT
562	LCD_DATA9/DG3/PG1	INPUT
561	LCD_DATA8/DG2/PG0	CONTROL
560	LCD_DATA8/DG2/PG0	OUTPUT
559	LCD_DATA8/DG2/PG0	INPUT
558	LCD_DATA7/DG1/BT_DATA7/PI4	CONTROL
557	LCD_DATA7/DG1/BT_DATA7/PI4	OUTPUT
556	LCD_DATA7/DG1/BT_DATA7/PI4	INPUT
555	LCD_DATA6/DG0/BT_DATA6/PI3	CONTROL
554	LCD_DATA6/DG0/BT_DATA6/PI3	OUTPUT
553	LCD_DATA6/DG0/BT_DATA6/PI3	INPUT
552	LCD_DATA5/DB5/BT_DATA5/PI2	CONTROL
551	LCD_DATA5/DB5/BT_DATA5/PI2	OUTPUT
550	LCD_DATA5/DB5/BT_DATA5/PI2	INPUT
549	LCD_DATA4/DB4/BT_DATA4/PI1	CONTROL
548	LCD_DATA4/DB4/BT_DATA4/PI1	OUTPUT
547	LCD_DATA4/DB4/BT_DATA4/PI1	INPUT
546	LCD_DATA3/DB3/BT_DATA3	CONTROL
545	LCD_DATA3/DB3/BT_DATA3	OUTPUT
544	LCD_DATA3/DB3/BT_DATA3	INPUT

Number	Pin Name	I/O*
543	LCD_DATA2/DB2/BT_DATA2	CONTROL
542	LCD_DATA2/DB2/BT_DATA2	OUTPUT
541	LCD_DATA2/DB2/BT_DATA2	INPUT
540	LCD_DATA1/DB1/BT_DATA1	CONTROL
539	LCD_DATA1/DB1/BT_DATA1	OUTPUT
538	LCD_DATA1/DB1/BT_DATA1	INPUT
537	LCD_DATA0/DB0/BT_DATA0	CONTROL
536	LCD_DATA0/DB0/BT_DATA0	OUTPUT
535	LCD_DATA0/DB0/BT_DATA0	INPUT
534	LCD_CL2/DE_V/PH3	CONTROL
533	LCD_CL2/DE_V/PH3	OUTPUT
532	LCD_CL2/DE_V/PH3	INPUT
531	LCD_DON/DCLKOUT/PH2	CONTROL
530	LCD_DON/DCLKOUT/PH2	OUTPUT
529	LCD_DON/DCLKOUT/PH2	INPUT
528	PI0/COM/CDE	CONTROL
527	PI0/COM/CDE	OUTPUT
526	PI0/COM/CDE	INPUT
525	RDY	INPUT
524	—	INTERNAL
523	NMI	INPUT
522	BACK	CONTROL
521	BACK	OUTPUT
520	BACK	INPUT
519	RD	CONTROL
518	RD	OUTPUT
517	RD	INPUT
516	CS3	CONTROL
515	CS3	OUTPUT
514	CS3	INPUT
513	BREQ	INPUT

Number	Pin Name	I/O*
512	$\overline{BS}$	CONTROL
511	$\overline{BS}$	OUTPUT
510	CS0	CONTROL
509	CS0	OUTPUT
508	CS0	INPUT
507	$\overline{ASEBRKAK/BRKACK/TCLK/PC1}$	CONTROL
506	$\overline{ASEBRKAK/BRKACK/TCLK/PC1}$	OUTPUT
505	$\overline{ASEBRKAK/BRKACK/TCLK/PC1}$	INPUT
504	A25/PB7/ $\overline{DREQ0/RTS0}$	CONTROL
503	A25/PB7/ $\overline{DREQ0/RTS0}$	OUTPUT
502	A25/PB7/ $\overline{DREQ0/RTS0}$	INPUT
501	A24/PB6/ $\overline{DACK0/CTS0}$	CONTROL
500	A24/PB6/ $\overline{DACK0/CTS0}$	OUTPUT
499	A24/PB6/ $\overline{DACK0/CTS0}$	INPUT
498	A17	CONTROL
497	A17	OUTPUT
496	A17	INPUT
495	A23/PB5/ $\overline{DTEND0/RTS1}$	CONTROL
494	A23/PB5/ $\overline{DTEND0/RTS1}$	OUTPUT
493	A23/PB5/ $\overline{DTEND0/RTS1}$	INPUT
492	A21/PB3	CONTROL
491	A21/PB3	OUTPUT
490	A21/PB3	INPUT
489	A20/PB2	CONTROL
488	A20/PB2	OUTPUT
487	A20/PB2	INPUT
486	A22/PB4/ $\overline{CTS1}$	CONTROL
485	A22/PB4/ $\overline{CTS1}$	OUTPUT
484	A22/PB4/ $\overline{CTS1}$	INPUT
483	A19/PB1	CONTROL
482	A19/PB1	OUTPUT
481	A19/PB1	INPUT

Number	Pin Name	I/O*
480	A18/PB0	CONTROL
479	A18/PB0	OUTPUT
478	A18/PB0	INPUT
477	D15	CONTROL
476	D15	OUTPUT
475	D15	INPUT
474	D14	CONTROL
473	D14	OUTPUT
472	D14	INPUT
471	D1	CONTROL
470	D1	OUTPUT
469	D1	INPUT
468	D0	CONTROL
467	D0	OUTPUT
466	D0	INPUT
465	D13	CONTROL
464	D13	OUTPUT
463	D13	INPUT
462	D12	CONTROL
461	D12	OUTPUT
460	D12	INPUT
459	D3	CONTROL
458	D3	OUTPUT
457	D3	INPUT
456	D2	CONTROL
455	D2	OUTPUT
454	D2	INPUT
453	D11	CONTROL
452	D11	OUTPUT
451	D11	INPUT
450	D10	CONTROL
449	D10	OUTPUT

Number	Pin Name	I/O*
448	D10	INPUT
447	D5	CONTROL
446	D5	OUTPUT
445	D5	INPUT
444	D4	CONTROL
443	D4	OUTPUT
442	D4	INPUT
441	D9	CONTROL
440	D9	OUTPUT
439	D9	INPUT
438	D6	CONTROL
437	D6	OUTPUT
436	D6	INPUT
435	D7	CONTROL
434	D7	OUTPUT
433	D7	INPUT
432	D8	CONTROL
431	D8	OUTPUT
430	D8	INPUT
429	DQMLL	CONTROL
428	DQMLL	OUTPUT
427	DQMLL	INPUT
426	DQMUL	CONTROL
425	DQMUL	OUTPUT
424	DQMUL	INPUT
423	DQMUU	CONTROL
422	DQMUU	OUTPUT
421	DQMUU	INPUT
420	D16	CONTROL
419	D16	OUTPUT
418	D16	INPUT
417	DQMLU	CONTROL

Number	Pin Name	I/O*
416	DQMLU	OUTPUT
415	DQMLU	INPUT
414	D17	CONTROL
413	D17	OUTPUT
412	D17	INPUT
411	D18	CONTROL
410	D18	OUTPUT
409	D18	INPUT
408	D19	CONTROL
407	D19	OUTPUT
406	D19	INPUT
405	D31	CONTROL
404	D31	OUTPUT
403	D31	INPUT
402	D30	CONTROL
401	D30	OUTPUT
400	D30	INPUT
399	D20	CONTROL
398	D20	OUTPUT
397	D20	INPUT
396	D21	CONTROL
395	D21	OUTPUT
394	D21	INPUT
393	D22	CONTROL
392	D22	OUTPUT
391	D22	INPUT
390	D28	CONTROL
389	D28	OUTPUT
388	D28	INPUT
387	D29	CONTROL
386	D29	OUTPUT
385	D29	INPUT



Number	Pin Name	I/O*
384	A15	CONTROL
383	A15	OUTPUT
382	A15	INPUT
381	D23	CONTROL
380	D23	OUTPUT
379	D23	INPUT
378	D26	CONTROL
377	D26	OUTPUT
376	D26	INPUT
375	A13	CONTROL
374	A13	OUTPUT
373	A16	CONTROL
372	A16	OUTPUT
371	A16	INPUT
370	D27	CONTROL
369	D27	OUTPUT
368	D27	INPUT
367	D24	CONTROL
366	D24	OUTPUT
365	D24	INPUT
364	A10	CONTROL
363	A10	OUTPUT
362	A14	CONTROL
361	A14	OUTPUT
360	D25	CONTROL
359	D25	OUTPUT
358	D25	INPUT
357	A4	CONTROL
356	A4	OUTPUT
355	A11	CONTROL
354	A11	OUTPUT

Number	Pin Name	I/O*
353	A5	CONTROL
352	A5	OUTPUT
351	R/W	CONTROL
350	R/W	OUTPUT
349	A8	CONTROL
348	A8	OUTPUT
347	A12	CONTROL
346	A12	OUTPUT
345	CKE	CONTROL
344	CKE	OUTPUT
343	RAS	CONTROL
342	RAS	OUTPUT
341	CLKOUT	CONTROL
340	CLKOUT	OUTPUT
339	A9	CONTROL
338	A9	OUTPUT
337	A6	CONTROL
336	A6	OUTPUT
335	A7	CONTROL
334	A7	OUTPUT
333	CAS	CONTROL
332	CAS	OUTPUT
331	CS1	CONTROL
330	CS1	OUTPUT
329	CS1	INPUT
328	CS2	CONTROL
327	CS2	OUTPUT
326	CS2	INPUT
325	A0	CONTROL
324	A0	OUTPUT
323	D47/IDECS0	CONTROL
322	D47/IDECS0	OUTPUT

Number	Pin Name	I/O*
321	D47/ $\overline{\text{IDECS0}}$	INPUT
320	A3	CONTROL
319	A3	OUTPUT
318	A1	CONTROL
317	A1	OUTPUT
316	D45/ $\overline{\text{IODACK}}$	CONTROL
315	D45/ $\overline{\text{IODACK}}$	OUTPUT
314	D45/ $\overline{\text{IODACK}}$	INPUT
313	D46/ $\overline{\text{IDECST1}}$	CONTROL
312	D46/ $\overline{\text{IDECST1}}$	OUTPUT
311	D46/ $\overline{\text{IDECST1}}$	INPUT
310	D33/PF6	CONTROL
309	D33/PF6	OUTPUT
308	D33/PF6	INPUT
307	A2	CONTROL
306	A2	OUTPUT
305	D44/ $\overline{\text{IDEINT}}$	CONTROL
304	D44/ $\overline{\text{IDEINT}}$	OUTPUT
303	D44/ $\overline{\text{IDEINT}}$	INPUT
302	D43/ $\overline{\text{IDEIORDY}}$	CONTROL
301	D43/ $\overline{\text{IDEIORDY}}$	OUTPUT
300	D43/ $\overline{\text{IDEIORDY}}$	INPUT
299	D42/ $\overline{\text{IDEIORD}}$	CONTROL
298	D42/ $\overline{\text{IDEIORD}}$	OUTPUT
297	D42/ $\overline{\text{IDEIORD}}$	INPUT
296	D32/PF7	CONTROL
295	D32/PF7	OUTPUT
294	D32/PF7	INPUT
293	D40/ $\overline{\text{IDEIOWR}}$	CONTROL
292	D40/ $\overline{\text{IDEIOWR}}$	OUTPUT
291	D40/ $\overline{\text{IDEIOWR}}$	INPUT
290	D41/ $\overline{\text{IODREQ}}$	CONTROL

Number	Pin Name	I/O*
289	D41/IODREQ	OUTPUT
288	D41/IODREQ	INPUT
287	D35/IDEA0	CONTROL
286	D35/IDEA0	OUTPUT
285	D35/IDEA0	INPUT
284	D37/IDEA1	CONTROL
283	D37/IDEA1	OUTPUT
282	D37/IDEA1	INPUT
281	D39/IDED14	CONTROL
280	D39/IDED14	OUTPUT
279	D39/IDED14	INPUT
278	D34/PF5	CONTROL
277	D34/PF5	OUTPUT
276	D34/PF5	INPUT
275	D36/IDEA2	CONTROL
274	D36/IDEA2	OUTPUT
273	D36/IDEA2	INPUT
272	D63/IDED1	CONTROL
271	D63/IDED1	OUTPUT
270	D63/IDED1	INPUT
269	D38/IDED15	CONTROL
268	D38/IDED15	OUTPUT
267	D38/IDED15	INPUT
266	D62/IDED0	CONTROL
265	D62/IDED0	OUTPUT
264	D62/IDED0	INPUT
263	$\overline{WE2}$ /DQM64UL	CONTROL
262	$\overline{WE2}$ /DQM64UL	OUTPUT
261	$\overline{WE0}$ /DQM64LL	CONTROL
260	$\overline{WE0}$ /DQM64LL	OUTPUT
259	D60/IDED2	CONTROL
258	D60/IDED2	OUTPUT

Number	Pin Name	I/O*
257	D60/IDED2	INPUT
256	$\overline{\text{WE}}3/\text{DQM64UU}$	CONTROL
255	$\overline{\text{WE}}3/\text{DQM64UU}$	OUTPUT
254	$\overline{\text{WE}}1/\text{DQM64LU}$	CONTROL
253	$\overline{\text{WE}}1/\text{DQM64LU}$	OUTPUT
252	D61/IDED3	CONTROL
251	D61/IDED3	OUTPUT
250	D61/IDED3	INPUT
249	D48/IDED13	CONTROL
248	D48/IDED13	OUTPUT
247	D48/IDED13	INPUT
246	D59/IDED5	CONTROL
245	D59/IDED5	OUTPUT
244	D59/IDED5	INPUT
243	D58/IDED4	CONTROL
242	D58/IDED4	OUTPUT
241	D58/IDED4	INPUT
240	D49/IDED12	CONTROL
239	D49/IDED12	OUTPUT
238	D49/IDED12	INPUT
237	D51/IDED10	CONTROL
236	D51/IDED10	OUTPUT
235	D51/IDED10	INPUT
234	D50/IDED11	CONTROL
233	D50/IDED11	OUTPUT
232	D50/IDED11	INPUT
231	D56/IDED6	CONTROL
230	D56/IDED6	OUTPUT
229	D56/IDED6	INPUT
228	D52/IDED9	CONTROL
227	D52/IDED9	OUTPUT
226	D52/IDED9	INPUT

Number	Pin Name	I/O*
225	D53/IDED8	CONTROL
224	D53/IDED8	OUTPUT
223	D53/IDED8	INPUT
222	D57/IDED7	CONTROL
221	D57/IDED7	OUTPUT
220	D57/IDED7	INPUT
219	D54/IDERST	CONTROL
218	D54/IDERST	OUTPUT
217	D54/IDERST	INPUT
216	D55/DIRECTION	CONTROL
215	D55/DIRECTION	OUTPUT
214	D55/DIRECTION	INPUT
213	WOL/PF2/IDEA0_M	CONTROL
212	WOL/PF2/IDEA0_M	OUTPUT
211	WOL/PF2/IDEA0_M	INPUT
210	SSISCK2/PC3	CONTROL
209	SSISCK2/PC3	OUTPUT
208	SSISCK2/PC3	INPUT
207	SSIDATA2/PC2	CONTROL
206	SSIDATA2/PC2	OUTPUT
205	SSIDATA2/PC2	INPUT
204	SSIWS2/PC4	CONTROL
203	SSIWS2/PC4	OUTPUT
202	SSIWS2/PC4	INPUT
201	LNKSTA/PF3/IDECS0_M	CONTROL
200	LNKSTA/PF3/IDECS0_M	OUTPUT
199	LNKSTA/PF3/IDECS0_M	INPUT
198	EXOUT/PF4/IDECS1_M	CONTROL
197	EXOUT/PF4/IDECS1_M	OUTPUT
196	EXOUT/PF4/IDECS1_M	INPUT
195	AUDIO_CLK2/PC5	CONTROL
194	AUDIO_CLK2/PC5	OUTPUT

Number	Pin Name	I/O*
193	AUDIO_CLK2/PC5	INPUT
192	CRS/PD7/IDEA1_M	CONTROL
191	CRS/PD7/IDEA1_M	OUTPUT
190	CRS/PD7/IDEA1_M	INPUT
189	COL/PE7/IDEA2_M	CONTROL
188	COL/PE7/IDEA2_M	OUTPUT
187	COL/PE7/IDEA2_M	INPUT
186	TX_ER/PD6/IDEIOWR_M	CONTROL
185	TX_ER/PD6/IDEIOWR_M	OUTPUT
184	TX_ER/PD6/IDEIOWR_M	INPUT
183	MII_TXD3/SSIDATA5/IODACK_M/PD0	CONTROL
182	MII_TXD3/SSIDATA5/IODACK_M/PD0	OUTPUT
181	MII_TXD3/SSIDATA5/IODACK_M/PD0	INPUT
180	MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1	CONTROL
179	MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1	OUTPUT
178	MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1	INPUT
177	RX_ER/PE6/IODREQ_M	CONTROL
176	RX_ER/PE6/IODREQ_M	OUTPUT
175	RX_ER/PE6/IODREQ_M	INPUT
174	MII_TXD1/SSIWS5/IDEIORD_M/PD2	CONTROL
173	MII_TXD1/SSIWS5/IDEIORD_M/PD2	OUTPUT
172	MII_TXD1/SSIWS5/IDEIORD_M/PD2	INPUT
171	SSIDATA3/PH4	CONTROL
170	SSIDATA3/PH4	OUTPUT
169	SSIDATA3/PH4	INPUT
168	MII_TXD0/SSISCK5/IDEIORDY_M/PD3	CONTROL
167	MII_TXD0/SSISCK5/IDEIORDY_M/PD3	OUTPUT
166	MII_TXD0/SSISCK5/IDEIORDY_M/PD3	INPUT
165	TX_EN/PD4/IDED0_M	CONTROL
164	TX_EN/PD4/IDED0_M	OUTPUT
163	TX_EN/PD4/IDED0_M	INPUT
162	SSIWS3/PH6	CONTROL

Number	Pin Name	I/O*
161	SSIWS3/PH6	OUTPUT
160	SSIWS3/PH6	INPUT
159	TX_CLK/PD5/IDED15_M	CONTROL
158	TX_CLK/PD5/IDED15_M	OUTPUT
157	TX_CLK/PD5/IDED15_M	INPUT
156	RX_CLK/PE5/IDED1_M	CONTROL
155	RX_CLK/PE5/IDED1_M	OUTPUT
154	RX_CLK/PE5/IDED1_M	INPUT
153	RX_DV/PE4/IDED14_M	CONTROL
152	RX_DV/PE4/IDED14_M	OUTPUT
151	RX_DV/PE4/IDED14_M	INPUT
150	SSISCK3/PH5	CONTROL
149	SSISCK3/PH5	OUTPUT
148	SSISCK3/PH5	INPUT
147	IRQ0/DTEND1	CONTROL
146	IRQ0/DTEND1	OUTPUT
145	IRQ0/DTEND1	INPUT
144	MII_RXD0/SSIWS4/IDED2_M/PE3	CONTROL
143	MII_RXD0/SSIWS4/IDED2_M/PE3	OUTPUT
142	MII_RXD0/SSIWS4/IDED2_M/PE3	INPUT
141	MII_RXD1/SSISCK4/IDED13_M/PE2	CONTROL
140	MII_RXD1/SSISCK4/IDED13_M/PE2	OUTPUT
139	MII_RXD1/SSISCK4/IDED13_M/PE2	INPUT
138	AUDIO_CLK3/PH7	CONTROL
137	AUDIO_CLK3/PH7	OUTPUT
136	AUDIO_CLK3/PH7	INPUT
135	MII_RXD2/SSIDATA4/IDED3_M/PE1	CONTROL
134	MII_RXD2/SSIDATA4/IDED3_M/PE1	OUTPUT
133	MII_RXD2/SSIDATA4/IDED3_M/PE1	INPUT
132	IRQOUT/DREQ1	CONTROL
131	IRQOUT/DREQ1	OUTPUT
130	IRQOUT/DREQ1	INPUT



Number	Pin Name	I/O*
129	MII_RXD3/AUDIO_CLK4/IDED12_M/PE0	CONTROL
128	MII_RXD3/AUDIO_CLK4/IDED12_M/PE0	OUTPUT
127	MII_RXD3/AUDIO_CLK4/IDED12_M/PE0	INPUT
126	MDC/PF0/IDED4_M	CONTROL
125	MDC/PF0/IDED4_M	OUTPUT
124	MDC/PF0/IDED4_M	INPUT
123	MDIO/PF1/IDED11_M	CONTROL
122	MDIO/PF1/IDED11_M	OUTPUT
121	MDIO/PF1/IDED11_M	INPUT
120	AUDIO_CLK0/PC7	CONTROL
119	AUDIO_CLK0/PC7	OUTPUT
118	AUDIO_CLK0/PC7	INPUT
117	SSIWS0	CONTROL
116	SSIWS0	OUTPUT
115	SSIWS0	INPUT
114	STATUS1/RTS2/PA7	CONTROL
113	STATUS1/RTS2/PA7	OUTPUT
112	STATUS1/RTS2/PA7	INPUT
111	SSISCK0	CONTROL
110	SSISCK0	OUTPUT
109	SSISCK0	INPUT
108	AUDIO_CLK1/PC6	CONTROL
107	AUDIO_CLK1/PC6	OUTPUT
106	AUDIO_CLK1/PC6	INPUT
105	STATUS0/CTS2/PA6	CONTROL
104	STATUS0/CTS2/PA6	OUTPUT
103	STATUS0/CTS2/PA6	INPUT
102	SSIDATA0	CONTROL
101	SSIDATA0	OUTPUT
100	SSIDATA0	INPUT
99	SSISCK1	CONTROL
98	SSISCK1	OUTPUT

Number	Pin Name	I/O*
97	SSISCK1	INPUT
96	PJ7/IDED10_M	CONTROL
95	PJ7/IDED10_M	OUTPUT
94	PJ7/IDED10_M	INPUT
93	SSIWS1	CONTROL
92	SSIWS1	OUTPUT
91	SSIWS1	INPUT
90	PJ6/IDED5_M	CONTROL
89	PJ6/IDED5_M	OUTPUT
88	PJ6/IDED5_M	INPUT
87	$\overline{\text{FRE}}/\text{PA4}$	CONTROL
86	$\overline{\text{FRE}}/\text{PA4}$	OUTPUT
85	$\overline{\text{FRE}}/\text{PA4}$	INPUT
84	SSIDATA1	CONTROL
83	SSIDATA1	OUTPUT
82	SSIDATA1	INPUT
81	PJ5/IDED9_M	CONTROL
80	PJ5/IDED9_M	OUTPUT
79	PJ5/IDED9_M	INPUT
78	PJ4/IDED6_M	CONTROL
77	PJ4/IDED6_M	OUTPUT
76	PJ4/IDED6_M	INPUT
75	PJ2/IDED8_M	CONTROL
74	PJ2/IDED8_M	OUTPUT
73	PJ2/IDED8_M	INPUT
72	PJ3/IDED7_M	CONTROL
71	PJ3/IDED7_M	OUTPUT
70	PJ3/IDED7_M	INPUT
69	$\overline{\text{FEW}}/\text{PA3}$	CONTROL
68	$\overline{\text{FEW}}/\text{PA3}$	OUTPUT
67	$\overline{\text{FEW}}/\text{PA3}$	INPUT
66	$\overline{\text{FCE}}/\text{PA5}$	CONTROL

Number	Pin Name	I/O*
65	$\overline{\text{FCE}}/\text{PA5}$	OUTPUT
64	$\overline{\text{FCE}}/\text{PA5}$	INPUT
63	PJ1/IDERST_M	CONTROL
62	PJ1/IDERST_M	OUTPUT
61	PJ1/IDERST_M	INPUT
60	PJ0/DIRECTION_M	CONTROL
59	PJ0/DIRECTION_M	OUTPUT
58	PJ0/DIRECTION_M	INPUT
57	MODE7/FD6	CONTROL
56	MODE7/FD6	OUTPUT
55	MODE7/FD6	INPUT
54	FALE/PC0	CONTROL
53	FALE/PC0	OUTPUT
52	FALE/PC0	INPUT
51	MODE3/FD3	CONTROL
50	MODE3/FD3	OUTPUT
49	MODE3/FD3	INPUT
48	MODE5/FD5	CONTROL
47	MODE5/FD5	OUTPUT
46	MODE5/FD5	INPUT
45	TXD2/PA2	CONTROL
44	TXD2/PA2	OUTPUT
43	TXD2/PA2	INPUT
42	MODE2/FD2	CONTROL
41	MODE2/FD2	OUTPUT
40	MODE2/FD2	INPUT
39	MODE4/FD4	CONTROL
38	MODE4/FD4	OUTPUT
37	MODE4/FD4	INPUT
36	MODE8/FD7	CONTROL
35	MODE8/FD7	OUTPUT

Number	Pin Name	I/O*
34	MODE8/FD7	INPUT
33	MODE1/FD1	CONTROL
32	MODE1/FD1	OUTPUT
31	MODE1/FD1	INPUT
30	RXD2/PA1	CONTROL
29	RXD2/PA1	OUTPUT
28	RXD2/PA1	INPUT
27	SCK2/PA0	CONTROL
26	SCK2/PA0	OUTPUT
25	SCK2/PA0	INPUT
24	SCL	OUTPUT
23	SCL	INPUT
22	SDA	OUTPUT
21	SDA	INPUT
20	RXD1/AUDATA2	CONTROL
19	RXD1/AUDATA2	OUTPUT
18	RXD1/AUDATA2	INPUT
17	WDTOVF/IRQ1/AUDCK/DACK1	CONTROL
16	WDTOVF/IRQ1/AUDCK/DACK1	OUTPUT
15	WDTOVF/IRQ1/AUDCK/DACK1	INPUT
14	MODE0/FD0	CONTROL
13	MODE0/FD0	OUTPUT
12	MODE0/FD0	INPUT
11	RXD0/AUDATA0	CONTROL
10	RXD0/AUDATA0	OUTPUT
9	RXD0/AUDATA0	INPUT
8	TXD1/AUDATA3	CONTROL
7	TXD1/AUDATA3	OUTPUT
6	TXD1/AUDATA3	INPUT
5	TXD0/AUDATA1	CONTROL
4	TXD0/AUDATA1	OUTPUT
3	TXD0/AUDATA1	INPUT

Number	Pin Name	I/O*
2	SCK1/FR/ $\overline{B}$	CONTROL
1	SCK1/FR/ $\overline{B}$	OUTPUT
0	SCK1/FR/ $\overline{B}$	INPUT
To TDO		

## 31.5 Operation

### 31.5.1 TAP Control

Figure 31.3 shows the internal states of the TAP controller. The state transitions basically conform to the JTAG standard.

- State transitions occur according to the TMS value at the rising edge of the TCK signal.
- The TDI value is sampled at the rising edge of the TCK signal and shifted at the falling edge of the TCK signal.
- The TDO value is changed at the falling edge of the TCK signal. The TDO signal is in a Hi-Z state other than in the Shift-DR or Shift-IR state.
- A transition to the Test-Logic-Reset by clearing  $\overline{\text{TRST}}$  to 0 is performed asynchronously with the TCK signal.

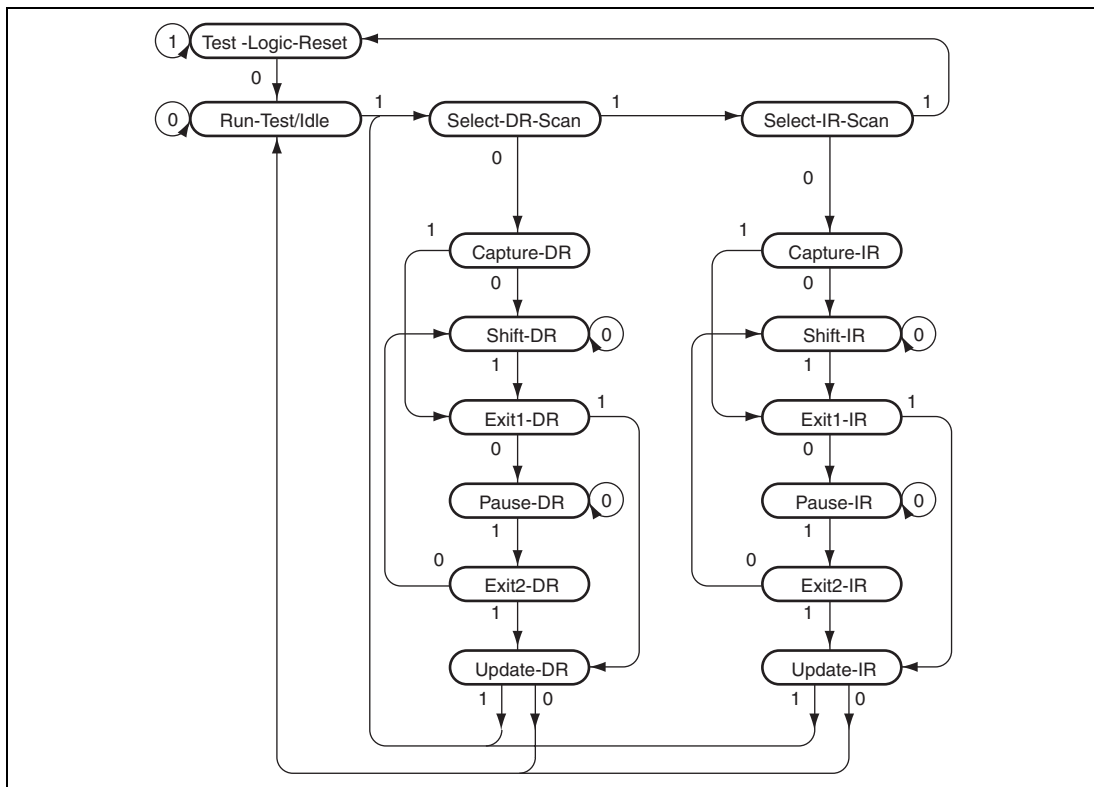
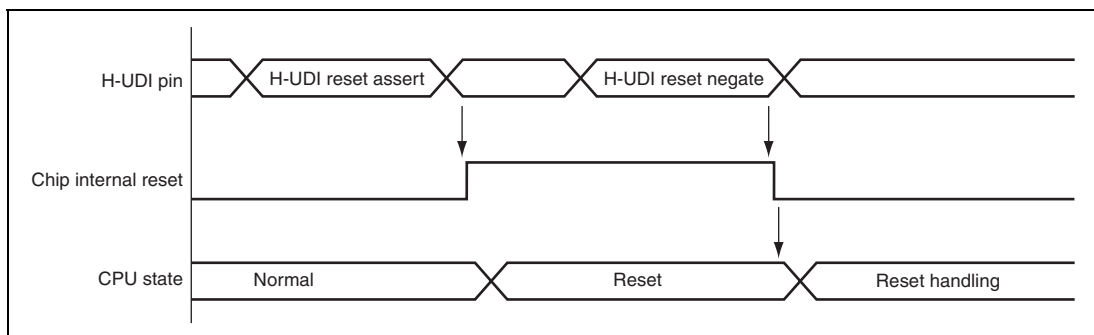


Figure 31.3 TAP Controller State Transitions

### 31.5.2 H-UDI Reset

A power-on reset is generated by the SDIR command. After the H-UDI reset assert command has been sent from the H-UDI pin, sending the H-UDI reset negate command resets the CPU (see Figure 31.4). The required time between the H-UDI reset assert and H-UDI reset negate commands is the same as the time for holding the reset pin low in order to reset this LSI by a power-on reset.



**Figure 31.4 H-UDI Reset**

### 31.5.3 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting the appropriate command in SDIR from the H-UDI. An H-UDI interrupt request signal is asserted when the INTREQ bit in SDINT is set to 1 by setting the appropriate command. Since the interrupt request signal is not negated until the INTREQ bit is cleared to 0 by software, it is not possible to lose the interrupt request. While an H-UDI interrupt command is set in SDIR, SDINT is connected between the TDI and TDO pins.

## 31.6 Usage Notes

Once an SDIR command is set, it will be changed only by an assertion of the  $\overline{\text{TRST}}$  signal, making the TAP controller Test-Logic-Reset state, or writing other commands from the H-UDI.

The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.





## Section 32 List of Registers

### 32.1 Register Addresses

- The on-chip I/O registers of this LSI are described by functional module, in order of the corresponding section numbers.
- Access to reserved addresses which are not described in this list is disabled. Operation or continued operation is not guaranteed when these registers are accessed.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given, on the presumption of a big-endian system.
- Entries under Access size indicates numbers of bits.
- For details on each register, refer to the description of the register in the corresponding section.

**Table 32.1 Register Configuration**

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
Exception handling	TRAPA exception register	TRA	R/W	H'FF00 0020	H'1F00 0020	32	
	Exception event register	EXPEVT	R/W	H'FF00 0024	H'1F00 0024	32	
	Interrupt event register	INTEVT	R/W	H'FF00 0028	H'1F00 0028	32	
	Non-support detection exception register	EXPMASK	R/W	H'FF2F 0004	H'1F2F 0004	32	
MMU	Page table entry high register	PTEH	R/W	H'FF00 0000	H'1F00 0000	32	
	Page table entry low register	PTL	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	
Cache	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	
On-chip memory	On-chip memory control register	RAMCR	R/W	H'FF00 0074*	H'1F00 0074*	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
CPG	Frequency control register	FRQCR	R	H'FFC8 0000	H'1FC8 0000	32	
	PLL control register	PLLCR	R/W	H'FFC8 0024	H'1FC8 0024	32	
	VDC2 clock control register	VDC2CLKCR	R/W	H'FFC8 0004	H'1FC8 0004	32	
WDT	Watchdog timer stop time register	WDTST	R/W	H'FFCC 0000	H'1FCC 0000	32	
	Watchdog timer control/status register	WDTCSR	R/W	H'FFCC 0004	H'1FCC 0004	32	
	Watchdog timer base stop time register	WDTBST	R/W	H'FFCC 0008	H'1FCC 0008	32	
	Watchdog timer counter	WDCNT	R	H'FFCC 0010	H'1FCC 0010	32	
	Watchdog timer base counter	WDTBCNT	R	H'FFCC 0018	H'1FCC 0018	32	
DMAC	DMA source address register 0	SAR0	R/W	H'FF60 8020	H'1F60 8020	32* <sup>3</sup>	
	DMA destination address register 0	DAR0	R/W	H'FF60 8024	H'1F60 8024	32* <sup>3</sup>	
	DMA transfer count register 0	TCR0	R/W	H'FF60 8028	H'1F60 8028	32* <sup>3</sup>	
	DMA channel control register 0	CHCR0	R/W* <sup>1</sup>	H'FF60 802C	H'1F60 802C	32* <sup>3</sup>	
	DMA source address register 1	SAR1	R/W	H'FF60 8030	H'1F60 8030	32* <sup>3</sup>	
	DMA destination address register 1	DAR1	R/W	H'FF60 8034	H'1F60 8034	32* <sup>3</sup>	
	DMA transfer count register 1	TCR1	R/W	H'FF60 8038	H'1F60 8038	32* <sup>3</sup>	
	DMA channel control register 1	CHCR1	R/W* <sup>1</sup>	H'FF60 803C	H'1F60 803C	32* <sup>3</sup>	
	DMA source address register 2	SAR2	R/W	H'FF60 8040	H'1F60 8040	32* <sup>3</sup>	
	DMA destination address register 2	DAR2	R/W	H'FF60 8044	H'1F60 8044	32* <sup>3</sup>	
	DMA transfer count register 2	TCR2	R/W	H'FF60 8048	H'1F60 8048	32* <sup>3</sup>	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
DMAC	DMA channel control register 2	CHCR2	R/W* <sup>1</sup>	H'FF60 804C	H'1F60 804C	32* <sup>3</sup>	
	DMA source address register 3	SAR3	R/W	H'FF60 8050	H'1F60 8050	32* <sup>3</sup>	
	DMA destination address register 3	DAR3	R/W	H'FF60 8054	H'1F60 8054	32* <sup>3</sup>	
	DMA transfer count register 3	TCR3	R/W	H'FF60 8058	H'1F60 8058	32* <sup>3</sup>	
	DMA channel control register 3	CHCR3	R/W* <sup>1</sup>	H'FF60 805C	H'1F60 805C	32* <sup>3</sup>	
	DMA operation register 0	DMAOR0	R/W* <sup>2</sup>	H'FF60 8060	H'1F60 8060	16* <sup>3</sup>	
	DMA source address register 4	SAR4	R/W	H'FF60 8070	H'1F60 8070	32* <sup>3</sup>	
	DMA destination address register 4	DAR4	R/W	H'FF60 8074	H'1F60 8074	32* <sup>3</sup>	
	DMA transfer count register 4	TCR4	R/W	H'FF60 8078	H'1F60 8078	32* <sup>3</sup>	
	DMA channel control register 4	CHCR4	R/W* <sup>1</sup>	H'FF60 807C	H'1F60 807C	32* <sup>3</sup>	
	DMA source address register 5	SAR5	R/W	H'FF60 8080	H'1F60 8080	32* <sup>3</sup>	
	DMA destination address register 5	DAR5	R/W	H'FF60 8084	H'1F60 8084	32* <sup>3</sup>	
	DMA transfer count register 5	TCR5	R/W	H'FF60 8088	H'1F60 8088	32* <sup>3</sup>	
	DMA channel control register 5	CHCR5	R/W* <sup>1</sup>	H'FF60 808C	H'1F60 808C	32* <sup>3</sup>	
	DMA source address register B0	SARB0	R/W	H'FF60 8120	H'1F60 8120	32* <sup>3</sup>	
	DMA destination address register B0	DARB0	R/W	H'FF60 8124	H'1F60 8124	32* <sup>3</sup>	
	DMA transfer count register B0	TCRB0	R/W	H'FF60 8128	H'1F60 8128	32* <sup>3</sup>	
	DMA source address register B1	SARB1	R/W	H'FF60 8130	H'1F60 8130	32* <sup>3</sup>	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
DMAC	DMA destination address register B1	DARB1	R/W	H'FF60 8134	H'1F60 8134	32* <sup>3</sup>	
	DMA transfer count register B1	TCRB1	R/W	H'FF60 8138	H'1F60 8138	32* <sup>3</sup>	
	DMA source address register B2	SARB2	R/W	H'FF60 8140	H'1F60 8140	32* <sup>3</sup>	
	DMA destination address register B2	DARB2	R/W	H'FF60 8144	H'1F60 8144	32* <sup>3</sup>	
	DMA transfer count register B2	TCRB2	R/W	H'FF60 8148	H'1F60 8148	32* <sup>3</sup>	
	DMA source address register B3	SARB3	R/W	H'FF60 8150	H'1F60 8150	32* <sup>3</sup>	
	DMA destination address register B3	DARB3	R/W	H'FF60 8154	H'1F60 8154	32* <sup>3</sup>	
	DMA transfer count register B3	TCRB3	R/W	H'FF60 8158	H'1F60 8158	32* <sup>3</sup>	
	DMA extended resource selector 0	DMARS0	R/W	H'FF60 9000	H'1F60 9000	16* <sup>3</sup>	
	DMA extended resource selector 1	DMARS1	R/W	H'FF60 9004	H'1F60 9004	16* <sup>3</sup>	
	DMA extended resource selector 2	DMARS2	R/W	H'FF60 9008	H'1F60 9008	16* <sup>3</sup>	
INTC	Interrupt control register 0	ICR0	R/W	H'FFD0 0000	H'1FD0 0000	32	
	Interrupt control register 1	ICR1	R/W	H'FFD0 001C	H'1FD0 001C	32	
	Interrupt priority register	INTPRI	R/W	H'FFD0 0010	H'1FD0 0010	32	
	Interrupt source register	INTREQ	R/W	H'FFD0 0024	H'1FD0 0024	32	
	Interrupt mask register	INTMSK	R/W	H'FFD0 0044	H'1FD0 0044	32	
	Interrupt mask clear register	INTMSKCLR	R/W	H'FFD0 0064	H'1FD0 0064	32	
	NMI flag control register	NMIFCR	R/W	H'FFD0 00C0	H'1FD0 00C0	32	
	User interrupt mask level register	USERIMASK	R/W	H'FFD3 0000	H'1FD3 0000	32	
	Interrupt priority register 0	INT2PRI0	R/W	H'FFD4 0000	H'1FD4 0000	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
INTC	Interrupt priority register 1	INT2PRI1	R/W	H'FFD4 0004	H'1FD4 0004	32	
	Interrupt priority register 2	INT2PRI2	R/W	H'FFD4 0008	H'1FD4 0008	32	
	Interrupt priority register 3	INT2PRI3	R/W	H'FFD4 000C	H'1FD4 000C	32	
	Interrupt priority register 4	INT2PRI4	R/W	H'FFD4 0010	H'1FD4 0010	32	
	Interrupt priority register 5	INT2PRI5	R/W	H'FFD4 0014	H'1FD4 0014	32	
	Interrupt priority register 6	INT2PRI6	R/W	H'FFD4 0018	H'1FD4 0018	32	
	Interrupt priority register 7	INT2PRI7	R/W	H'FFD4 001C	H'1FD4 001C	32	
	Interrupt priority register 8	INT2PRI8	R/W	H'FFD4 00A0	H'1FD4 00A0	32	
	Interrupt priority register 9	INT2PRI9	R/W	H'FFD4 00A4	H'1FD4 00A4	32	
	Interrupt priority register 10	INT2PRI10	R/W	H'FFD4 00A8	H'1FD4 00A8	32	
	Interrupt priority register 11	INT2PRI11	R/W	H'FFD4 00AC	H'1FD4 00AC	32	
	Interrupt priority register 12	INT2PRI12	R/W	H'FFD4 00B0	H'1FD4 00B0	32	
	Interrupt source register 0 (mask state is not affected)	INT2A0	R	H'FFD4 0030	H'1FD4 0030	32	
	Interrupt source register 01 (mask state is not affected)	INT2A01	R	H'FFD4 00C0	H'1FD4 00C0	32	
	Interrupt source register 1 (mask state is affected)	INT2A1	R	H'FFD4 0034	H'1FD4 0034	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
INTC	Interrupt source register 11 (mask state is affected)	INT2A11	R	H'FFD4 00C4	H'1FD4 00C4	32	
	Interrupt mask register	INT2MSKR	R/W	H'FFD4 0038	H'1FD4 0038	32	
	Interrupt mask register 1	INT2MSKR1	R/W	H'FFD4 00D0	H'1FD4 00D0	32	
	Interrupt mask clear register	INT2MSKCR	W	H'FFD4 003C	H'1FD4 003C	32	
	Interrupt mask clear register 1	INT2MSKCR1	W	H'FFD4 00D4	H'1FD4 00D4	32	
	Individual module interrupt source register 0	INT2B0	R	H'FFD4 0040	H'1FD4 0040	32	
	Individual module interrupt source register 2	INT2B2	R	H'FFD4 0048	H'1FD4 0048	32	
	Individual module interrupt source register 3	INT2B3	R	H'FFD4 004C	H'1FD4 004C	32	
	Individual module interrupt source register 4	INT2B4	R	H'FFD4 0050	H'1FD4 0050	32	
	Individual module interrupt source register 5	INT2B5	R	H'FFD4 0054	H'1FD4 0054	32	
	Individual module interrupt source register 6	INT2B6	R	H'FFD4 0058	H'1FD4 0058	32	
	Individual module interrupt source register 7	INT2B7	R	H'FFD4 005C	H'1FD4 005C	32	
	GPIO interrupt set register	INT2GPIC	R/W	H'FFD4 0090	H'1FD4 0090	32	
TMU	Timer output control register	TOCR	R/W	H'FFD8 0000	H'1FD8 0000	8	
	Timer start register 0	TSTR0	R/W	H'FFD8 0004	H'1FD8 0004	8	
	Timer constant register 0	TCOR0	R/W	H'FFD8 0008	H'1FD8 0008	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
TMU	Timer counter 0	TCNT0	R/W	H'FFD8 000C	H'1FD8 000C	32	
	Timer control register 0	TCR0	R/W	H'FFD8 0010	H'1FD8 0010	16	
	Timer constant register 1	TCOR1	R/W	H'FFD8 0014	H'1FD8 0014	32	
	Timer counter 1	TCNT1	R/W	H'FFD8 0018	H'1FD8 0018	32	
	Timer control register 1	TCR1	R/W	H'FFD8 001C	H'1FD8 001C	16	
	Timer constant register 2	TCOR2	R/W	H'FFD8 0020	H'1FD8 0020	32	
	Timer counter 2	TCNT2	R/W	H'FFD8 0024	H'1FD8 0024	32	
	Timer control register 2	TCR2	R/W	H'FFD8 0028	H'1FD8 0028	16	
	Input capture register 2	TCPR2	R	H'FFD8 002C	H'1FD8 002C	32	
	Timer start register 1	TSTR1	R/W	H'FFDC 0004	H'1FDC 0004	8	
	Timer constant register 3	TCOR3	R/W	H'FFDC 0008	H'1FDC 0008	32	
	Timer counter 3	TCNT3	R/W	H'FFDC 000C	H'1FDC 000C	32	
	Timer control register 3	TCR3	R/W	H'FFDC 0010	H'1FDC 0010	16	
	Timer constant register 4	TCOR4	R/W	H'FFDC 0014	H'1FDC 0014	32	
	Timer counter 4	TCNT4	R/W	H'FFDC 0018	H'1FDC 0018	32	
	Timer control register 4	TCR4	R/W	H'FFDC 001C	H'1FDC 001C	16	
	Timer constant register 5	TCOR5	R/W	H'FFDC 0020	H'1FDC 0020	32	
	Timer counter 5	TCNT5	R/W	H'FFDC 0024	H'1FDC 0024	32	
	Timer control register 5	TCR5	R/W	H'FFDC 0028	H'1FDC 0028	16	
SCIF	Serial mode register_0	SCSMR_0	R/W	H'FFE00000	H'1FE00000	16	
	Bit rate register_0	SCBRR_0	R/W	H'FFE00004	H'1FE00004	8	
	Serial control register_0	SCSCR_0	R/W	H'FFE00008	H'1FE00008	16	
	Transmit FIFO data register_0	SCFTDR_0	W	H'FFE0000C	H'1FE0000C	8	
	Serial status register_0	SCFSR_0	R/ (W)*4	H'FFE00010	H'1FE00010	16	
	Receive FIFO data register_0	SCFRDR_0	R	H'FFE00014	H'1FE00014	8	



Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
SCIF	FIFO control register_0	SCFCR_0	R/W	H'FFE00018	H'1FE00018	16	
	FIFO data count register_0	SCFDR_0	R	H'FFE0001C	H'1FE0001C	16	
	Serial port register_0	SCSPTR_0	R/W	H'FFE00020	H'1FE00020	16	
	Line status register_0	SCLSR_0	R/ (W)* <sup>5</sup>	H'FFE00024	H'1FE00024	16	
	Serial extension mode register_0	SCEMR_0	R/W	H'FFE00028	H'1FE00028	16	
	Serial mode register_1	SCSMR_1	R/W	H'FFE10000	H'1FE10000	16	
	Bit rate register_1	SCBRR_1	R/W	H'FFE10004	H'1FE10004	8	
	Serial control register_1	SCSCR_1	R/W	H'FFE10008	H'1FE10008	16	
	Transmit FIFO data register_1	SCFTDR_1	W	H'FFE1000C	H'1FE1000C	8	
	Serial status register_1	SCFSR_1	R/ (W)* <sup>4</sup>	H'FFE10010	H'1FE10010	16	
	Receive FIFO data register_1	SCFRDR_1	R	H'FFE10014	H'1FE10014	8	
	FIFO control register_1	SCFCR_1	R/W	H'FFE10018	H'1FE10018	16	
	FIFO data count register_1	SCFDR_1	R	H'FFE1001C	H'1FE1001C	16	
	Serial port register_1	SCSPTR_1	R/W	H'FFE10020	H'1FE10020	16	
	Line status register_1	SCLSR_1	R/ (W)* <sup>5</sup>	H'FFE10024	H'1FE10024	16	
	Serial extension mode register_1	SCEMR_1	R/W	H'FFE10028	H'1FE10028	16	
	Serial mode register_2	SCSMR_2	R/W	H'FFE20000	H'1FE20000	16	
	Bit rate register_2	SCBRR_2	R/W	H'FFE20004	H'1FE20004	8	
	Serial control register_2	SCSCR_2	R/W	H'FFE20008	H'1FE20008	16	
	Transmit FIFO data register_2	SCFTDR_2	W	H'FFE2000C	H'1FE2000C	8	
	Serial status register_2	SCFSR_2	R/ (W)* <sup>4</sup>	H'FFE20010	H'1FE20010	16	
	Receive FIFO data register_2	SCFRDR_2	R	H'FFE20014	H'1FE20014	8	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
SCIF	FIFO control register_2	SCFCR_2	R/W	H'FFE20018	H'1FE20018	16	
	FIFO data count register_2	SCFDR_2	R	H'FFE2001C	H'1FE2001C	16	
	Serial port register_2	SCSPTR_2	R/W	H'FFE20020	H'1FE20020	16	
	Line status register_2	SCLSR_2	R/ (W)* <sup>5</sup>	H'FFE20024	H'1FE20024	16	
	Serial extension mode register_2	SCEMR_2	R/W	H'FFE20028	H'1FE20028	16	
IIC	Slave control register	ICSCR	R/W	H'FFE7 0000	H'1FF7 0000	8	
	Master control register	ICMCR	R/W	H'FFE7 0004	H'1FF7 0004	8	
	Slave status register	ICSSR	R/ (W)* <sup>6</sup>	H'FFE7 0008	H'1FF7 0008	8	
	Master status register	ICMSR	R/ (W)* <sup>7</sup>	H'FFE7 000C	H'1FF7 000C	8	
	Slave interrupt enable register	ICSIER	R/W	H'FFE7 0010	H'1FF7 0010	8	
	Master interrupt enable register	ICMIER	R/W	H'FFE7 0014	H'1FF7 0014	8	
	Clock control register	ICCCR	R/W	H'FFE7 0018	H'1FF7 0018	8	
	Slave address register	ICSAR	R/W	H'FFE7 001C	H'1FF7 001C	8	
	Master address register	ICMAR	R/W	H'FFE7 0020	H'1FF7 0020	8	
	Receive data register	ICRXD	R/W	H'FFE7 0024	H'1FF7 0024	8	
	Transmit data register	ICTXD	R/W	H'FFE7 0024	H'1FF7 0024	8	
SSI_DMACH	DMA mode register 0	SSIDMMR0	R/W	H'FF40 1000	H'1F40 1000	32	
	RDMA transfer source address register 0	SSIRDMAADR0	R/W	H'FF40 1008	H'1F40 1008	32	
	RDMA transfer word count register 0	SSIRDMCNTR0	R/W	H'FF40 1010	H'1F40 1010	32	
	WDMA transfer destination address register 0	SSIWDMADR0	R/W	H'FF40 1018	H'1F40 1018	32	
	WDMA transfer word count register 0	SSIWDMCNTR0	R/W	H'FF40 1020	H'1F40 1020	32	
	DMA control register 0	SSIDMCOR0	R/W	H'FF40 1028	H'1F40 1028	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
SSI_DMAC0	Transmit suspension block counter 0	SSISTPBLCNT0	R/W	H'FF40 1030	H'1F40 1030	32	
	Transmit suspension transfer data register 0	SSISTPDR0	R/W	H'FF40 1038	H'1F40 1038	32	
	Block count source register 0	SSIBLCNTSR0	R/W	H'FF40 1040	H'1F40 1040	32	
	Block counter 0	SSIBLCNT0	R/W	H'FF40 1048	H'1F40 1048	32	
	n-times block transfer interrupt count source register 0	SSIBLNCNTSR0	R/W	H'FF40 1050	H'1F40 1050	32	
	n-times block counter 0	SSIBLNCNT0	R/W	H'FF40 1058	H'1F40 1058	32	
	DMA mode register 1	SSIDMMR1	R/W	H'FF40 1060	H'1F40 1060	32	
	RDMA transfer source address register 1	SSIRDMADR1	R/W	H'FF40 1068	H'1F40 1068	32	
	RDMA transfer word count register 1	SSIRDMCNTR1	R/W	H'FF40 1070	H'1F40 1070	32	
	WDMA transfer destination address register 1	SSIWDMADR1	R/W	H'FF40 1078	H'1F40 1078	32	
	WDMA transfer word count register 1	SSIWDMCNTR1	R/W	H'FF40 1080	H'1F40 1080	32	
	DMA control register 1	SSIDMCOR1	R/W	H'FF40 1088	H'1F40 1088	32	
	Transmit suspension block counter 1	SSISTPBLCNT1	R/W	H'FF40 1090	H'1F40 1090	32	
	Transmit suspension transfer data register 1	SSISTPDR1	R/W	H'FF40 1098	H'1F40 1098	32	
	Block count source register 1	SSIBLCNTSR1	R/W	H'FF40 10A0	H'1F40 10A0	32	
	Block counter 1	SSIBLCNT1	R/W	H'FF40 10A8	H'1F40 10A8	32	
	n-times block transfer interrupt count source register 1	SSIBLNCNTSR1	R/W	H'FF40 10B0	H'1F40 10B0	32	
	n-times block counter 1	SSIBLNCNT1	R/W	H'FF40 10B8	H'1F40 10B8	32	
	DMA mode register 2	SSIDMMR2	R/W	H'FF40 10C0	H'1F40 10C0	32	
	RDMA transfer source address register 2	SSIRDMADR2	R/W	H'FF40 10C8	H'1F40 10C8	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
SSI_DMAC0	RDMA transfer word count register 2	SSIRDMCNTR2	R/W	H'FF40 10D0	H'1F40 10D0	32	
	WDMA transfer destination address register 2	SSIWDMADR2	R/W	H'FF40 10D8	H'1F40 10D8	32	
	WDMA transfer word count register 2	SSIWDMCNTR2	R/W	H'FF40 10E0	H'1F40 10E0	32	
	DMA control register 2	SSIDMCOR2	R/W	H'FF40 10E8	H'1F40 10E8	32	
	Transmit suspension block counter 2	SSISTPBLCNT2	R/W	H'FF40 10F0	H'1F40 10F0	32	
	Transmit suspension transfer data register 2	SSISTPDR2	R/W	H'FF40 10F8	H'1F40 10F8	32	
	Block count source register 2	SSIBLCNTR2	R/W	H'FF40 1100	H'1F40 1100	32	
	Block counter 2	SSIBLCNT2	R/W	H'FF40 1108	H'1F40 1108	32	
	n-times block transfer interrupt count source register 2	SSIBLNCNTR2	R/W	H'FF40 1110	H'1F40 1110	32	
	n-times block counter 2	SSIBLNCNT2	R/W	H'FF40 1118	H'1F40 1118	32	
	DMA operation register 0	SSIDMAOR0	R/W	H'FF40 1180	H'1F40 1180	32	
	Interrupt status register 0	SSIDMINTSR0	R/W	H'FF40 1188	H'1F40 1188	32	
	Interrupt mask register 0	SSIDMINTMR0	R/W	H'FF40 1190	H'1F40 1190	32	
SSI_DMAC1	DMA mode register 3	SSIDMMR3	R/W	H'FF50 1000	H'1F50 1000	32	
	RDMA transfer source address register 3	SSIRDMADR3	R/W	H'FF50 1008	H'1F50 1008	32	
	RDMA transfer word count register 3	SSIRDMCNTR3	R/W	H'FF50 1010	H'1F50 1010	32	
	WDMA transfer destination address register 3	SSIWDMADR3	R/W	H'FF50 1008	H'1F50 1018	32	
	WDMA transfer word count register 3	SSIWDMCNTR3	R/W	H'FF50 1020	H'1F50 1020	32	
	DMA control register 3	SSIDMCOR3	R/W	H'FF50 1028	H'1F50 1028	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
SSI_DMACH1	Transmit suspension block counter 3	SSISTPBLCNT3	R/W	H'FF50 1030	H'1F50 1030	32	
	Transmit suspension transfer data register 3	SSISTPDR3	R/W	H'FF50 1038	H'1F50 1038	32	
	Block count source register 3	SSIBLCNTR3	R/W	H'FF50 1040	H'1F50 1040	32	
	Block counter 3	SSIBLCNT3	R/W	H'FF50 1048	H'1F50 1048	32	
	n-times block transfer interrupt count source register 3	SSIBLNCNTR3	R/W	H'FF50 1050	H'1F50 1050	32	
	n-times block counter 3	SSIBLNCNT3	R/W	H'FF50 1058	H'1F50 1058	32	
	DMA mode register 4	SSIDMMR4	R/W	H'FF50 1060	H'1F50 1060	32	
	RDMA transfer source address register 4	SSIRDMADR4	R/W	H'FF50 1068	H'1F50 1068	32	
	RDMA transfer word count register 4	SSIRDMCNTR4	R/W	H'FF50 1070	H'1F50 1070	32	
	WDMA transfer destination address register 4	SSIWDMADR4	R/W	H'FF50 1078	H'1F50 1078	32	
	WDMA transfer word count register 4	SSIWDMCNTR4	R/W	H'FF50 1080	H'1F50 1080	32	
	DMA control register 4	SSIDMCOR4	R/W	H'FF50 1088	H'1F50 1088	32	
	Transmit suspension block counter 4	SSISTPBLCNT4	R/W	H'FF50 1090	H'1F50 1090	32	
	Transmit suspension transfer data register 4	SSISTPDR4	R/W	H'FF50 1098	H'1F50 1098	32	
	Block count source register 4	SSIBLCNTR4	R/W	H'FF50 10A0	H'1F50 10A0	32	
	Block counter 4	SSIBLCNT4	R/W	H'FF50 10A8	H'1F50 10A8	32	
	n-times block transfer interrupt count source register 4	SSIBLNCNTR4	R/W	H'FF50 10B0	H'1F50 10B0	32	
	n-times block counter 4	SSIBLNCNT4	R/W	H'FF50 10B8	H'1F50 10B8	32	
	DMA mode register 5	SSIDMMR5	R/W	H'FF50 10C0	H'1F50 10C0	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
SSI_DMACH1	RDMA transfer source address register 5	SSIRDMADR5	R/W	H'FF50 10C8	H'1F50 10C8	32	
	RDMA transfer word count register 5	SSIRDMCNTR5	R/W	H'FF50 10D0	H'1F50 10D0	32	
	WDMA transfer destination address register 5	SSIWDMADR5	R/W	H'FF50 10D8	H'1F50 10D8	32	
	WDMA transfer word count register 5	SSIWDMCNTR5	R/W	H'FF50 10E0	H'1F50 10E0	32	
	DMA control register 5	SSIDMCOR5	R/W	H'FF50 10E8	H'1F50 10E8	32	
	Transmit suspension block counter 5	SSISTPBLCNT5	R/W	H'FF50 10F0	H'1F50 10F0	32	
	Transmit suspension transfer data register 5	SSISTPDR5	R/W	H'FF50 10F8	H'1F50 10F8	32	
	Block count source register 5	SSIBLCNTR5	R/W	H'FF50 1100	H'1F50 1100	32	
	Block counter 5	SSIBLCNT5	R/W	H'FF50 1108	H'1F50 1108	32	
	n-times block transfer interrupt count source register 5	SSIBLNCNTR5	R/W	H'FF50 1110	H'1F50 1110	32	
	n-times block counter 5	SSIBLNCNT5	R/W	H'FF50 1118	H'1F50 1118	32	
	DMA operation register 1	SSIDMAOR1	R/W	H'FF50 1180	H'1F50 1180	32	
	Interrupt status register 1	SSIDMINTSR1	R/W	H'FF50 1188	H'1F50 1188	32	
	Interrupt mask register 1	SSIDMINTMR1	R/W	H'FF50 1190	H'1F50 1190	32	
SSI_CH0 to 5	Control register 0	SSICR0	R/W	H'FF40 2000	H'1F40 2000	32	
	Status register 0	SSISR0	R/W* <sup>8</sup>	H'FF40 2004	H'1F40 2004	32	
	Transmit data register 0	SSITDR0	R/W	H'FF40 2008	H'1F40 2008	32	
	Receive data register 0	SSIRDR0	R	H'FF40 200C	H'1F40 200C	32	
	Control register 1	SSICR1	R/W	H'FF40 3000	H'1F40 3000	32	
	Status register 1	SSISR1	R/W* <sup>8</sup>	H'FF40 3004	H'1F40 3004	32	
	Transmit data register 1	SSITDR1	R/W	H'FF40 3008	H'1F40 3008	32	
	Receive data register 1	SSIRDR1	R	H'FF40 300C	H'1F40 300C	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
SSI_CH0 to 5	Control register 2	SSICR2	R/W	H'FF40 4000	H'1F40 4000	32	
	Status register 2	SSISR2	R/W* <sup>8</sup>	H'FF40 4004	H'1F40 4004	32	
	Transmit data register 2	SSITDR2	R/W	H'FF40 4008	H'1F40 4008	32	
	Receive data register 2	SSIRDR2	R	H'FF40 400C	H'1F40 400C	32	
	Control register 3	SSICR3	R/W	H'FF50 2000	H'1F50 2000	32	
	Status register 3	SSISR3	R/W* <sup>8</sup>	H'FF50 2004	H'1F50 2004	32	
	Transmit data register 3	SSITDR3	R/W	H'FF50 2008	H'1F50 2008	32	
	Receive data register 3	SSIRDR3	R	H'FF50 200C	H'1F50 200C	32	
	Control register 4	SSICR4	R/W	H'FF50 3000	H'1F50 3000	32	
	Status register 4	SSISR4	R/W* <sup>8</sup>	H'FF50 3004	H'1F50 3004	32	
	Transmit data register 4	SSITDR4	R/W	H'FF50 3008	H'1F50 3008	32	
	Receive data register 4	SSIRDR4	R	H'FF50 300C	H'1F50 300C	32	
	Control register 5	SSICR5	R/W	H'FF50 4000	H'1F50 4000	32	
	Status register 5	SSISR5	R/W* <sup>8</sup>	H'FF50 4004	H'1F50 4004	32	
	Transmit data register 5	SSITDR5	R/W	H'FF50 4008	H'1F50 4008	32	
	Receive data register 5	SSIRDR5	R	H'FF50 400C	H'1F50 400C	32	
EtherC	EtherC mode register	ECMR	R/W	H'FEF0 0100*	H'1EF0 0100*	32	
	EtherC status register	ECSR	R/W	H'FEF0 0110*	H'1EF0 0110*	32	
	EtherC interrupt permission register	ECSIPR	R/W	H'FEF0 0118*	H'1EF0 0118*	32	
	Receive frame length register	RFLR	R/W	H'FEF0 0108*	H'1EF0 0108*	32	
	PHY interface register	PIR	R/W	H'FEF0 0120*	H'1EF0 0120*	32	
	MAC address high register	MAHR	R/W	H'FEF0 01C0*	H'1EF0 01C0*	32	
	MAC address low register	MALR	R/W	H'FEF0 01C8*	H'1EF0 01C8*	32	
	PHY status register	PSR	R	H'FEF0 0128*	H'1EF0 0128*	32	
	Transmit retry over counter register	TROCR	R/W	H'FEF0 01D0*	H'1EF0 01D0*	32	
	Delayed collision detect counter register	CDCR	R/W	H'FEF0 01D4*	H'1EF0 01D4*	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
EtherC	Lost carrier counter register	LCCR	R/W	H'FEF0 01D8*	H'1EF0 01D8*	32	
	Carrier not detect counter register	CNDCR	R/W	H'FEF0 01DC*	H'1EF0 01DC*	32	
	CRC error frame receive counter register	CEFCR	R/W	H'FEF0 01E4*	H'1EF0 01E4*	32	
	Frame receive error counter register	FRECR	R/W	H'FEF0 01E8*	H'1EF0 01E8*	32	
	Too-short frame receive counter register	TSFRCCR	R/W	H'FEF0 01EC*	H'1EF0 01EC*	32	
	Too-long frame receive counter register	TLFRCCR	R/W	H'FEF0 01F0*	H'1EF0 01F0*	32	
	Residual-bit frame receive counter register	RFCR	R/W	H'FEF0 01F4*	H'1EF0 01F4*	32	
	Multicast address frame receive counter register	MAFCR	R/W	H'FEF0 01F8*	H'1EF0 01F8*	32	
	IPG register	IPGR	R/W	H'FEF0 0150*	H'1EF0 0150*	32	
	Automatic PAUSE frame register	APR	R/W	H'FEF0 0154*	H'1EF0 0154*	32	
	Manual PAUSE frame register	MPR	R/W	H'FEF0 0158*	H'1EF0 0158*	32	
	Automatic PAUSE frame retransmit count register	TPAUSER	R/W	H'FEF0 0164*	H'1EF0 0164*	32	
	Random number generation counter upper limit setting register	RDMLR	R/W	H'FEF0 0140*	H'1EF0 0140*	32	
	PAUSE Frame Receive Counter Register	RFCF	R/W	H'FEF0 0160*	H'1EF0 0160*	32	
	PAUSE frame retransmit counter register	TPAUSECR	R/W	H'FEF0 0168*	H'1EF0 0168*	32	
	Broadcast frame receive count setting register	BCFRR	R/W	H'FEF0 016C*	H'1EF0 016C*	32	



Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
E-DMAC	E-DMAC mode register	EDMR	R/W	H'FEF0 0000*	H'1EF0 0000*	32	
	E-DMAC transmit request register	EDTRR	R/W	H'FEF0 0008*	H'1EF0 0008*	32	
	E-DMAC receive request register	EDRRR	R/W	H'FEF0 0010*	H'1EF0 0010*	32	
	Transmit descriptor list start address register	TDLAR	R/W	H'FEF0 0018*	H'1EF0 0018*	32	
	Receive descriptor list start address register	RDLAR	R/W	H'FEF0 0020*	H'1EF0 0020*	32	
	EtherC/E-DMAC status register	EESR	R/W	H'FEF0 0028*	H'1EF0 0028*	32	
	EtherC/E-DMAC status interrupt permission register	EESIPR	R/W	H'FEF0 0030*	H'1EF0 0030*	32	
	Transmit/receive status copy enable register	TRSCER	R/W	H'FEF0 0038*	H'1EF0 0038*	32	
	Receive missed-frame counter register	RMFCR	R	H'FEF0 0040*	H'1EF0 0040*	32	
	Transmit FIFO threshold register	TFTR	R/W	H'FEF0 0048*	H'1EF0 0048*	32	
	FIFO depth register	FDR	R/W	H'FEF0 0050*	H'1EF0 0050*	32	
	Receiving method control register	RMCR	R/W	H'FEF0 0058*	H'1EF0 0058*	32	
	Transmit FIFO underrun counter	TFUCR	R/W	H'FEF0 0064*	H'1EF0 0064*	32	
	Receive FIFO overflow counter	RFOCR	R/W	H'FEF0 0068*	H'1EF0 0068*	32	
	Receive buffer write address register	RBWAR	R	H'FEF0 00C8*	H'1EF0 00C8*	32	
	Receive descriptor fetch address register	RDFAR	R	H'FEF0 00CC*	H'1EF0 00CC*	32	
	Transmit buffer read address register	TBRAR	R	H'FEF0 00D4*	H'1EF0 00D4*	32	
	Transmit descriptor fetch address register	TDFAR	R	H'FEF0 00D8*	H'1EF0 00D8*	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
E-DMAC	Flow control start FIFO threshold setting register	FCFTR	R/W	H'FEF0 0070*	H'1EF0 0070*	32	
	Receive data padding insert register	RPADIR	R/W	H'FEF0 0078*	H'1EF0 0078*	32	
	Transmit interrupt setting register	TRIMD	R/W	H'FEF0 007C*	H'1EF0 007C*	32	
	Independent output signal setting register	IOSR	R/W	H'FEF0 006C*	H'1EF0 006C*	32	
USB	System configuration control register	SYSCFG	R/W	H'FE40 0000		16	
	CPU bus wait setting register	BUSWAIT	R/W	H'FE40 0002		16	
	System configuration status register	SYSSTS	R	H'FE40 0004		16	
	Device state control register	DVSTCTR	R/W	H'FE40 0008		16	
	Test mode register	TESTMODE	R/W	H'FE40 000C		16	
	DMA0-FIFO bus configuration register	D0FBCFG	R/W	H'FE40 0010		16	
	DMA1-FIFO bus configuration register	D1FBCFG	R/W	H'FE40 0012		16	
	CFIFO port register	CFIFO	R/W	H'FE40 0014		8, 16, 32	
	D0FIFO port register	D0FIFO	R/W	H'FE40 0018		8, 16, 32	
				H'FE40 0180			
	D1FIFO port register	D1FIFO	R/W	H'FE40 001C		8, 16, 32	
				H'FE40 01C0			
	CFIFO port select register	CFIFOSEL	R/W	H'FE40 0020		16	
	CFIFO port control register	CFIFOCTR	R/W	H'FE40 0022		16	
	D0FIFO port select register	D0FIFOSEL	R/W	H'FE40 0028		16	
	D0FIFO port control register	D0FIFOCTR	R/W	H'FE40 002A		16	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
USB	D1FIFO port select register	D1FIFOSEL	R/W	H'FE40 002C		16	
	D1FIFO port control register	D1FIFOCTR	R/W	H'FE40 002E		16	
	Interrupt enable register 0	INTENB0	R/W	H'FE40 0030		16	
	Interrupt enable register 1	INTENB1	R/W	H'FE40 0032		16	
	BRDY interrupt enable register	BRDYENB	R/W	H'FE40 0036		16	
	NRDY interrupt enable register	NRDYENB	R/W	H'FE40 0038		16	
	BEMP interrupt enable register	BEMPENB	R/W	H'FE40 003A		16	
	SOF output configuration register	SOFCFG	R/W	H'FE40 003C		16	
	Interrupt status register 0	INTSTS0	R/W	H'FE40 0040		16	
	Interrupt status register 1	INTSTS1	R/W	H'FE40 0042		16	
	BRDY interrupt status register	BRDYSTS	R/W	H'FE40 0046		16	
	NRDY interrupt status register	NRDYSTS	R/W	H'FE40 0048		16	
	BEMP interrupt status register	BEMPSTS	R/W	H'FE40 004A		16	
	Frame number register	FRMNUM	R/W	H'FE40 004C		16	
	μFrame number register	UFRMNUM	R/W	H'FE40 004E		16	
	USB address register	USBADDR	R	H'FE40 0050		16	
	USB request type register	USBREQ	R	H'FE40 0054		16	
	USB request value register	USBVAL	R	H'FE40 0056		16	
	USB request index register	USBINDX	R	H'FE40 0058		16	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
USB	USB request length register	USBLENG	R	H'FE40 005A		16	
	DCP configuration register	DCPCFG	R/W	H'FE40 005C		16	
	DCP maximum packet size register	DCPMAXP	R/W	H'FE40 005E		16	
	DCP control register	DPCPTR	R/W	H'FE40 0060		16	
	Pipe window select register	PIPESEL	R/W	H'FE40 0064		16	
	Pipe configuration register	PIPECFG	R/W	H'FE40 0068		16	
	Pipe buffer setting register	PIPEBUF	R/W	H'FE40 006A		16	
	Pipe maximum packet size register	PIPEMAXP	R/W	H'FE40 006C		16	
	Pipe cycle control register	PIPEPERI	R/W	H'FE40 006E		16	
	Pipe 1 control register	PIPE1CTR	R/W	H'FE40 0070		16	
	Pipe 2 control register	PIPE2CTR	R/W	H'FE40 0072		16	
	Pipe 3 control register	PIPE3CTR	R/W	H'FE40 0074		16	
	Pipe 4 control register	PIPE4CTR	R/W	H'FE40 0076		16	
	Pipe 5 control register	PIPE5CTR	R/W	H'FE40 0078		16	
	Pipe 6 control register	PIPE6CTR	R/W	H'FE40 007A		16	
	Pipe 7 control register	PIPE7CTR	R/W	H'FE40 007C		16	
	Pipe 8 control register	PIPE8CTR	R/W	H'FE40 007E		16	
	Pipe 9 control register	PIPE9CTR	R/W	H'FE40 0080		16	
	Pipe 1 transaction counter enable register	PIPE1TRE	R/W	H'FE40 0090		16	
	Pipe 1 transaction counter register	PIPE1TRN	R/W	H'FE40 0092		16	
	Pipe 2 transaction counter enable register	PIPE2TRE	R/W	H'FE40 0094		16	
	Pipe 2 transaction counter register	PIPE2TRN	R/W	H'FE40 0096		16	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
USB	Pipe 3 transaction counter enable register	PIPE3TRE	R/W	H'FE40 0098		16	
	Pipe 3 transaction counter register	PIPE3TRN	R/W	H'FE40 009A		16	
	Pipe 4 transaction counter enable register	PIPE4TRE	R/W	H'FE40 009C		16	
	Pipe 4 transaction counter register	PIPE4TRN	R/W	H'FE40 009E		16	
	Pipe 5 transaction counter enable register	PIPE5TRE	R/W	H'FE40 00A0		16	
	Pipe 5 transaction counter register	PIPE5TRN	R/W	H'FE40 00A2		16	
	Device address 0 configuration register	DEVADD0	R/W	H'FE40 00D0		16	
	Device address 1 configuration register	DEVADD1	R/W	H'FE40 00D2		16	
	Device address 2 configuration register	DEVADD2	R/W	H'FE40 00D4		16	
	Device address 3 configuration register	DEVADD3	R/W	H'FE40 00D6		16	
	Device address 4 configuration register	DEVADD4	R/W	H'FE40 00D8		16	
	Device address 5 configuration register	DEVADD5	R/W	H'FE40 00DA		16	
	Device address 6 configuration register	DEVADD6	R/W	H'FE40 00DC		16	
	Device address 7 configuration register	DEVADD7	R/W	H'FE40 00DE		16	
	Device address 8 configuration register	DEVADD8	R/W	H'FE40 00E0		16	
	Device address 9 configuration register	DEVADD9	R/W	H'FE40 00E2		16	
	Device address A configuration register	DEVADDA	R/W	H'FE40 00E4		16	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
LCDC	Palette data register 00 to FF	LDPR00 to LDPRFF	R/W	H'FFE3 0000 to H'FFE3 03FC	H'1FE3 0000 to H'1FE3 03FC	32	
	LCDC input clock register	LDICKR	R/W	H'FFE3 0400	H'1FE3 0400	16	
	LCDC module type register	LDMTR	R/W	H'FFE3 0402	H'1FE3 0402	16	
	LCDC data format register	LDDFR	R/W	H'FFE3 0404	H'1FE3 0404	16	
	LCDC data fetch start address register for upper display panel	LDSARU	R/W	H'FFE3 0408	H'1FE3 0408	32	
	LCDC data fetch start address register for lower display panel	LDSARL	R/W	H'FFE3 040C	H'1FE3 040C	32	
	LCDC fetch data line address offset register for display panel	LDLAOR	R/W	H'FFE3 0410	H'1FE3 0410	16	
	LCDC palette control register	LDPALCR	R/W	H'FFE3 0412	H'1FE3 0412	16	
	LCDC horizontal character number register	LDHCNR	R/W	H'FFE3 0414	H'1FE3 0414	16	
	LCDC horizontal synchronization signal register	LDHSYNR	R/W	H'FFE3 0416	H'1FE3 0416	16	
	LCDC vertical displayed line number register	LDVDLNR	R/W	H'FFE3 0418	H'1FE3 0418	16	
	LCDC vertical total line number register	LDVTLNR	R/W	H'FFE3 041A	H'1FE3 041A	16	
	LCDC vertical synchronization signal register	LDVSYNR	R/W	H'FFE3 041C	H'1FE3 041C	16	
	LCDC AC modulation signal toggle line number register	LDACLNR	R/W	H'FFE3 041E	H'1FE3 041E	16	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
LCDC	LDCD interrupt control register	LDINTR	R/W	H'FFE3 0420	H'1FE3 0420	16	
	LDCD power management mode register	LDPMMR	R/W	H'FFE3 0424	H'1FE3 0424	16	
	LDCD power supply sequence period register	LDPSPR	R/W	H'FFE3 0426	H'1FE3 0426	16	
	LDCD control register	LDCNTR	R/W	H'FFE3 0428	H'1FE3 0428	16	
	LDCD user specified interrupt control register	LDUINTR	R/W	H'FFE3 0434	H'1FE3 0434	16	
	LDCD user specified interrupt line number register	LDUINLNR	R/W	H'FFE3 0436	H'1FE3 0436	16	
	LDCD memory access interval number register	LDLIRNR	R/W	H'FFE3 0440	H'1FE3 0440	16	
VDC2 graphics block 1	Graphics block control register	GRCMEN1	R/W	H'FFEC 0000*	H'1FEC 0000*	32/16/8	
	Bus control register	GRCBUSCNT1	R/W	H'FFEC 0004*	H'1FEC 0004*	32/16/8	
	Reserved	—	R	H'FFEC 0008*	H'1FEC 0008*	32/16/8	
	Reserved	—	R	H'FFEC 000C*	H'1FEC 000C*	32/16/8	
	Reserved	—	R	H'FFEC 0300*	H'1FEC 0300*	32/16/8	
	Reserved	—	R	H'FFEC 0304*	H'1FEC 0304*	32/16/8	
	Graphic image base address register	GROPSADR1	R/W	H'FFEC 0308*	H'1FEC 0308*	32/16/8	
	Graphic image area register	GROPSWH1	R/W	H'FFEC 030C*	H'1FEC 030C*	32/16/8	
	Graphic image line offset register	GROPSOFST1	R/W	H'FFEC 0310*	H'1FEC 0310*	32/16/8	
	Graphic image start position register	GROPDPHV1	R/W	H'FFEC 0314*	H'1FEC 0314*	32/16/8	
	Reserved	—	R	H'FFEC 0318*	H'1FEC 0318*	32/16/8	
	Reserved	—	R	H'FFEC 031C*	H'1FEC 031C*	32/16/8	
	Reserved	—	R	H'FFEC 0320*	H'1FEC 0320*	32/16/8	
	Reserved	—	R	H'FFEC 0324*	H'1FEC 0324*	32/16/8	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
VDC2 graphics block 1	Reserved	—	R	H'FFEC 0328*	H'1FEC 0328*	32/16/8	
	Color register for outside of graphic image area	GROPBASERG B1	R/W	H'FFEC 032C*	H'1FEC 032C*	32/16/8	
VDC2 graphics block 2	Graphics block control register	GRCMEN2	R/W	H'FFED 0000*	H'1FED 0000*	32/16/8	
	Bus control register	GRCBUSCNT2	R/W	H'FFED 0004*	H'1FED 0004*	32/16/8	
	Reserved	—	R	H'FFED 0008*	H'1FED 0008*	32/16/8	
	Reserved	—	R	H'FFED 000C*	H'1FED 000C*	32/16/8	
	Reserved	—	R	H'FFED 0300*	H'1FED 0300*	32/16/8	
	Reserved	—	R	H'FFED 0304*	H'1FED 0304*	32/16/8	
	Graphic image base address register	GROPSADR2	R/W	H'FFED 0308*	H'1FED 0308*	32/16/8	
	Graphic image area register	GROPSWH2	R/W	H'FFED 030C*	H'1FED 030C*	32/16/8	
	Graphic image line offset register	GROPSOFST2	R/W	H'FFED 0310*	H'1FED 0310*	32/16/8	
	Graphic image start position register	GRODPHV2	R/W	H'FFED 0314*	H'1FED 0314*	32/16/8	
	$\alpha$ control area register	GROPEWH2	R/W	H'FFED 0318*	H'1FED 0318*	32/16/8	
	$\alpha$ control area start position register	GROPEDPHV2	R/W	H'FFED 031C*	H'1FED 031C*	32/16/8	
	$\alpha$ control register	GROPEDPA2	R/W	H'FFED 0320*	H'1FED 0320*	32/16/8	
	Chroma-key control register	GROPCRKY0_2	R/W	H'FFED 0324*	H'1FED 0324*	32/16/8	
	Chroma-key color register	GROPCRKY1_2	R/W	H'FFED 0328*	H'1FED 0328*	32/16/8	
	Color register for outside of graphic image area	GROPBASERG B2	R/W	H'FFED 032C*	H'1FED 032C*	32/16/8	
VDC2 graphics block 3	Graphics block control register	GRCMEN3	R/W	H'FFEE 0000*	H'1FEE 0000*	32/16/8	
	Bus control register	GRCBUSCNT3	R/W	H'FFEE 0004*	H'1FEE 0004*	32/16/8	
	Reserved	—	R	H'FFEE 0008*	H'1FEE 0008*	32/16/8	
	Reserved	—	R	H'FFEE 000C*	H'1FEE 000C*	32/16/8	
	Reserved	—	R	H'FFEE 0300*	H'1FEE 0300*	32/16/8	



Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
VDC2 graphics block 3	Reserved	—	R	H'FFEE 0304*	H'1FEE 0304*	32/16/8	
	Graphic image base address register	GROPSADR3	R/W	H'FFEE 0308*	H'1FEE 0308*	32/16/8	
	Graphic image area register	GROPSWH3	R/W	H'FFEE 030C*	H'1FEE 030C*	32/16/8	
	Graphic image line offset register	GROPSOFST3	R/W	H'FFEE 0310*	H'1FEE 0310*	32/16/8	
	Graphic image start position register	GROPDPHV3	R/W	H'FFEE 0314*	H'1FEE 0314*	32/16/8	
	$\alpha$ control area register	GROPEWH3	R/W	H'FFEE 0318*	H'1FEE 0318*	32/16/8	
	$\alpha$ control area start position register	GROPEDPHV3	R/W	H'FFEE 031C*	H'1FEE 031C*	32/16/8	
	$\alpha$ control register	GROPEDPA3	R/W	H'FFEE 0320*	H'1FEE 0320*	32/16/8	
	Chroma-key control register	GROPCRKY0_3	R/W	H'FFEE 0324*	H'1FEE 0324*	32/16/8	
	Chroma-key color register	GROPCRKY1_3	R/W	H'FFEE 0328*	H'1FEE 0328*	32/16/8	
	Color register for outside of graphic image area	GROPBASERGB3	R/W	H'FFEE 032C*	H'1FEE 032C*	32/16/8	
VDC2 graphics block 4	Graphics block control register	GRCMEN4	R/W	H'FFEF 0000*	H'1FEF 0000*	32/16/8	
	Bus control register	GRCBUSCNT4	R/W	H'FFEF 0004*	H'1FEF 0004*	32/16/8	
	Reserved	—	R	H'FFEF 0008*	H'1FEF 0008*	32/16/8	
	Reserved	—	R	H'FFEF 000C*	H'1FEF 000C*	32/16/8	
	Reserved	—	R	H'FFEF 0300*	H'1FEF 0300*	32/16/8	
	Reserved	—	R	H'FFEF 0304*	H'1FEF 0304*	32/16/8	
	Graphic image base address register	GROPSADR4	R/W	H'FFEF 0308*	H'1FEF 0308*	32/16/8	
	Graphic image area register	GROPSWH4	R/W	H'FFEF 030C*	H'1FEF 030C*	32/16/8	
	Graphic image line offset register	GROPSOFST4	R/W	H'FFEF 0310*	H'1FEF 0310*	32/16/8	
	Graphic image start position register	GROPDPHV4	R/W	H'FFEF 0314*	H'1FEF 0314*	32/16/8	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
VDC2 graphics block 4	$\alpha$ control area register	GROPEWH4	R/W	H'FFEF 0318*	H'1FEF 0318*	32/16/8	
	$\alpha$ control area start position register	GROPEDPHV4	R/W	H'FFEF 031C*	H'1FEF 031C*	32/16/8	
	$\alpha$ control register	GROPEDPA4	R/W	H'FFEF 0320*	H'1FEF 0320*	32/16/8	
	Chroma-key control register	GROPCRKY0_4	R/W	H'FFEF 0324*	H'1FEF 0324*	32/16/8	
	Chroma-key color register	GROPCRKY1_4	R/W	H'FFEF 0328*	H'1FEF 0328*	32/16/8	
	Color register for outside of graphic image area	GROPBASERGB4	R/W	H'FFEF 032C*	H'1FEF 032C*	32/16/8	
VDC2 display control block	SG mode register	SGMODE	R/W	H'FFEB 0000*	H'1FEB 0000*	32/16/8	
	Interrupt output control register	SGINTCNT	R/W	H'FFEB 0004*	H'1FEB 0004*	32/16/8	
	Sync signal control register	SYNCNT	R/W	H'FFEB 0008*	H'1FEB 0008*	32/16/8	
	External sync signal input timing control register	EXTSYNCNT	R/W	H'FFEB 000C*	H'1FEB 000C*	32/16/8	
	Reserved	—	R	H'FFEB 0100*	H'1FEB 0100*	32/16/8	
	Sync signal size register	SYNSIZE	R/W	H'FFEB 0104*	H'1FEB 0104*	32/16/8	
	Vertical sync signal timing control register	VSYNCTIM	R/W	H'FFEB 0108*	H'1FEB 0108*	32/16/8	
	Horizontal sync signal timing control register	HSYNTIM	R/W	H'FFEB 010C*	H'1FEB 010C*	32/16/8	
	Gate clock signal timing control register	CLSTIM	R/W	H'FFEB 0110*	H'1FEB 0110*	32/16/8	
	Sampling start signal timing control register	SPLTIM	R/W	H'FFEB 0118*	H'1FEB 0118*	32/16/8	
	Gate control signal timing control register	COMTIM	R/W	H'FFEB 011C*	H'1FEB 011C*	32/16/8	
	SGDE area start position register	SGDESTART	R/W	H'FFEB 0120*	H'1FEB 0120*	32/16/8	
	SGDE area size register	SGDESIZE	R/W	H'FFEB 0124*	H'1FEB 0124*	32/16/8	
	CDE chroma-key color register	CDECRKY	R/W	H'FFEB 0128*	H'1FEB 0128*	32/16/8	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
VDC2 display control block	Reserved	—	R	H'FFEB 0148*	H'1FEB 0148*	32/16/8	
	T-1004 control register	T1004CNT	R/W	H'FFEB 0200*	H'1FEB 0200*	32/16/8	
	T-1004 video start position register	T1004OFFSET	R/W	H'FFEB 0204*	H'1FEB 0204*	32/16/8	
	Reserved	—	R	H'FFEB 0208*	H'1FEB 0208*	32/16/8	
	Reserved	—	R	H'FFEB 020C*	H'1FEB 020C*	32/16/8	
FLCTL	Common control register	FLCMNCR	R/W	H'FFE9 0000	H'1FE9 0000	32	
	Command control register	FLCMDRCR	R/W	H'FFE9 0004	H'1FE9 0004	32	
	Command code register	FLCMCDR	R/W	H'FFE9 0008	H'1FE9 0008	32	
	Address register	FLADR	R/W	H'FFE9 000C	H'1FE9 000C	32	
	Address register 2	FLADR2	R/W	H'FFE9 003C	H'1FE9 003C	32	
	Data register	FLDATAR	R/W	H'FFE9 0010	H'1FE9 0010	32	
	Data counter register	FLDTCNTR	R/W	H'FFE9 0014	H'1FE9 0014	32	
	Interrupt DMA control register	FLINTDMACR	R/W	H'FFE9 0018	H'1FE9 0018	32	
	Ready busy timeout setting register	FLBSYTMR	R/W	H'FFE9 001C	H'1FE9 001C	32	
	Ready busy timeout counter	FLBSYCNT	R	H'FFE9 0020	H'1FE9 0020	32	
	Data FIFO register	FLDTFIFO	R/W	H'FFE9 0024/ H'FFE9 0050	H'1FE9 0024/ H'1FE9 0050	32	
	Control code FIFO register	FLECFIFO	R/W	H'FFE9 0028/ H'FFE9 0060	H'1FE9 0028/ H'1FE9 0060	32	
	Transfer control register	FLTRCR	R/W	H'FFE9 002C	H'1FE9 002C	8	
SRC	SRC input data register	SRCID	R/W	H'FFF3 0000	H'1FF3 0000	16, 32	
	SRC output data register	SRCOD	R	H'FFF3 0004	H'1FF3 0004	16, 32	
	SRC input data control register	SRCIDCTRL	R/W	H'FFF3 0008	H'1FF3 0008	16	
	SRC output data control register	SRCODCTRL	R/W	H'FFF3 000A	H'1FF3 000A	16	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
SRC	SRC control register	SRCCTRL	R/W	H'FFF3 000C	H'1FF3 000C	16	
	SRC status register	SRCSTAT	R/ (W)* <sup>9</sup>	H'FFF3 000E	H'1FF3 000E	16	
GPIO	Port A control register	PTIO_A	R/W	H'FFF1 0000*	H'1FF1 0000*	16* <sup>10</sup>	
	Port B control register	PTIO_B	R/W	H'FFF1 0004*	H'1FF1 0004*	16* <sup>10</sup>	
	Port C control register	PTIO_C	R/W	H'FFF1 0008*	H'1FF1 0008*	16* <sup>10</sup>	
	Port D control register	PTIO_D	R/W	H'FFF1 000C*	H'1FF1 000C*	16* <sup>10</sup>	
	Port E control register	PTIO_E	R/W	H'FFF1 0010*	H'1FF1 0010*	16* <sup>10</sup>	
	Port F control register	PTIO_F	R/W	H'FFF1 0014*	H'1FF1 0014*	16* <sup>10</sup>	
	Port G control register	PTIO_G	R/W	H'FFF1 0018*	H'1FF1 0018*	16* <sup>10</sup>	
	Port H control register	PTIO_H	R/W	H'FFF1 001C*	H'1FF1 001C*	16* <sup>10</sup>	
	Port I control register	PTIO_I	R/W	H'FFF1 0020*	H'1FF1 0020*	16* <sup>10</sup>	
	Port J control register	PTIO_J	R/W	H'FFF1 0024*	H'1FF1 0024*	16* <sup>10</sup>	
	Port A data register	PTDAT_A	R/W	H'FFF1 0040*	H'1FF1 0040*	16* <sup>10</sup>	
	Port B data register	PTDAT_B	R/W	H'FFF1 0044*	H'1FF1 0044*	16* <sup>10</sup>	
	Port C data register	PTDAT_C	R/W	H'FFF1 0048*	H'1FF1 0048*	16* <sup>10</sup>	
	Port D data register	PTDAT_D	R/W	H'FFF1 004C*	H'1FF1 004C*	16* <sup>10</sup>	
	Port E data register	PTDAT_E	R/W	H'FFF1 0050*	H'1FF1 0050*	16* <sup>10</sup>	
	Port F data register	PTDAT_F	R/W	H'FFF1 0054*	H'1FF1 0054*	16* <sup>10</sup>	
	Port G data register	PTDAT_G	R/W	H'FFF1 0058*	H'1FF1 0058*	16* <sup>10</sup>	
	Port H data register	PTDAT_H	R/W	H'FFF1 005C*	H'1FF1 005C*	16* <sup>10</sup>	
	Port I data register	PTDAT_I	R/W	H'FFF1 0060*	H'1FF1 0060*	16* <sup>10</sup>	
	Port J data register	PTDAT_J	R/W	H'FFF1 0064*	H'1FF1 0064*	16* <sup>10</sup>	
	Input-pin pull-up control register	PTPUL_SPCL	R/W	H'FFF1 00E0*	H'1FF1 00E0*	16* <sup>10</sup>	
	Pin select register A	PTSEL_A	R/W	H'FFF1 0080*	H'1FF1 0080*	16* <sup>10</sup>	
	Pin select register B	PTSEL_B	R/W	H'FFF1 0084*	H'1FF1 0084*	16* <sup>10</sup>	
	Pin select register C	PTSEL_C	R/W	H'FFF1 0088*	H'1FF1 0088*	16* <sup>10</sup>	
	Pin select register D	PTSEL_D	R/W	H'FFF1 008C*	H'1FF1 008C*	16* <sup>10</sup>	
	Pin select register E	PTSEL_E	R/W	H'FFF1 0090*	H'1FF1 0090*	16* <sup>10</sup>	
	Pin select register F	PTSEL_F	R/W	H'FFF1 0094*	H'1FF1 0094*	16* <sup>10</sup>	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
GPIO	Pin select register G	PTSEL_G	R/W	H'FFF1 0098*	H'1FF1 0098*	16* <sup>10</sup>	
	Pin select register H	PTSEL_H	R/W	H'FFF1 009C*	H'1FF1 009C*	16* <sup>10</sup>	
	Pin select register I	PTSEL_I	R/W	H'FFF1 00A0*	H'1FF1 00A0*	16* <sup>10</sup>	
	Pin select register J	PTSEL_J	R/W	H'FFF1 00A4*	H'1FF1 00A4*	16* <sup>10</sup>	
	Pin select register K	PTSEL_K	R/W	H'FFF1 00A8*	H'1FF1 00A8*	16* <sup>10</sup>	
	Pin select register P	PTSEL_P	R/W	H'FFF1 00AC*	H'1FF1 00AC*	16* <sup>10</sup>	
	Pin select register R	PTSEL_R	R/W	H'FFF1 00B0*	H'1FF1 00B0*	16* <sup>10</sup>	
	Pin select register S	PTSEL_S	R/W	H'FFF1 00B4*	H'1FF1 00B4*	16* <sup>10</sup>	
	Hi-Z register A	PTHIZ_A	R/W	H'FFF1 00E8*	H'1FF1 00E8*	16* <sup>10</sup>	
	Hi-Z register B	PTHIZ_B	R/W	H'FFF1 00EC*	H'1FF1 00EC*	16* <sup>10</sup>	
	Special select register	PTSEL_SPCL	R/W	H'FFF1 00F0	H'1FF1 00F0	16* <sup>10</sup>	
Power-down mode	Standby control register	STBCR	R/W	H'FFC8 0020	H'1FC8 0020	32	
	Module stop register 0	MSTPCR0	R/W	H'FFC8 0030	H'1FC8 0030	32	
	Module stop register 1	MSTPCR1	R/W	H'FFC8 0038	H'1FC8 0038	32	
UBC	Match condition setting register 0	CBR0	R/W	H'FF20 0000	H'1F20 0000	32	
	Match operation setting register 0	CRR0	R/W	H'FF20 0004	H'1F20 0004	32	
	Match address setting register 0	CAR0	R/W	H'FF20 0008	H'1F20 0008	32	
	Match address mask setting register 0	CAMR0	R/W	H'FF20 000C	H'1F20 000C	32	
	Match condition setting register 1	CBR1	R/W	H'FF20 0020	H'1F20 0020	32	
	Match operation setting register 1	CRR1	R/W	H'FF20 0024	H'1F20 0024	32	
	Match address setting register 1	CAR1	R/W	H'FF20 0028	H'1F20 0028	32	
	Match address mask setting register 1	CAMR1	R/W	H'FF20 002C	H'1F20 002C	32	
	Match data setting register 1	CDR1	R/W	H'FF20 0030	H'1F20 0030	32	

Module	Register Name	Abbreviation	R/W	P4 Area Address*	Area 7 Address*	Access Size	Remarks
UBC	Match data mask setting register 1	CDMR1	R/W	H'FF20 0034	H'1F20 0034	32	
	Execution count break register 1	CETR1	R/W	H'FF20 0038	H'1F20 0038	32	
	Channel match flag register	CCMFR	R/W	H'FF20 0600	H'1F20 0600	32	
	Break control register	CBCR	R/W	H'FF20 0620	H'1F20 0620	32	

Notes: \* The P4 area addresses shown here are the P4 area addresses in the virtual address space. The area 7 addresses should be accessed via the area 7 in the physical address space using the TLB.

- Only 0 can be written to the HE and TE bits in CHCR after 1 is read from the bits to clear the flags.
- Only 0 can be written to the AE and NMIF bits in DMAOR after 1 is read from the bits to clear the flags.
- Do not access the registers with the access size not specified here.
- Only 0 can be written to the registers to clear the flags. In addition, bits 15 to 8, 3, and 2 are read-only bits allowing no write-accesses.
- Only 0 can be written to the registers to clear the flags. In addition, bits 15 to 1 are read-only bits allowing no write-accesses.
- Only 0 can be written to bits 4 to 0 to clear the flags.
- Only 0 can be written to bits 6 to 0 to clear the flags.
- All the bits except bits 27 and 26 in the registers are read-only bits; bits 27 and 26 allow both read- and write- accesses. For details, refer to section 18.3.17, Status Registers 0 to 5 (SSISR0 to SSISR5).
- Bits 15 to 3 are read-only bits. Only 0 can be written to bits 2 to 0 after 1 is read from the bits.
- The registers can only be accessed in 16-bit units; be sure to access the registers with the specified access size.
- For the standby control register, also refer to figure 9.1, Block Diagram of CPG.

Module	Register Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size	Remarks (Initial Value)
MCU	Version control register	VCR	R/W	H'FF80 0000		32	H'0B04 0000 0000 0000
	Memory interface mode register	MIM	R/W	H'FF80 0008		32	H'0000 0000 061A 0x40
	SDRAM control register	SCR	R/W	H'FF80 0010		32	H'0000 0000 0000 0000
	SDRAM timing register	STR	R/W	H'FF80 0018		32	H'0000 0000 00FF FFE7
	SDRAM row attribute register	SDRA	R/W	H'FF80 0030		32	H'0000 0000 0000 0200
	SDRAM mode register	SDMR	R	H'FFAx xxxx		32	—
	Arbitration mode register	AMR	R/W	H'FF80 0200		32	H'0000 0000 0400 0000
	Linear-to-tiled memory address translation control register 0	LTC0	R/W	H'FF80 0100		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address register 0	LTAD0	R/W	H'FF80 0108		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address mask register 0	LTAM0	R/W	H'FF80 0110		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation control register 1	LTC1	R/W	H'FF80 0118		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address register 1	LTAD1	R/W	H'FF80 0120		32	H'0000 0000 0000 0000

Module	Register Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size	Remarks (Initial Value)
MCU	Linear-to-tiled memory address translation area start address mask register 1	LTAM1	R/W	H'FF80 0128		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation control register 2	LTC2	R/W	H'FF80 0130		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address register 2	LTAD2	R/W	H'FF80 0138		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address mask register 2	LTAM2	R/W	H'FF80 0140		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation control register 3	LTC3	R/W	H'FF80 0148		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address register 3	LTAD3	R/W	H'FF80 0150		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address mask register 3	LTAM3	R/W	H'FF80 0158		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation control register 4	LTC4	R/W	H'FF80 0160		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address register 4	LTAD4	R/W	H'FF80 0168		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address mask register 4	LTAM4	R/W	H'FF80 0170		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation control register 5	LTC5	R/W	H'FF80 0178		32	H'0000 0000 0000 0000



Module	Register Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size	Remarks (Initial Value)
MCU	Linear-to-tiled memory address translation area start address register 5	LTAD5	R/W	H'FF80 0180		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address mask register 5	LTAM5	R/W	H'FF80 0188		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation control register 6	LTC6	R/W	H'FF80 0190		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address register 6	LTAD6	R/W	H'FF80 0198		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address mask register 6	LTAM6	R/W	H'FF80 01A0		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation control register 7	LTC7	R/W	H'FF80 01A8		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address register 7	LTAD7	R/W	H'FF80 01B0		32	H'0000 0000 0000 0000
	Linear-to-tiled memory address translation area start address mask register 7	LTAM7	R/W	H'FF80 01B8		32	H'0000 0000 0000 0000
	Request mask setting register	RQM	R/W	H'FF80 0218		32	H'0000 0000 0000 0000
	Bus control register	BCR	R/W	H'FF80 1000		32	H'0000 0000 3800 0000
	CS0 bus control register	CS0BCR	R/W	H'FF80 2000		32	H'0000 0000 7777 7x80

Module	Register Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size	Remarks (Initial Value)
MCU	CS0 wait control register	CS0WCR	R/W	H'FF80 2008		32	H'0000 0000 7777 770F
	CS3 bus control register	CS3BCR	R/W	H'FF80 2030		32	H'0000 0000 7777 7380
	CS3 wait control register	CS3WCR	R/W	H'FF80 2038		32	H'0000 0000 7777 770F

---

Module	Register Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Register Access Size*	Remarks
ATAPI	ATAPI status	ATAPI_STATU S	R/W	H'FFF0 0084		32	
	Interrupt enable	ATAPI_INT_EN ABLE	R/W	H'FFF0 0088		32	
	PIO timing	ATAPI_PIO_TI MING	R/W	H'FFF0 008C		32	
	Multiword DMA timing	ATAPI_MULTI_ TIMING	R/W	H'FFF0 0090		32	
	Ultra DMA timing	ATAPI_ULTRA_ TIMING	R/W	H'FFF0 0094		32	
	Descriptor table base address	ATAPI_DTB_A DR	R/W	H'FFF0 0098		32	
	DMA start address	ATAPI_DMA_ START_ADR	R/W	H'FFF0 009C		32	
	DMA transfer count	ATAPI_DMA_ TRANS_CNT	R/W	H'FFF0 00A0		32	
	ATAPI control 2	ATAPI_ CONTROL2	R/W	H'FFF0 00A4		32	
	Reserved		R	H'FFF0 00A8		32	
	Reserved		R	H'FFF0 00AC		32	
	ATAPI signal status	ATAPI_SIG_ST	R	H'FFF0 00B0		32	
	Byte swap	ATAPI_BYTE_ SWAP	R/W	H'FFF0 00BC		32	

Note: \* The above registers should be accessed in longword units (32 bits); byte and word accesses are prohibited.

Module	Register Name	Abbreviation	R/W	P4 Area Address* <sup>2</sup>	Area 7 Address* <sup>2</sup>	Access Size	Remarks (WPR)* <sup>1</sup>
G2D	System control	SCLR	R/W	H'FFEA 0000	H'1FEA 0000	32	×
	Status	SR	R	H'FFEA 0004	H'1FEA 0004	32	×
	Status register clear	SRCR	W	H'FFEA 0008	H'1FEA 0008	32	×
	Interrupt enable	IER	R/W	H'FFEA 000C	H'1FEA 000C	32	O
	Interrupt command ID	ICIDR	R	H'FFEA 0010	H'1FEA 0010	32	×
	Return address 0	RTN0R	R	H'FFEA 0040	H'1FEA 0040	32	O
	Return address 1	RTN1R	R	H'FFEA 0044	H'1FEA 0044	32	O
	Display list start address	DLSAR	R/W	H'FFEA 0048	H'1FEA 0048	32	×
	2-dimensional source area start address	SSAR	R/W	H'FFEA 004C	H'1FEA 004C	32	O
	Rendering start address	RSAR	R/W	H'FFEA 0050	H'1FEA 0050	32	O
	Work area start address	WSAR	R/W	H'FFEA 0054	H'1FEA 0054	32	O
	Source stride	SSTRR	R/W	H'FFEA 0058	H'1FEA 0058	32	O
	Destination stride	DSTRR	R/W	H'FFEA 005C	H'1FEA 005C	32	O
	Endian conversion control	ENDCVR	R/W	H'FFEA 0060	H'1FEA 0060	32	×
	Source transparent color	STCR	R/W	H'FFEA 0080	H'1FEA 0080	32	O
	Destination transparent color	DTCR	R/W	H'FFEA 0084	H'1FEA 0084	32	O
	Alpha value	ALPHR	R/W	H'FFEA 0088	H'1FEA 0088	32	O
	Color offset	COFSR	R/W	H'FFEA 008C	H'1FEA 008C	32	O
	Rendering control	RCLR	R/W	H'FFEA 00C0	H'1FEA 00C0	32	O
	Command status	CSTR	R	H'FFEA 00C4	H'1FEA 00C4	32	×
	Current pointer	CURR	R	H'FFEA 00C8	H'1FEA 00C8	32	×
	Local offset	LCOR	R	H'FFEA 00CC	H'1FEA 00CC	32	×
	System clipping area MAX	SCLMAR	R	H'FFEA 00D0	H'1FEA 00D0	32	O
	User clipping area MIN	UCLMIR	R	H'FFEA 00D4	H'1FEA 00D4	32	O
	User clipping area MAX	UCLMAR	R	H'FFEA 00D8	H'1FEA 00D8	32	O
	Relative user clipping area MIN	RUCLMIR	R	H'FFEA 00DC	H'1FEA 00DC	32	O
	Relative user clipping area MAX	RUCLMAR	R	H'FFEA 00E0	H'1FEA 00E0	32	O

Module	Register Name	Abbreviation	R/W	P4 Area Address* <sup>2</sup>	Area 7 Address* <sup>2</sup>	Access Size	Remarks (WPR)* <sup>1</sup>
G2D	Rendering control 2	RCL2R	R/W	H'FFEA 00F0	H'1FEA 00F0	32	O
	Pattern offset	POFSR	R/W	H'FFEA 00F8	H'1FEA 00F8	32	O
	Coordinate transformation control	GTRCR	R/W	H'FFEA 0100	H'1FEA 0100	32	O
	Matrix parameter A	MTRAR	R/W	H'FFEA 0104	H'1FEA 0104	32	O
	Matrix parameter B	MTRBR	R/W	H'FFEA 0108	H'1FEA 0108	32	O
	Matrix parameter C	MTRCR	R/W	H'FFEA 010C	H'1FEA 010C	32	O
	Matrix parameter D	MTRDR	R/W	H'FFEA 0110	H'1FEA 0110	32	O
	Matrix parameter E	MTRER	R/W	H'FFEA 0114	H'1FEA 0114	32	O
	Matrix parameter F	MTRFR	R/W	H'FFEA 0118	H'1FEA 0118	32	O
	Matrix parameter G	MTRGR	R/W	H'FFEA 011C	H'1FEA 011C	32	O
	Matrix parameter H	MTRHR	R/W	H'FFEA 0120	H'1FEA 0120	32	O
	Matrix parameter I	MTRIR	R/W	H'FFEA 0124	H'1FEA 0124	32	O
	Coordinate transformation offset X	GTROFSXR	R/W	H'FFEA 0128	H'1FEA 0128	32	O
	Coordinate transformation offset Y	GTROFSYR	R/W	H'FFEA 012C	H'1FEA 012C	32	O
	Z clipping area MIN	ZCLPMINR	R/W	H'FFEA 0130	H'1FEA 0130	32	O
	Z clipping area MAX	ZCLPMAXR	R/W	H'FFEA 0134	H'1FEA 0134	32	O
	Z saturation value MIN	ZSATVMINR	R/W	H'FFEA 0138	H'1FEA 0138	32	O

Notes: 1. WPR command setting

O: Possible

×: Impossible

2. The P4 area addresses shown here are the P4 area addresses in the virtual address space. The area 7 addresses should be accessed via the area 7 in the physical address space using the TLB.

If any address not specified here is written to, operation is not guaranteed.

Module	Register Name	Abbreviation	R/W	P4 Area Address* <sup>1</sup>	Area 7 Address* <sup>1</sup>	Access Size	Remarks (Initial Value)* <sup>2</sup>
H-UDI	Instruction register	SDIR	R	H'FC11 0000	H'1C11 0000	16	H'0EFF
	Interrupt source register	SDINT	R/W	H'FC11 0018	H'1C11 0018	16	H'0000
	Boundary scan register	SDBSR	—	—	—	—	—
	Bypass register	SDBPR	—	—	—	—	Undefined

- Notes: 1. The P4 area addresses shown here are the P4 area addresses in the virtual address space. The area 7 addresses should be accessed via the area 7 in the physical address space using the TLB.
2. Registers are initialized when the TRST pin level is low or TAP is in the Test-Logic-Reset state.

## 32.2 Register States in Each Operation Mode

**Table 32.2 Register States in Each Operation Mode (1)**

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
Exception handling	TRA	Undefined	Retained	Retained
	EXPEVT	H'0000 0000	Retained	Retained
	INTEVT	Undefined	Retained	Retained
	EXPMASK	H'0000 0000	Retained	Retained
MMU	PTEH	Undefined	Retained	Retained
	PTEL	Undefined	Retained	Retained
	TTB	Undefined	Retained	Retained
	TEA	Undefined	Retained	Retained
	MMUCR	H'0000 0000	Retained	Retained
	PTEA	H'0000 xxx0	Retained	Retained
	PASCR	H'0000 0000	Retained	Retained
	IRMCR	H'0000 0000	Retained	Retained
Cache	CCR	H'0000 0000	Retained	Retained
	QACR0	Undefined	Retained	Retained
	QACR1	Undefined	Retained	Retained
	RAMCR	H'0000 0000	Retained	Retained
On-chip memory	RAMCR	H'0000 0000	Retained	Retained
CPG	FRQCR	H'x032 0044* <sup>1</sup>	Retained	Retained
	PLLCR	H'0000 E001	Retained	Retained
	VDC2CLKCR	H'0000 0080	Retained	Retained
DMAC	SAR0	Undefined	Retained	Retained
	DAR0	Undefined	Retained	Retained
	TCR0	Undefined	Retained	Retained
	CHCR0	H'4000 0000	Retained	Retained
	SAR1	Undefined	Retained	Retained
	DAR1	Undefined	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
DMAC	TCR1	Undefined	Retained	Retained
	CHCR1	H'4000 0000	Retained	Retained
	SAR2	Undefined	Retained	Retained
	DAR2	Undefined	Retained	Retained
	TCR2	Undefined	Retained	Retained
	CHCR2	H'4000 0000	Retained	Retained
	SAR3	Undefined	Retained	Retained
	DAR3	Undefined	Retained	Retained
	TCR3	Undefined	Retained	Retained
	CHCR3	H'4000 0000	Retained	Retained
	DMAOR0	H'0000	Retained	Retained
	SAR4	Undefined	Retained	Retained
	DAR4	Undefined	Retained	Retained
	TCR4	Undefined	Retained	Retained
	CHCR4	H'4000 0000	Retained	Retained
	SAR5	Undefined	Retained	Retained
	DAR5	Undefined	Retained	Retained
	TCR5	Undefined	Retained	Retained
	CHCR5	H'4000 0000	Retained	Retained
	SARB0	Undefined	Retained	Retained
	DARB0	Undefined	Retained	Retained
	TCRB0	Undefined	Retained	Retained
	SARB1	Undefined	Retained	Retained
	DARB1	Undefined	Retained	Retained
	TCRB1	Undefined	Retained	Retained
	SARB2	Undefined	Retained	Retained
	DARB2	Undefined	Retained	Retained
	TCRB2	Undefined	Retained	Retained
	SARB3	Undefined	Retained	Retained
	DARB3	Undefined	Retained	Retained
	TCRB3	Undefined	Retained	Retained



Module	Register Abbreviation	Power-on Reset	Sleep	Standby
DMAC	DMARS0	H'0000	Retained	Retained
	DMARS1	H'0000	Retained	Retained
	DMARS2	H'0000	Retained	Retained
MCU	VCR	H'0B04 0000 0000 0000	Retained	Retained
	MIM	H'0000 0000 061A 0x40	Retained	Retained
	SCR	H'0000 0000 0000 0000	Retained	Retained
	STR	H'0000 0000 00FF FFE7	Retained	Retained
	SDRA	H'0000 0000 0000 0200	Retained	Retained
	SDMR	—	Retained	Retained
	AMR	H'0000 0000 0400 0000	Retained	Retained
	LTC0	H'0000 0000 0000 0000	Retained	Retained
	LTAD0	H'0000 0000 0000 0000	Retained	Retained
	LTAM0	H'0000 0000 0000 0000	Retained	Retained
	LTC1	H'0000 0000 0000 0000	Retained	Retained
	LTAD1	H'0000 0000 0000 0000	Retained	Retained
	LTAM1	H'0000 0000 0000 0000	Retained	Retained
	LTC2	H'0000 0000 0000 0000	Retained	Retained
	LTAD2	H'0000 0000 0000 0000	Retained	Retained
	LTAM2	H'0000 0000 0000 0000	Retained	Retained
	LTC3	H'0000 0000 0000 0000	Retained	Retained
	LTAD3	H'0000 0000 0000 0000	Retained	Retained
	LTAM3	H'0000 0000 0000 0000	Retained	Retained
	LTC4	H'0000 0000 0000 0000	Retained	Retained
	LTAD4	H'0000 0000 0000 0000	Retained	Retained
	LTAM4	H'0000 0000 0000 0000	Retained	Retained
	LTC5	H'0000 0000 0000 0000	Retained	Retained
	LTAD5	H'0000 0000 0000 0000	Retained	Retained
	LTAM5	H'0000 0000 0000 0000	Retained	Retained
	LTC6	H'0000 0000 0000 0000	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
MCU	LTAD6	H'0000 0000 0000 0000	Retained	Retained
	LTAM6	H'0000 0000 0000 0000	Retained	Retained
	LTC7	H'0000 0000 0000 0000	Retained	Retained
	LTAD7	H'0000 0000 0000 0000	Retained	Retained
	LTAM7	H'0000 0000 0000 0000	Retained	Retained
	RQM	H'0000 0000 0000 0000	Retained	Retained
	BCR	H'0000 0000 3800 0000	Retained	Retained
	CS0BCR	H'0000 0000 7777 7x80	Retained	Retained
	CS0WCR	H'0000 0000 7777 770F	Retained	Retained
	CS3BCR	H'0000 0000 7777 7380	Retained	Retained
	CS3WCR	H'0000 0000 7777 770F	Retained	Retained
INTC	ICR0	H'x000 0000	Retained	Retained
	ICR1	H'0000 0000	Retained	Retained
	INTPRI	H'0000 0000	Retained	Retained
	INTREQ	H'0000 0000	Retained	Retained
	INTMSK	H'FF00 0000	Retained	Retained
	INTMSKCLR	H'0000 0000	Retained	Retained
	NMIFCR	H'x000 0000	Retained	Retained
	USERIMASK	H'0000 0000	Retained	Retained
	INT2PRI0	H'0000 0000	Retained	Retained
	INT2PRI1	H'0000 0000	Retained	Retained
	INT2PRI2	H'0000 0000	Retained	Retained
	INT2PRI3	H'0000 0000	Retained	Retained
	INT2PRI4	H'0000 0000	Retained	Retained
	INT2PRI5	H'0000 0000	Retained	Retained
	INT2PRI6	H'0000 0000	Retained	Retained
	INT2PRI7	H'0000 0000	Retained	Retained
	INT2PRI8	H'0000 0000	Retained	Retained
	INT2PRI9	H'0000 0000	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
INTC	INT2PRI10	H'0000 0000	Retained	Retained
	INT2PRI11	H'0000 0000	Retained	Retained
	INT2PRI12	H'0000 0000	Retained	Retained
	INT2A0	H'xxxx xxxx	Retained	Retained
	INT2A01	H'xxxx xxxx	Retained	Retained
	INT2A1	H'0000 0000	Retained	Retained
	INT2A11	H'0000 0000	Retained	Retained
	INT2MSKR	H'FFFF FFFF	Retained	Retained
	INT2MSKR1	H'FFFF FFFF	Retained	Retained
	INT2MSKCR	H'0000 0000	Retained	Retained
	INT2MSKCR1	H'0000 0000	Retained	Retained
	INT2B0	H'xxxx xxxx	Retained	Retained
	INT2B2	H'xxxx xxxx	Retained	Retained
	INT2B3	H'xxxx xxxx	Retained	Retained
	INT2B4	H'xxxx xxxx	Retained	Retained
	INT2B5	H'xxxx xxxx	Retained	Retained
	INT2B6	H'xxxx xxxx	Retained	Retained
	INT2B7	H'xxxx xxxx	Retained	Retained
	INT2GPIC	H'0000 0000	Retained	Retained
TMU	TOCR	H'00	Retained	Retained
	TSTR0	H'00	Retained	Retained
	TCOR0	H'FFFF FFFF	Retained	Retained
	TCNT0	H'FFFF FFFF	Retained	Retained
	TCR0	H'0000	Retained	Retained
	TCOR1	H'FFFF FFFF	Retained	Retained
	TCNT1	H'FFFF FFFF	Retained	Retained
	TCR1	H'0000	Retained	Retained
	TCOR2	H'FFFF FFFF	Retained	Retained
	TCNT2	H'FFFF FFFF	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
TMU	TCR2	H'0000	Retained	Retained
	TCPR2	H'xxxx xxxx	Retained	Retained
	TSTR1	H'00	Retained	Retained
	TCOR3	H'FFFF FFFF	Retained	Retained
	TCNT3	H'FFFF FFFF	Retained	Retained
	TCR3	H'0000	Retained	Retained
	TCOR4	H'FFFF FFFF	Retained	Retained
	TCNT4	H'FFFF FFFF	Retained	Retained
	TCR4	H'0000	Retained	Retained
	TCOR5	H'FFFF FFFF	Retained	Retained
	TCNT5	H'FFFF FFFF	Retained	Retained
	TCR5	H'0000	Retained	Retained
SCIF	SCSMR_0	H'0000	Retained	Retained
	SCBRR_0	H'FF	Retained	Retained
	SCSCR_0	H'0000	Retained	Retained
	SCFTDR_0	Undefined	Retained	Retained
	SCFSR_0	H'0060	Retained	Retained
	SCFRDR_0	Undefined	Retained	Retained
	SCFCR_0	H'0000	Retained	Retained
	SCFDR_0	H'0000	Retained	Retained
	SCSPTR_0	H'0050	Retained	Retained
	SCLSR_0	H'0000	Retained	Retained
	SCEMR_0	H'0000	Retained	Retained
	SCSMR_1	H'0000	Retained	Retained
	SCBRR_1	H'FF	Retained	Retained
	SCSCR_1	H'0000	Retained	Retained
	SCFTDR_1	Undefined	Retained	Retained
	SCFSR_1	H'0060	Retained	Retained
	SCFRDR_1	Undefined	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
SCIF	SCFCR_1	H'0000	Retained	Retained
	SCFDR_1	H'0000	Retained	Retained
	SCSPTR_1	H'0050	Retained	Retained
	SCLSR_1	H'0000	Retained	Retained
	SCEMR_1	H'0000	Retained	Retained
	SCSMR_2	H'0000	Retained	Retained
	SCBRR_2	H'FF	Retained	Retained
	SCSCR_2	H'0000	Retained	Retained
	SCFTDR_2	Undefined	Retained	Retained
	SCFSR_2	H'0060	Retained	Retained
	SCFRDR_2	Undefined	Retained	Retained
	SCFCR_2	H'0000	Retained	Retained
	SCFDR_2	H'0000	Retained	Retained
	SCSPTR_2	H'0050	Retained	Retained
	SCLSR_2	H'0000	Retained	Retained
	SCEMR_2	H'0000	Retained	Retained
IIC	ICSCR	H'00	Retained	Retained
	ICMCR	H'x0	Retained	Retained
	ICSSR	H'00	Retained	Retained
	ICMSR	H'00	Retained	Retained
	ICSIER	H'00	Retained	Retained
	ICMIER	H'00	Retained	Retained
	ICCCR	H'00	Retained	Retained
	ICSAR	H'00	Retained	Retained
	ICMAR	H'00	Retained	Retained
	ICRXD	H'00	Retained	Retained
	ICTXD	H'00	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
ATAPI	ATAPI_CONTROL	H'0000 0020	Retained	Retained
	ATAPI_STATUS	H'0000 0000	Retained	Retained
	ATAPI_INT_ENABLE	H'0000 0000	Retained	Retained
	ATAPI_PIO_TIMING	H'0000 0000	Retained	Retained
	ATAPI_MULTI_TIMING	H'0000 0000	Retained	Retained
	ATAPI_ULTRA_TIMING	H'0000 0000	Retained	Retained
	ATAPI_DTB_ADR	H'0000 0000	Retained	Retained
	ATAPI_DMA_START_ADR	H'0000 0000	Retained	Retained
	ATAPI_DMA_TRANS_CNT	H'0000 0000	Retained	Retained
	ATAPI_CONTROL2	H'0000 0000	Retained	Retained
	ATAPI_SIG_ST	H'0000 000x	Retained	Retained
	ATAPI_BYTE_SWAP	H'0000 0000	Retained	Retained
SSI_DMAC0	SSIDMMR0	H'0000 0000	Retained	Retained
	SSIRDMADR0	H'0000 0000	Retained	Retained
	SSIRDMCNTR0	H'0000 0000	Retained	Retained
	SSIWDMADR0	H'0000 0000	Retained	Retained
	SSIWDMCNTR0	H'0000 0000	Retained	Retained
	SSIDMCOR0	H'0000 0000	Retained	Retained
	SSISTPBLCNT0	H'0000 0000	Retained	Retained
	SSISTPDRO	H'0000 0000	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
SSI_DMAC0	SSIBLCNTR0	H'0000 0000	Retained	Retained
	SSIBLCNT0	H'0000 0000	Retained	Retained
	SSIBLNCNTR0	H'0000 0000	Retained	Retained
	SSIBLNCNT0	H'0000 0000	Retained	Retained
	SSIDMMR1	H'0000 0000	Retained	Retained
	SSIRDMADR1	H'0000 0000	Retained	Retained
	SSIRDMCNTR1	H'0000 0000	Retained	Retained
	SSIWDMADR1	H'0000 0000	Retained	Retained
	SSIWDMCNTR1	H'0000 0000	Retained	Retained
	SSIDMCOR1	H'0000 0000	Retained	Retained
	SSISTPBLCNT1	H'0000 0000	Retained	Retained
	SSISTPDR1	H'0000 0000	Retained	Retained
	SSIBLCNTR1	H'0000 0000	Retained	Retained
	SSIBLCNT1	H'0000 0000	Retained	Retained
	SSIBLNCNTR1	H'0000 0000	Retained	Retained
	SSIBLNCNT1	H'0000 0000	Retained	Retained
	SSIDMMR2	H'0000 0000	Retained	Retained
	SSIRDMADR2	H'0000 0000	Retained	Retained
	SSIRDMCNTR2	H'0000 0000	Retained	Retained
	SSIWDMADR2	H'0000 0000	Retained	Retained
	SSIWDMCNTR2	H'0000 0000	Retained	Retained
	SSIDMCOR2	H'0000 0000	Retained	Retained
	SSISTPBLCNT2	H'0000 0000	Retained	Retained
	SSISTPDR2	H'0000 0000	Retained	Retained

<b>Module</b>	<b>Register Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
SSI_DMAC0	SSIBLCNTR2	H'0000 0000	Retained	Retained
	SSIBLCNT2	H'0000 0000	Retained	Retained
	SSIBLNCNTR2	H'0000 0000	Retained	Retained
	SSIBLNCNT2	H'0000 0000	Retained	Retained
	SSIDMAOR0	H'0000 0000	Retained	Retained
	SSIDMINTSR0	H'0101 0101	Retained	Retained
	SSIDMINTMR0	H'1F1F 1F1F	Retained	Retained
SSI_DMAC1	SSIDMMR3	H'0000 0000	Retained	Retained
	SSIRDMADR3	H'0000 0000	Retained	Retained
	SSIRDMCNTR3	H'0000 0000	Retained	Retained
	SSIWDMADR3	H'0000 0000	Retained	Retained
	SSIWDMCNTR3	H'0000 0000	Retained	Retained
	SSIDMCOR3	H'0000 0000	Retained	Retained
	SSISTPBLCNT3	H'0000 0000	Retained	Retained
	SSISTPDR3	H'0000 0000	Retained	Retained
	SSIBLCNTR3	H'0000 0000	Retained	Retained
	SSIBLCNT3	H'0000 0000	Retained	Retained
	SSIBLNCNTR3	H'0000 0000	Retained	Retained
	SSIBLNCNT3	H'0000 0000	Retained	Retained
	SSIDMMR4	H'0000 0000	Retained	Retained
	SSIRDMADR4	H'0000 0000	Retained	Retained
	SSIRDMCNTR4	H'0000 0000	Retained	Retained
	SSIWDMADR4	H'0000 0000	Retained	Retained
	SSIWDMCNTR4	H'0000 0000	Retained	Retained



Module	Register Abbreviation	Power-on Reset	Sleep	Standby
SSI_DMAC1	SSIDMCOR4	H'0000 0000	Retained	Retained
	SSISTPBLCNT4	H'0000 0000	Retained	Retained
	SSISTPDR4	H'0000 0000	Retained	Retained
	SSIBLCNTR4	H'0000 0000	Retained	Retained
	SSIBLCNT4	H'0000 0000	Retained	Retained
	SSIBLNCNTR4	H'0000 0000	Retained	Retained
	SSIBLNCNT4	H'0000 0000	Retained	Retained
	SSIDMMR5	H'0000 0000	Retained	Retained
	SSIIRDMADR5	H'0000 0000	Retained	Retained
	SSIIRDMCNTR5	H'0000 0000	Retained	Retained
	SSIWDMADR5	H'0000 0000	Retained	Retained
	SSIWDMCNTR5	H'0000 0000	Retained	Retained
	SSIDMCOR5	H'0000 0000	Retained	Retained
	SSISTPBLCNT5	H'0000 0000	Retained	Retained
	SSISTPDR5	H'0000 0000	Retained	Retained
	SSIBLCNTR5	H'0000 0000	Retained	Retained
	SSIBLCNT5	H'0000 0000	Retained	Retained
	SSIBLNCNTR5	H'0000 0000	Retained	Retained
	SSIBLNCNT5	H'0000 0000	Retained	Retained
	SSIDMAOR1	H'0000 0000	Retained	Retained
	SSIDMINTSR1	H'0101 0101	Retained	Retained
	SSIDMINTMR1	H'1F1F 1F1F	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
SSI_CH0 to 5	SSICR0	H'0000 0000	Retained	Retained
	SSISR0	H'0210 A003	Retained	Retained
	SSITDR0	H'0000 0000	Retained	Retained
	SSIRDR0	H'0000 0000	Retained	Retained
	SSICR1	H'0000 0000	Retained	Retained
	SISR1	H'0210 A003	Retained	Retained
	SSITDR1	H'0000 0000	Retained	Retained
	SSIRDR1	H'0000 0000	Retained	Retained
	SSICR2	H'0000 0000	Retained	Retained
	SSISR2	H'0210 A003	Retained	Retained
	SSITDR2	H'0000 0000	Retained	Retained
	SSIRDR2	H'0000 0000	Retained	Retained
	SSICR3	H'0000 0000	Retained	Retained
	SSISR3	H'0210 A003	Retained	Retained
	SSITDR3	H'0000 0000	Retained	Retained
	SSIRDR3	H'0000 0000	Retained	Retained
	SSICR4	H'0000 0000	Retained	Retained
	SSISR4	H'0210 A003	Retained	Retained
	SITDR4	H'0000 0000	Retained	Retained
	SSIRDR4	H'0000 0000	Retained	Retained
	SSICR5	H'0000 0000	Retained	Retained
	SSISR5	H'0210 A003	Retained	Retained
	SSITDR5	H'0000 0000	Retained	Retained
	SSIRDR5	H'0000 0000	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
EtherC	ECMR	H'0000 0000	Retained	Retained
	ECSR	H'0000 0000	Retained	Retained
	ECSIPR	H'0000 0000	Retained	Retained
	RFLR	H'0000 0000	Retained	Retained
	PIR	H'0000 000x	Retained	Retained
	MAHR	H'0000 0000	Retained	Retained
	MALR	H'0000 0000	Retained	Retained
	PSR	H'0000 000x	Retained	Retained
	TROCR	H'0000 000x	Retained	Retained
	CDCR	H'0000 0000	Retained	Retained
	LCCR	H'0000 0000	Retained	Retained
	CNDCR	H'0000 0000	Retained	Retained
	CEFCR	H'0000 0000	Retained	Retained
	FRECR	H'0000 0000	Retained	Retained
	TSFRCR	H'0000 0000	Retained	Retained
	TLFRCR	H'0000 0000	Retained	Retained
	RFCR	H'0000 0000	Retained	Retained
	MAFCR	H'0000 0000	Retained	Retained
	IPGR	H'0000 0000	Retained	Retained
	APR	H'0000 0000	Retained	Retained
	MPR	H'0000 0000	Retained	Retained
	TPAUSER	H'0000 0000	Retained	Retained
	BCFRR	H'0000 0000	Retained	Retained

<b>Module</b>	<b>Register Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
EDMAC	EDMR	H'0000 0000	Retained	Retained
	EDTRR	H'0000 0000	Retained	Retained
	EDRRR	H'0000 0000	Retained	Retained
	TDLAR	H'0000 0000	Retained	Retained
	RDLAR	H'0000 0000	Retained	Retained
	EESR	H'0000 0000	Retained	Retained
	EESIPR	H'0000 0000	Retained	Retained
	TRSCER	H'0000 0000	Retained	Retained
	RMFCR	H'0000 0000	Retained	Retained
	TFTR	H'0000 0000	Retained	Retained
	FDR	H'0000 0707	Retained	Retained
	RMCR	H'0000 0000	Retained	Retained
	TFUCR	H'0000 0000	Retained	Retained
	RFOCR	H'0000 0000	Retained	Retained
	RBWAR	H'0000 0000	Retained	Retained
	RDFAR	H'0000 0000	Retained	Retained
	TBRAR	H'0000 0000	Retained	Retained
	TDFAR	H'0000 0000	Retained	Retained
	FCFTR	H'0007 0007	Retained	Retained
	RPADIR	H'0000 0000	Retained	Retained
	TRIMD	H'0000 0000	Retained	Retained

---

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
USB	SYSCFG	H'0000	Retained	Retained
	BUSWAIT	H'000F	Retained	Retained
	SYSSTS	H'040x	Retained	Retained
	DVSTCTR	H'0000	Retained	Retained
	TESTMODE	H'0000	Retained	Retained
	D0FBCFG	H'0000	Retained	Retained
	D1FBCFG	H'0000	Retained	Retained
	CFIFO	H'0000 0000	Retained	Retained
	D0FIFO	H'0000 0000	Retained	Retained
	D1FIFO	H'0000 0000	Retained	Retained
	CFIFOSEL	H'0000	Retained	Retained
	CFIFOCTR	H'0000	Retained	Retained
	D0FIFOSEL	H'0000	Retained	Retained
	D0FIFOCTR	H'0000	Retained	Retained
	D1FIFOSEL	H'0000	Retained	Retained
	D1FIFOCTR	H'0000	Retained	Retained
	INTENB0	H'0000	Retained	Retained
	INTENB1	H'0000	Retained	Retained
	BRDYENB	H'0000	Retained	Retained
	NRDYENB	H'0000	Retained	Retained
	BEMPENB	H'0000	Retained	Retained
	SOFCFG	H'0000	Retained	Retained

<b>Module</b>	<b>Register Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
USB	INTSTS0	H'0000	Retained	Retained
	INTSTS1	H'0000	Retained	Retained
	BRDYSTS	H'0000	Retained	Retained
	NRDYSTS	H'0000	Retained	Retained
	BEMPSTS	H'0000	Retained	Retained
	FRMNUM	H'0000	Retained	Retained
	UFRMNUM	H'0000	Retained	Retained
	USBADDR	H'0000	Retained	Retained
	USBREQ	H'0000	Retained	Retained
	USBVAL	H'0000	Retained	Retained
	USBINDX	H'0000	Retained	Retained
	USBLENG	H'0000	Retained	Retained
	DCPCFG	H'0000	Retained	Retained
	DCPMAXP	H'0040	Retained	Retained
	DCPCTR	H'0040	Retained	Retained
	PIPESEL	H'0000	Retained	Retained
	PIPECFG	H'0000	Retained	Retained
	PIPEBUF	H'0000	Retained	Retained
	PIPEMAXP	H'0000	Retained	Retained
	PIPEPERI	H'0000	Retained	Retained
	PIPE1CTR	H'0000	Retained	Retained
	PIPE2CTR	H'0000	Retained	Retained
	PIPE3CTR	H'0000	Retained	Retained
	PIPE4CTR	H'0000	Retained	Retained
	PIPE5CTR	H'0000	Retained	Retained
	PIPE6CTR	H'0000	Retained	Retained
	PIPE7CTR	H'0000	Retained	Retained
	PIPE8CTR	H'0000	Retained	Retained
	PIPE9CTR	H'0000	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
USB	PIPE1TRE	H'0000	Retained	Retained
	PIPE1TRN	H'0000	Retained	Retained
	PIPE2TRE	H'0000	Retained	Retained
	PIPE2TRN	H'0000	Retained	Retained
	PIPE3TRE	H'0000	Retained	Retained
	PIPE3TRN	H'0000	Retained	Retained
	PIPE4TRE	H'0000	Retained	Retained
	PIPE4TRN	H'0000	Retained	Retained
	PIPE5TRE	H'0000	Retained	Retained
	PIPE5TRN	H'0000	Retained	Retained
	DEVADD0	H'0000	Retained	Retained
	DEVADD1	H'0000	Retained	Retained
	DEVADD2	H'0000	Retained	Retained
	DEVADD3	H'0000	Retained	Retained
	DEVADD4	H'0000	Retained	Retained
	DEVADD5	H'0000	Retained	Retained
	DEVADD6	H'0000	Retained	Retained
	DEVADD7	H'0000	Retained	Retained
	DEVADD8	H'0000	Retained	Retained
	DEVADD9	H'0000	Retained	Retained
	DEVADDA	H'0000	Retained	Retained

<b>Module</b>	<b>Register Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
LCDC	LDPR00 to LDPRFF	Undefined	Retained	Retained
	LDICKR	H'1101	Retained	Retained
	LDMTR	H'0109	Retained	Retained
	LDDFR	H'000C	Retained	Retained
	LDSARU	H'04000000	Retained	Retained
	LDSARL	H'04000000	Retained	Retained
	LDLAOR	H'0280	Retained	Retained
	LDPALCR	H'0000	Retained	Retained
	LDHCNR	H'4F52	Retained	Retained
	LDHSYNR	H'0050	Retained	Retained
	LDVDLNR	H'01DF	Retained	Retained
	LDVTLNR	H'01DF	Retained	Retained
	LDVSYNR	H'01DF	Retained	Retained
	LDACLNR	H'000C	Retained	Retained
	LDINTR	H'0000	Retained	Retained
	LDPMMR	H'0010	Retained	Retained
	LDPSPR	H'F60F	Retained	Retained
	LDCNTR	H'0000	Retained	Retained
	LDUINTR	H'0000	Retained	Retained
	LDUINTLNR	H'004F	Retained	Retained
	LDLIRNR	H'0000	Retained	Retained



Module	Register Abbreviation	Power-on Reset	Sleep	Standby
G2D	SCLR	H'8000 0000	Retained	Retained
	SR	Undefined	Retained	Retained
	SRCR	H'0000 0000	Retained	Retained
	IER	H'0000 0000	Retained	Retained
	ICIDR	Undefined	Retained	Retained
	RTN0R	Undefined	Retained	Retained
	RTN1R	Undefined	Retained	Retained
	DLSAR	Undefined	Retained	Retained
	SSAR	Undefined	Retained	Retained
	RSAR	Undefined	Retained	Retained
	WSAR	Undefined	Retained	Retained
	SSTRR	Undefined	Retained	Retained
	DSTRR	Undefined	Retained	Retained
	ENDCVR	H'0000 0000	Retained	Retained
	STCR	Undefined	Retained	Retained
	DTCR	Undefined	Retained	Retained
	ALPHR	Undefined	Retained	Retained
	COFSR	Undefined	Retained	Retained
	RCLR	H'0000 0000	Retained	Retained
	CSTR	Undefined	Retained	Retained
	CURR	Undefined	Retained	Retained
	LCOR	Undefined	Retained	Retained
	SCLMAR	Undefined	Retained	Retained

<b>Module</b>	<b>Register Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
G2D	UCLMIR	Undefined	Retained	Retained
	UCLMAR	Undefined	Retained	Retained
	RUCLMIR	Undefined	Retained	Retained
	RUCLMAR	Undefined	Retained	Retained
	RCL2R	H'0000 4004	Retained	Retained
	POFSR	H'0000 0000	Retained	Retained
	GTRCR	H'0000 0000	Retained	Retained
	MTRAR	Undefined	Retained	Retained
	MTRBR	Undefined	Retained	Retained
	MTRCR	Undefined	Retained	Retained
	MTRDR	Undefined	Retained	Retained
	MTRER	Undefined	Retained	Retained
	MTRFR	Undefined	Retained	Retained
	MTRGR	Undefined	Retained	Retained
	MTRHR	Undefined	Retained	Retained
	MTRIR	Undefined	Retained	Retained
	GTROFSXR	Undefined	Retained	Retained
	GTROFSYR	Undefined	Retained	Retained
	ZCLPMINR	Undefined	Retained	Retained
	ZCLPMAXR	Undefined	Retained	Retained
	ZSATVMINR	Undefined	Retained	Retained

<b>Module</b>	<b>Register Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
VDC2 graphics block 1	GRCMEN1	H'0000 0000	Retained	Retained
	GRCBUSCNT1	H'0000 0000	Retained	Retained
	GROPSADR1	H'0000 0000	Retained	Retained
	GROPSWH1	H'0000 0000	Retained	Retained
	GROPSOFST1	H'0000 0000	Retained	Retained
	GROPDPHV1	H'0000 0000	Retained	Retained
	GROPBASERGB1	H'0000 0000	Retained	Retained
VDC2 graphics block 2	GRCMEN2	H'0000 0000	Retained	Retained
	GRCBUSCNT2	H'0000 0000	Retained	Retained
	GROPSADR2	H'0000 0000	Retained	Retained
	GROPSWH2	H'0000 0000	Retained	Retained
	GROPSOFST2	H'0000 0000	Retained	Retained
	GROPDPHV2	H'0000 0000	Retained	Retained
	GROPEWH2	H'0000 0000	Retained	Retained
	GROPEDPHV2	H'0000 0000	Retained	Retained
	GROPEDPA2	H'0000 0000	Retained	Retained
	GROPCRKY0_2	H'0000 0000	Retained	Retained
	GROPCRKY1_2	H'0000 0000	Retained	Retained
	GROPBASERGB2	H'0000 0000	Retained	Retained

<b>Module</b>	<b>Register Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
VDC2 graphics block 3	GRCMEN3	H'0000 0000	Retained	Retained
	GRCBUSCNT3	H'0000 0000	Retained	Retained
	GROPSADR3	H'0000 0000	Retained	Retained
	GROPSWH3	H'0000 0000	Retained	Retained
	GROPSOFST3	H'0000 0000	Retained	Retained
	GROPDPHV3	H'0000 0000	Retained	Retained
	GROPEWH3	H'0000 0000	Retained	Retained
	GROPEDPHV3	H'0000 0000	Retained	Retained
	GROPEDPA3	H'0000 0000	Retained	Retained
	GROPCRKY0_3	H'0000 0000	Retained	Retained
	GROPCRKY1_3	H'0000 0000	Retained	Retained
	GROPBASERGB3	H'0000 0000	Retained	Retained
VDC2 graphics block 4	GRCMEN4	H'0000 0000	Retained	Retained
	GRCBUSCNT4	H'0000 0000	Retained	Retained
	GROPSADR4	H'0000 0000	Retained	Retained
	GROPSWH4	H'0000 0000	Retained	Retained
	GROPSOFST4	H'0000 0000	Retained	Retained
	GROPDPHV4	H'0000 0000	Retained	Retained
	GROPEWH4	H'0000 0000	Retained	Retained
	GROPEDPHV4	H'0000 0000	Retained	Retained
	GROPEDPA4	H'0000 0000	Retained	Retained
	GROPCRKY0_4	H'0000 0000	Retained	Retained
	GROPCRKY1_4	H'0000 0000	Retained	Retained
	GROPBASERGB4	H'0000 0000	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
VDC2 display control block	SGMODE	H'0000 0000	Retained	Retained
	SGINTCNT	H'0000 0000	Retained	Retained
	SYNCNT	H'0000 0000	Retained	Retained
	EXTSYNCNT	H'0000 0000	Retained	Retained
	SYNSIZE	H'020D 035A	Retained	Retained
	VSYNCTIM	H'0000 0001	Retained	Retained
	HSYNCTIM	H'0000 000A	Retained	Retained
	CLSTIM	H'0000 0000	Retained	Retained
	SPLTIM	H'0000 0000	Retained	Retained
	COMTIM	H'0000 0000	Retained	Retained
	SGDESTART	H'0000 0000	Retained	Retained
	SGDESIZE	H'0000 0000	Retained	Retained
	CDECRKY	H'0000 0000	Retained	Retained
	T1004CNT	H'0000 0000	Retained	Retained
	T1004OFFSET	H'0000 0000	Retained	Retained
FLCTL	FLCMNCR	Initialized	Retained	Retained
	FLCMDRCR	Initialized	Retained	Retained
	FLCMCDR	Initialized	Retained	Retained
	FLADR	Initialized	Retained	Retained
	FLADR2	Initialized	Retained	Retained
	FLDATAR	Initialized	Retained	Retained
	FLDTCNTR	Initialized	Retained	Retained
	FLINTDMACR	Initialized	Retained	Retained
	FLBSYTMR	Initialized	Retained	Retained
	FLBSYCNT	Initialized	Retained	Retained
	FLDTFIFO	Initialized	Retained	Retained
	FLECFIFO	Initialized	Retained	Retained
	FLTRCR	Initialized	Retained	Retained

<b>Module</b>	<b>Register Abbreviation</b>	<b>Power-on Reset</b>	<b>Sleep</b>	<b>Standby</b>
SRC	SRCID	H'0000 0000	Retained	Retained
	SRCOD	H'0000 0000	Retained	Retained
	SRCIDCTRL	H'0000	Retained	Retained
	SRCODCTRL	H'0000	Retained	Retained
	SRCCTRL	H'0000	Retained	Retained
	SRCSTAT	H'0002	Retained	Retained
GPIO	PTIO_A	H'0000	Retained	Retained
	PTIO_B	H'0000	Retained	Retained
	PTIO_C	H'0000	Retained	Retained
	PTIO_D	H'0000	Retained	Retained
	PTIO_E	H'0000	Retained	Retained
	PTIO_F	H'0000	Retained	Retained
	PTIO_G	H'0000	Retained	Retained
	PTIO_H	H'0000	Retained	Retained
	PTIO_I	H'0002	Retained	Retained
	PTIO_J	H'AAAA	Retained	Retained
	PTDAT_A	H'0000	Retained	Retained
	PTDAT_B	H'0000	Retained	Retained
	PTDAT_C	H'0000	Retained	Retained
	PTDAT_D	H'0000	Retained	Retained
	PTDAT_E	H'0000	Retained	Retained
	PTDAT_F	H'0000	Retained	Retained
	PTDAT_G	H'0000	Retained	Retained
	PTDAT_H	H'0000	Retained	Retained
	PTDAT_I	H'0000	Retained	Retained
	PTDAT_J	H'0000	Retained	Retained
	PTPUL_SPCL	H'0000	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
GPIO	PTSEL_A	H'0000	Retained	Retained
	PTSEL_B	H'0000	Retained	Retained
	PTSEL_C	H'0000	Retained	Retained
	PTSEL_D	H'0000	Retained	Retained
	PTSEL_E	H'0000	Retained	Retained
	PTSEL_F	H'0000	Retained	Retained
	PTSEL_G	H'0000	Retained	Retained
	PTSEL_H	H'0000	Retained	Retained
	PTSEL_I	H'0000	Retained	Retained
	PTSEL_J	H'0000	Retained	Retained
	PTSEL_K	H'0000	Retained	Retained
	PTSEL_P	H'0000	Retained	Retained
	PTSEL_R	H'0000	Retained	Retained
	PTSEL_S	H'0000	Retained	Retained
	PTHIZ_A	H'0000	Retained	Retained
	PTHIZ_B	H'0000	Retained	Retained
	PTSEL_SPCL	H'0000	Retained	Retained
Power-down mode* <sup>2</sup>	STBCR	H'0000 0000	Retained	Retained
	MSTPCR0	H'0000 0000	Retained	Retained
	MSTPCR1	H'0000 0000	Retained	Retained
UBC	CBR0	H'2000 0000	Retained	Retained
	CRR0	H'0000 2000	Retained	Retained
	CAR0	Undefined	Retained	Retained
	CAMR0	Undefined	Retained	Retained
	CBR1	H'2000 0000	Retained	Retained
	CRR1	H'0000 2000	Retained	Retained
	CAR1	Undefined	Retained	Retained
	CAMR1	Undefined	Retained	Retained
	CDR1	Undefined	Retained	Retained

Module	Register Abbreviation	Power-on Reset	Sleep	Standby
UBC	CDMR1	Undefined	Retained	Retained
	CETR1	Undefined	Retained	Retained
	CCMFR	H'0000 0000	Retained	Retained
	CBCR	H'0000 0000	Retained	Retained
H-UDI	SDIR	H'0EFF	Retained	Retained
	SDINT	H'0000	Retained	Retained

Legend:

O: Initialized

—: Retained

Notes: 1. The initial value in the clock-operating mode that is selected according to the MODE0, MODE1, and MODE2 settings.

2. For the standby control register, also refer to figure 9.1, Block Diagram of CPG.

Module	Register Abbreviation	Power-on Reset		Sleep	Standby
		By RESET pin	By WDT/H-UDI		
WDT	WDTST	H'0000 0000	Retained	Retained	Retained
	WDTCSR	H'0000 0000	Retained	Retained	Retained
	WDTBST	H'0000 0000	Retained	Retained	Retained
	WDTCNT	H'0000 0000	Retained	Retained	Retained
	WDTBCNT	H'0000 0000	Retained	Retained	Retained



## Section 33 Electrical Characteristics

### 33.1 Absolute Maximum Ratings\*1\*2

**Table 33.1 Absolute Maximum Ratings**

Item	Symbol	Value	Unit
Power supply voltage (I/O)	$V_{DDQ}, V_{DDQ\_USB}$	−0.3 to 4.6	V
Power supply voltage (internal)	$V_{DD}, V_{DD\_PLL1}, V_{DD\_PLL2}, V_{DD\_USB}, UV12$	−0.3 to 1.7	V
Power supply voltage (analog 3.3-V)	$V_{DDQA\_USB}$	−0.3 to 4.6	V
Power supply voltage (analog 1.2-V)	$V_{DDA\_USB}$	−0.3 to 1.7	V
Input voltage	$V_{in}$	−0.3 to $V_{DDQ} + 0.3^{*3}$	V
Vbus input voltage	$V_{bus}$	−0.3 to 5.5	V
Operating temperature	$T_{opr}$	−20 to 85 (regular specifications) −40 to 85 (wide temperature specifications)	°C
Storage temperature	$T_{stg}$	−55 to 125	°C

- Caution:
- 1 Permanent damage to the LSI may result if absolute maximum ratings are exceeded.
  - 2 Permanent damage to the LSI may result if all the VSS are not connected to GND.
  - 3 Do not exceed the maximum power supply voltage.

33.2
 Power-on/Power-off Sequence

The sequences for turning on and off power supplies are shown below together with recommended values.

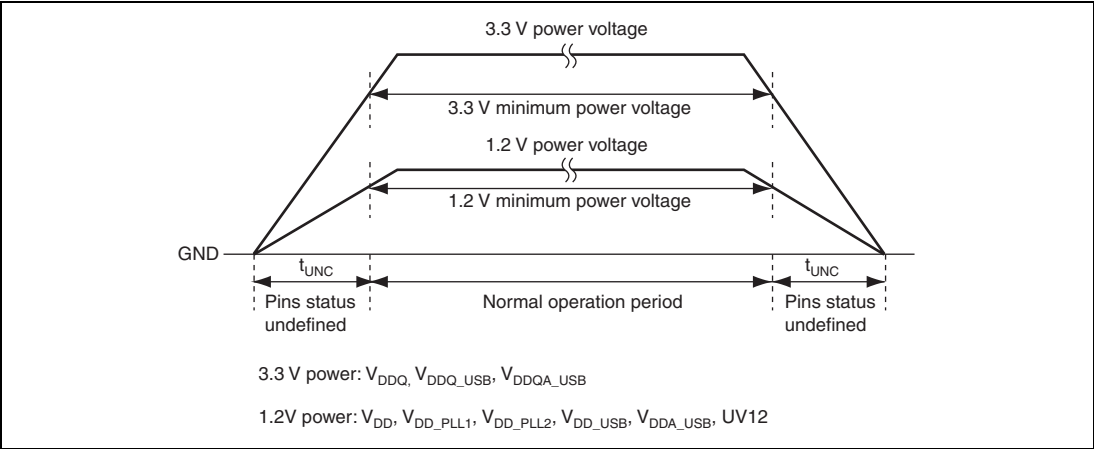


Figure 33.1
 Power-on/Power-off Sequence

Table 33.2
 Recommended Time for Power-on/Power-off Sequence

Item	Symbol	Max.	Unit
State undefined time	$t_{UNC}$	100	ms

Note: The table shown above is the maximum values, so they represent guidelines rather than strict requirements.

Either the 3.3-V power supply or the 1.2-V power does not matter to be turned on or turned off. An undefined time appears until either of the power supply, which turns on later, reaches above the minimum voltage and after it has reached below the minimum voltage. During these periods, pin and internal states become undefined. Design the system so that these undefined states do not cause an overall malfunction.

### 33.3 DC Characteristics

**Table 33.3 DC Characteristics [Common Items]**

Conditions:  $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage (I/O)		$V_{DDQ}$ $V_{DDQ\_USB}$	3.0	3.3	3.6	V	
Power supply voltage (Internal)		$V_{DD}$ , $V_{DD\_PLL1}$ , $V_{DD\_PLL2}$ , $V_{DD\_USB}$ UV12	1.15	1.25	1.35	V	
Power supply voltage (analog 3.3-V)		$V_{DDQA\_USB}$	3.0	3.3	3.6	V	
Power supply voltage (analog 1.2-V)		$V_{DDA\_USB}$	1.15	1.25	1.35	V	
Supply current	Normal operation	$I_{DDQ} + I_{DDQ\_USB}$	—	—	150	mA	Lck= 324 MHz SHck= Bck=108 MHz Pck= 54 MHz
		$I_{DD} + I_{DD\_PLL1} +$ $I_{DD\_PLL2} +$ $I_{DD\_USB} + \text{UV12}$	—	—	850	mA	
	Sleep mode	$I_{DDQ} + I_{DDQ\_USB}$	—	—	150	mA	
		$I_{DD} + I_{DD\_PLL1} +$ $I_{DD\_PLL2} +$ $I_{DD\_USB} + \text{UV12}$	—	—	650	mA	
	Refresh standby	$I_{DDQ} + I_{DDQ\_USB}$	—	—	15	mA	
		$I_{DD} + I_{DD\_PLL1} +$ $I_{DD\_PLL2} +$ $I_{DD\_USB} + \text{UV12}$	—	—	450	mA	
Supply current (USB)t	Normal operation	$I_{DDQA\_USB}$	—	—	15	mA	
		$I_{DDA\_USB}$	—	—	20	mA	
	Refresh standby	$I_{DDQA\_USB}$	—	—	600	μA	
		$I_{DDA\_USB}$	—	—	600	μA	

**Table 33.4 DC Characteristics [Excluding the Pins Related to USB Transceiver and I<sup>2</sup>C]**Conditions:  $T_a = -20$  to  $85^{\circ}\text{C}$ ,  $-40$  to  $85^{\circ}\text{C}$ 

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input voltage	$V_{IH}$	$V_{CCQ}$ $\times$ 0.9	—	$V_{CCQ} +$ 0.3	V	$V_{CCQ} = 3.0$ to 3.6 V
PRESET, TRST, NMI, MODE8/FD7, MODE7/FD6, MODE5/FD5, MODE4/FD4, MODE3/FD3, MODE2/FD2, MODE1/FD1, MODE0/FD0, IRQ0/DTEND1, WDTOVF/IRQ1/AUDCK/DACK1, D44, IDEINT, MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1, STATUS1/RTS2/PA7 STATUS0/CTS2/PA6, FCE/PA5, FRE/PA4, FEW/PA3, TXD2/PA2, RXD2/PA1, SCK2/PA0, A25/PB7/DREQ0/TRS0, A24/PB6/DACK0/CTS0, A23/PB5/DTEND0/RTS1, A22/PB4/CTS1, A21/PB3, A20/PB2, A19/PB1, A18/PB0, ASEBRKAK/BRKACK/TCLK/PC1, EXTAL, XIN		2.0	—	$V_{CCQ} +$ 0.3		

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input voltage	PRESET, TRST, NMI, MODE8/FD7, MODE7/FD6, MODE5/FD5, MODE4/FD4, MODE3/FD3, MODE2/FD2, MODE1/FD1, MODE0/FD0, IRQ0/DTEND1, WDTOVF/IRQ1/AUDCK/DACK1, D44,IDEINT, MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1, STATUS1/RTS2/PA7 STATUS0/CTS2/PA6, FCE/PA5, FRE/PA4, FEW/PA3, TXD2/PA2, RXD2/PA1, SCK2/PA0, A25/PB7/DREQ0/TRS0, A24/PB6/DACK0/CTS0, A23/PB5/DTEND0/RTS1, A22/PB4/CTS1, A21/PB3, A20/PB2, A19/PB1, A18/PB0, ASEBRKAK/BRKACK/TCLK/PC1, ESTAL, XIN	$V_{IL}$	-0.3	—	$V_{CCQ}$ $\times 0.1$	V	$V_{CCQ} = 3.0$ to 3.6 V
	Input pins other than above	$V_{IL}$	-0.3	—	$V_{CCQ}$ $\times 0.2$	V	
Input leakage current	All input pins	$ I_{in} $	—	—	1.0	$\mu A$	$V_{in} = 0.5$ to $V_{CCQ} - 0.5$ V
Three-state leakage current	All input/output pins, output pins (off status)	$ I_{sti} $	—	—	1.0		
Output voltage	All output pins	$V_{OH}$	2.4	—	—	V	$V_{CCQ} = 3.0$ V $I_{OH} = 2$ mA
	All output pins	$V_{OL}$	—	—	0.55	V	$V_{DDQ} = 3.0$ V $I_{OL} = 2$ mA
Pull-up resistance	All pins	$R_{pull}$	20	60	180	k $\Omega$	
Pin capacitance	Others	$C_L$	—	—	10	pF	

Note: Supply current values are the values measured when all of the output pins and pins are unloaded in the conditions;  $HV_H$  (minimum) =  $V_{CCQ} - 0.5$  or  $V_{IL}$  (maximum) = 0.5 V.

**Table 33.5 DC Characteristics [Pins Related to I<sup>2</sup>C\*]**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage	$V_{CCQ}$	3.0	3.3	3.6	V	
Input high voltage	$V_{IH}$	$V_{CCQ} \times 0.7$	—	$V_{CCQ} + 0.3$	V	
Input low voltage	$V_{IL}$	-0.3	—	$V_{CCQ} \times 0.3$	V	
Output low voltage	$V_{OL}$	—	—	0.4	V	$I_{OL} = 3.0 \text{ mA}$
Output low permissible current	$I_{OL}$	—	—	10	mA	

Note: \* Pins related to I<sup>2</sup>C: SCL, SDA (open-drain pins)

**Table 33.6 DC Characteristics [Pins Related to USB (1)]**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reference resistance	$R_{REF}$	$5.6 \Omega \pm 1\%$			V	
Input high voltage (VBUS)	$V_{IH}$	4.02	—	5.25	V	
Input low voltage (VBUS)	$V_{IL}$	0.0	—	1.0	V	
Input high voltage (XIN)	$V_{IH}$	$V_{DDQ} - 0.5$	—	$V_{DDQ} + 0.3$	V	
Input low voltage (XIN)	$V_{IL}$	-0.3	—	0.5	V	

**Table 33.7 DC Characteristics [Pins Related to USB (2)  
(for Full-Speed/High-Speed Common Items)]**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
DP pull-up resistance (when the function is selected)	$R_{pu}$	0.900	—	1.575	k $\Omega$	In idle mode
		1.425	—	3.090	k $\Omega$	In transmit/receive mode
DP/DM pull-down resistance (when the host is selected)	$R_{pd}$	14.25	—	24.80	k $\Omega$	

**Table 33.8 DC Characteristics [Pins Related to USB (3) (for Full Speed)]**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$V_{IH}$	2.0	—	—	V	
Input low voltage	$V_{IL}$	—	—	0.8	V	
Differential input sensitivity	$V_{DI}$	0.2	—	—	V	(DP) – (DM)
Differential common mode range	$V_{CM}$	0.8	—	2.5	V	
Output high voltage	$V_{OH}$	2.8	—	3.6	V	$I_{OH} = 5$ mA
Output low voltage	$V_{OL}$	0.0	—	0.3	V	$I_{OL} = 5$ mA
Single-ended receiver threshold voltage	$V_{SE}$	0.8	—	2.0	V	
Crossover voltage range	$V_{CRS}$	1.3	—	2.0	V	$C_L = 50$ pF

Note: \* Referring to the DP and DM pins

**Table 33.9 DC Characteristics [Pins Related to USB (4) (for High Speed)]**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Differential common mode range	$V_{\text{HSDI}}$	0.15			V	
Squelch-detected threshold voltage (differential voltage)	$V_{\text{HSSQ}}$	100	—	150	mV	
Common mode voltage range	$V_{\text{HSCM}}$	−50	—	500	mV	
Idle state	$V_{\text{HSOI}}$	−10.0	—	10.0	mV	
Output high voltage	$V_{\text{HSOH}}$	360	—	440	mV	
Output low voltage	$V_{\text{HSOL}}$	−10.0	—	10.0	mV	
Chirp J output voltage (differential)	$V_{\text{CHIRPJ}}$	700	—	1000	mV	
Chirp K output voltage (differential)	$V_{\text{CHIRPK}}$	−900	—	−500	mV	

Note: Pins related to USB: DP, DM

**Table 33.10 DC Characteristics [Pins Related to USB (5) (for Low Speed)]**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$V_{\text{LSOH}}$	2.8	—	—	V	$I_{\text{OH}} = 200 \mu\text{A}$
Input low voltage	$V_{\text{LSOL}}$	—	—	0.3	V	$I_{\text{OL}} = 2 \text{ mA}$

Note: Pins related to USB: DP, DM



**Table 33.11 Output Permissible Current Value**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^{\circ}\text{C}$ ,  $-40$  to  $85^{\circ}\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit
Output low-level capacitance current	$I_{OL}$	—	—	2	mA
Output low-level capacitance current (total)	$\Sigma I_{OL}$	—	—	120	
Output high-level capacitance current	$-I_{OL}$	—	—	2	mA
Output high-level capacitance current (total)	$\Sigma   -I_{OH}  $	—	—	40	

Caution: To protect the LSI's reliability, do not exceed the output current values in table 33.11.

## 33.4 AC Characteristics

Signals input to this LSI are basically handled as signals in synchronization with a clock. The setup and hold times for input pins must be followed.

### 33.4.1 Clock and Control Signal Timing

**Table 33.12 Clock and Control Signal Timing**

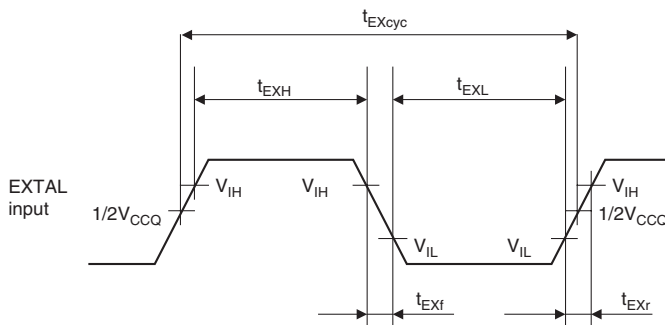
Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^{\circ}\text{C}$ ,  $-40$  to  $85^{\circ}\text{C}$

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency <sup>*1</sup>	$f_{EX}$	24	32.4	MHz	
EXTAL clock input cycle time	$t_{EXcyc}$	30.8	42	ns	33.2
EXTAL clock input low-level pulse width	$t_{EXL}$	7	—	ns	
EXTAL clock input high-level pulse width	$t_{EXH}$	7	—	ns	

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input rise time	$t_{EXr}$	—	4	ns	33.2
EXTAL clock input fall time	$t_{EXf}$	—	4	ns	
CLKOUT clock output* <sup>2</sup>	$t_{op}$	80	108	MHz	
CLKOUT clock output cycle time	$t_{CLKOUTcyc}$	9.26	12.5	ns	33.3
CLKOUT clock output low-level pulse width	$t_{CLKOUTL}$	2	—	ns	
CLKOUT clock output high-level pulse width	$t_{CLKOUTH}$	2	—	ns	
CLKOUT clock output rise time	$t_{CLKOUTr}$	—	3	ns	
CLKOUT clock output fall time	$t_{CLKOUTf}$	—	3	ns	
MDn reset set up time	$t_{MDRS}$	30	—	ms	33.5
MDn reset hold time	$t_{MDRH}$	20	—	ns	
PRESET $\overline{}$ assert time	$T_{RESPW}$	30	—	ms	33.4, 33.5
Power-on oscillation settling time	$T_{OSC}$	60	—	$\mu$ s	33.4
TRST $\overline{}$ reset hold time	$T_{TRSTRH}$	20	—	ns	

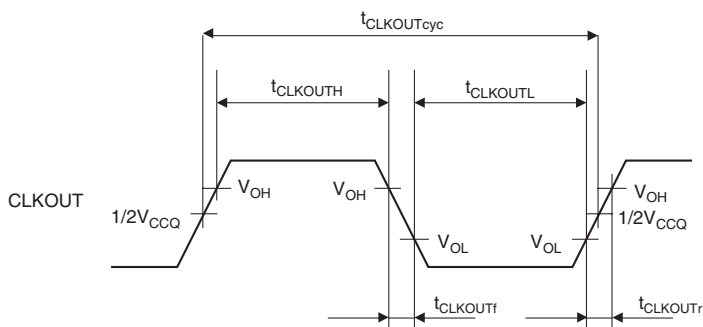
Notes: 1. The maximum frequency indicates 32.4 MHz when the crystal oscillator connects to EXTAL and XTAL. The tank circuit is required as the external circuit when the three-dimensional overtone oscillator.

2. The maximum connection load capacitance to the CLOKOUT pin is 50 pF.

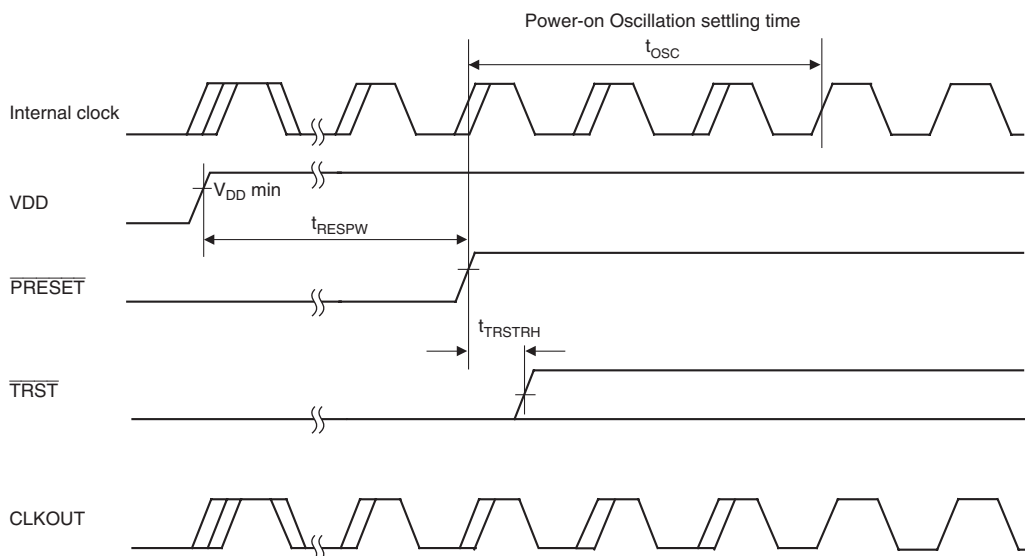


Note: When the clock is input on the EXTAL pin.

**Figure 33.2 EXTAL Clock Input Timing**



**Figure 33.3 CLKOUT Clock Output Timing (1)**



Note: Oscillation settling time when the internal oscillator is used.

**Figure 33.4 Power-On Oscillation Settling Time**

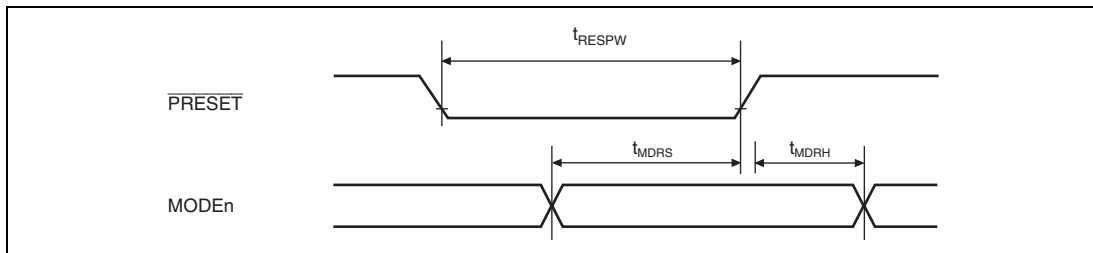


Figure 33.5 MODE Pin Setup / Hold Timing

### 33.4.2 Control Signal Timing

Table 33.13 Control Signal Timing

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Max.	Unit	Figure
WDTOVF delay time	$t_{\text{WOVD}}$	—	$t_{\text{cyc}} + 9$	ns	33.6
STATUS0, STATUS1 delay time	$t_{\text{STD}}$	—	$t_{\text{cyc}} + 10$	ns	
BREQ setup time	$t_{\text{BREQS}}$	3	—	ns	33.7
BREQ hold time	$t_{\text{BREQH}}$	1.5	—	ns	
BACK delay time	$t_{\text{BACKD}}$	—	7	ns	
Bus 3-state delay time	$t_{\text{BOFF}}$	—	10	ns	
Bus buffer on time	$t_{\text{BON}}$	—	10	ns	

Note:  $t_{\text{cyc}}$  is a cycle time of CLKOUT clock.

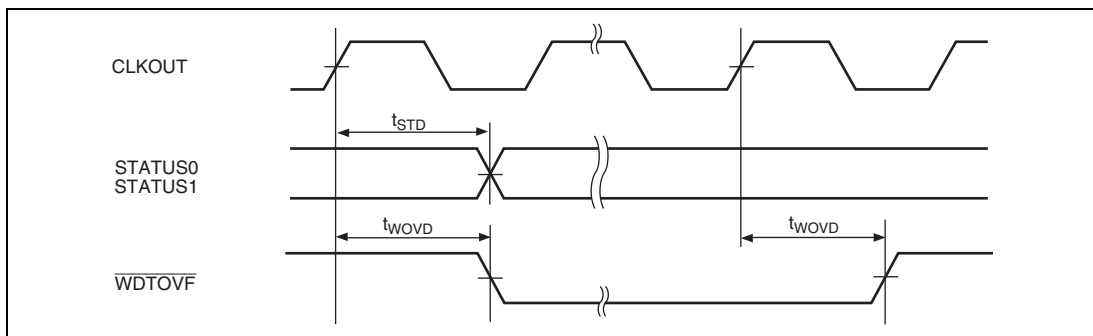


Figure 33.6 Pin Drive Timing in Standby

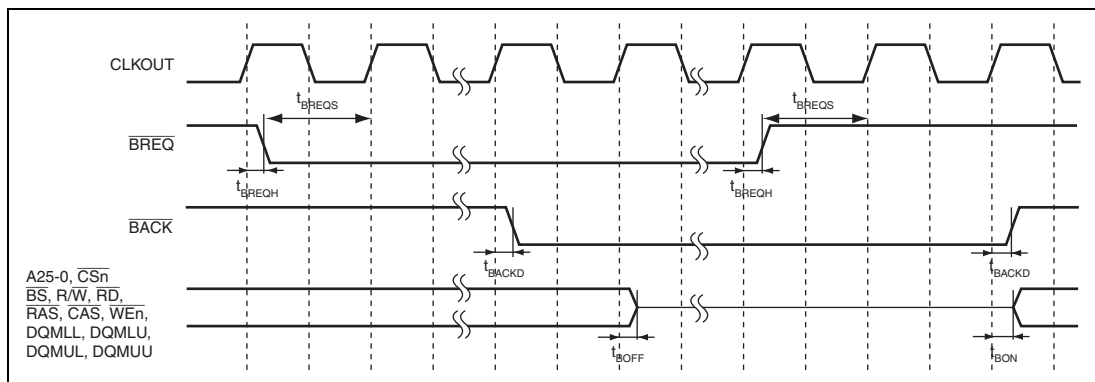


Figure 33.7 Control Signal Timing

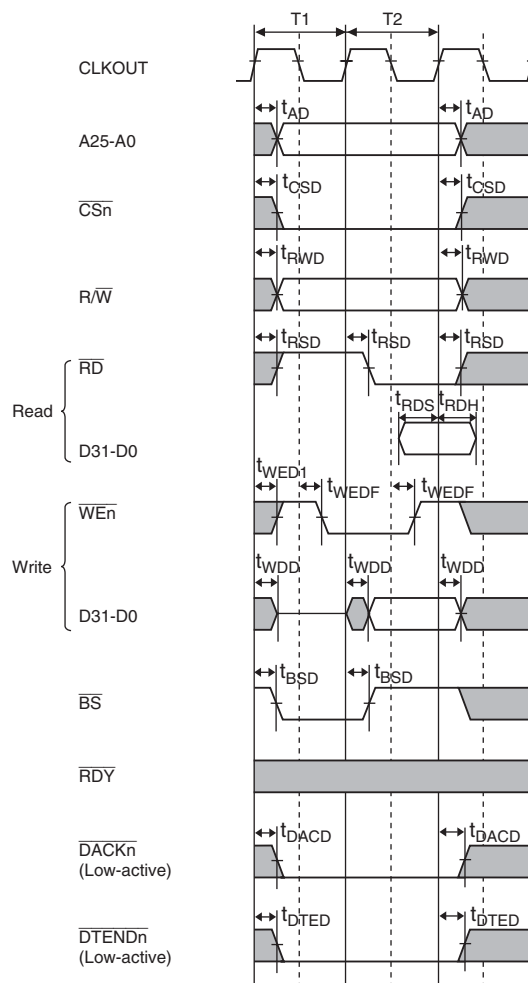
### 33.4.3 Bus Timing

**Table 33.14 Bus Timing**

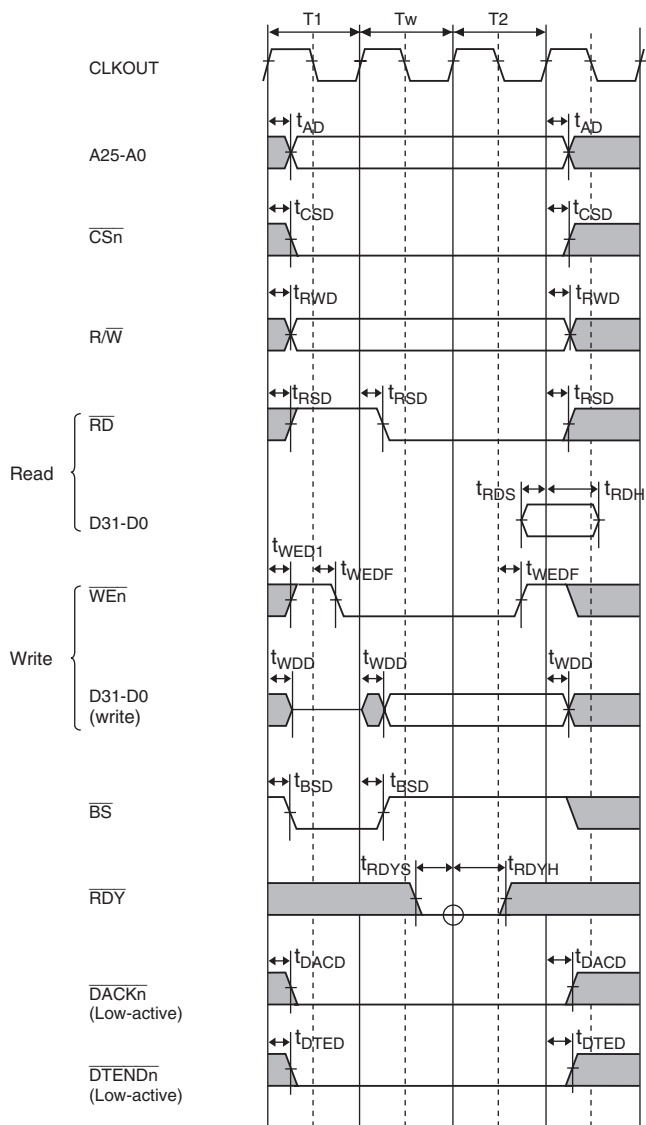
Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Max.	Unit	Figure
Address delay time	$t_{AD}$	1.0	7.0	ns	33.8 to 33.11
$\overline{BS}$ delay time	$t_{BSD}$	1.0	7.0	ns	
$\overline{CSn}$ delay time	$t_{CSD}$	1.0	7.0	ns	33.9, 33.10
R/W delaytime	$t_{RWD}$	1.0	7.0	ns	
$\overline{RD}$ delay time	$t_{RSD}$	1.0	7.0	ns	33.12 to 33.23
Read data setup time	$t_{RDS}$	3.0	—	ns	
Read data hold time	$t_{RDH}$	1.5	—	ns	
$\overline{WEn}$ delay time (falling edge)*	$t_{WEDF}$	—	7.0	ns	
$\overline{WEn}$ delay time	$T_{WED1}$	1.0	7.0	ns	
Write data delay time	$t_{WDD}$	1.0	7.0	ns	
$\overline{RDY}$ setup time	$t_{RDYS}$	3.0	—	ns	
$\overline{RDY}$ hold time	$t_{RDYH}$	1.5	—	ns	
RAS delay time	$t_{RASD}$	1.0	7.0	ns	
$\overline{CAS}$ delay time	$t_{CASD}$	1.0	7.0	ns	
$\overline{CKE}$ delay time	$t_{CKED}$	1.0	7.0	ns	
$\overline{DQM}$ delay time	$t_{DQMD}$	1.0	7.0	ns	

Note: This indicates the delay time for the CLKOUT rising-edge.

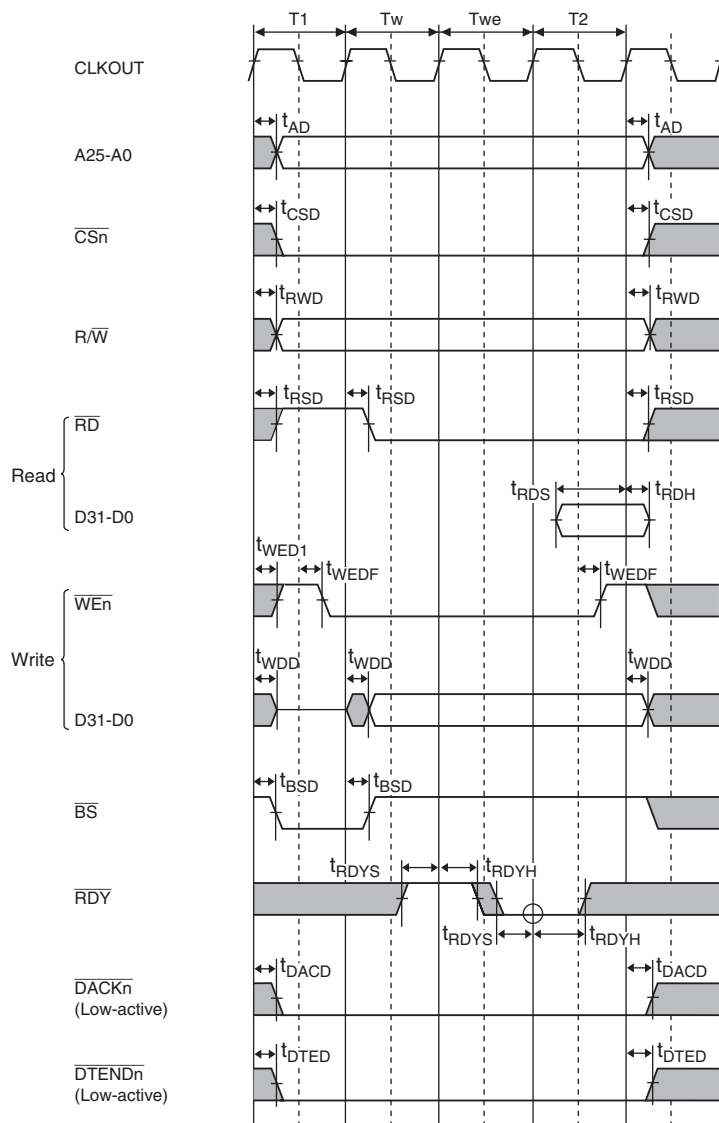


**Figure 33.8 Basic Bus Cycle in SRAM Bus Cycle (No Wait Cycle)**

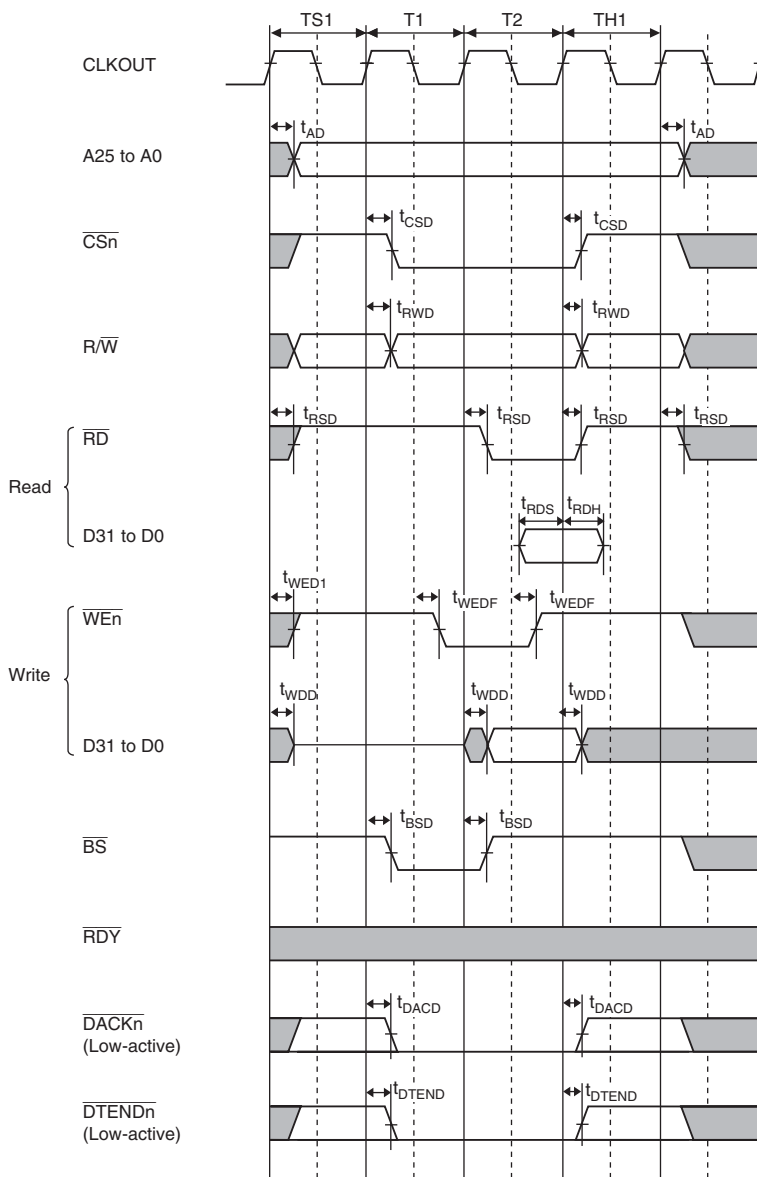


**Figure 33.9 Basic Bus Cycle in SRAM Bus Cycle (One Internal Wait Cycle)**

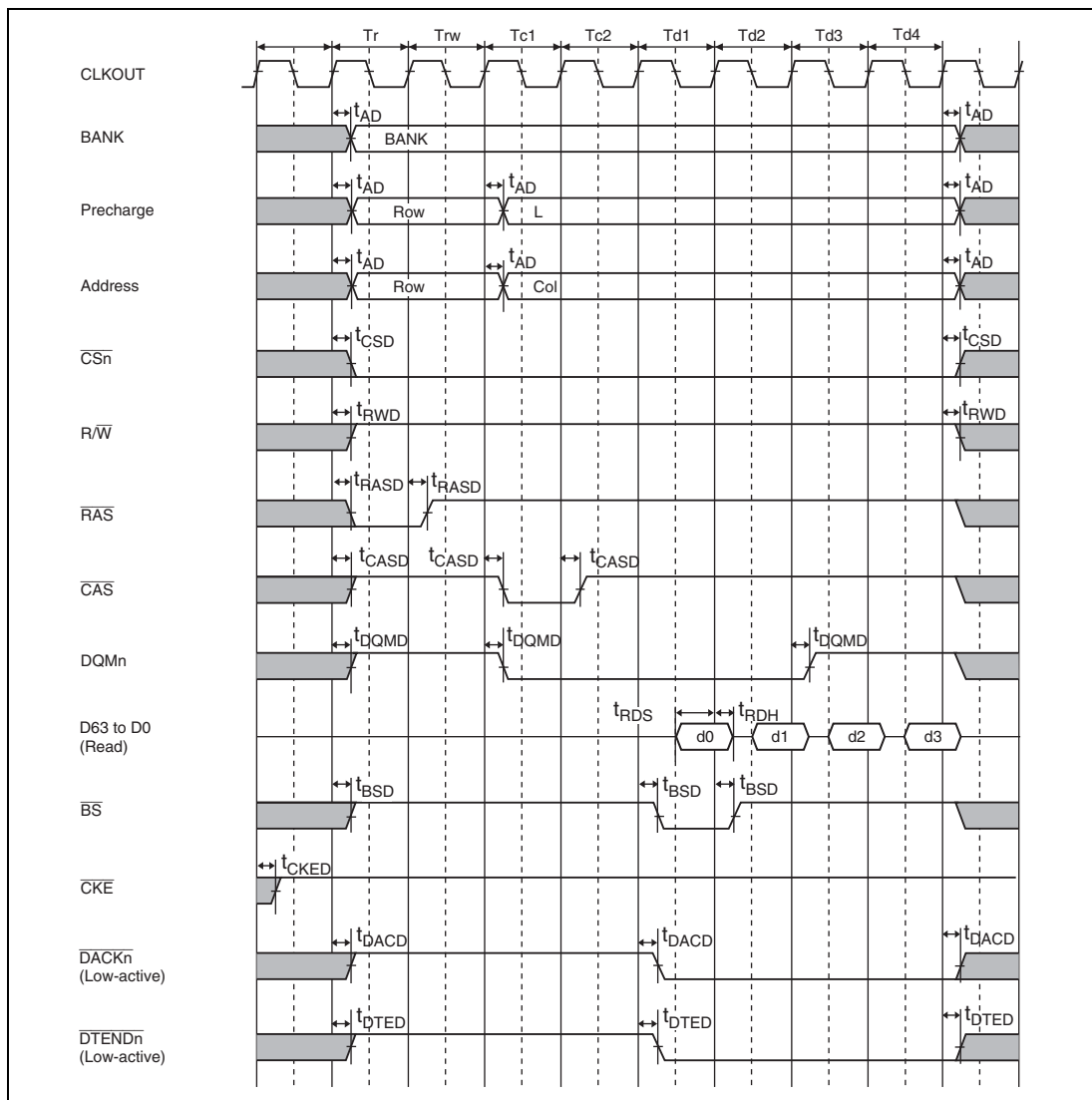




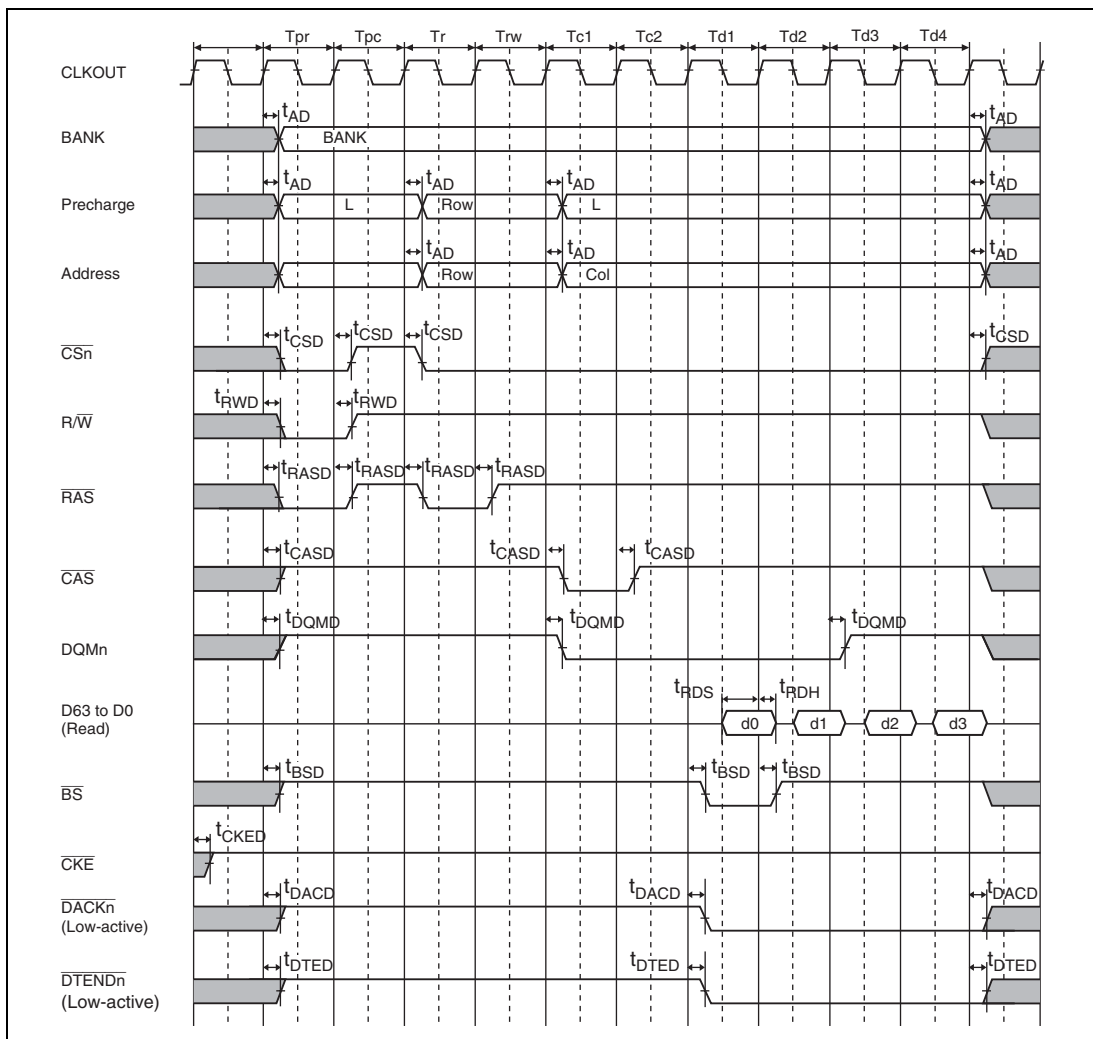
**Figure 33.10 Basic Bus Cycle in SRAM Bus Cycle  
(Internal Wait Cycle + One External Wait Cycle)**



**Figure 33.11 Basic Bus Cycle in SRAM Bus Cycle**  
 (No Wait Cycle, Address Setup / Hold Time Insert, AnS= 1, AnH= 1)

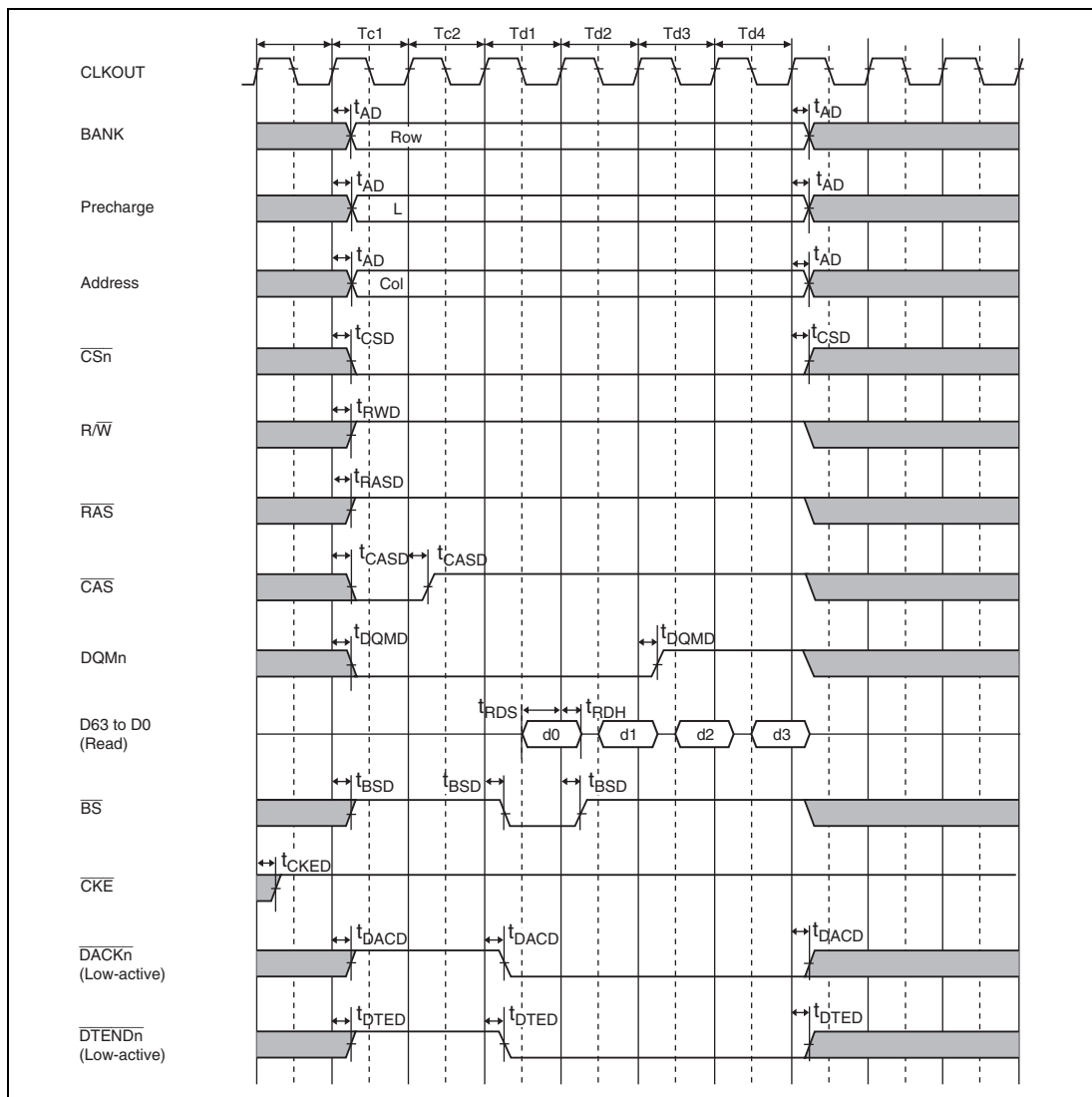


**Figure 33.12 SRAM Bus Cycle in Bank Open Mode Read Bus Cycle (ACT-READ)**  
**(BOMODE[1:0]= 00, SCL[2:0]= 000, SRCD= 0, CAS Latency= 2cyc, IRCD= 2cyc)**

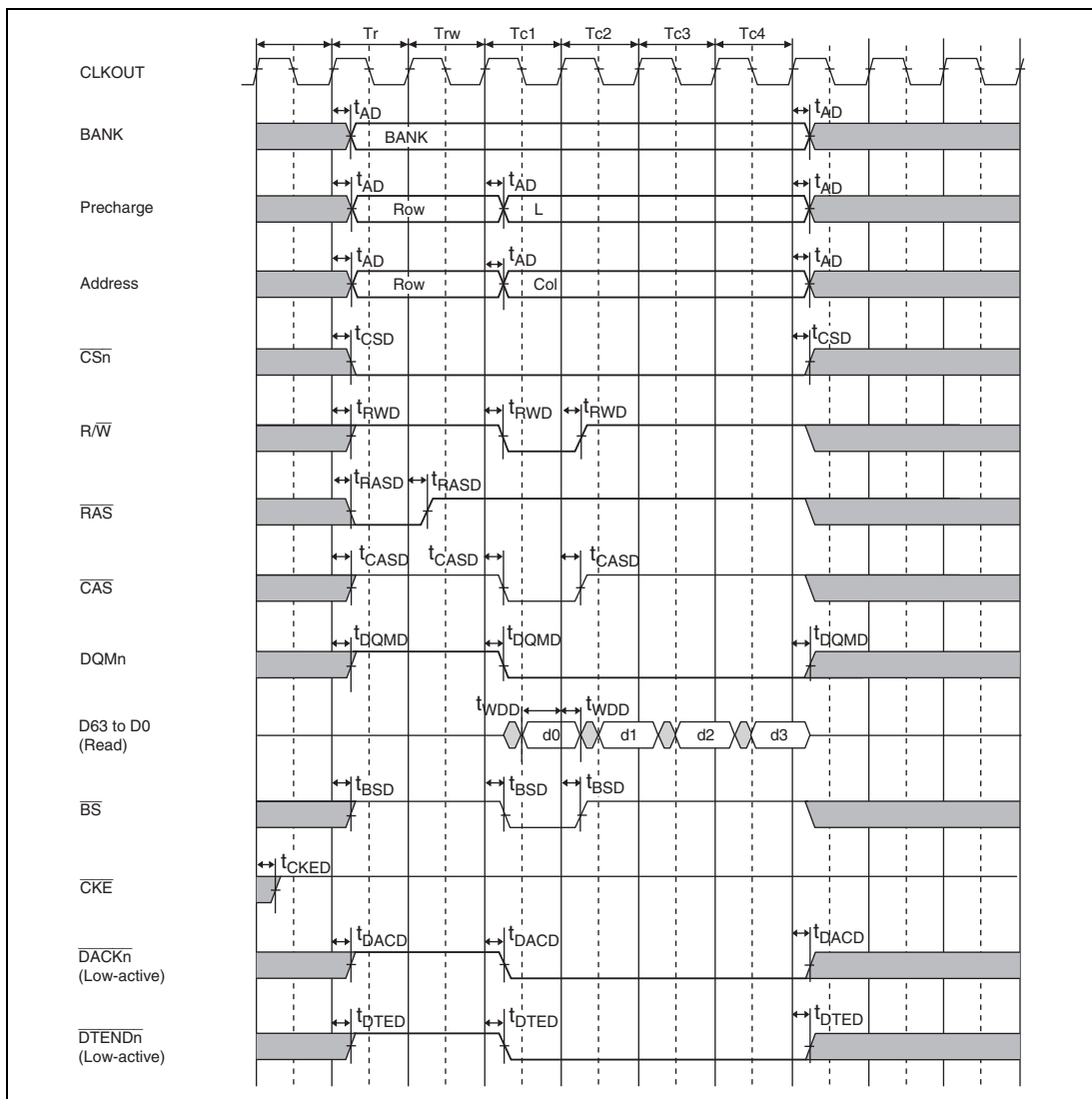


**Figure 33.13 SRAM Bus Cycle in Bank Open Mode Pre-charge Read Bus Cycle (PRE-ACT-READ)**

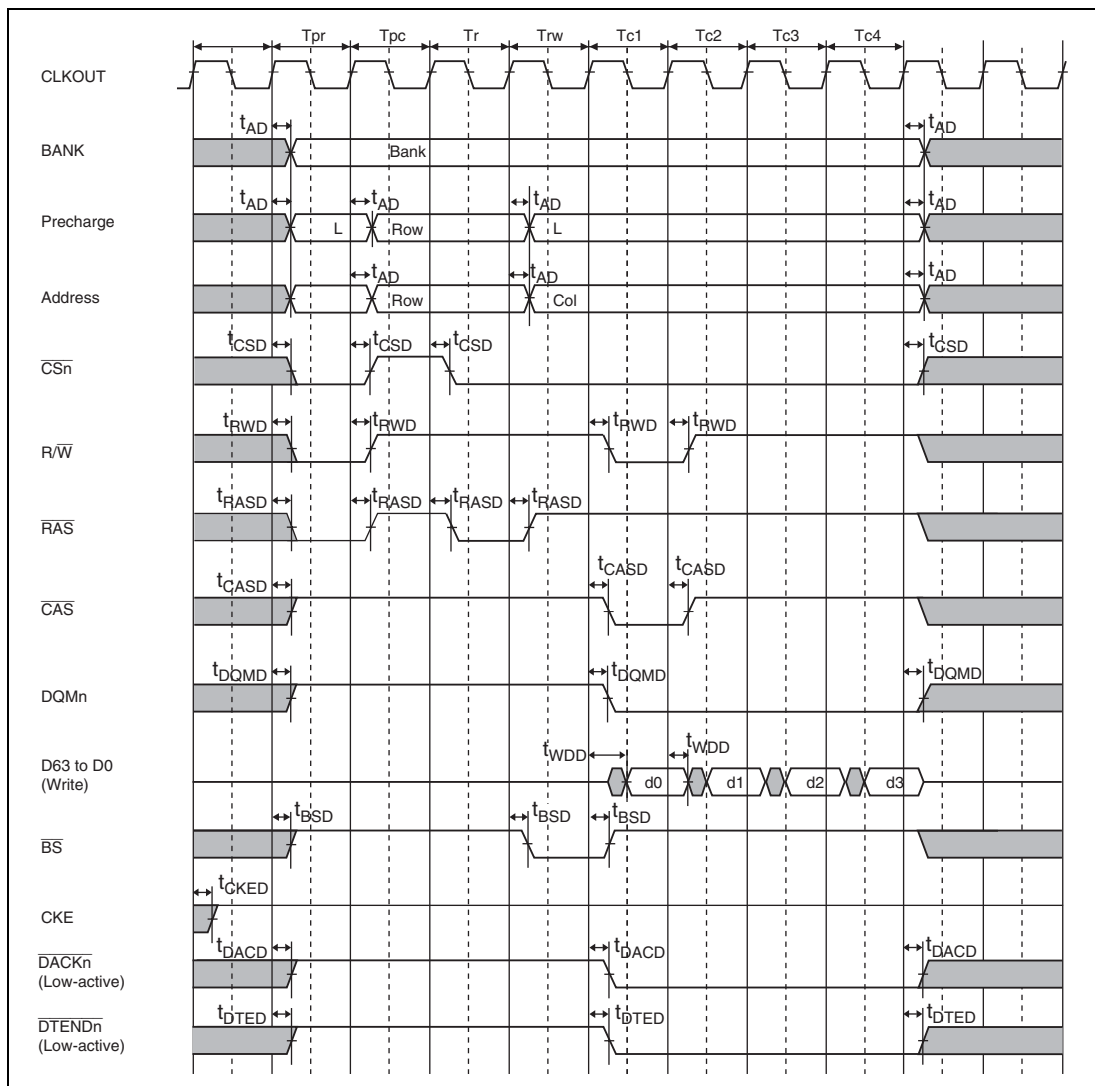
(BOMODE[1:0]= 00, SRP[1:0]= 00, SCL[2:0]= 000, SRCD=0, IRP= 2cyc,  
CAS Latency= 2cyc, IRCD= 2cyc)



**Figure 33.14 SRAM Bus Cycle in Bank Open Mode Read Bus Cycle (Read)**  
**(BOMODE[1:0]= 00, SCL[2:0]= 000, SRCD=0, CAS Latency= 2cyc, IRCD= 2cyc)**

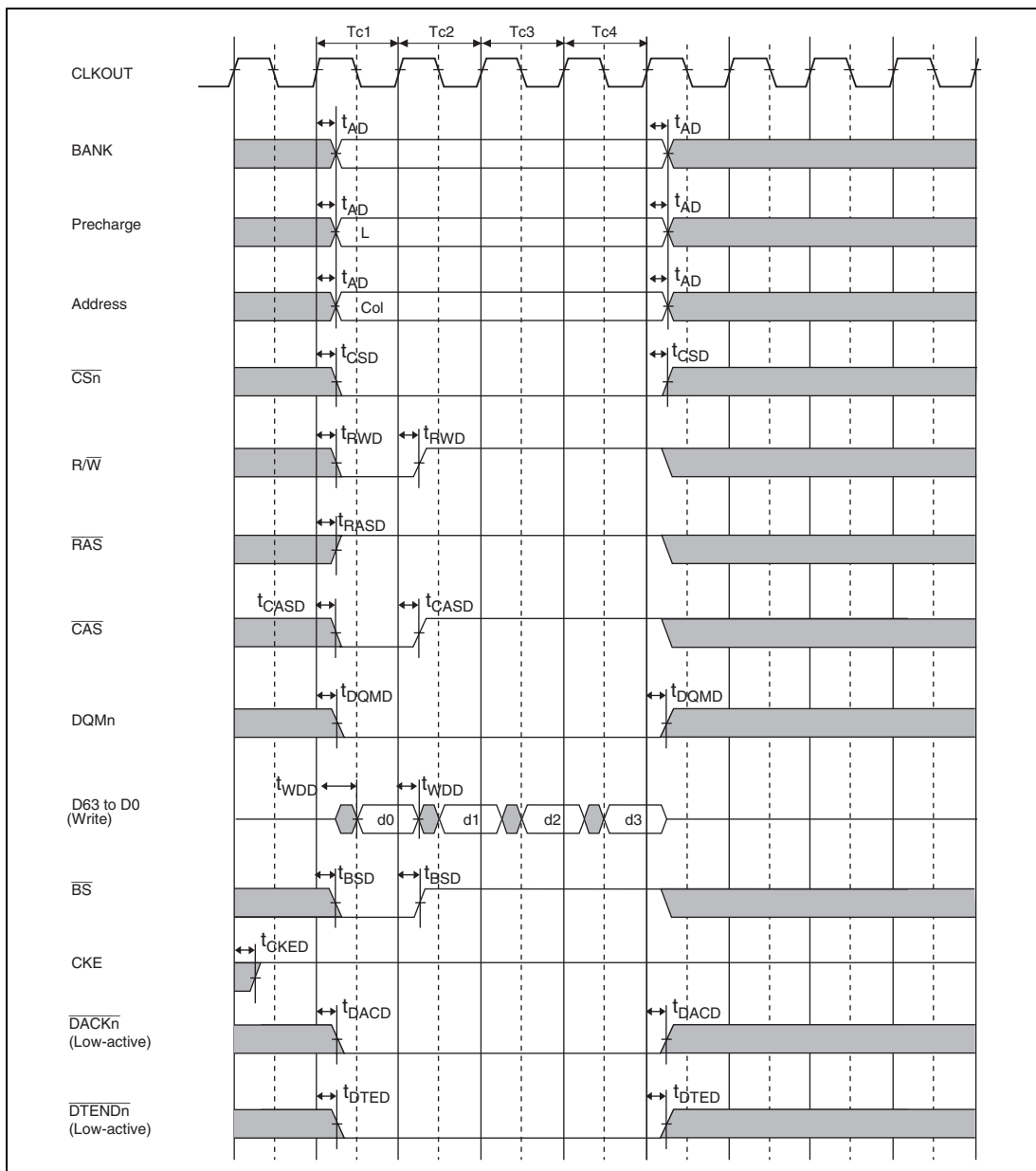


**Figure 33.15 SRAM Bus Cycle in Bank Open Mode Write Bus Cycle (ACT-WRITE)**  
**(BOMODE[1:0]= 00, SRCD= 0, IRCD= 2cyc)**



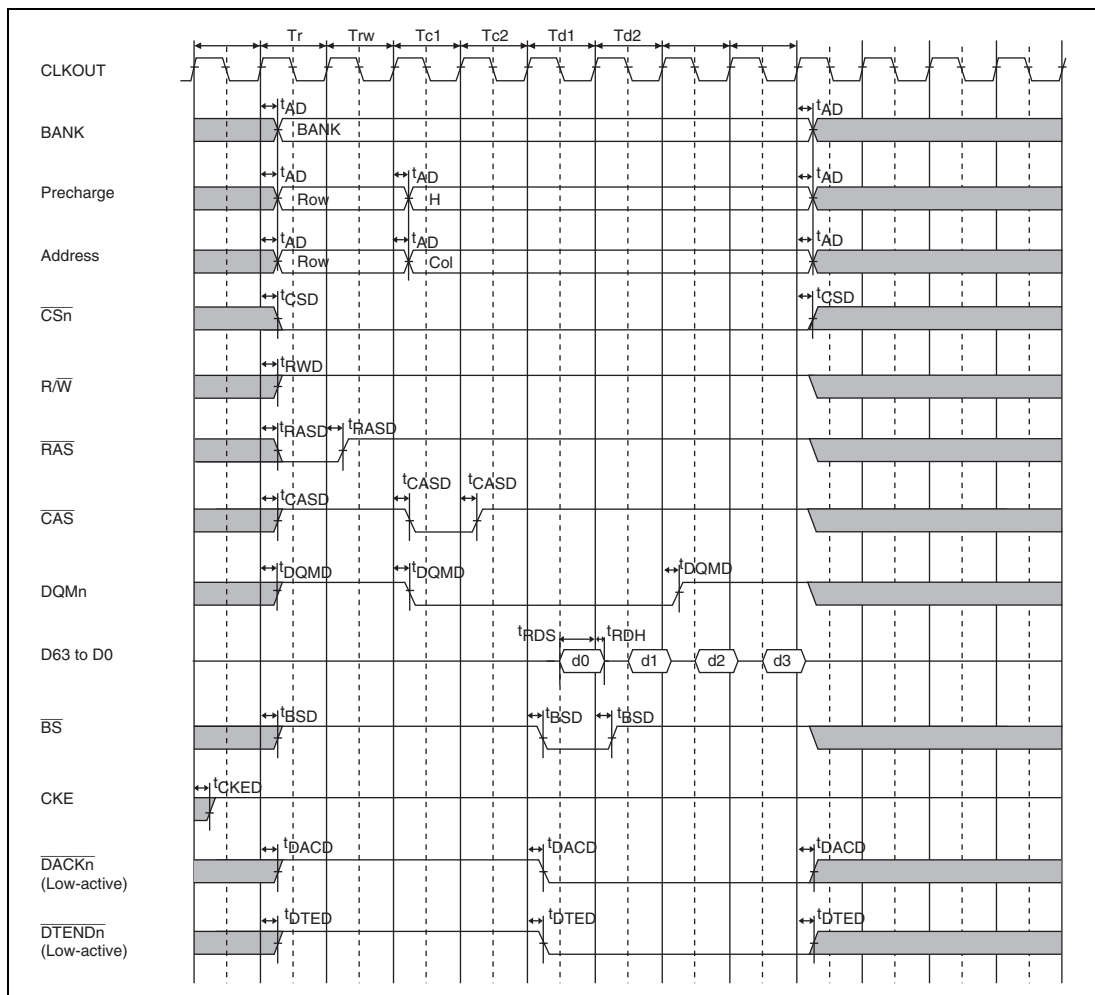
**Figure 33.16 SRAM Bus Cycle in Bank Open Mode Pre-charge Write Bus Cycle (PRE-ACT-SRITE)**

(BOMODE[1:0]= 00, SRP[1:0]= 00, SRCD=0, IRP= 2cyc, IRCD= 2cyc)



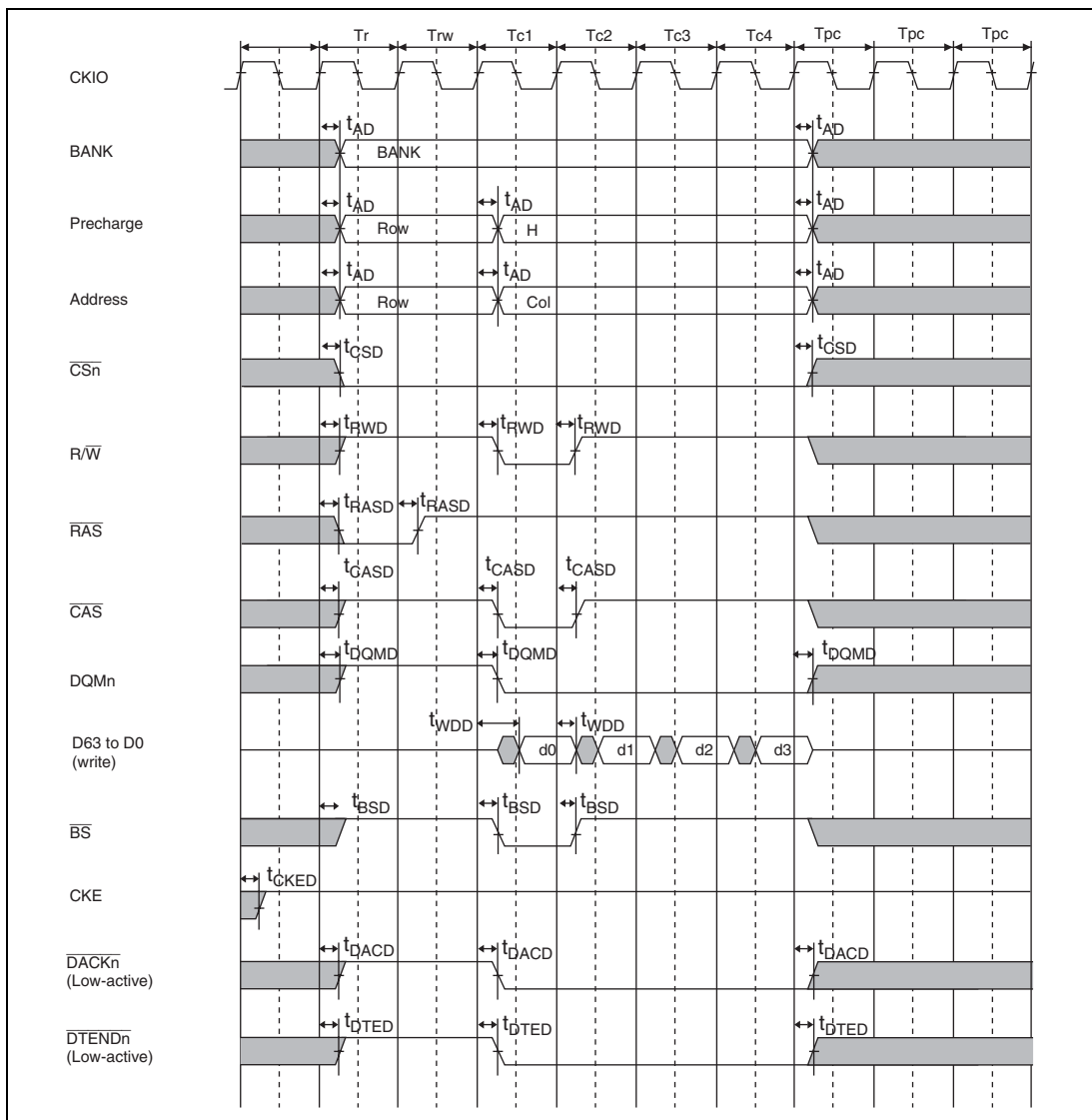
**Figure 33.17 SRAM Bus Cycle in Bank Open Mode Write Bus Cycle (WRITE)**  
**(BOMODE[1:0]= 00, SRCD= 0, IRCD= 2cyc)**



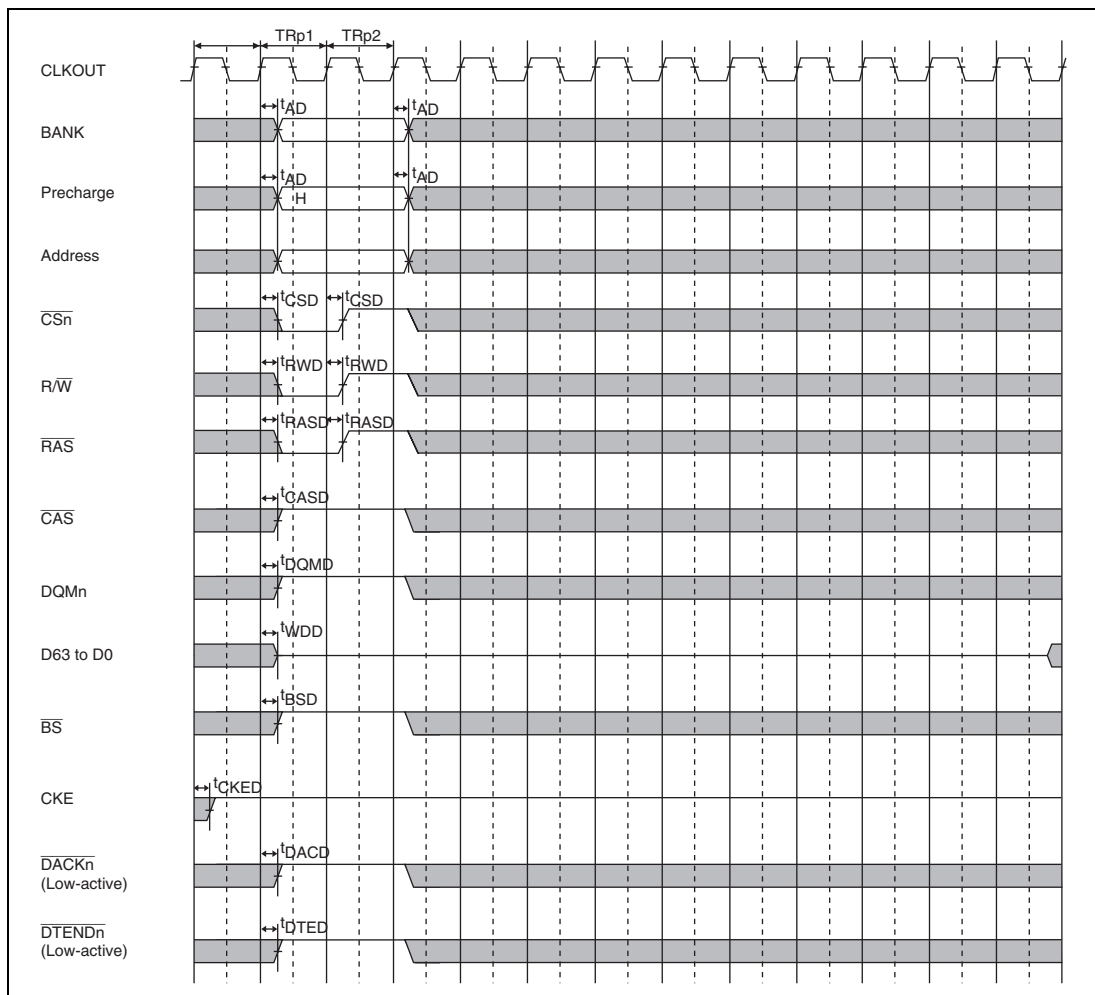


**Figure 33.18 SRAM Bus Cycle in Bank Close Mode Read Bus Cycle (ACT-READA)**

(BOMODE[1:0]= 1, SCL[2:0]= 000, SRCD= 0, IRP= 2cyc, CAS Latency= 2cyc)



**Figure 33.19 SRAM Bus Cycle in Bank Close Mode Write Bus Cycle (ACT-WRITEA)**  
**(BOMODE[1:0]= 00, SWR[1:0]= 00, SRP[1:0]= 00, SRCD= 0, IDAL= 4cyc, IRCD= 2cyc)**



**Figure 33.20 SRAM Bus Cycle in Pre-charge Cycle (PALL)**  
 (SRP[1:0]= 00, IRP= 2cyc)

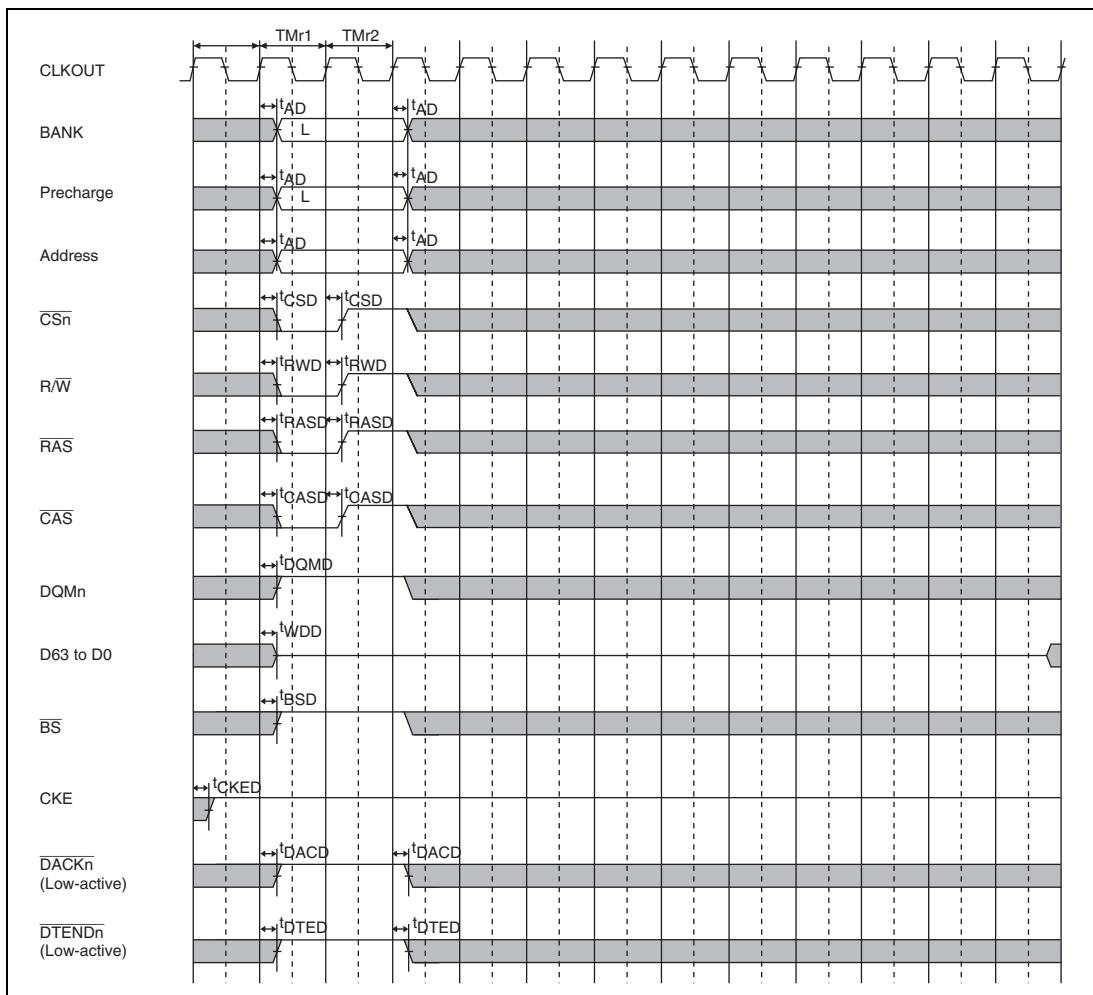
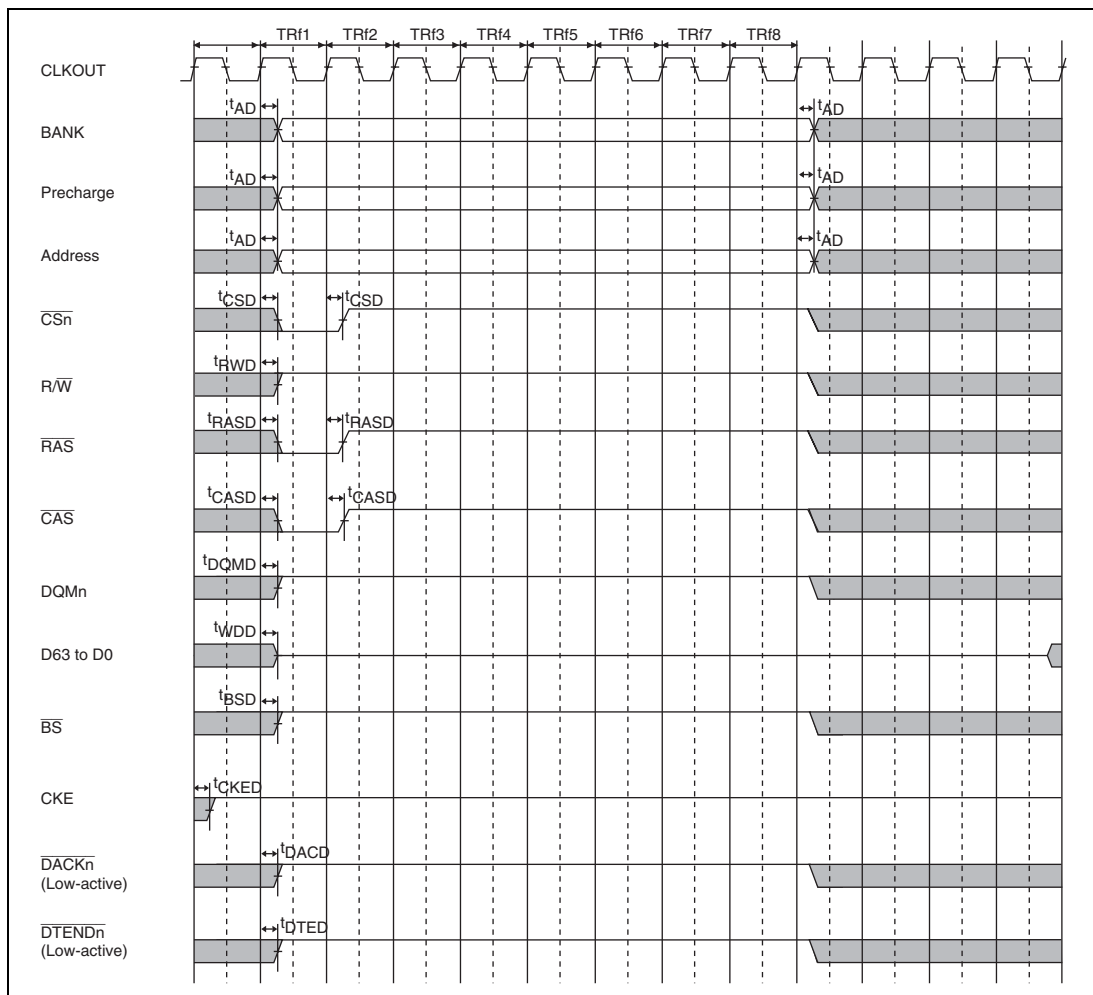


Figure 33.21 SRAM Bus Cycle in Mode Register Setting Cycle (MRS)



**Figure 33.22 SRAM Bus Cycle in Auto Refresh Cycle (REF)**  
(SRFC[2:0]= 000, IRC= 8cyc)

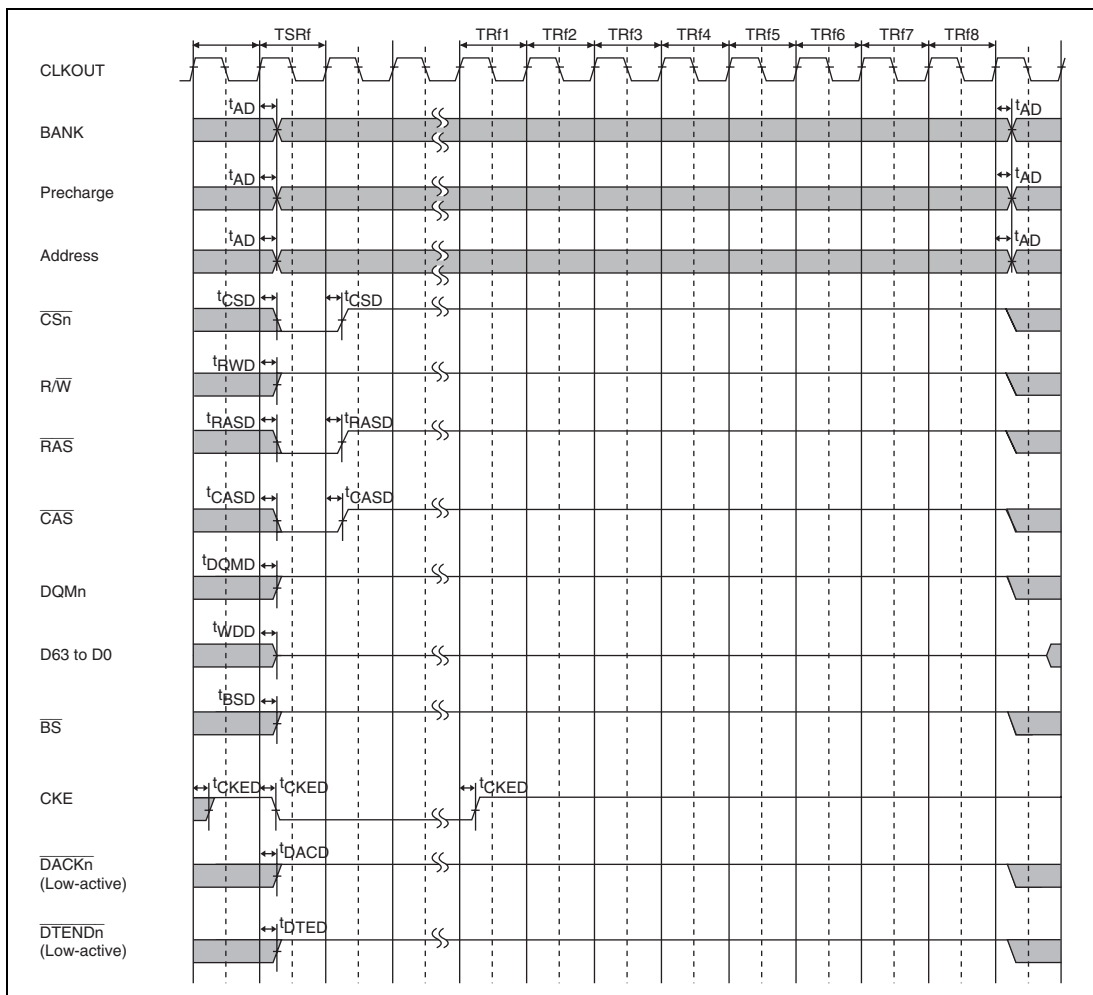


Figure 33.23 SRAM Bus Cycle in Refresh Cycle (SREF)

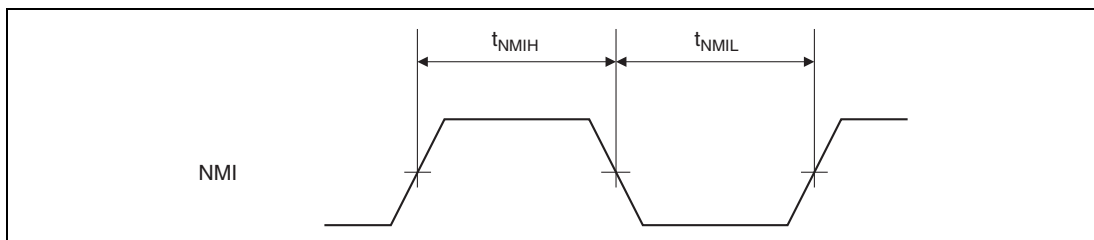
### 33.4.4 INTC Module Signal Timing

**Table 33.15 INTC Module Signal Timing**

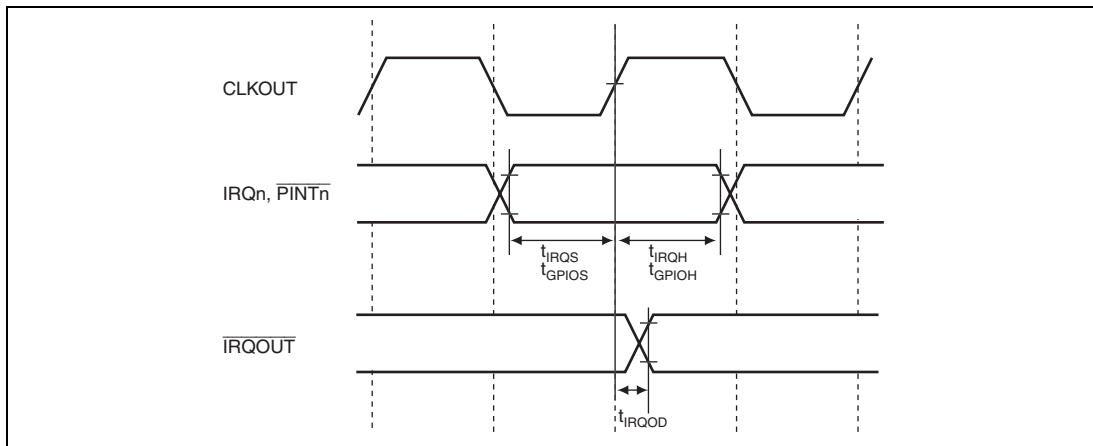
Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^{\circ}\text{C}$ ,  $-40$  to  $85^{\circ}\text{C}$

Item	Symbol	Min.	Max.	Unit	Figure	Note
NMI pulse width (high)	$t_{\text{NMIH}}$	5	—	$t_{\text{cyc}}$	33.24	Regular state, Sleep state
NMI pulse width (low)	$t_{\text{NMIL}}$	5	—	$t_{\text{cyc}}$		Regular state, sleep state
IRQ1, IRQ0 setup time	$t_{\text{IRQS}}$	8	—	ns	33.25	IRQ input
IRQ1, IRQ0 hold time	$t_{\text{IRQH}}$	3	—	ns		IRQ input
$\overline{\text{PINTn}}$ interrupt setup time	$t_{\text{GPIOS}}$	15	—	ns		GPIO interrupt input
$\overline{\text{PINTn}}$ interrupt hold time	$t_{\text{GPIOH}}$	8	—	ns		GPO interrupt input
$\overline{\text{IRQOUT}}$ output delay time	$t_{\text{IRQOD}}$	—	13	ns		$\overline{\text{IRQOUT}}$ output

Note:  $t_{\text{cyc}}$  is a cycle time of CLKOUT clock.



**Figure 33.24 NMI Input Timing**



**Figure 33.25**  $\overline{\text{IRQ}}, \overline{\text{PINTn}}$  Input,  $\overline{\text{IRQOUT}}$  Output Timing

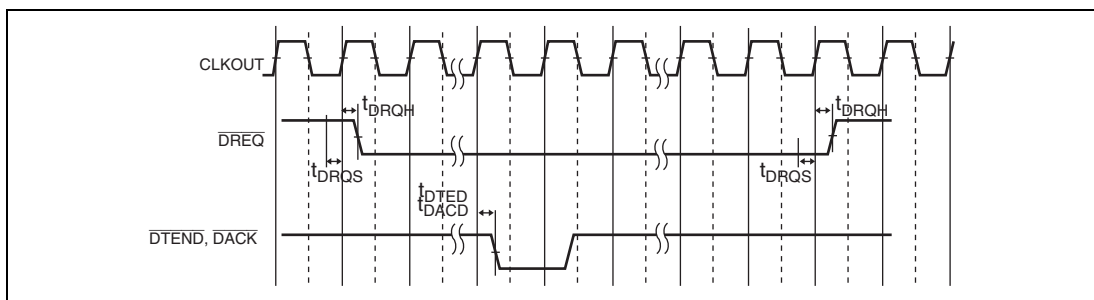


### 33.4.5 DMAC Module Signal Timing

**Table 33.16 DMAC Module Signal Timing**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Module	Item	Symbol	Min.	Max.	Unit	Figure
DMAC	$\overline{\text{DREQn}}$ setup time	$t_{\text{DRQS}}$	6	—	ns	33.26
	$\overline{\text{DREQn}}$ hold time	$t_{\text{DRQH}}$	4	—	ns	
	$\overline{\text{DTENDn}}$ delay time	$t_{\text{DTED}}$	1.0	8	ns	33.26,
	$\overline{\text{DACKn}}$ delay time	$t_{\text{DACD}}$	1.0	8	ns	33.8 to 33.23



**Figure 33.26  $\overline{\text{DREQ}}/\overline{\text{DTEND}}/\overline{\text{DACK}}$  Timing**

### 33.4.6 TMU Module Signal Timing

**Table 33.17 TMU Module Signal Timing**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Module	Item	Symbol	Min.	Max.	Unit	Figure
TMU	Timer clock pulse width (high)	$t_{\text{TCLKWH}}$	4	—	$t_{\text{Pcyc}}$	33.27
	Timer clock pulse width (low)	$t_{\text{TCLKWL}}$	4	—	$t_{\text{Pcyc}}$	
	Timer clock rising time	$t_{\text{TCLKr}}$	—	0.8	$t_{\text{Pcyc}}$	
	Timer clock falling time	$t_{\text{TCLKf}}$	—	0.8	$t_{\text{Pcyc}}$	

Note:  $t_{\text{pcyc}}$  indicates the peripheral clock (Pck) cycle.

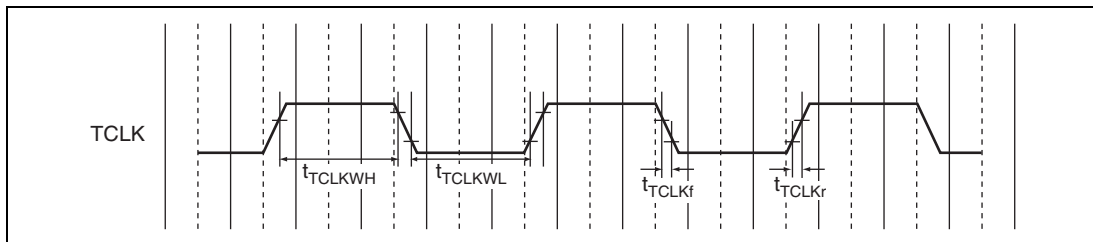


Figure 33.27 TCLK Input Timing

### 33.4.7 IIC Module Signal Timing

Table 33.18 I<sup>2</sup>C Module Signal Timing

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
SCL frequency	$t_{ICYC}$	0	—	400	kHz	33.28, 33.29
SCL, SDA input rise time	$t_{ICF}$	—	—	300	ns	RP-CB=
SDA bus free time	$t_{ICBF}$	1.3	—	—	$\mu\text{s}$	$257 \times 10^{-9}$ to $275 \times 10^{-9}$
SCL start condition input hold time	$t_{ICH}$	0.6	—	—	$\mu\text{s}$	$[\Omega \cdot \text{pF}]$ VPU= 3.3 V
SCL retransmit start condition setup time	$t_{ICS}$	0.6	—	—	$\mu\text{s}$	
SDA stop condition setup time	$t_{ICST}$	0.6	—	—	ns	
SDA setup time	$t_{DAS}$	100	—	—	ns	
SDA hold time	$t_{ICDH}$	0	—	0.9	ns	

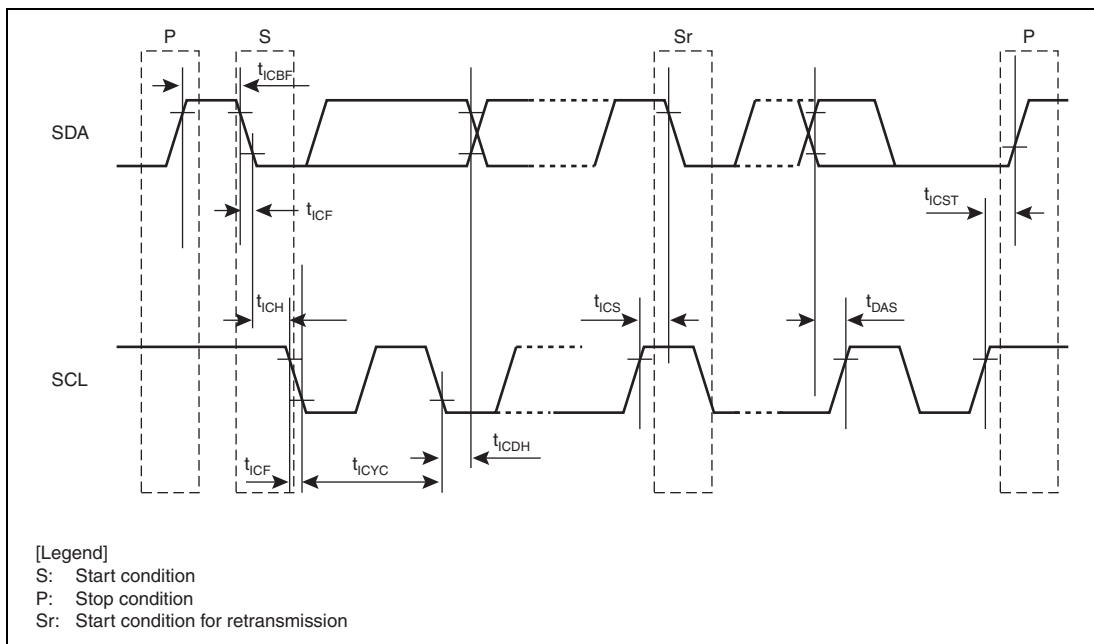


Figure 33.28 I²C Timing

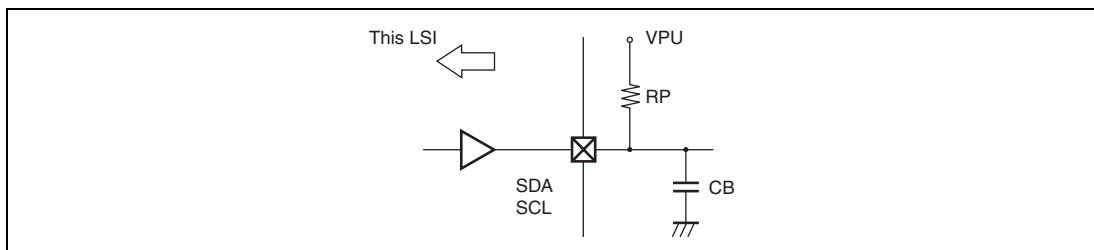


Figure 33.29 AC Characteristics Load Condition

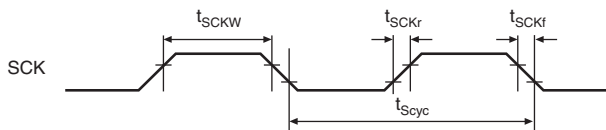
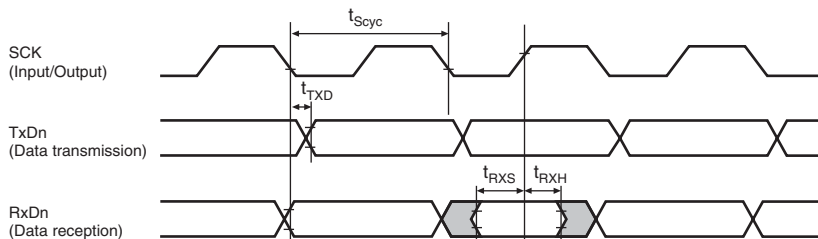
### 33.4.8 SCIF Module SignalTiming

**Table 33.19 SCIF Module Signal Timing**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle	Clocked synchronous	$t_{\text{Scyc}}$	12	—	$t_{\text{pcyc}}$	33.30
	Asynchronous		4	—	$t_{\text{pcyc}}$	
Input clock rising time		$t_{\text{SCKr}}$	—	1.5	$t_{\text{pcyc}}$	
Input clock falling time		$t_{\text{SCKf}}$	—	1.5	$t_{\text{pcyc}}$	
Input clock width		$t_{\text{SCKW}}$	0.4	0.6	$t_{\text{Scyc}}$	
Transfer data delay time (Clocked synchronous)		$t_{\text{TXD}}$	—	$3 \times t_{\text{pcyc}} + 15$	$t_{\text{pcyc}}$	33.31
Receive data setup time (Clocked synchronous)		$t_{\text{RXS}}$	$4 \times t_{\text{pcyc}} + 15$	—	ns	
Receive data hold time (Clocked synchronous)		$t_{\text{RXH}}$	100	—	ns	

Note:  $t_{\text{pcyc}}$  indicates the peripheral clock (ck) cycle.

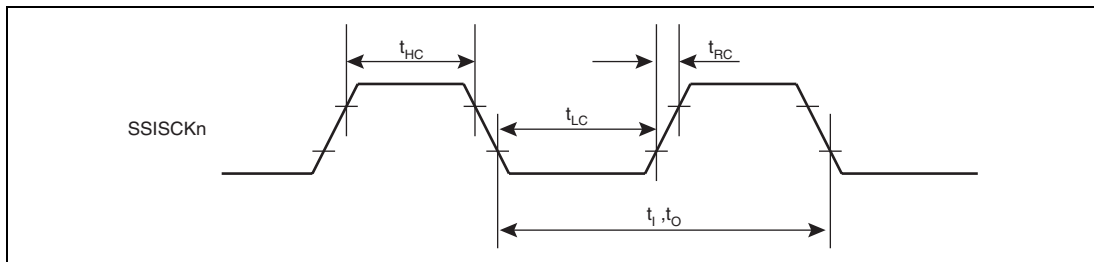

**Figure 33.30 SCK Input Clock Timing**

**Figure 33.31 SCIF Input/Output Timing in Clocked Synchronous Mode**

### 33.4.9 SSI Module Signal Timing

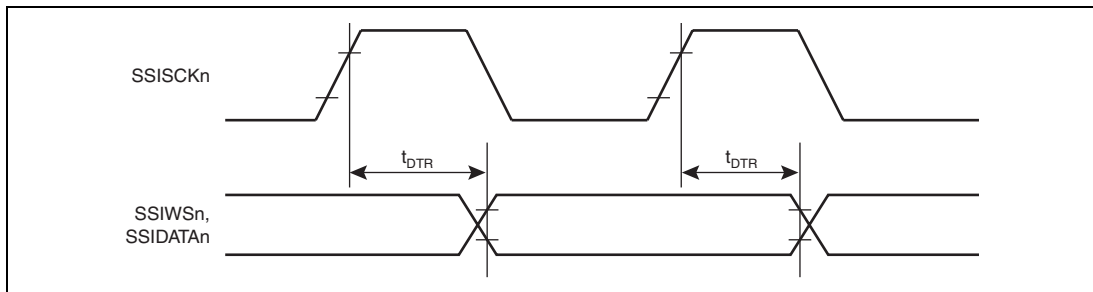
**Table 33.20 SSI Module Signal Timing**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

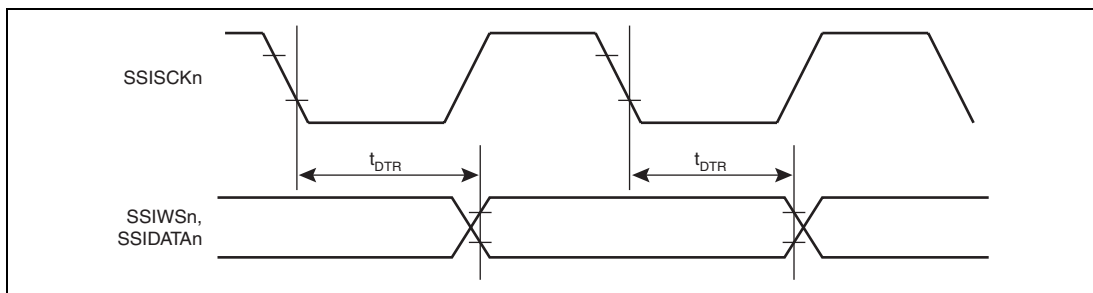
Item	Symbol	Min.	Typ.	Max.	Unit	Remarks	Figure
Output clock frequency	$t_o$	160	—	3364	ns	Output	33.32
Input clock frequency	$t_i$	160	—	3364	ns	Input	
Clock high	$t_{HC}$	40	—	—	ns	Bidirectional	
Clock low	$t_{LC}$	40	—	—	ns		
Clock rise time	$t_{RC}$	—	—	20	ns	Output (100 pF)	
Delay	$t_{DTR}$	—	—	30	ns	Transmit	33.33, 33.34
Setup time	$t_{SR}$	20	—	—	ns	Receive	33.35, 33.36
Hold time	$t_{HTR}$	10	—	—	ns	Receive	
AUDIO_CLK frequency	$f_{\text{AUDIO}}$	3.072	—	24.576	MHz		33.37



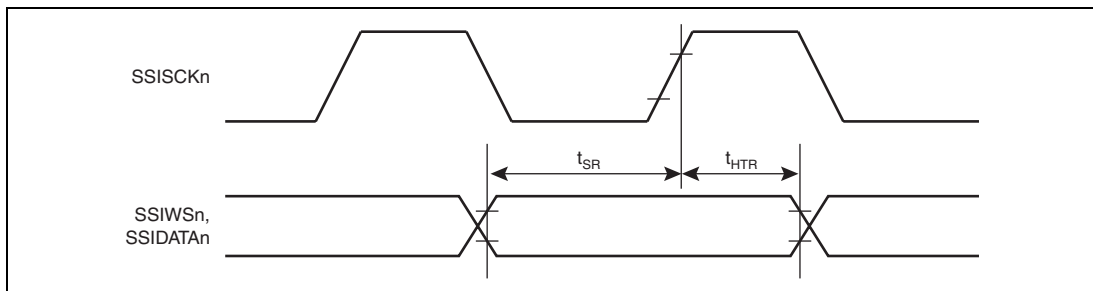
**Figure 33.32 Clock Input/Output Timing**



**Figure 33.33 SSI Transmit Timing (1)**



**Figure 33.34 SSI Transmit Timing (2)**



**Figure 33.35 SSI Receive Timing (1)**

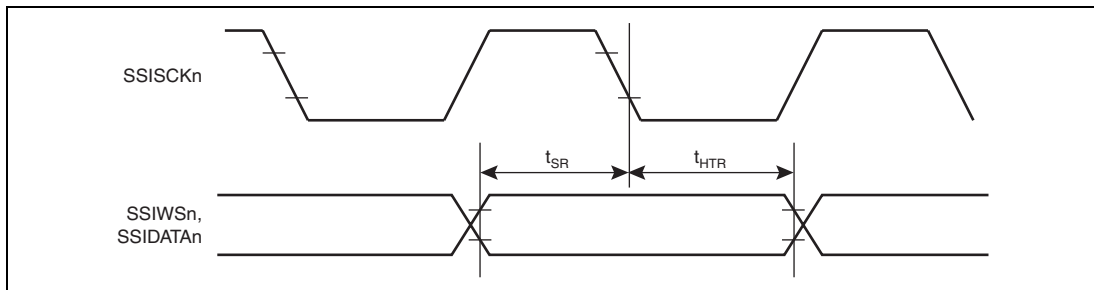


Figure 33.36 SSI Receive Timing (2)

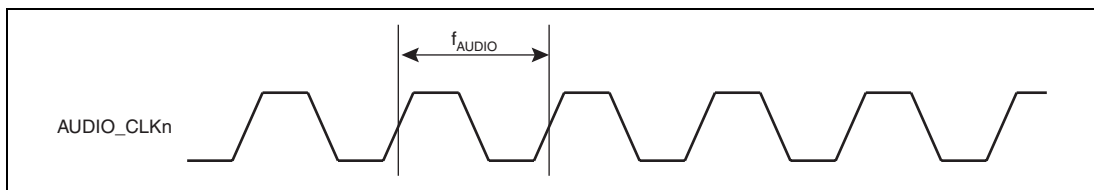


Figure 33.37 AUDIO\_CLK Timing

### 33.4.10 ATAPI Interface Module Signal timing

**Table 33.21 ATAPI Interface Resister Access Timing in PIO Transmission**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^{\circ}\text{C}$ ,  $-40$  to  $85^{\circ}\text{C}$

Symbol and Item for Register Access in PIO Transmission (max/min)			Mode 0 ns	Mode 1 ns	Mode 2 ns	Mode 3 ns	Mode 4 ns	Figure
t0	Cycle Time	(min)	600	383	330	180	120	33.38
t1	Address setup time	(min)	70	50	30	30	25	
t2	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ pulse width 8-bit	(min)	290	290	290	80	70	
t2i	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ recovery time	(min)	—	—	—	70	25	
t3	$\overline{\text{IDEIOWR}}$ data setup time	(min)	60	45	30	30	20	
t4	$\overline{\text{IDEIOWR}}$ data hold time	(min)	30	20	15	10	10	
t5	$\overline{\text{IDEIORD}}$ data setup time	(min)	50	35	20	20	20	
t6	$\overline{\text{IDEIORD}}$ data hold time	(min)	5	5	5	5	5	
t6z	$\overline{\text{IDEIORD}}$ three-state delay time	(max)	30	30	30	30	30	
t9	Address hold time	(min)	20	15	10	10	10	
tRD	IDEIORDY read data valid time	(min)	0	0	0	0	0	
tA	IDEIORDY setup time		35	35	35	35	35	
tB	IDEIORDY pulse time	(max)	1250	1250	1250	1250	1250	
tC	IDEIORDY time from negate to high-impedance	(max)	5	5	5	5	5	



**Table 33.22 ATAPI Interface Data Transmission Timing in PIO Transmission**

Symbol ,Item and Conditions for Data Transfer in PIO (max/min)			Mode 0 ns	Mode 1 ns	Mode 2 ns	Mode 3 ns	Mode 4 ns	Figure
t0	Cycle time	(min)	600	383	240	180	120	33.38
t1	Address setup time	(min)	70	50	30	30	25	
t2	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ pulse width 8-bit	(min)	290	290	290	80	70	
t2i	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ recovery time	(min)	—	—	—	70	25	
t3	$\overline{\text{IDEIOWR}}$ data setup time	(min)	60	45	30	30	20	
t4	$\overline{\text{IDEIOWR}}$ data hold time	(min)	30	20	15	10	10	
t5	$\overline{\text{IDEIORD}}$ data setup time	(min)	50	35	20	20	20	
t6	$\overline{\text{IDEIORD}}$ data hold time	(min)	5	5	5	5	5	
t6z	$\overline{\text{IDEIORD}}\overline{\text{D3}}$ state delay time	(max)	30	30	30	30	30	
t9	Address hold time	(min)	20	15	10	10	10	
tRD	IDEIORDY read data valid time	(min)	0	0	0	0	0	
tA	IDEIORDY setup time		35	35	35	35	35	
tB	IDEIORDY pulse time	(max)	1250	1250	1250	1250	1250	
tC	Time from negate to high-impedance of IDEIORDY	(max)	5	5	5	5	5	

**Table 33.23 ATAPI Interface Multiword Transmission Timing**

Symbol and Item Multiword Transmission(max/min)			Mode 0 ns	Mode 1 ns	Mode 2 ns	Figure
t0	Cycle time	(min)	480	150	120	33.40 to 33.42
tD	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ pulse width	(min)	215	80	70	33.39 to 33.42
tE	$\overline{\text{IDEIORD}}$ data access time	(max)	150	60	50	
tF	$\overline{\text{IDEIORD}}$ data hold time	(min)	5	5	5	
tG	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ data setup time	(min)	100	30	20	
tH	$\overline{\text{IDEIOWR}}$ data hold time	(min)	20	15	10	
tI	$\overline{\text{IODACK}}$ setup time	(min)	0	0	0	33.39
tJ	$\overline{\text{IODACK}}$ hold time	(min)	20	5	5	33.41, 33.42
tKR	$\overline{\text{IDEIORD}}$ negate pulse width	(min)	50	50	25	33.40 to 33.42
tKW	$\overline{\text{IDEIOWR}}$ negate pulse width	(min)	215	50	25	33.40 to 33.42
tLR	$\overline{\text{IDEIORD}}$ IODREQ delay time	(max)	120	40	35	33.41
tLW	$\overline{\text{IDEIOWR}}$ IODREQ delay time	(max)	40	40	35	
tM	$\overline{\text{IDECs}}[1:0]$ setup time	(min)	50	30	25	33.39
tN	$\overline{\text{IDECs}}[1:0]$ hold time	(min)	15	10	10	33.41, 33.42
tZ	$\overline{\text{IODACK}}$ 3-state delay time	(max)	20	25	25	

**Table 33.24 ATAPI Interface Ultra-DMA Transmission Timing**

Symbol for Ultra-DMA	Mode 0 ns		Mode 1 ns		Mode 2 ns		Mode 3 ns		Mode 4 ns		Figure
	min	max	min	max	min	max	min	max	min	max	
t2CYCTYP	240		160		120		90		60		33.44
tCYC	112		73		54		39		25		33.44, 33.49
t2CYC	230		153		115		86		57		
tDS	15		10		7		7		5		
tDH	5		5		5		5		5		
tDVS	70		48		31		20		6.7		33.43, 33.44, 33.48, 33.49
tDVH	6.2		6.2		6.2		6.2		6.2		
tCS	15		10		7		7		5		
tCH	5		5		5		5		5		
tCVS	70		48		31		20		6.7		33.46, 33.47, 33.51, 33.52
tCVH	6.2		6.2		6.2		6.2		6.2		
tZFS	0		0		0		0		0		33.43
tDZFS	70		48		31		20		6.7		33.43, 33.48
tFS		230		200		170		130		120	33.43
tLI	0	150	0	150	0	150	0	100	0	100	33.46 to 33.48 33.51, 33.52
tMLI	20		20		20		20		20		33.46, 33.47, 33.51, 33.52
tUI	0		0		0		0		0		33.43, 33.48
tAZ		10		10		10		10		10	33.43, 33.46, 33.47
tZAH	20		20		20		20		20		33.46, 33.47
tZAD	0		0		0		0		0		33.43
tENV	20	70	20	70	20	70	20	55	20	55	33.43, 33.48
tRFS		75		70		60		60		60	33.45, 33.47, 33.50, 33.52

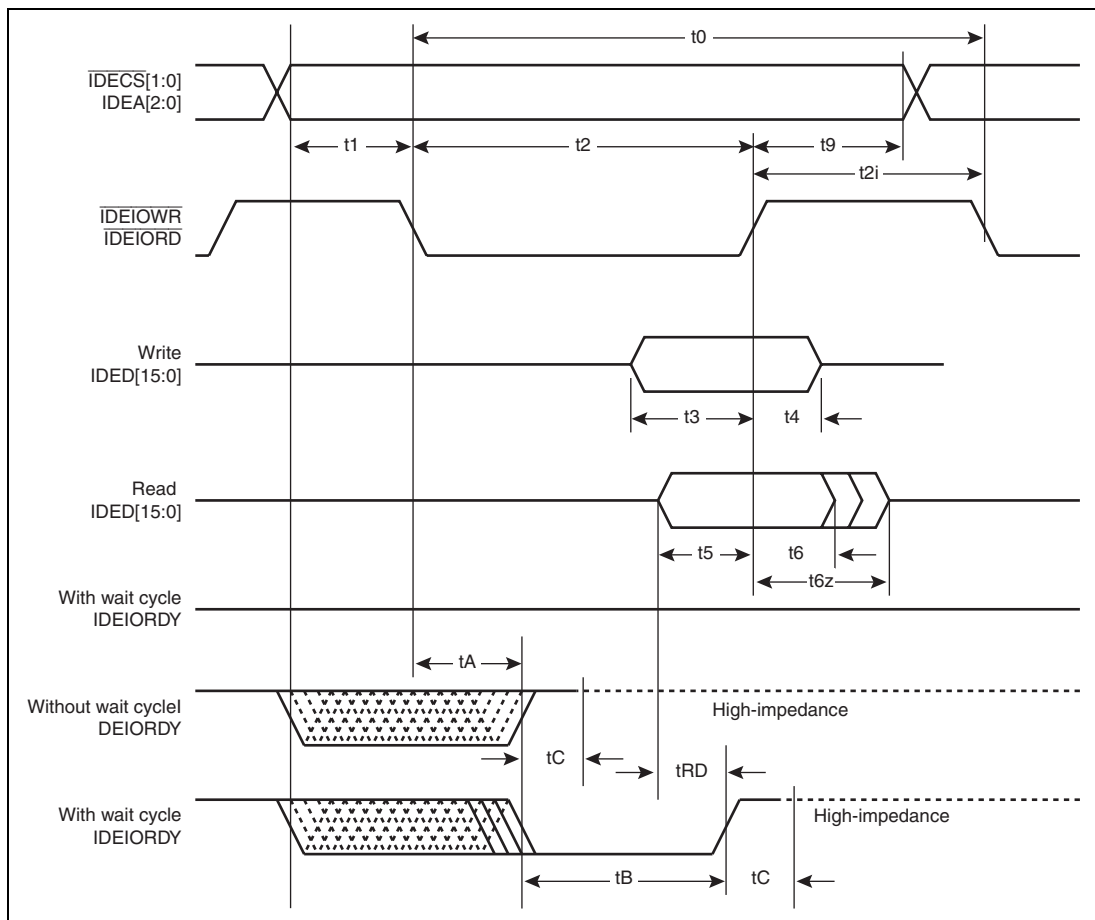
DMA	Mode 0 ns		Mode 1 ns		Mode 2 ns		Mode 3 ns		Mode 4 ns		Figure
	min	max	min	max	min	max	min	max	min	max	
tRP	160		125		100		100		100		
tIORDYZ		20		20		20		20		20	33.46, 33.47, 33.51, 33.52
tZIORDY	0		0		0		0		0		33.43, 33.48
tACK	20		20		20		20		20		33.43, 33.46 to 33.48, 33.51, 33.52
tSS	50		50		50		50		50		33.46, 33.51

**Table 33.25 Symbol for ATAPI Interface Ultra-DMA Transmission Timing**

<b>Symbol</b>	<b>Note</b>
t2CYCTYP	Average cycle time (2 cycles)
tCYC	Cycle time
t2CYC	Minimum cycle time (2 cycles)
tDS	Data setup time (receive side)
tDH	Data hold time (receive side)
tDVS	Data setup time (transfer side)
tDVH	Data hold time (transfer side)
tCS	CRC data setup time (receive side)
tCH	CRC data hold time (receive side)
tCVS	CRC setup time (transfer side)
tCVH	CRC hold time (transfer side)
tZFS	Setup time from the strove state to the drive state of the active signal (transfer side)
tDZFS	Setup time from the drive state to the first strove state of the active signal (transfer side)
tFS	Initial STROBE time
tLI	Interlock time with restriction
tMLI	Minimum interlock time
tUI	Interlock time without restriction
tAZ	Output release time
tZAH	Output delay time
tZAD	Output defined time (from release)
tENV	Envelope time
tRFS	Final STROBE time
tRP	Time till assert the STOP or negate the DMARQ
tIORDYZ	Time till release the IORDY
tZIORDY	Time till drive the STROBE
tACK	Time for $\overline{\text{DMACK}}$ setup/hold
tSS	Time for STROBE STOP

**Table 33.26 ATAPI Interface DIRECTION Timing**

Item or Symbol	Mode 0 ns		Mode 1 ns		Mode 2 ns		Mode 3 ns		Mode 4 ns		Figure
	min	max	min	max	min	max	min	max	min	max	
DIRECTION fall delay time in PIO write DIRECTION_WF	65	74	45	54	25	34	25	34	25	34	33.53
DIRECTION rise delay time in PIO write DIRECTION_WR	47	55	47	55	47	55	47	55	47	55	
Multiword DMA data out DIRECTION fall delay time tMDIRECTION_F	-3	5	-3	5	-3	5	—	—	—	—	33.55
Multiword DMA data-out DIRECTION rise delay time tMDIRECTION_R	7	15	7	15	7	15	—	—	—	—	
DIRECTION fall delay time in Ultra-DMA data- in CRC transmission tUDIRECTION_F(CRC)	116	124	76	84	56	64	46	54	36	44	33.57, 33.58
DIRECTION rise delay time in Ultra-DMA data- in CRC transmission tUDIRECTION_R(CRC)	17	25	17	25	17	25	17	25	17	25	
DIRECTION fall delay time in Ultra-DMA data- out tUDIRECTION_F	25	34	25	34	25	34	25	34	38	43	33.59
DIRECTION rise delay time in Ultra-DMA data- out tUDIRECTION_R	48	55	48	55	48	55	48	55	48	55	33.60, 33.61
Time from DIRECTION fall to turning ON the IDED data bus tDON	9	15	9	15	9	15	9	15	18	22	33.53, 33.55, 33.57 to 33.59
Time from IDED data bus OFF status to DIRECTION rise tDON tDOFF	6	14	6	14	6	14	6	14	6	14	33.53, 33.55, 33.57, 33.58, 33.60, 33.61



**Figure 33.38 PIO Data Transmission In-between Devices**

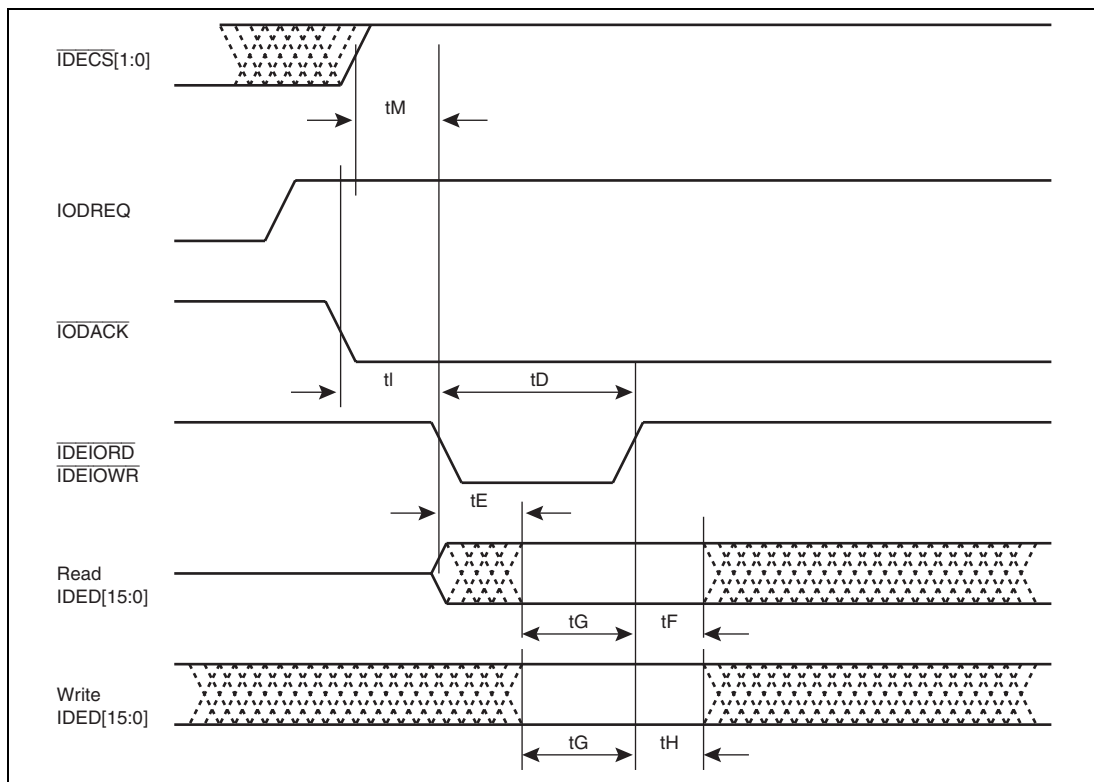


Figure 33.39 Multiword DMA Data Transmission Start



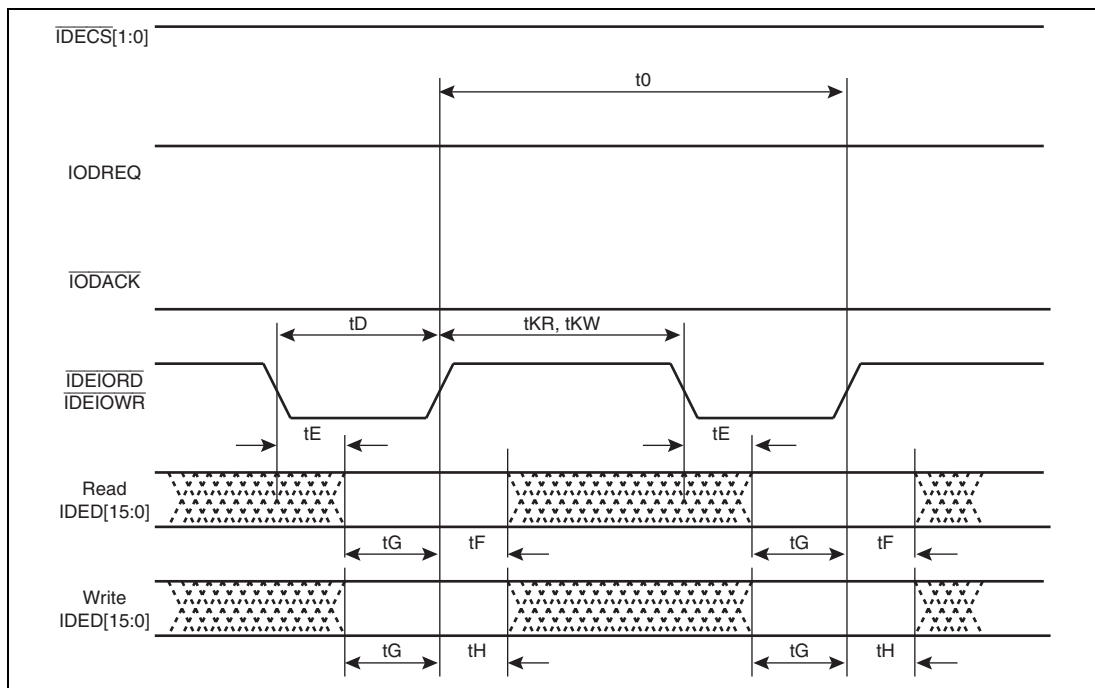
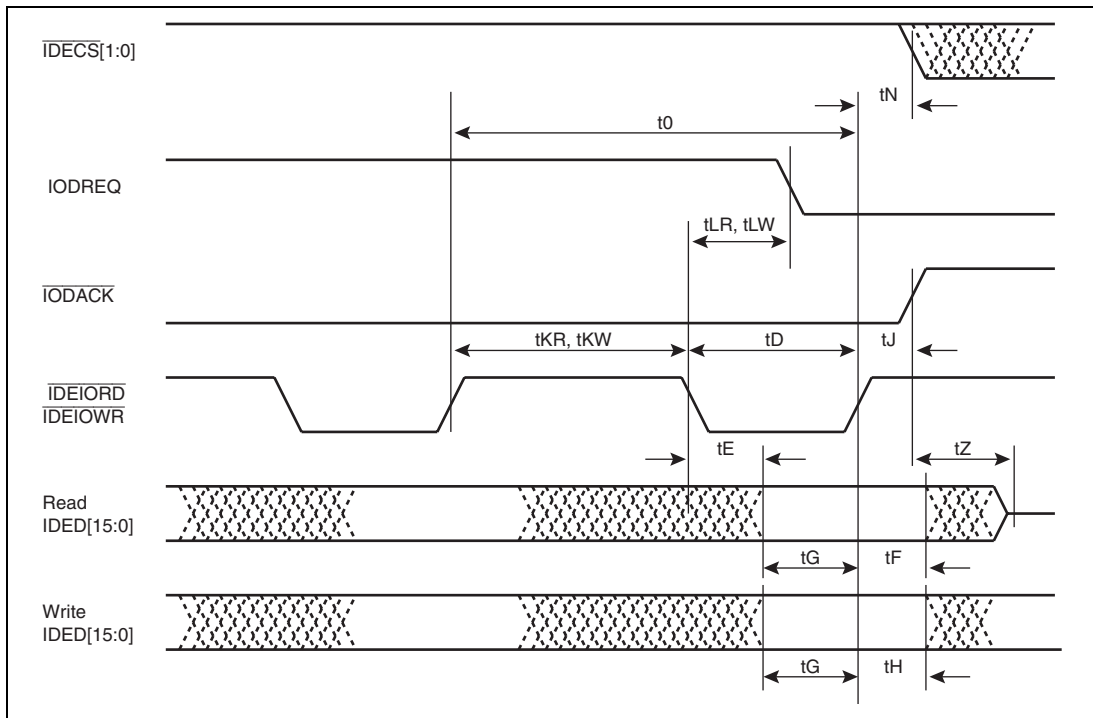
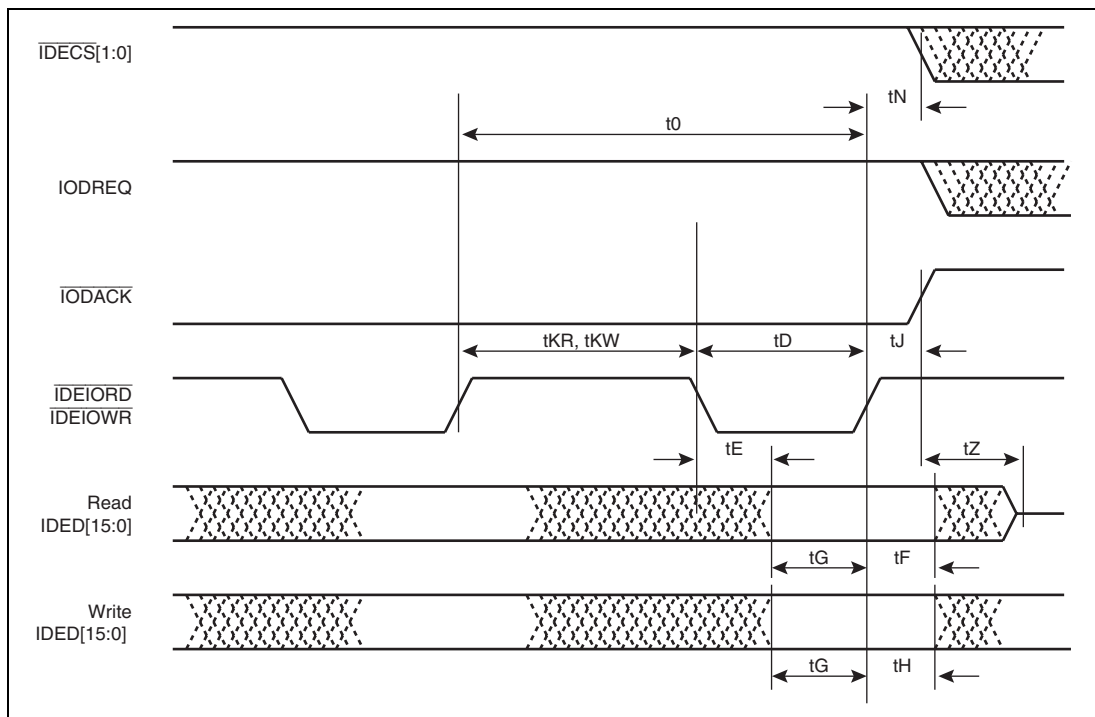


Figure 33.40 Multiword Data Transmission



**Figure 33.41 End of Multiword Data Transmission from Device**



**Figure 33.42 End of Multiword Data Transmission from Host**

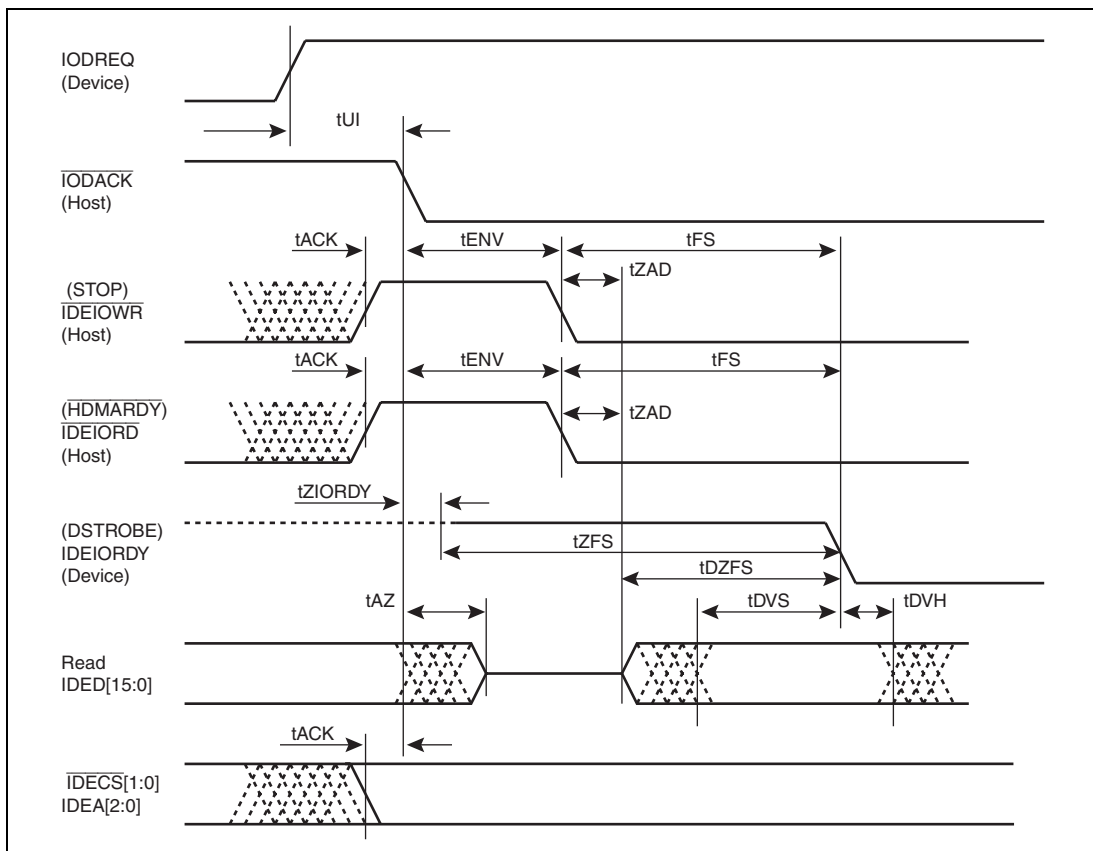


Figure 33.43 Ultra-DMA Data In-burst Start

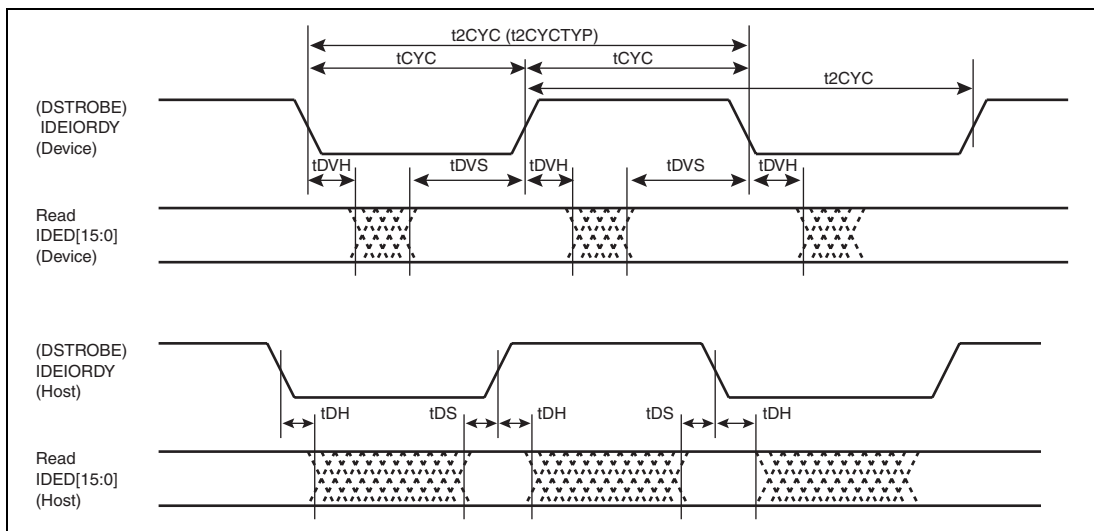


Figure 33.44 Ultra-DMA Data In-burst

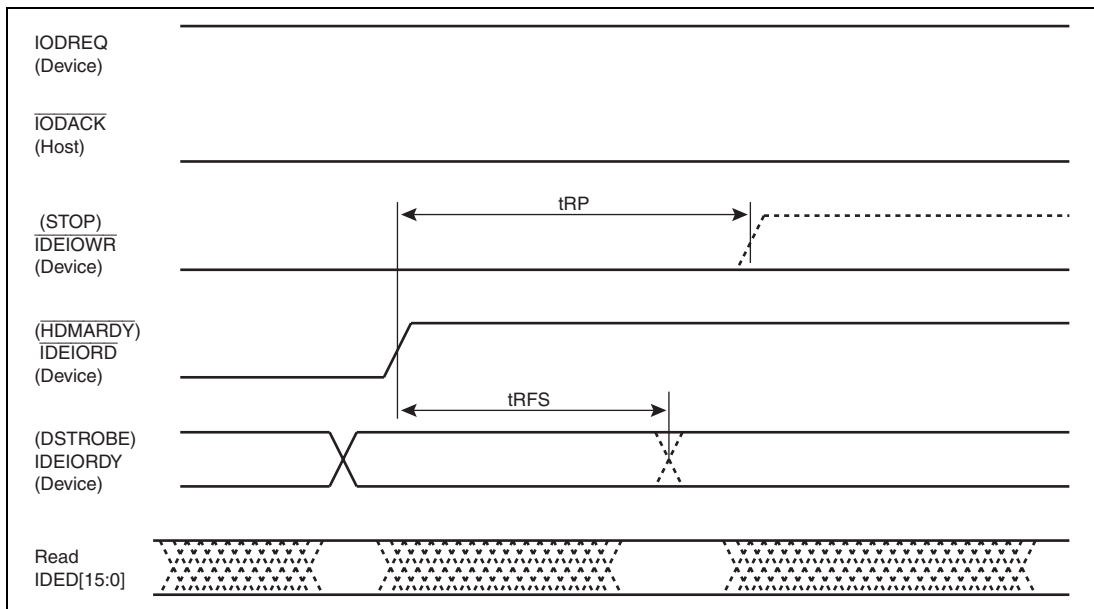


Figure 33.45 Ultra-DMA Data In-burst from Host Pause

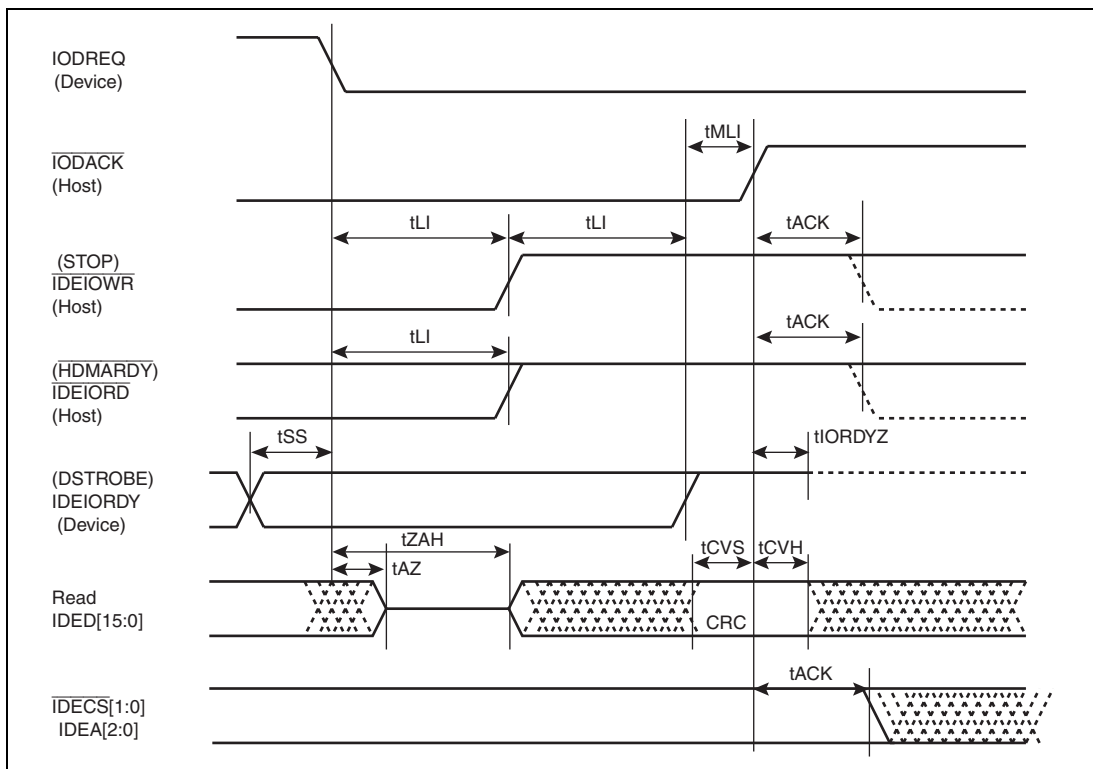
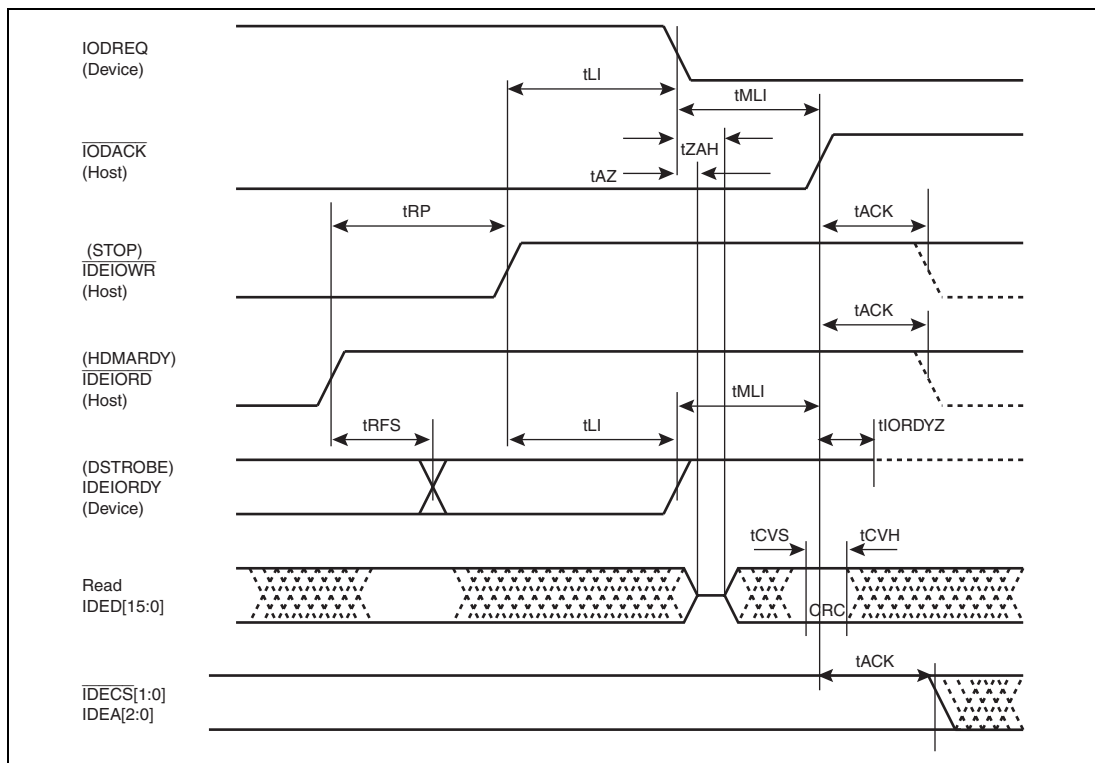


Figure 33.46 End of Ultra-DMA Data In-burst from Device



**Figure 33.47 End of Ultra-DMA Data In-burst from Host**

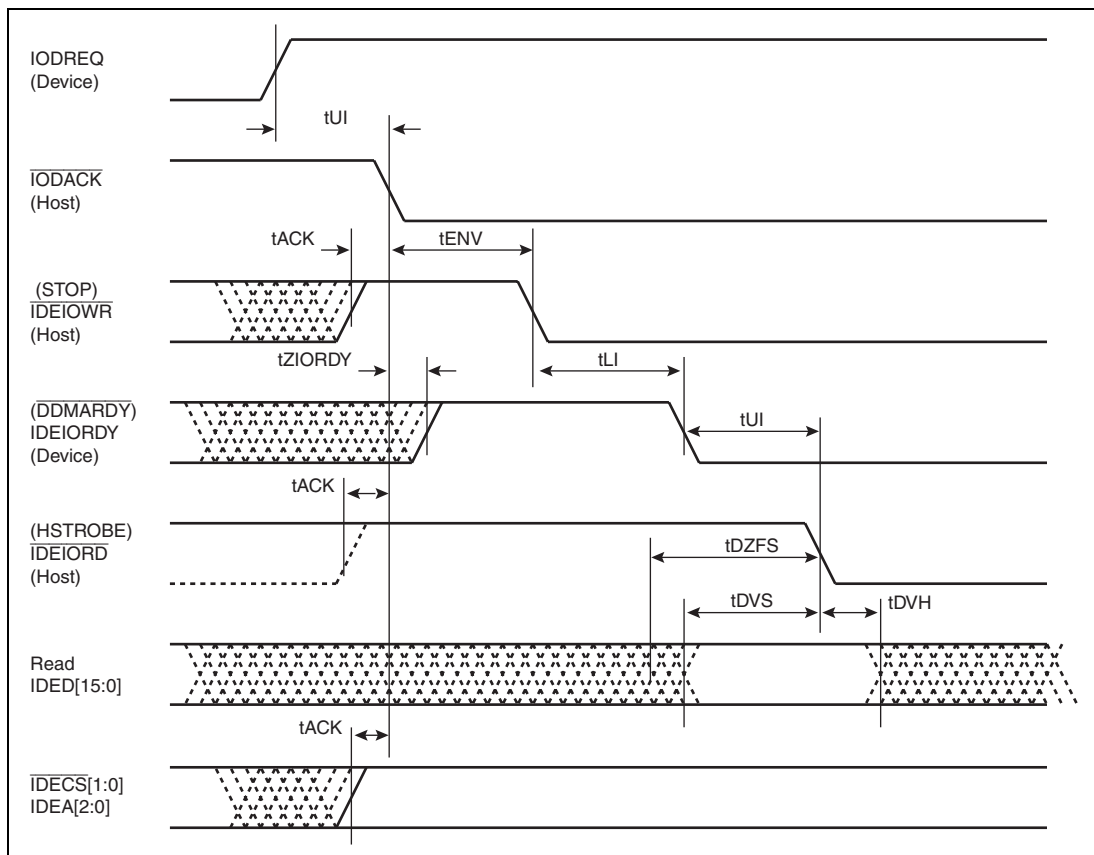


Figure 33.48 Ultra-DMA Data Out-burst Start



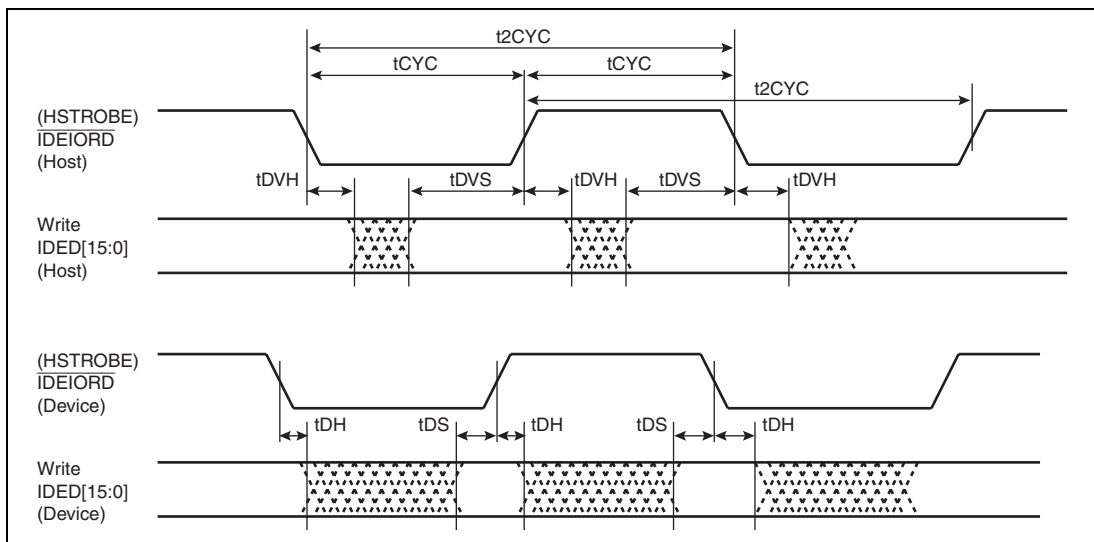


Figure 33.49 Ultra-DMA Data Out-burst

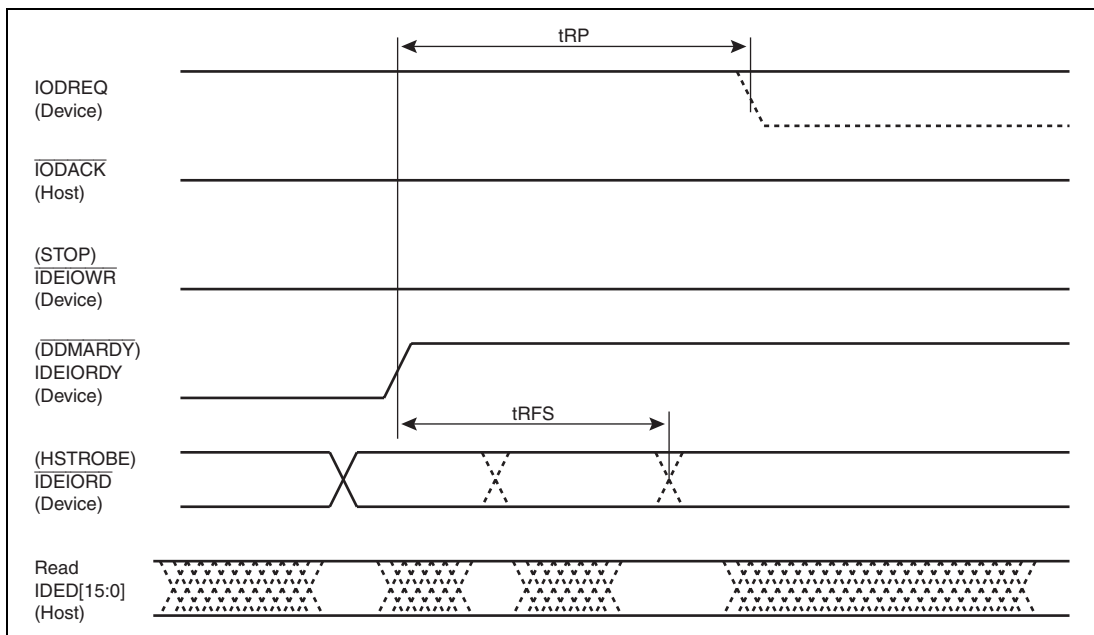


Figure 33.50 Ultra-DMA Data Out-burst from Device Pause

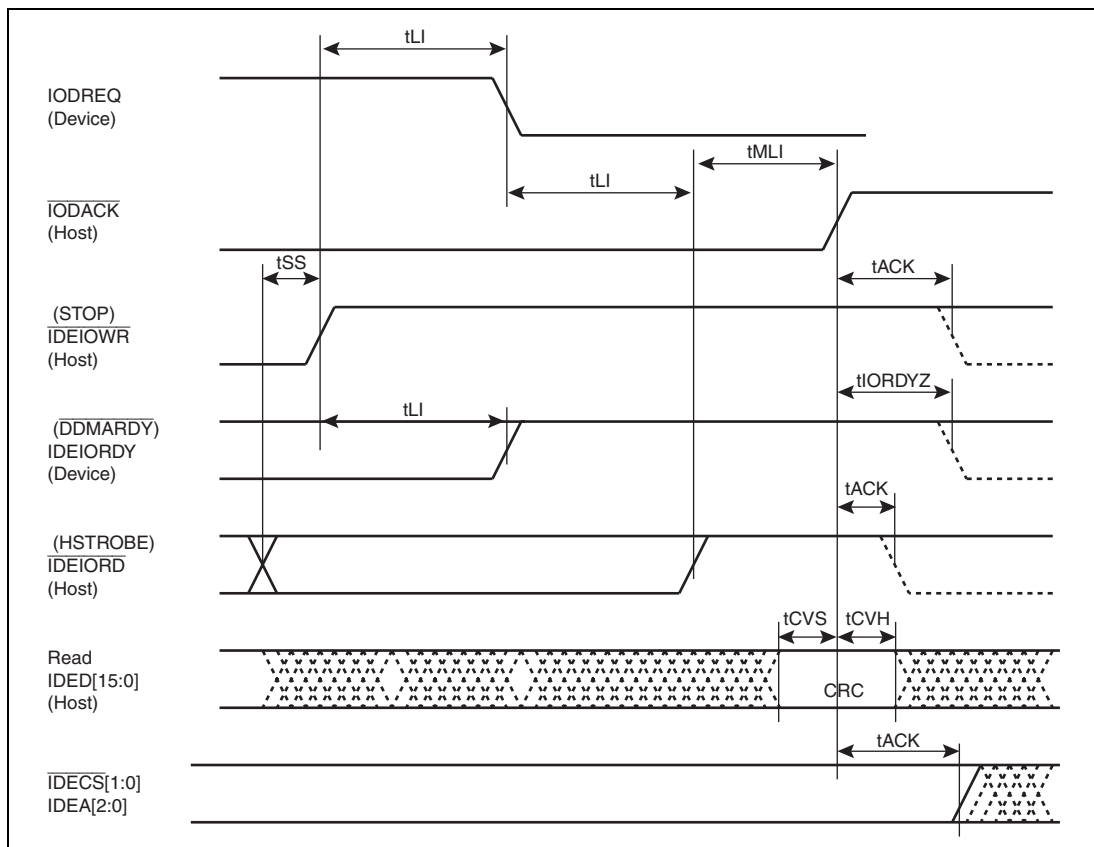
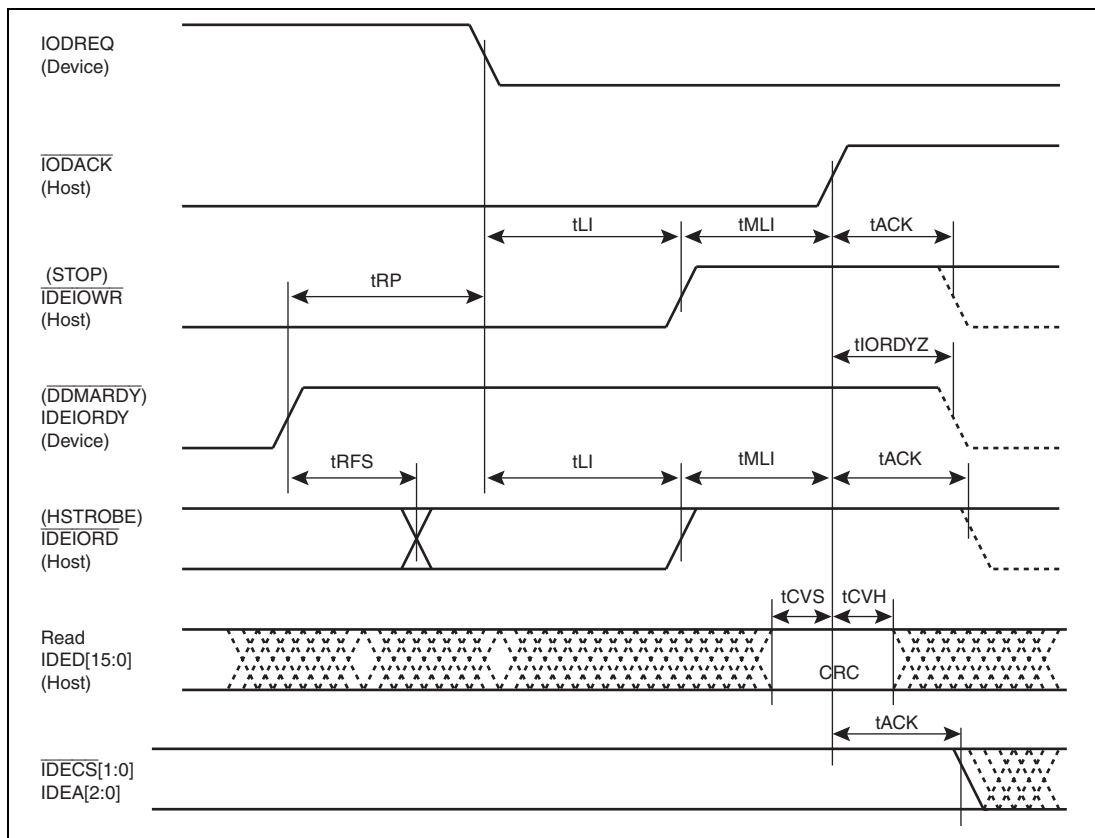
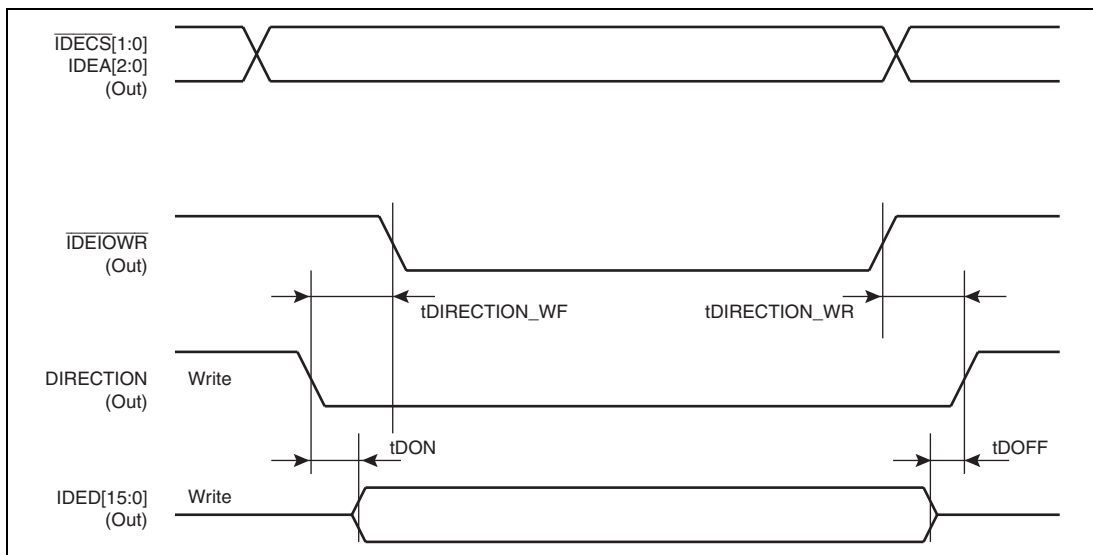


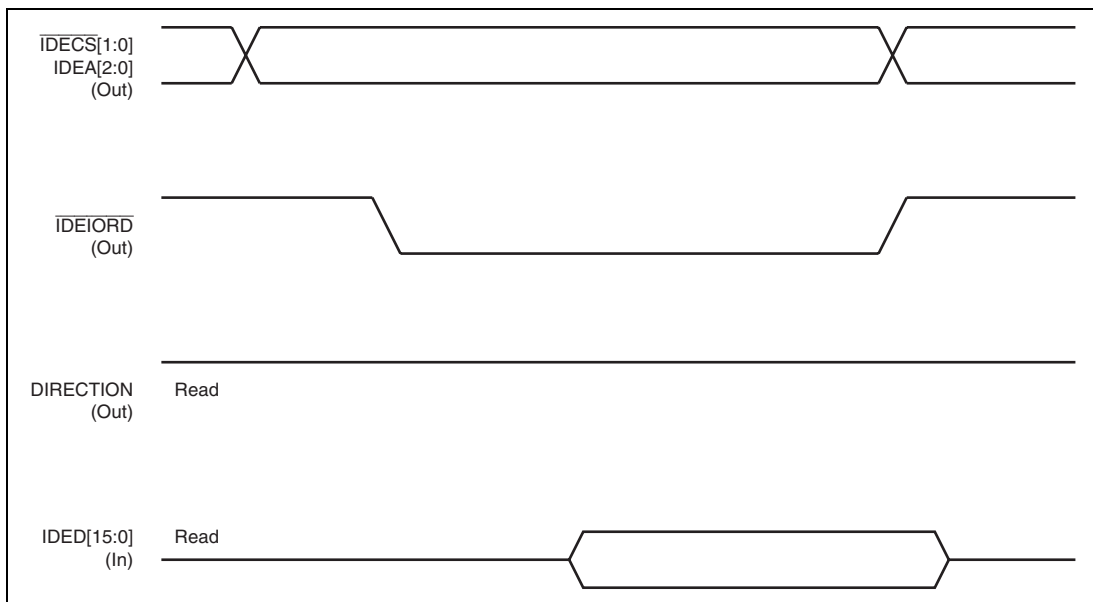
Figure 33.51 End of Ultra-DMA Data Out-burst from Host



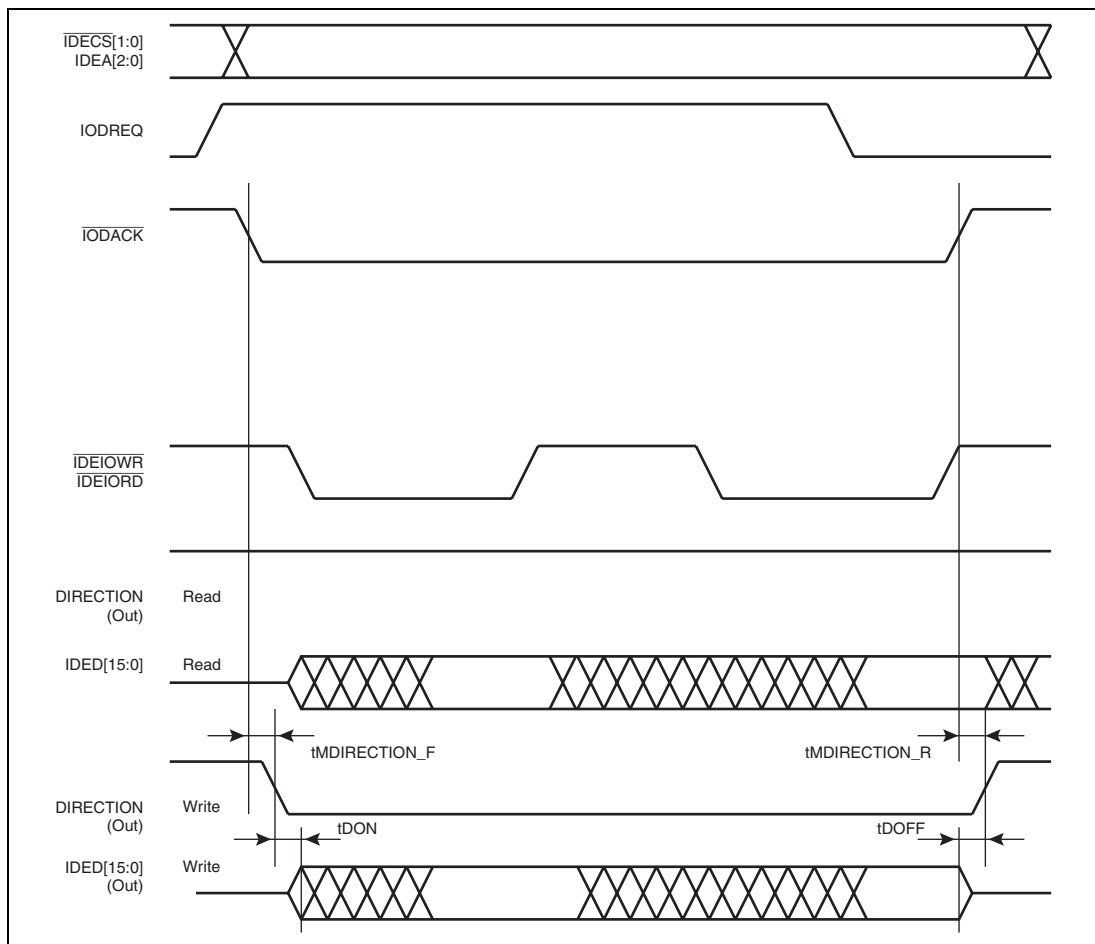
**Figure 33.52 End of Ultra-DMA Data Out-burst from Device**



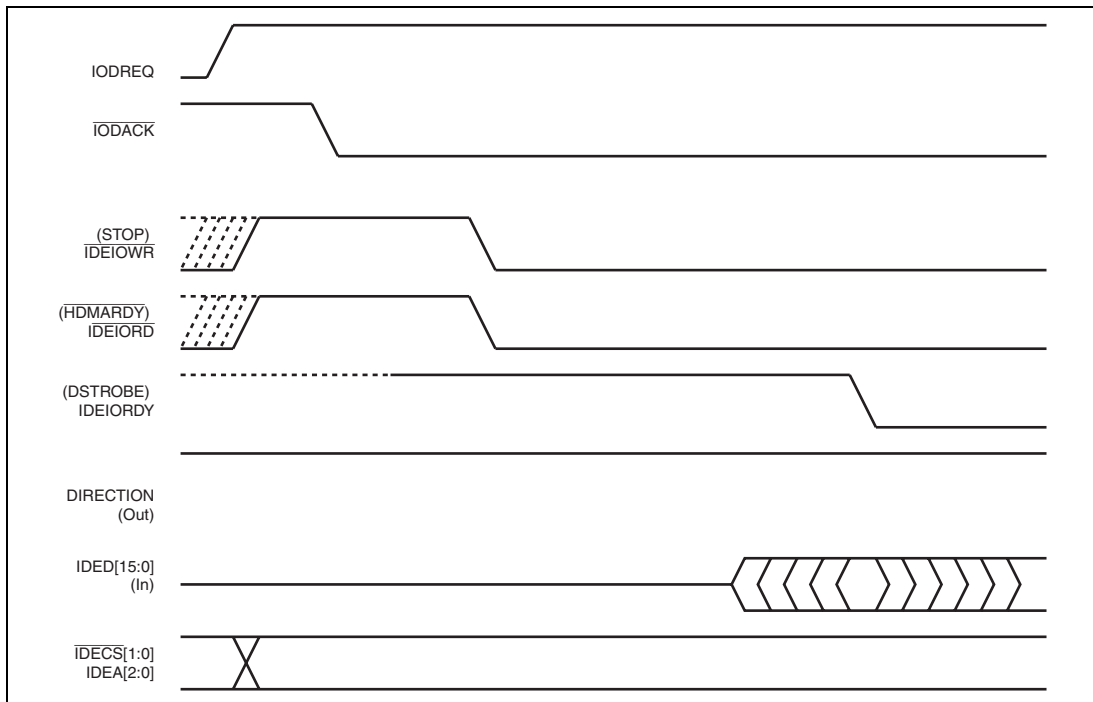
**Figure 33.53 PIO Data Transmission (DIRECTIO) to Device**



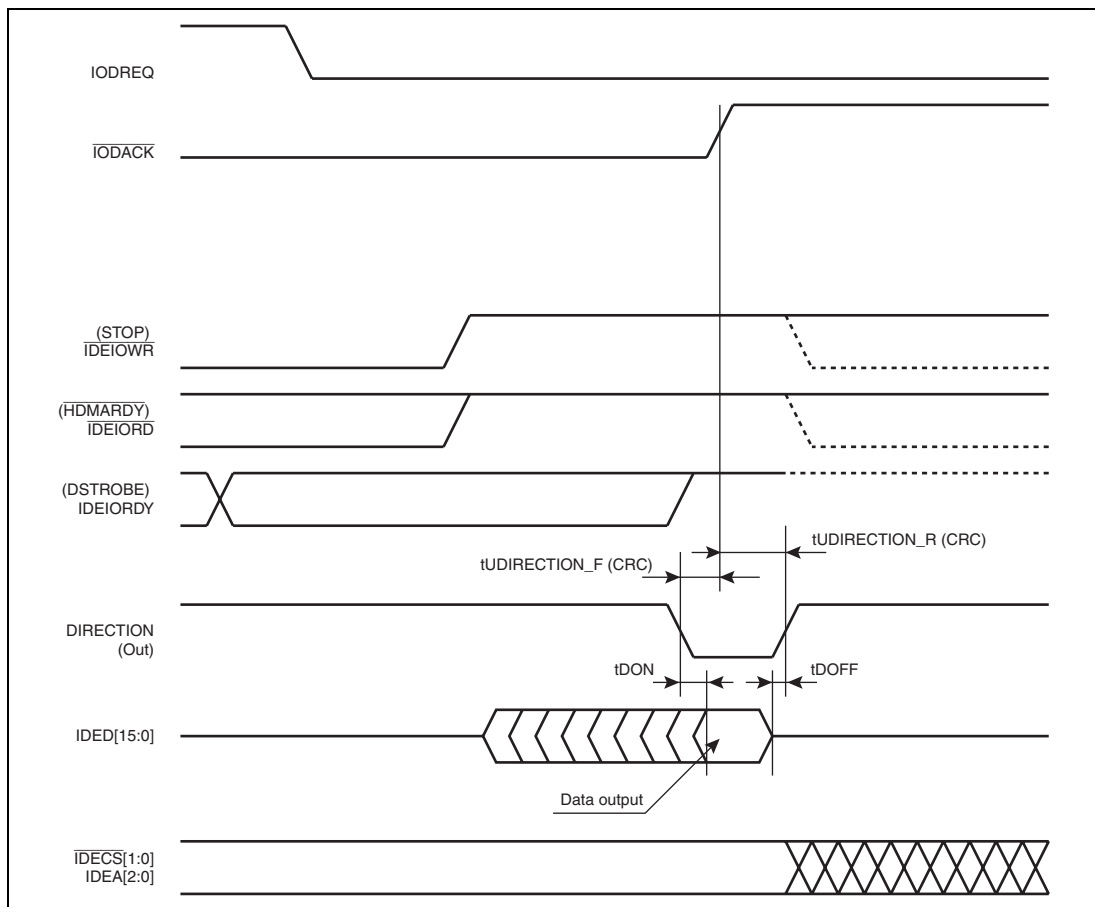
**Figure 33.54 PIO Data Transmission (DIRECTIO) from Device**



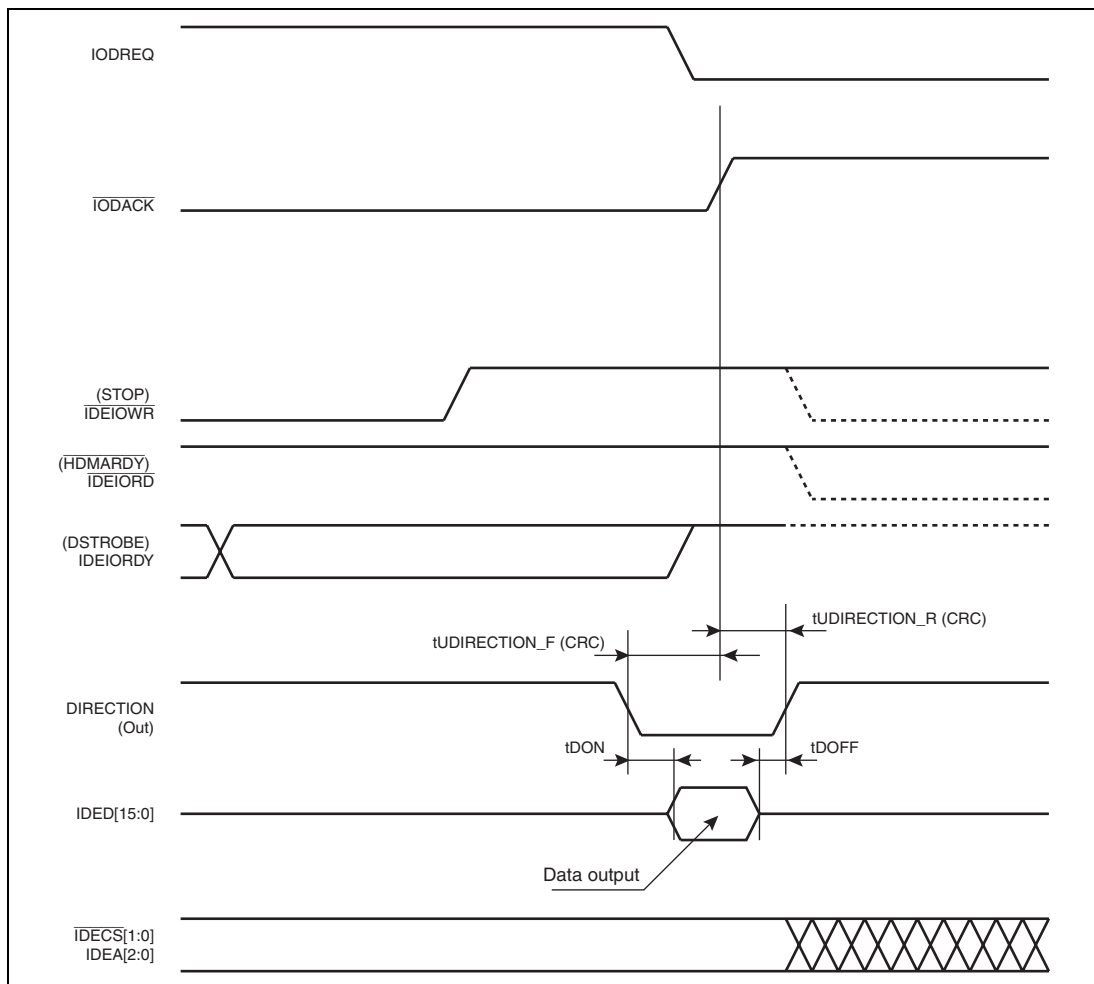
**Figure 33.55 Multiword DMA Transmission (DIRECTION)**



**Figure 33.56 Ultra-DMA Transmission Data In-burst Start(DIRECTION)**

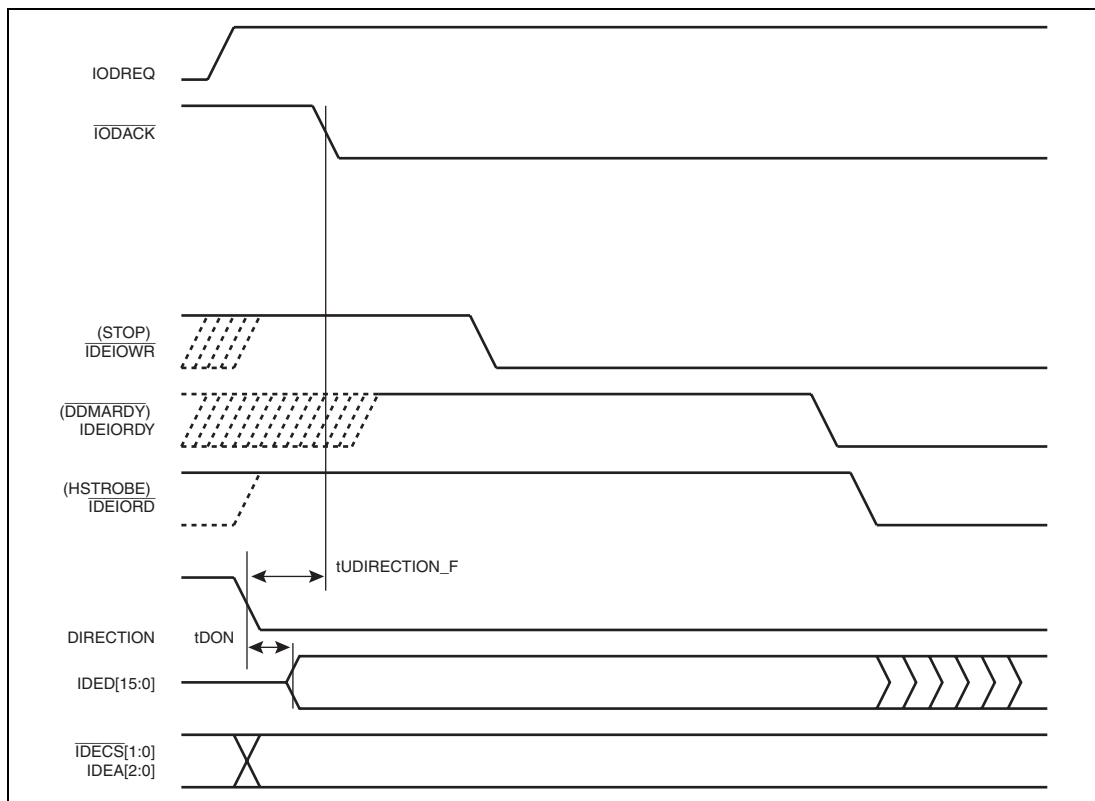


**Figure 33.57 End of Ultra-DMA Transmission Data In-burst from Device (DIRECTION)**

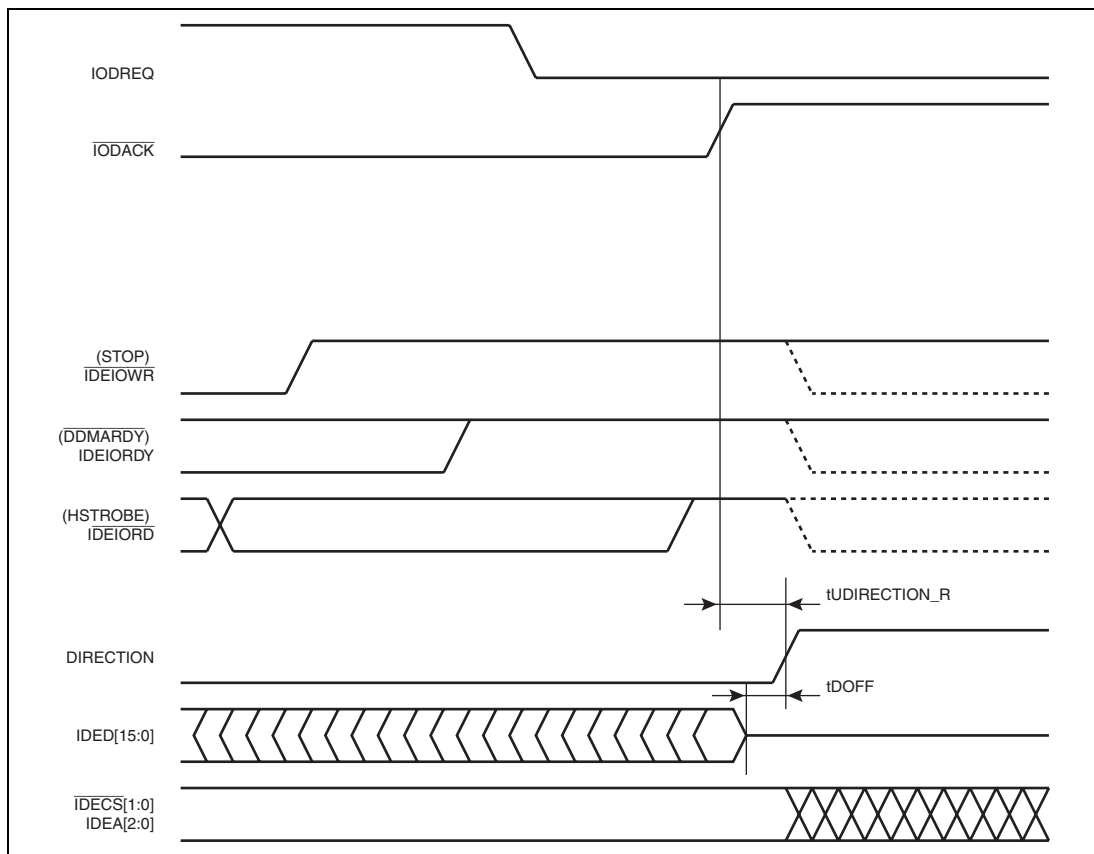


**Figure 33.58 End of Ultra-DMA Transmission Data In-burst from Host (DIRECTION)**

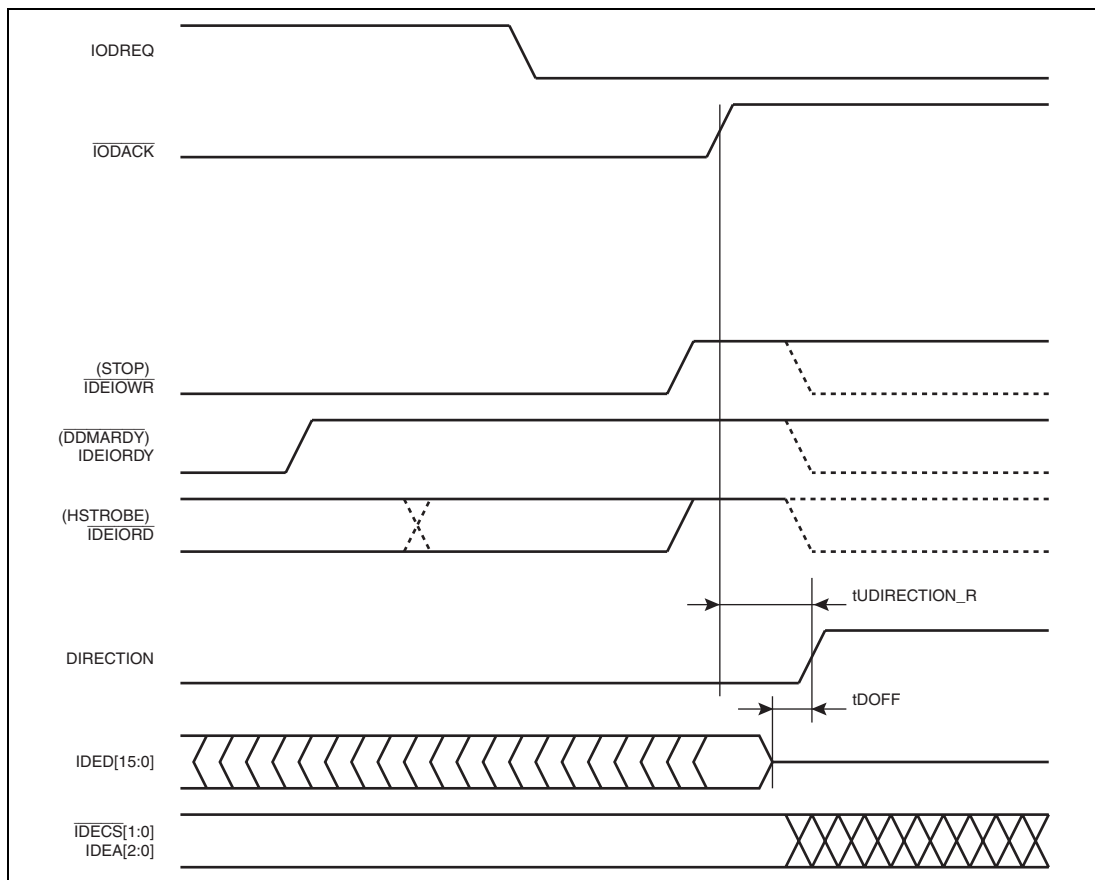




**Figure 33.59 Ultra-DMA Transmission Data Out-burst Start (DIRECTION)**



**Figure 33.60 End of Ultra-DMA Transmission Data Out-burst from Host (DIRECTION)**



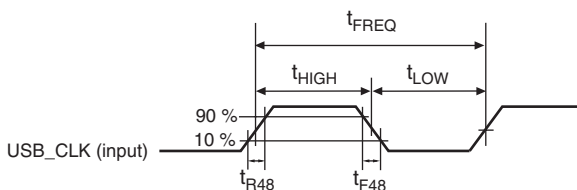
**Figure 33.61 End of Ultra-DMA Transmission Data Out-burst from Device (DIRECTION)**

### 33.4.11 USB Module Signal Timing

**Table 33.27 USB Module Clock Timing**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Max.	Unit	Figure
USB_CLK external input clock frequency (48 MHz)	$t_{\text{FREQ}}$	47.9	48.1	MHz	33.62
Clock rising time	$t_{\text{R48}}$	—	2	ns	
Clock falling time	$t_{\text{F48}}$	—	2	ns	
Duty ( $t_{\text{HIGH}}/t_{\text{LOW}}$ )	$t_{\text{DUTY}}$	90	110	%	


**Figure 33.62 USB Clock Timing**
**Table 33.28 USB Electrical Characteristics (for Full Speed)**

Item	Symbol	Min.	Max.	Unit	Condition* <sup>1</sup>
Transition time (rising)* <sup>2</sup>	$t_{\text{R}}$	4	20	ns	CL= 50 pF
Transition time (falling)* <sup>2</sup>	$t_{\text{F}}$	4	20	ns	CL= 50 pF
Rising/ falling time matching	$t_{\text{RFM}}$	90	111	%	(TR/TF)
Output signal crossover voltage	$V_{\text{CRS}}$	1.3	2.0	V	—

Notes: The values are measured under the condition that the capacitor for edge control  $C_{\text{EDGE}} = 47\text{pF}$  is connected to the serial resistor  $R_s = 45\Omega$ .

1. The condition is that CL = 50pF unless otherwise specified.
2. Within 10 to 90% of the signal voltage.

**Table 33.29 USB Electrical Characteristics (for Low Speed)**

Item	Symbol	Min.	Max.	Unit	Condition
Transition time (rising)	$t_R$	75	300	ns	
Transition time (falling)	$t_F$	75	300	ns	
Rising/ falling time matching	$t_{RFM}$	80	125	%	(TR/TF)
Output signal crossover voltage	$V_{CRS}$	1.3	2.0	V	—

Notes: The values are measured under the condition that the capacitor for edge control  $C_{EDGE} = 47\text{pF}$  is connected to the serial resistor  $R_s = 22\Omega$ . GPIO Signal Timing.

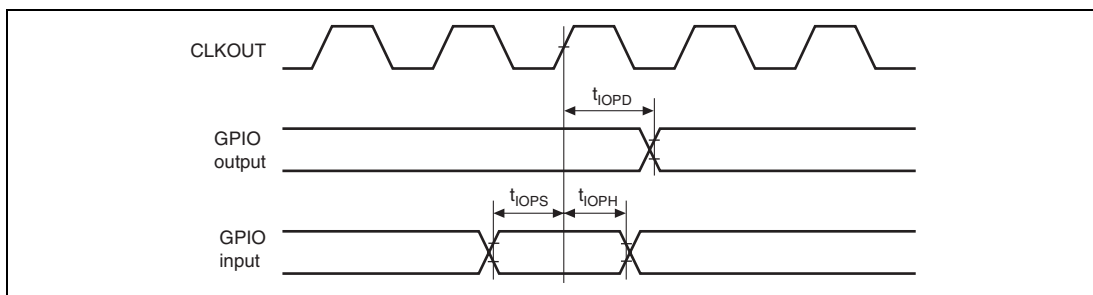
\* Within 10 to 90% of signal voltage.

### 33.4.12 GPIO Signal Timing

**Table 33.30 GPIO Signal Timing**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Max.	Unit	Figure
GPIO output delay time	$t_{IOPK}$	—	17	ns	33.63
GPIO input setup time	$t_{IOPS}$	17	—	ns	
GPIO input hold time	$t_{IOPH}$	$T_{CLKOUT\text{cyc}}$	—	ns	

**Figure 33.63 GPIO Timing**

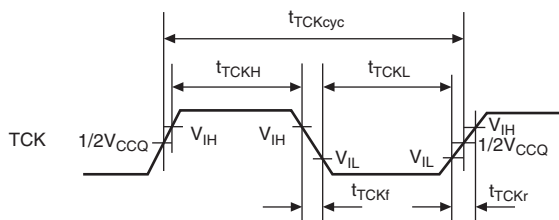
### 33.4.13 H-UDI Module Signal Timing

**Table 33.31 H-UDI Module Signal Timing**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

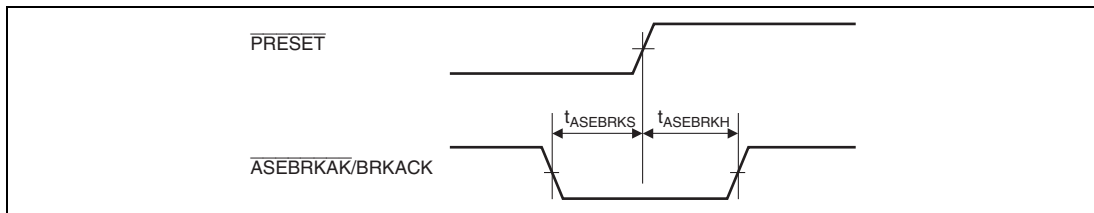
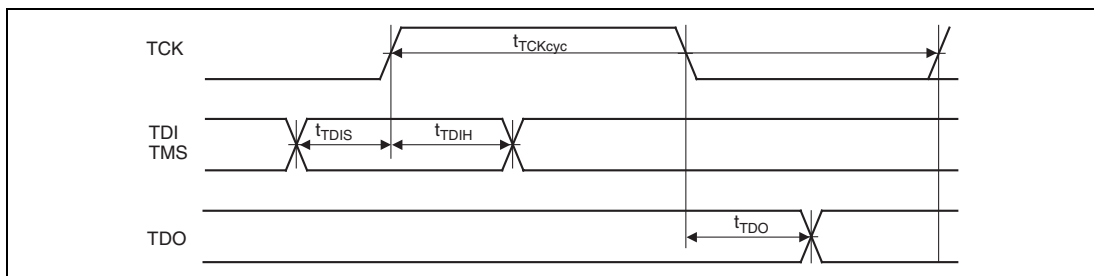
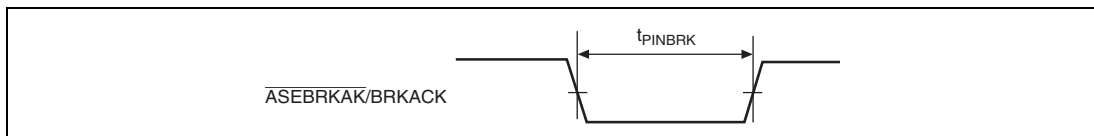
Module	Item	Symbol	Min.	Max.	Unit	Figure
H-UDI	Input clock cycle	$t_{\text{TCKcyc}}$	50	—	ns	33.64, 33.66
	Input clock pulse width (High)	$t_{\text{TCKH}}$	15	—	ns	33.64
	Input clock pulse width (Low)	$t_{\text{TCKL}}$	15	—	ns	
	Input clock rise time	$t_{\text{TCKr}}$	—	10	ns	
	Input clock fall time	$t_{\text{TCKf}}$	—	10	ns	
	ASEBRKAK/BRKACK setup time	$t_{\text{ASEBRKS}}$	10	—	$t_{\text{cyc}}$	33.65
	ASEBRKAK/BRKACK hold time	$t_{\text{ASEBRKH}}$	10	—	$t_{\text{cyc}}$	
	TDI/TMS setup time	$t_{\text{TDIS}}$	15	—	ns	33.66
	TDI/TMS hold time	$t_{\text{TDIH}}$	15	—	ns	
	TDO data delay time	$t_{\text{TDO}}$	0	15	ns	
	ASEBRKAK/BRKACK pulse width	$t_{\text{PINBRK}}$	2	—	$t_{\text{Pcyc}}$	33.67

- Notes: 1.  $t_{\text{cyc}}$  indicates the CLKOUT clock cycle.  
 2.  $t_{\text{pcyc}}$  indicates the peripheral clock (Pck) cycle.



Note: When the clock is input on the TCK pin.

**Figure 33.64 TCK Input Timing**

Figure 33.65 **PRESET Hold Timing**Figure 33.66 **H-UDI Data Transmission Timing**Figure 33.67 **ASEBRKAK/BRKACK Pin Break Timing**

### 33.4.14 EtherC Module Signal Timing

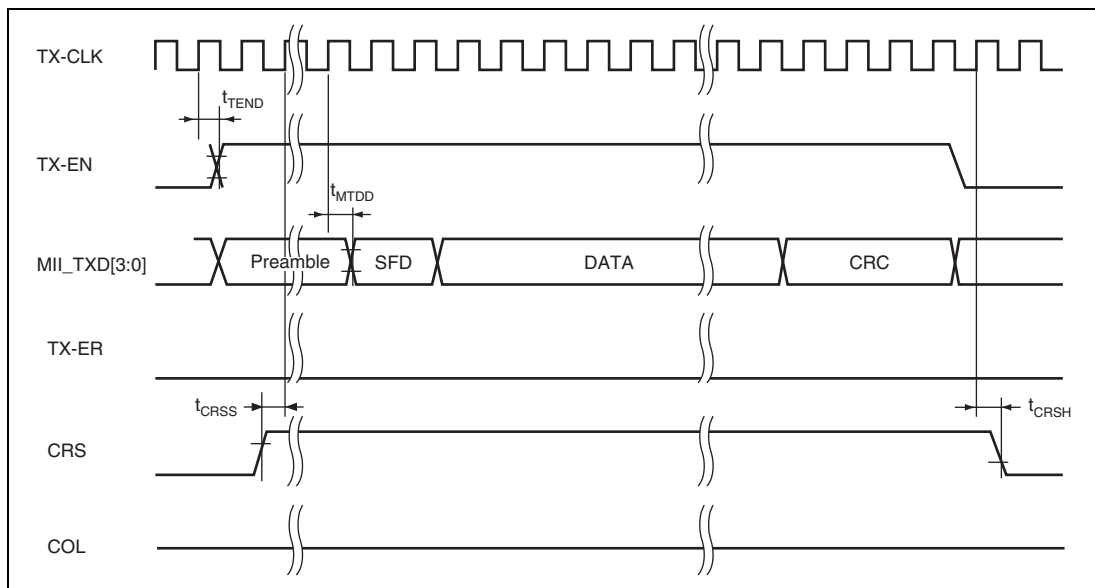
**Table 33.32 Ether Net Controller Timing**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^{\circ}\text{C}$ ,  $-40$  to  $85^{\circ}\text{C}$

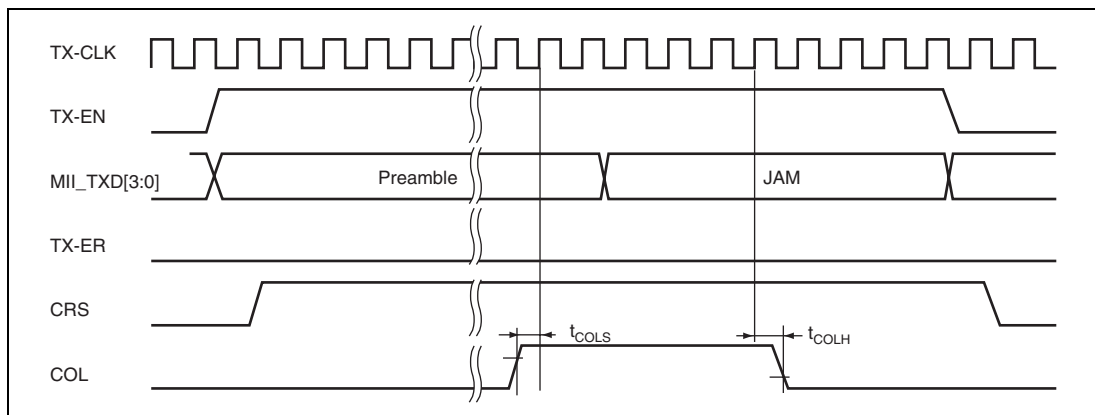
Item	Symbol	Min.	Max.	Unit	Figure
TX-CLK cycle time	$t_{\text{Tcyc}}$	40	—	ns	33.68
TX-EN output delay time	$t_{\text{TENd}}$	1	20	ns	
MII_TXD[3:0] output delay time	$t_{\text{MTDd}}$	1	20	ns	
CRS setup time	$t_{\text{CRSs}}$	10	—	ns	33.69
CRS hold time	$t_{\text{CRSh}}$	10	—	ns	
COL setup time	$t_{\text{COLs}}$	10	—	ns	
COL hold time	$t_{\text{COLh}}$	10	—	ns	33.70
RX-CLK cycle time	$t_{\text{Rcyc}}$	40	—	ns	
RX-DV setup time	$t_{\text{RDVs}}$	10	—	ns	
RX-DV hold time	$t_{\text{RDVh}}$	10	—	ns	33.71
MII_RXD[3:0] setup time	$t_{\text{MRDs}}$	10	—	ns	
MII_RXD[3:0] hold time	$t_{\text{MRDh}}$	10	—	ns	
RX-ER setup time	$t_{\text{RERs}}$	10	—	ns	33.72
RX-ER hold time	$t_{\text{RERh}}$	10	—	ns	
MDIO setup time	$t_{\text{MDIOs}}$	10	—	ns	
MDIO hold time	$t_{\text{MDIOh}}$	10	—	ns	33.73
MDIO output data hold time*	$t_{\text{MDIOdh}}$	5	18	ns	
WOL output delay time	$t_{\text{WOLd}}$	1	20	ns	
EXOUT output delay time	$t_{\text{EXOUTd}}$	1	20	ns	33.75

Note: \* Operate the internal register (PIR) in PHY block to meet the requirement of this specification.

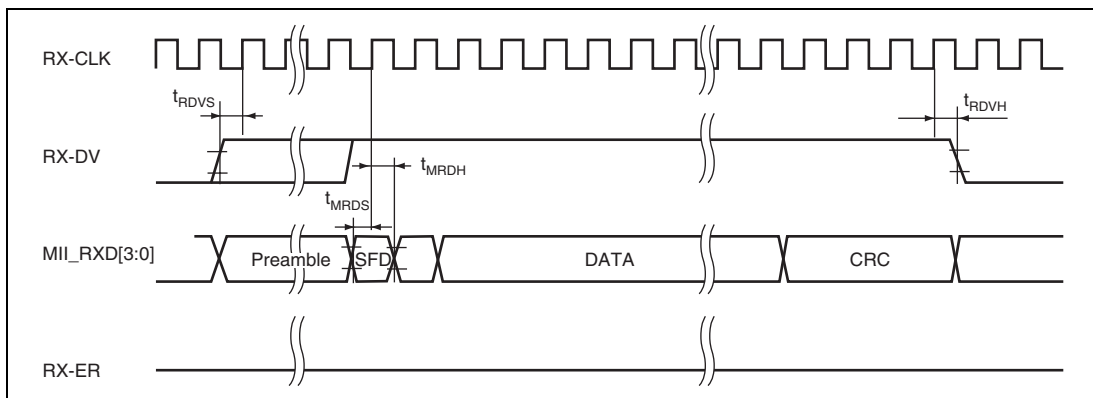




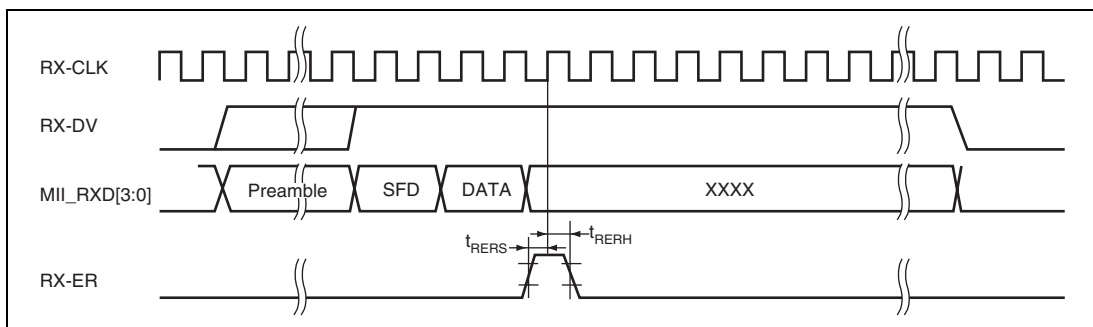
**Figure 33.68 MII Transmission Timing (during Normal Operation)**



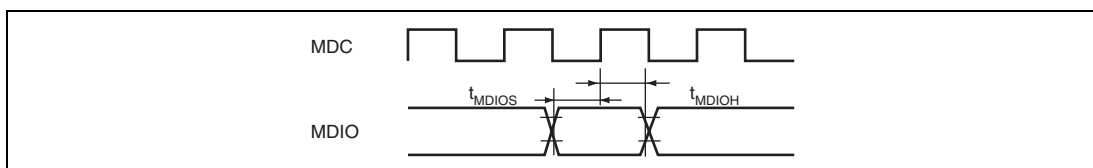
**Figure 33.69 MII Transmission Timing (in the Event of a Collision)**



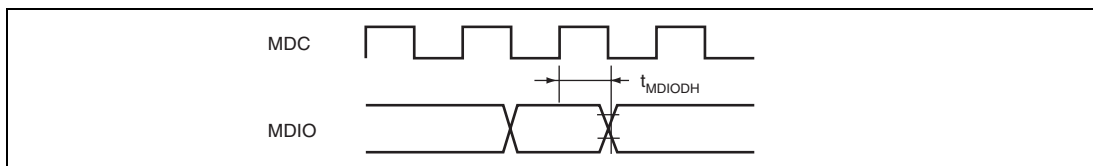
**Figure 33.70 MII Receive Timing (during Normal Operation)**



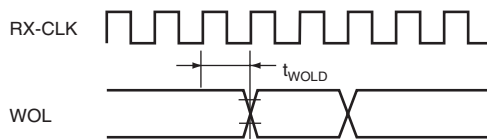
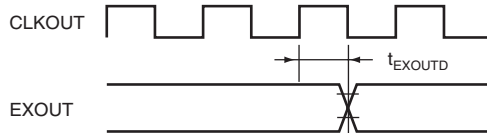
**Figure 33.71 MII Receive Timing (in the Event of a Collision)**



**Figure 33.72 MDIO Input Timing**



**Figure 33.73 MDIO Output Timing**

**Figure 33.74 WOL Output Timing****Figure 33.75 EXOUT Output Timing**

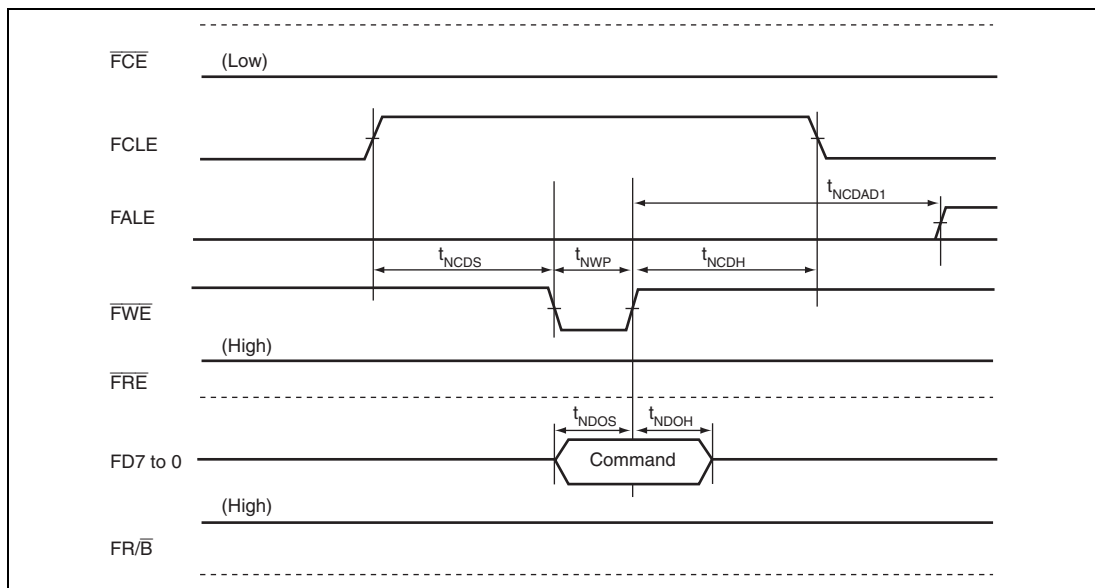
### 33.4.15 FLCTL Module Signal Timing

**Table 33.33 NAND Flush Memory Interface Timing**

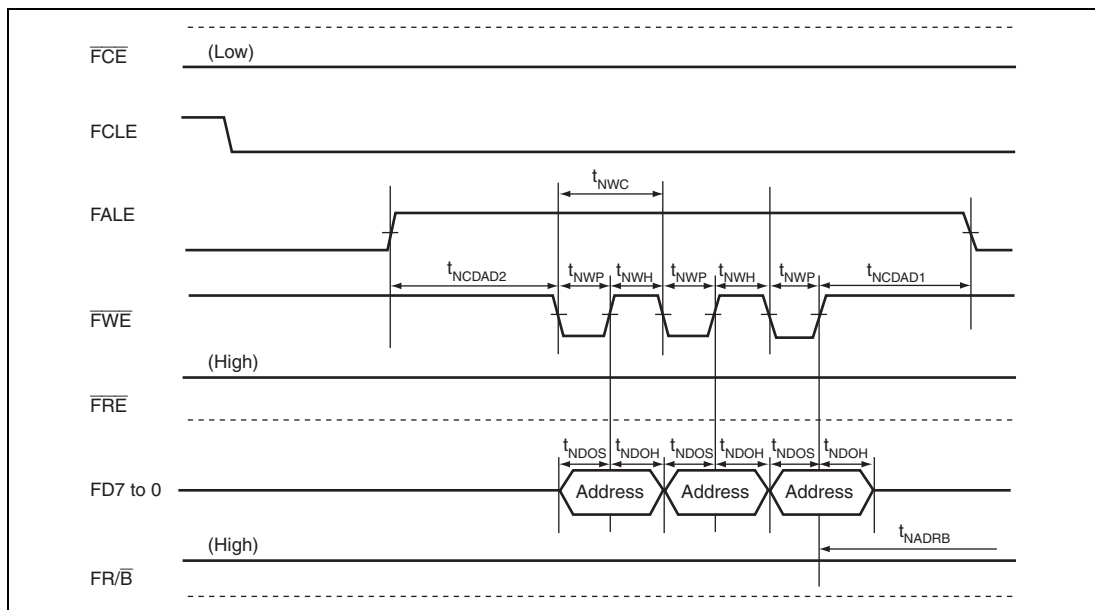
Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^{\circ}\text{C}$ ,  $-40$  to  $85^{\circ}\text{C}$

Item	Symbol	Min.	Max.	Unit	Figure
Command output setup time	$t_{\text{NCDS}}$	$2 \times t_{\text{fcyc}} - 10$	—	ns	33.76, 33.80
Command output hold time	$t_{\text{NCDH}}$	$1.5 \times t_{\text{fcyc}} - 5$	—	ns	
Data output setup time	$t_{\text{NDOS}}$	$0.5 \times t_{\text{fcyc}} - 5$	—	ns	33.76, 33.77,
Data output hold time	$t_{\text{NDOH}}$	$0.5 \times t_{\text{fcyc}} - 10$	—	ns	33.79, 33.80
Command address transmission time 1	$t_{\text{NCAD1}}$	$1.5 \times t_{\text{fcyc}} - 10$	—	ns	33.76, 33.77
Command address transmission time 2	$t_{\text{NCAD2}}$	$2 \times t_{\text{fcyc}} - 10$	—	ns	33.77
FEW cycle time	$t_{\text{NWC}}$	$t_{\text{fcyc}} - 5$	—	ns	33.77, 33.79
FEW low pulse width	$t_{\text{NWP}}$	$0.5 \times t_{\text{fcyc}} - 5$	—	ns	33.76, 33.77,
					33.79, 33.80
FEW high pulse width	$t_{\text{NWH}}$	$0.5 \times t_{\text{fcyc}} - 5$	—	ns	33.77, 33.79
Address ready/busy transmission time	$t_{\text{NADRB}}$	—	$32 \times t_{\text{pcyc}}$	ns	33.77, 33.78
Ready/busy data read transmission time 1	$t_{\text{NRBDR1}}$	$1.5 \times t_{\text{fcyc}}$	—	ns	33.78
Ready/busy data read transmission time 2	$t_{\text{NRBDR2}}$	$32 \times t_{\text{pcyc}}$	—	ns	
FRE cycle time	$t_{\text{NSCC}}$	$t_{\text{fcyc}} - 5$	—	ns	
FRE low pulse width	$t_{\text{NSP}}$	$0.5 \times t_{\text{fcyc}} - 5$	—	ns	33.78, 33.80
FRE high pulse width	$t_{\text{NSPH}}$	$0.5 \times t_{\text{fcyc}} - 5$	—	ns	33.78
Read data setup time	$t_{\text{NRDS}}$	14	—	ns	33.78, 33.80
Read data hold time	$t_{\text{NRDH}}$	0	—	ns	
Data write setup time time	$t_{\text{NDWS}}$	$32 \times t_{\text{pcyc}}$	—	ns	33.79
Command status read transmission time	$t_{\text{NCDSR}}$	$4 \times t_{\text{fcyc}}$	—	ns	33.80
Command output off status read transmission time	$t_{\text{NCDFSR}}$	$3.5 \times t_{\text{fcyc}}$	—	ns	
Status read setup time	$t_{\text{NSTS}}$	$2.5 \times t_{\text{fcyc}}$	—	ns	

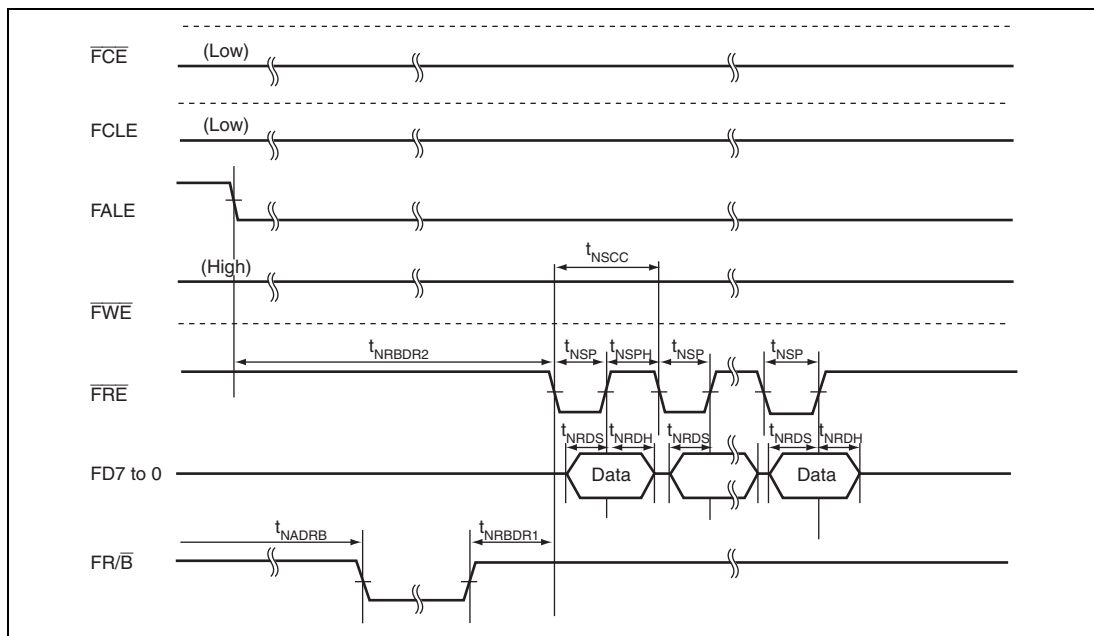
Note:  $t_{\text{cyc}}$  indicates one cycle time of the FLCTL clock.



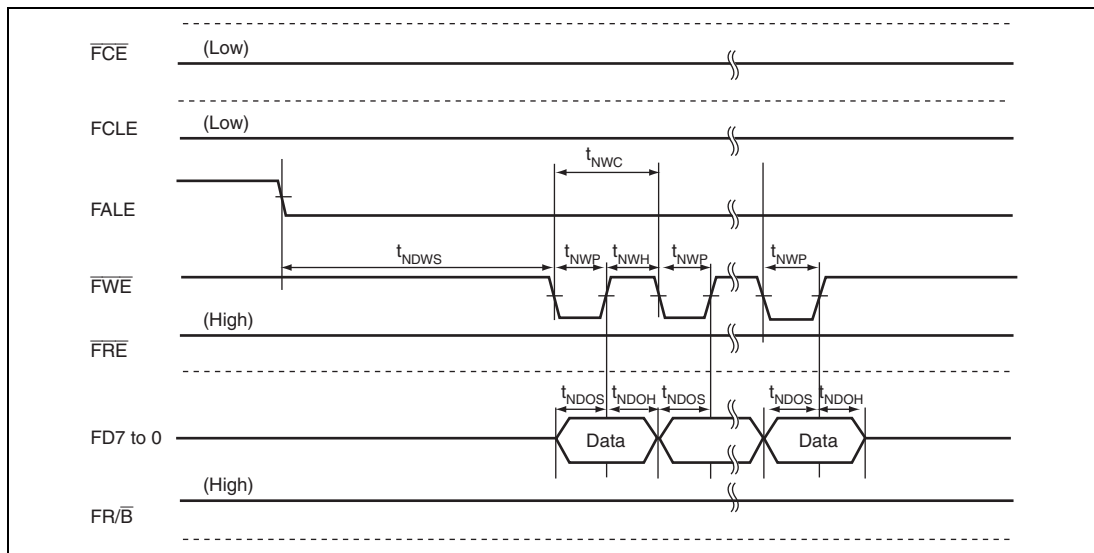
**Figure 33.76 NAND Flash Memory Command Issue Timing**



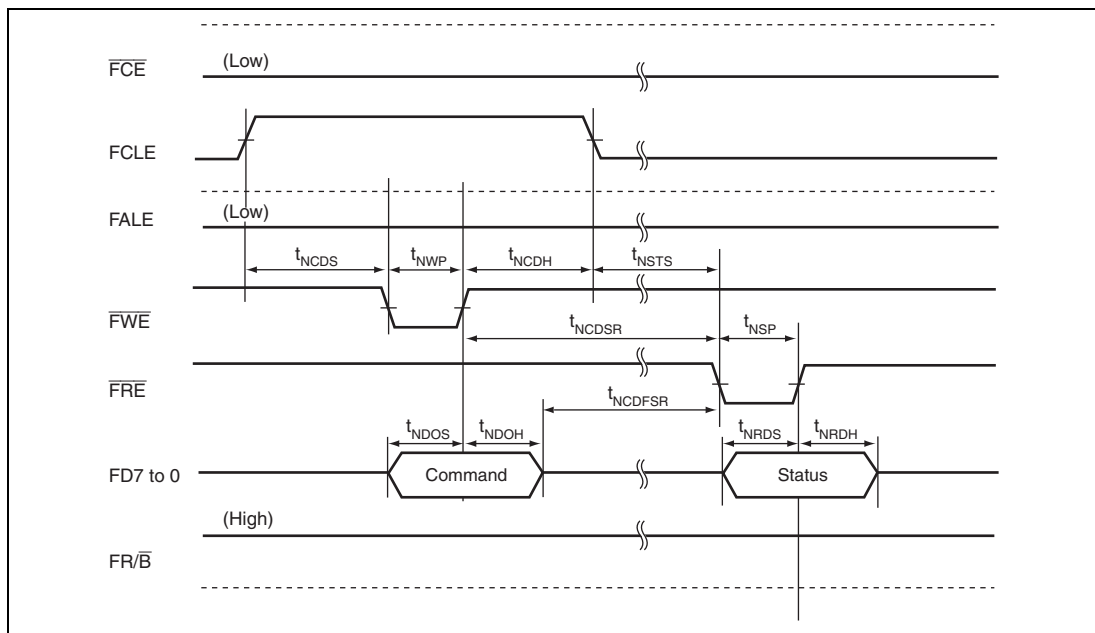
**Figure 33.77 NAND Flash Memory Address Issue Timing**



**Figure 33.78 NAND Flash Memory Data Read Timing**



**Figure 33.79 NAND Flash Memory Data Write Timing**



**Figure 33.80 NAND Flash Memory Status Read Timing**

### 33.4.16 LCDC Module Signal Timing

**Table 33.34 LCDC Module Signal timing**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Max.	Unit	Figure
LCD_CLK input clock frequency	$t_{\text{FREQ}}$	—	54	MHz	
LCD_CLK input clock rise time	$t_r$	—	3	ns	
LCD_CLK input clock fall time	$t_f$	—	3	ns	
LCD_CLK input clock duty	$t_{\text{DUTY}}$	90	110	%	
Clock (LCD_CL2) cycle time	$t_{\text{CC}}$	25	—	ns	33.81
Clock (LCD_CL2) high pulse width	$t_{\text{CHW}}$	7	—	ns	
Clock (LCD_CL2) low pulse width	$t_{\text{CLW}}$	7	—	ns	
Clock (LCD_CL2) transition time (rise/fall)	$t_{\text{CT}}$	—	3	ns	
Data (LCD_DATA) delay time	$t_{\text{DDdo}}$	-3.5	3	ns	
Display permission (LCK_CL1) delay time	$t_{\text{IDdo}}$	-3.5	3	ns	
Horizontal synchronized signal (LCD_CL1) delay time	$t_{\text{HDdo}}$	-3.5	3	ns	
Vertical synchronized signal (LCD_FLM) delay time	$t_{\text{VDdo}}$	-3.5	3	ns	



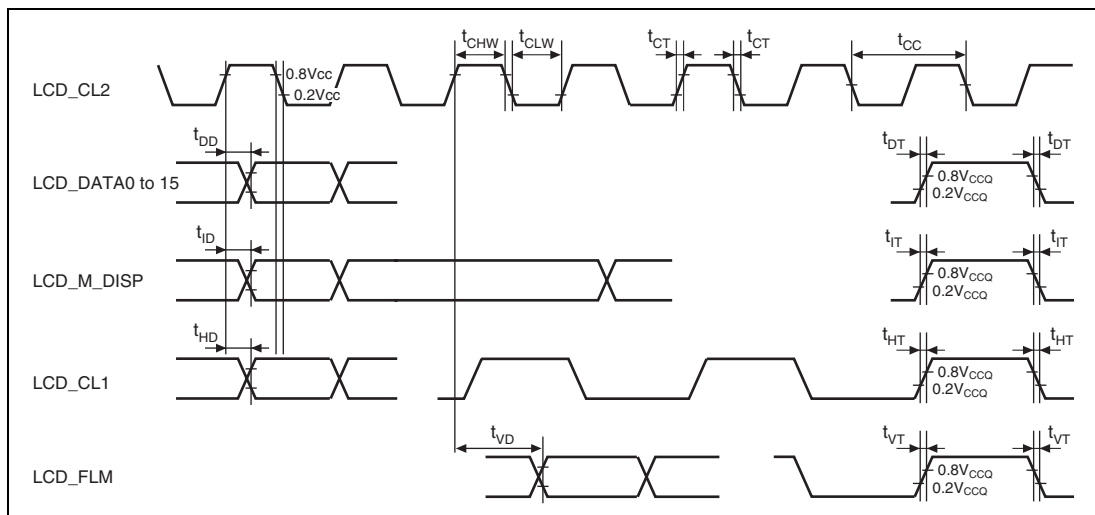


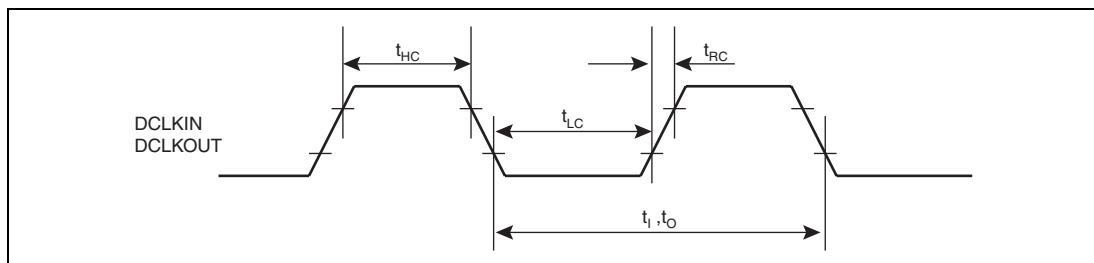
Figure 33.81 LCDC Module Signal Timing

### 33.4.17 VDC2 Module Signal Timing

**Table 33.35 VDC2 Module Signal Timing**

Conditions: 3.3-V power supply= 3.0 to 3.6 V, 1.2-V power supply= 1.15 to 1.35,  
 $T_a = -20$  to  $85^\circ\text{C}$ ,  $-40$  to  $85^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Note	Figure
Output clock frequency	$t_o$	18.5	—	158	ns	Output	33.82
Input clock frequency	$t_i$	18.5	—	158	ns	Input	
Clock high	$t_{HC}$	6	—	—	ns	Input/ output	
Clock low	$t_{LC}$	6	—	—	ns		
Clock rise time	$t_{RC}$	—	—	3	ns	Output (100 pF)	
Delay	Internal synchronization mode $t_{DTRI}$	—	—	5	ns	Transmit	33.83, 33.84
	External synchronization mode $t_{DTRO}$	—	—	20	ns		
Setup time	$t_{SR}$	5	—	—	ns	Receive	33.85, 33.86
Hold time	$t_{HTR}$	5	—	—	ns	Receive	33.85 to 33.86


**Figure 33.82 Clock Input/Output Timing**

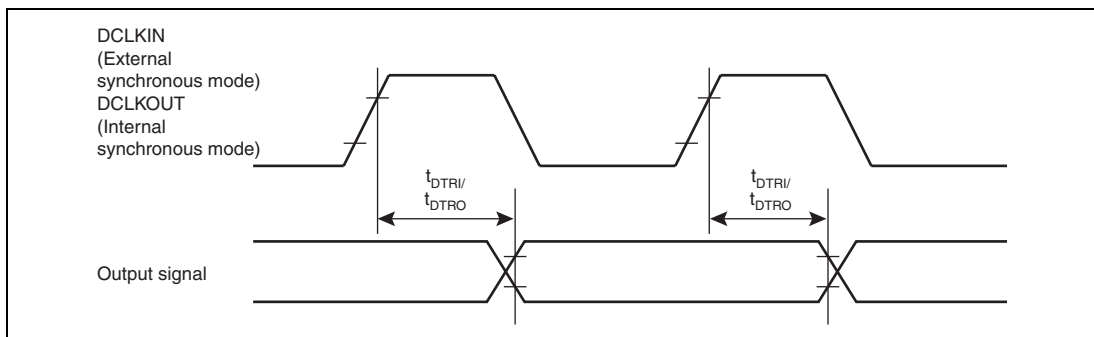


Figure 33.83 VDC2 Transmission Timing (1)

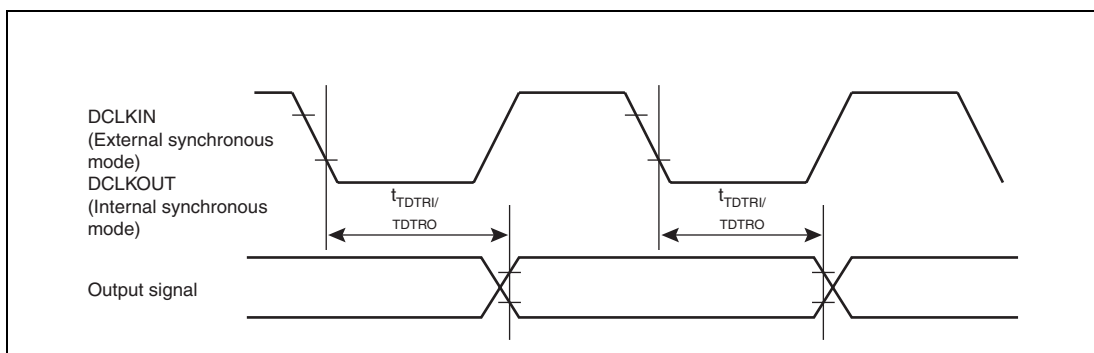


Figure 33.84 VDC2 Transmission Timing (2)

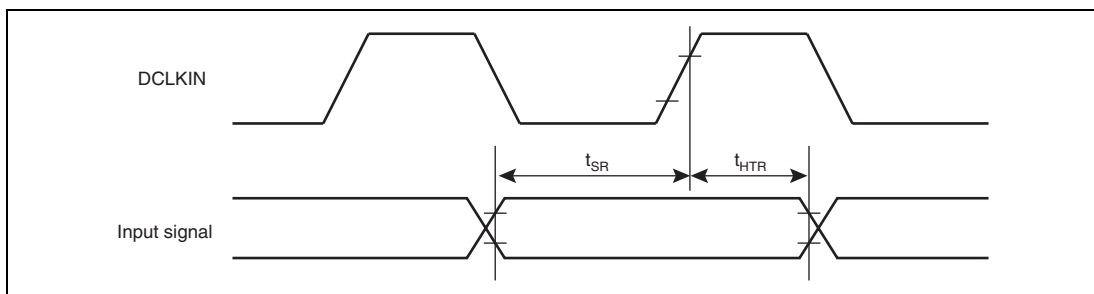
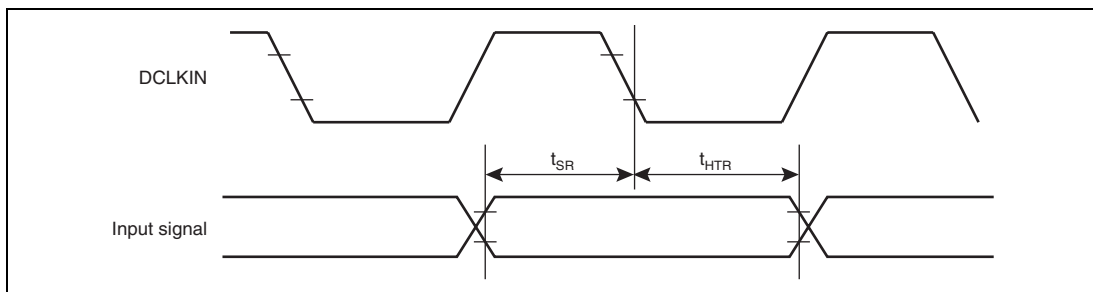


Figure 33.85 VDC2 Reception Timing (1)



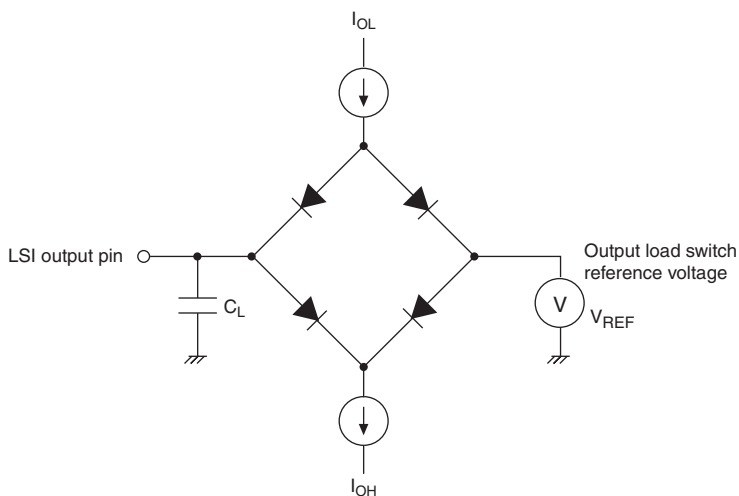
**Figure 33.86 VDC2 Reception Timing (2)**

### 33.5 AC Characteristics Measurement Conditions

AC characteristics measurement conditions as follows.

- I/O signal reference level:  $\frac{V^*}{2}$
- Input pulse level:  $V_{SSQ}$  to  $V^*$
- Input rise and fall times: 1 ns

Note: \*  $V:V_{DDQ}$  ( $V_{DDQ} = 3.0$  to  $3.6$  V)



- Note:
1.  $C_L$  is the total value that includes the capacitance of the measuring tools.  
The individual pins are set as 30 pF.
  2.  $I_{OL} = 3$  mA (IIC pin)  
2 mA (other than IIC pin)  
 $I_{OH} = -2$  mA

**Figure 33.87 Output Load Circuit**



# Appendix

## A. CPU Operation Mode Register (CPUOPM)

The CPUOPM is used to control the CPU operation mode. This register can be read from or written to the address H'FF2F0000 in P4 area or H'1F2F0000 in 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	1	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 6	—	H'000000F	R	Reserved The write value must be the initial value.
5	RABD	1	R/W	Speculative execution bit for subroutine return 0: Instruction fetch for subroutine return is issued speculatively. When this bit is set to 0, refer to appendix C, Speculative Execution for Subroutine Return. 1: Instruction fetch for subroutine return is not issued speculatively.
4	—	0	R	Reserved The write value must be the initial value.
3	INTMU	0	R/W	Interrupt mode switch bit 0: SR.IMASK is not changed when an interrupt is accepted. 1: SR.IMASK is changed to the accepted interrupt level.
2 to 0	—	All 0	R	Reserved The write value must be the initial value.



## B. Instruction Prefetching and Its Side Effects

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

	Address	Instruction	
	:	:	
	H'03FF FFF8	ADD R1,R4	← PC (Program Counter)
	H'03FF FFFA	JMP @R2	
	H'03FF FFFC	NOP	
Area 0	H'03FF FFEE	NOP	
Area 1	H'4000 0000		
	H'4000 0002		← Instruction prefetch address

**Figure B.1 Instruction Prefetch**

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

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

### Instruction Prefetch Side Effects

1. It is possible that an external bus access caused by an instruction prefetch may result in misoperation of an external device, such as a FIFO, connected to the area concerned.
2. If there is no device to reply to an external bus request caused by an instruction prefetch, hang-up will occur.

### Remedies

1. These illegal instruction fetches can be avoided by using the MMU.
2. The problem can be avoided by not locating program code in the last 64 bytes of any area.

## C. Speculative Execution for Subroutine Return

The SH-4A has the mechanism to issue an instruction fetch speculatively when returning from subroutine. By issuing an instruction fetch speculatively, the execution cycles to return from subroutine may be shortened.

This function is enabled by setting 0 to the bit 5 (RABD) of CPU Operation Mode register (CPUOPM). But this speculative instruction fetch may issue the access to the address that should not be accessed from the program. Therefore, a bus access to an unexpected area or an internal instruction address error may cause a problem. As for the effect of this bus access to unexpected memory area, refer to appendix B, Instruction Prefetching and Its Side Effects.

**Usage Condition:** When the speculative execution for subroutine return is enabled, the RTS instruction should be used to return to the address set in PR by the JSR, BSR, or BSRF instructions. It can prevent the access to unexpected address and avoid the problem.

## D. Version Registers (PVR, PRR)

The SH-4A 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 bit 7 to bit 0 of PVR register and the bit 3 to bit 0 of PRR register should be masked by the software.

**Table D.1 Register Configuration**

Register Name	Abbr.	R/W	P4 Address	Area 7 Address	Size
Processor version register	PVR	R	H'FF000030	H'1F000030	32
Product register	PRR	R	H'FF000044	H'1F000044	32

### Processor Version Register (PVR):

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CHIP								VER							
Initial value:	0	0	0	1	0	0	0	0	0	0	1	1	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
	CUT								—	—	—	—	—	—	—	—
Initial value:	0	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	Bit Name	Initial Value	R/W	Description
31 to 24	CHIP	H'10	R	Processor Family The read value is always H'10 in the SH-4A.
23 to 16	VER	H'30	R	Major Version This value is changed when performing major enhancement of the architecture to the SH-4A.
15 to 8	CUT	H'08	R	Minor Version This value is changed when performing minor enhancement of the architecture to the SH-4A.
7 to 0	—	Undefined	R	This value is undefined. It should be masked by software when using it.

**Product Register (PRR):**

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
	Product								CUT				—	—	—	—
Initial value:	0	0	0	1	0	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 16	—	All 0	R	Reserved  For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
15 to 8	Product	H'11	R	Major Version  This value is changed when performing major enhancement of the product.
7 to 4	CUT	Undefined	R	Minor Version  This value is changed when performing minor enhancement of the product.
3 to 0	—	Undefined	R	This value is undefined. It should be masked by software when using it.

## E. Pin State

**Table E.1 Pin States in Reset, Power-Down State, and Bus-Released State**

BGA ball no.	Pin Name	I/O	Power-on Reset	Refresh Standby	Sleep	Hi-Z Setting*	Bus Release
B1	XIN	XI	XI	XI	XI	XI	XI
C1	XOUT	XO	XO	XO	XO	XO	XO
D3	WOL/PF2/IDEA0_M	O/IO/O	O	K/G/K	O/IO/O	Z/IO/Z	O/IO/O
E3	SSISCK2/PC3	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
F4	SSIDATA2/PC2	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
G4	SSIWS2/PC4	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
D2	LNKSTA/PF3/ IDECS0_M	I/IO/O	I	I/G/K	I/IO/O	Z/IO/Z	I/IO/O
D1	EXOUT/PF4/ IDECS1_M	O/IO/O	O	K/G/K	O/IO/O	Z/IO/Z	O/IO/O
F3	AUDIO_CLK2/PC5	I/IO	I	I/G	I/IO	Z/IO	I/IO
E2	CRS/PD7/IDEA1_M	I/IO/O	I	I/G/K	I/IO/O	Z/IO/Z	I/IO/O
E1	COL/PE7/IDEA2_M	I/IO/O	I	I/G/K	I/IO/O	Z/IO/Z	I/IO/O
G3	TX_ER/PD6/ IDEIOWR_M	O/IO/O	O	K/G/K	O/IO/O	Z/IO/Z	O/IO/O
F2	MII_TXD3/SSIDATA5/ IODACK_M/PD0	O/IO/O/IO	O	K/K/K/G	O/IO/O/IO	Z/Z/Z/IO	O/IO/O/IO
F1	MII_TXD2/AUDIO_CLK 5/IDEINT_M/PD1	O/I/I/IO	O	K/I/I/G	O/I/I/IO	Z/Z/Z/IO	O/I/I/IO
H4	MPMD	I	PI	PI	PI	PI	PI
H3	RX_ER/PE6/IODREQ_ M	I/IO/I	I	I/G/I	I/IO/I	Z/IO/Z	I/IO/I
G2	MII_TXD1/SSIWS5/ IDEIORD_M/PD2	O/IO/O/IO	O	K/K/K/G	O/IO/O/IO	Z/Z/Z/IO	O/IO/O/IO
J4	SSIDATA3/PH4	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
G1	MII_TXD0/SSISCK5/ID EIORDY_M/ PD3	O/IO/I/IO	O	K/K/I/G	O/IO/I/IO	Z/Z/Z/IO	O/IO/I/IO
H2	TX_EN/PD4/IDED0_M	O/IO/IO	O	K/G/IO	O/IO/IO	Z/IO/Z	O/IO/IO
J3	SSIWS3/PH6	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO

BGA ball no.	Pin Name	I/O	Power-on Reset	Refresh Standby	Sleep	Hi-Z Setting*	Bus Release
H1	TX_CLK/PD5/IDED15_M	I/O/I/O	I	I/G/I/O	I/O/I/O	Z/I/O/Z	I/O/I/O
J2	RX_CLK/PE5/IDED1_M	I/O/I/O	I	I/G/I/O	I/O/I/O	Z/I/O/Z	I/O/I/O
J1	RX_DV/PE4/IDED14_M	I/O/I/O	I	I/G/I/O	I/O/I/O	Z/I/O/Z	I/O/I/O
K3	SSISCK3/PH5	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
K4	IRQ0/DTEND1	I/O	I	I/O	I/O	I/Z	I/O
K2	MII_RXD0/SSIWS4/IDE D2_M/PE3	I/O/I/O/I/O	I	I/K/I/O/G	I/O/I/O/I/O	Z/Z/Z/IO	I/O/I/O/I/O
K1	MII_RXD1/SSISCK4/IDE D13_M/PE2	I/O/I/O/I/O	I	I/K/I/O/G	I/O/I/O/I/O	Z/Z/Z/IO	I/O/I/O/I/O
L3	AUDIO_CLK3/PH7	I/O	I	I/G	I/O	Z/IO	I/O
L2	MII_RXD2/SSIDATA4/IDE D3_M/PE1	I/O/I/O/I/O	I	I/K/I/O/G	I/O/I/O/I/O	Z/Z/Z/IO	I/O/I/O/I/O
L4	IRQOUT/DREQ1	O/I	O	K/I	O/I	O/Z	O/I
L1	MII_RXD3/AUDIO_CLK 4/IDED12_M/PE0	I/I/O/I/O	I	I/I/O/G	I/I/O/I/O	Z/Z/Z/IO	I/I/O/I/O
M2	MDC/PF0/IDED4_M	O/I/O/I/O	O	K/G/I/O	O/I/O/I/O	Z/IO/Z	O/I/O/I/O
M1	MDIO/PF1/IDED11_M	IO/IO/IO	I	K/G/I/O	IO/IO/IO	Z/IO/Z	IO/IO/IO
N1	AUDIO_CLK0/PC7	I/O	I	I/G	I/O	Z/IO	I/O
M3	SSIWS0	IO	I	K	IO	Z	IO
M4	STATUS1/RTS2/PA7	O/I/O/I/O	L	L/K/G	H/I/O/I/O	L/Z/IO	H/I/O/I/O
N2	SSISCK0	IO	I	K	IO	Z	IO
P1	AUDIO_CLK1/PC6	I/O	I	I/G	I/O	Z/IO	I/O
N4	STATUS0/CTS2/PA6	O/I/O/I/O	L	H/K/G	L/I/O/I/O	L/Z/IO	L/I/O/I/O
N3	SSIDATA0	IO	I	K	IO	Z	IO
P2	SSISCK1	IO	I	K	IO	Z	IO
R1	PJ7/IDED10_M	IO/IO	I	G/I/O	IO/IO	IO/Z	IO/IO
P3	SSIWS1	IO	I	K	IO	Z	IO
R2	PJ6/IDED5_M	IO/IO	I	G/I/O	IO/IO	IO/Z	IO/IO
P4	FRĒ/PA4	O/I/O	O	K/G	O/I/O	Z/IO	O/I/O
R3	SSIDATA1	IO	I	K	IO	Z	IO

BGA ball no.	Pin Name	I/O	Power-on Reset	Refresh Standby	Sleep	Hi-Z Setting*	Bus Release
T1	PJ5/IDED9_M	IO/IO	I	G/IO	IO/IO	IO/Z	IO/IO
T2	PJ4/IDED6_M	IO/IO	I	G/IO	IO/IO	IO/Z	IO/IO
U1	PJ2/IDED8_M	IO/IO	I	G/IO	IO/IO	IO/Z	IO/IO
U2	PJ3/IDED7_M	IO/IO	I	G/IO	IO/IO	IO/Z	IO/IO
T3	FWE/PA3	O/IO	O	K/G	O/IO	Z/IO	O/IO
R4	FCE/PA5	O/IO	O	K/G	O/IO	Z/IO	O/IO
V1	PJ1/IDERST_M	IO/O	I	G/K	IO/O	IO/Z	IO/O
V2	PJ0/DIRECTION_M	IO/O	I	G/K	IO/O	IO/Z	IO/O
U3	MODE7/FD6	I/IO	Z	-/K	-/IO	-/Z	-/IO
T4	FALE/PC0	O/IO	O	K/G	O/IO	Z/IO	O/IO
W1	MODE3/FD3	I/IO	Z	-/K	-/IO	-/Z	-/IO
V3	MODE5/FD5	I/IO	Z	-/K	-/IO	-/Z	-/IO
Y1	TXD2/PA2	O/IO	Z	K/G	O/IO	Z/IO	O/IO
W2	MODE2/FD2	I/IO	Z	-/K	-/IO	-/Z	-/IO
V4	MODE4/FD4	I/IO	Z	-/K	-/IO	-/Z	-/IO
U4	MODE8/FD7	I/IO	Z	-/K	-/IO	-/Z	-/IO
W3	MODE1/FD1	I/IO	Z	-/K	-/IO	-/Z	-/IO
Y2	RXD2/PA1	I/IO	I	I/G	I/IO	Z/IO	I/IO
AA1	SCK2/PA0	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
AB3	SDA	IO	Z	IO	IO	Z	IO
AB2	SCL	IO	Z	IO	IO	Z	IO
Y4	RXD1/AUDATA2	I/IO	I	I/K	I/IO	Z/IO	I/IO
W5	WDTOVF/IRQ1/AUDC K/DACK1	O/I/IO/O	O	O/I/K/O	O/I/IO/O	O/I/IO/Z	O/I/IO/O
AA3	MODE0/FD0	I/IO	Z	-/K	-/IO	-/Z	-/IO
Y5	RXD0/AUDATA0	I/IO	I	I/K	I/IO	Z/IO	I/IO
AA4	TXD1/AUDATA3	O/IO	Z	K/K	O/IO	Z/IO	O/IO
W6	TDO	O	Z	O	O	O	O
AA5	TXD0/AUDATA1	O/IO	Z	K/K	O/IO	Z/IO	O/IO
AB4	SCK1/FR $\bar{B}$	IO/I	I	K/I	IO/I	Z/Z	IO/I

BGA ball no.	Pin Name	I/O	Power-on Reset	Refresh Standby	Sleep	Hi-Z Setting*	Bus Release
Y6	TMS	I	PI	PI	PI	PI	PI
W7	TRST	I	PI	PI	PI	PI	PI
AA6	TDI	I	PI	PI	PI	PI	PI
AB5	SCK0/AUDSYNC/FCL E	IO/IO/O	I	K/K/K	IO/IO/O	Z/IO/Z	IO/IO/O
AB6	TCK	I	PI	PI	PI	PI	PI
Y8	LCD_VEP_WC/DR5/P H0	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
AA8	LCD_FLM/VSYNC/SP S/EX_VSYNC/BT_VSY NC	O/IO/O	O	K/K/K	O/IO/O	Z/Z/Z	O/IO/O
AB8	LCD_CL1/HSYNC/SPL /EX_HSYNC/BT_HSY NC	O/IO/O	O	K/K/K	O/IO/O	Z/Z/Z	O/IO/O
AA9	LCD_M_DISP/DE_C/D E_H/BT_DE_C	O/O/O	O	K/K/K	O/O/O	Z/Z/Z	O/O/O
Y9	LCD_VCP_WC/DR4/P H1	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
AB9	LCD_CLK/DCLKIN	I/I	I	I/I	I/I	Z/Z	I/I
Y10	LCD_DATA15/DR3/PG 7	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
AA10	LCD_DATA14/DR2/PG 6	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
AB10	LCD_DATA13/DR1/PG 5	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
Y11	LCD_DATA12/DR0/PG 4	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
AA11	LCD_DATA11/DG5/PG 3	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
AB11	LCD_DATA10/DG4/PG 2	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
Y12	LCD_DATA9/DG3/PG1	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
AA12	LCD_DATA8/DG2/PG0	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
AB12	LCD_DATA7/DG1/BT_ DATA7/PI4	O/O/O/IO	O	K/K/K/G	O/O/O/IO	Z/Z/Z/IO	O/O/O/IO



BGA ball no.	Pin Name	I/O	Power-on Reset	Refresh Standby	Sleep	Hi-Z Setting*	Bus Release
Y13	LCD_DATA6/DG0/BT_D ATA6/PI3	O/O/O/IO	O	K/K/K/G	O/O/O/IO	Z/Z/Z/IO	O/O/O/IO
AA13	LCD_DATA5/DB5/BT_D ATA5/PI2	O/O/O/IO	O	K/K/K/G	O/O/O/IO	Z/Z/Z/IO	O/O/O/IO
AB13	LCD_DATA4/DB4/BT_D ATA4/PI1	O/O/O/IO	O	K/K/K/G	O/O/O/IO	Z/Z/Z/IO	O/O/O/IO
Y14	LCD_DATA3/DB3/BT_D ATA3	O/O/O	O	K/K/K	O/O/O	Z/Z/Z	O/O/O
AA14	LCD_DATA2/DB2/BT_D ATA2	O/O/O	O	K/K/K	O/O/O	Z/Z/Z	O/O/O
AB14	LCD_DATA1/DB1/BT_D ATA1	O/O/O	O	K/K/K	O/O/O	Z/Z/Z	O/O/O
AA15	LCD_DATA0/DB0/BT_D ATA0	O/O/O	O	K/K/K	O/O/O	Z/Z/Z	O/O/O
Y15	LCD_CL2/DE_V/PH3	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
AB15	LCD_DON/DCLKOUT/P H2	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
Y16	PI0/COM/CDE	IO/O	I	G/K	IO/O	IO/Z	IO/O
AB17	RDY	I	I	MI	MI	MI	MI
AA17	NMI	I	I	I	I	I	I
AA18	BACK	O	O	K	O	O	O
Y17	RD	O	O	K	O	O	MZ
W17	CS3	O	O	K	O	O	MZ
Y19	BREQ	I	I	MI	MI	MI	MI
AB21	EXTAL	XI	XI	XI	XI	XI	XI
AB22	XTAL	XO	XO	XO	XO	XO	XO
Y20	BS	O	O	K	O	O	MZ
Y22	PRESET	I	I	I	I	I	I
W18	CS0	O	O	K	O	O	MZ
W21	ASEBRKAK/BRKACK/T CLK/PC1	O/I/IO	A	A/I/G	IO/I/IO	IO/Z/IO	IO/I/IO
V19	A25/PB7/DREQ0/RTS0	O/IO/I/IO	O	K/G/I/K	O/IO/I/IO	O/IO/Z/Z	MZ/IO/I/IO
W22	A24/PB6/DACK0/CTS0	O/IO/O/IO	O	K/G/O/K	O/IO/O/IO	O/IO/Z/Z	MZ/IO/O/IO

BGA ball no.	Pin Name	I/O	Power-on Reset	Refresh Standby	Sleep	Hi-Z Setting*	Bus Release
U19	A17	O	O	K	O	O	MZ
V21	A23/PB5/ $\overline{\text{DTEND0}}$ / $\overline{\text{RTS1}}$	O/IO/O/IO	O	K/G/O/K	O/IO/O/IO	O/IO/Z/Z	MZ/IO/O/IO
V20	A21/PB3	O/IO	O	K/G	O/IO	O/IO	MZ/IO
U22	A20/PB2	O/IO	O	K/G	O/IO	O/IO	MZ/IO
V22	A22/PB4/ $\overline{\text{CTS1}}$	O/IO/IO	O	K/G/K	O/IO/IO	O/IO/Z	MZ/IO/IO
U21	A19/PB1	O/IO	O	K/G	O/IO	O/IO	MZ/IO
U20	A18/PB0	O/IO	O	K/G	O/IO	O/IO	MZ/IO
T19	D15	IO	PZ	MZ	IO	IO	MZ
T20	D14	IO	PZ	MZ	IO	IO	MZ
T21	D1	IO	PZ	MZ	IO	IO	MZ
T22	D0	IO	PZ	MZ	IO	IO	MZ
R19	D13	IO	PZ	MZ	IO	IO	MZ
R20	D12	IO	PZ	MZ	IO	IO	MZ
R21	D3	IO	PZ	MZ	IO	IO	MZ
R22	D2	IO	PZ	MZ	IO	IO	MZ
P19	D11	IO	PZ	MZ	IO	IO	MZ
P20	D10	IO	PZ	MZ	IO	IO	MZ
P21	D5	IO	PZ	MZ	IO	IO	MZ
P22	D4	IO	PZ	MZ	IO	IO	MZ
N19	D9	IO	PZ	MZ	IO	IO	MZ
N22	D6	IO	PZ	MZ	IO	IO	MZ
N21	D7	IO	PZ	MZ	IO	IO	MZ
N20	D8	IO	PZ	MZ	IO	IO	MZ
M22	DQMLL	O	O	K	O	O	MZ
M21	DQMUL	O	O	K	O	O	MZ
M19	DQMUU	O	O	K	O	O	MZ
L22	D16	IO	PZ	MZ	IO	IO	MZ
M20	DQMLU	O	O	K	O	O	MZ
L21	D17	IO	PZ	MZ	IO	IO	MZ
K22	D18	IO	PZ	MZ	IO	IO	MZ

BGA ball no.	Pin Name	I/O	Power-on Reset	Refresh Standby	Sleep	Hi-Z Setting*	Bus Release
K21	D19	IO	PZ	MZ	IO	IO	MZ
L19	D31	IO	PZ	MZ	IO	IO	MZ
L20	D30	IO	PZ	MZ	IO	IO	MZ
J22	D20	IO	PZ	MZ	IO	IO	MZ
J21	D21	IO	PZ	MZ	IO	IO	MZ
H22	D22	IO	PZ	MZ	IO	IO	MZ
K20	D28	IO	PZ	MZ	IO	IO	MZ
K19	D29	IO	PZ	MZ	IO	IO	MZ
G22	A15	O	O	K	O	O	MZ
H21	D23	IO	PZ	MZ	IO	IO	MZ
J20	D26	IO	PZ	MZ	IO	IO	MZ
F22	A13	O	O	K	O	O	MZ
G21	A16	O	O	K	O	O	MZ
J19	D27	IO	PZ	MZ	IO	IO	MZ
H20	D24	IO	PZ	MZ	IO	IO	MZ
E22	A10	O	O	K	O	O	MZ
F21	A14	O	O	K	O	O	MZ
H19	D25	IO	PZ	MZ	IO	IO	MZ
D22	A4	O	O	K	O	O	MZ
G20	A11	O	O	K	O	O	MZ
E21	A5	O	O	K	O	O	MZ
D21	R/W	O	O	H	O	O	MZ
F20	A8	O	O	K	O	O	MZ
G19	A12	O	O	K	O	O	MZ
C22	CKE	O	O	K	O	O	MZ
C21	RAS	O	O	H	O	O	MZ
B22	CLKOUT	O	O	C	O	O	O
F19	A9	O	O	K	O	O	MZ
E20	A6	O	O	K	O	O	MZ
E19	A7	O	O	K	O	O	MZ

BGA ball no.	Pin Name	I/O	Power-on Reset	Refresh Standby	Sleep	Hi-Z Setting*	Bus Release
D20	CAS	O	O	K	O	O	MZ
A21	CS1	O	O	K	O	O	MZ
B20	CS2	O	O	K	O	O	MZ
C19	A0	O	O	K	O	O	MZ
D18	D47/IDECS0	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
A20	A3	O	O	K	O	O	MZ
B19	A1	O	O	K	O	O	MZ
D17	D45/IODACK	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
C18	D46/IDECS1	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
B18	D33/PF6	IO/IO	PZ	MZ/G	IO/IO	IO/IO	MZ/IO
A19	A2	O	O	K	O	O	MZ
C17	D44/IDEINT	IO/I	PZ	MZ/I	IO/I	IO/Z	MZ/I
D16	D43/IDEIORDY	IO/I	PZ	MZ/I	IO/I	IO/Z	MZ/I
C16	D42/IDEIORD	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
A18	D32/PF7	IO/IO	PZ	MZ/G	IO/IO	IO/IO	MZ/IO
C15	D40/IDEIOWR	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
D15	D41/IODREQ	IO/I	PZ	MZ/I	IO/I	IO/Z	MZ/I
B17	D35/IDEA0	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
B16	D37/IDEA1	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
B15	D39/IDED14	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
A17	D34/PF5	IO/IO	PZ	MZ/G	IO/IO	IO/IO	MZ/IO
A16	D36/IDEA2	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
D14	D63/IDED1	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
A15	D38/IDED15	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
C14	D62/IDED0	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
B14	WE2/DQM64UL	O/O	O	K/K	O/O	O/O	MZ/MZ
A14	WE0/DQM64LL	O/O	O	K/K	O/O	O/O	MZ/MZ
C13	D60/IDED2	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
B13	WE3/DQM64UU	O/O	O	K/K	O/O	O/O	MZ/MZ
A13	WE1/DQM64LU	O/O	O	K/K	O/O	O/O	MZ/MZ

BGA ball no.	Pin Name	I/O	Power-on Reset	Refresh Standby	Sleep	Hi-Z Setting*	Bus Release
D13	D61/IDED3	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
A12	D48/IDED13	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
D12	D59/IDED5	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
C12	D58/IDED4	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
B12	D49/IDED12	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
B11	D51/IDED10	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
A11	D50/IDED11	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
C11	D56/IDED6	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
A10	D52/IDED9	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
B10	D53/IDED8	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
D11	D57/IDED7	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
C10	D54/IDERST	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
D10	D55/DIRECTION	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
A7	DM	AIO	Z	Z	AIO	Z	AIO
A6	DP	AIO	Z	Z	AIO	Z	AIO
B6	VBUS	AI	AI	AI	AI	AI	AI
E8	REFRIN	AI	AI	AI	AI	AI	AI

Note: \* Indicated the pin states when setting the corresponding bit in the Hi-Z register A or B (PTHIZ\_A or PTHIZ\_B) in the GPIO to 1.

[Legend]

- I: Input
- O: Output
- IO: Input/output
- XI: XTAL input
- XO: XTAL output
- AI: Analog input
- AIO: Analog input/output
- Z: High impedance
- PI: Input and pulled up by an on-chip resistance
- PZ: High impedance and pulled up by an on-chip resistance
- H: High-level output
- L: Low-level output
- K: The state of a pin (both input and output) before transition to refresh standby mode is held.

- MI: Input buffer on, whether the on-chip pull-up resistance is on or off is depends on the corresponding register setting in the MCU.
- MZ: Input buffer off, output buffer off; whether the on-chip pull-up resistance is on or off is depends on the corresponding register setting in the MCU
- A: On-chip pull-up resistance is on. Input/output can be controlled by a register setting when MPMD is at a low level. Input when  $\overline{\text{TRST}}$  is at the low level or MPMD is at the high level.
- C: Clock output or low level output: depends on a register setting in the CPG.
- G: Whether this is an input or output and the on-chip resistance is on or off is depends on register settings in the GPIO.
- GPI: Input buffer on, output buffer off; whether the on-chip resistance is on or off depends on register settings in the GPIO.
- GPZ: Input buffer off, output buffer off; whether the on-chip resistance is on or off depends on register settings in the GPIO.
- : Invalid

## F. Pin Treatment When Not in Use

**Table F.1 Treatment of Unused Pins**

BGA ball no.	Pin Name	Treatment of Unused Pins*4
B1	XIN	Pull-down
C1	XOUT	Open
D3	WOL/PF2/IDEA0_M	Open
E3	SSISCK2/PC3	Pull-up
F4	SSIDATA2/PC2	Pull-up
G4	SSIWS2/PC4	Pull-up
D2	LNKSTA/PF3/IDECS0_M	Pull-down
D1	EXOUT/PF4/IDECS1_M	Open
F3	AUDIO_CLK2/PC5	Pull-up
E2	CRS/PD7/IDEA1_M	Pull-down
E1	COL/PE7/IDEA2_M	Pull-down
G3	TX_ER/PD6/IDEIOWR_M	Open
F2	MII_TXD3/SSIDATA5/IODACK_M/PD0	Open
F1	MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1	Open
H4	MPMD	Pull-up*2
H3	RX_ER/PE6/IODREQ_M	Pull-down
G2	MII_TXD1/SSIWS5/IDEIORD_M/PD2	Open
J4	SSIDATA3/PH4	Pull-up
G1	MII_TXD0/SSISCK5/IDEIORDY_M/PD3	Open
H2	TX_EN/PD4/IDED0_M	Open
J3	SSIWS3/PH6	Pull-up
H1	TX_CLK/PD5/IDED15_M	Pull-down
J2	RX_CLK/PE5/IDED1_M	Pull-down
J1	RX_DV/PE4/IDED14_M	Pull-down
K3	SSISCK3/PH5	Pull-up
K4	IRQ0/DTEND1	Pull-up
K2	MII_RXD0/SSIWS4/IDED2_M/PE3	Pull-down
K1	MII_RXD1/SSISCK4/IDED13_M/PE2	Pull-down

BGA ball no.	Pin Name	Treatment of Unused Pins <sup>*4</sup>
L3	AUDIO_CLK3/PH7	Pull-up
L2	MII_RXD2/SSIDATA4/IDED3_M/PE1	Pull-down
L4	IRQOUT/DREQ1	Open
L1	MII_RXD3/AUDIO_CLK4/IDED12_M/PE0	Pull-down
M2	MDC/PF0/IDED4_M	Open
M1	MDIO/PF1/IDED11_M	Pull-down
N1	AUDIO_CLK0/PC7	Pull-up
M3	SSIWS0	Pull-up
M4	STATUS1/RTS2/PA7	Open
N2	SSISCK0	Pull-up
P1	AUDIO_CLK1/PC6	Pull-up
N4	STATUS0/CTS2/PA6	Open
N3	SSIDATA0	Pull-up
P2	SSISCK1	Pull-up
R1	PJ7/IDED10_M	Pull-up
P3	SSIWS1	Pull-up
R2	PJ6/IDED5_M	Pull-up
P4	FRE/PA4	Open
R3	SSIDATA1	Pull-up
T1	PJ5/IDED9_M	Pull-up
T2	PJ4/IDED6_M	Pull-up
U1	PJ2/IDED8_M	Pull-up
U2	PJ3/IDED7_M	Pull-up
T3	FWE/PA3	Open
R4	FCE/PA5	Open
V1	PJ1/IDERST_M	Pull-up
V2	PJ0/DIRECTION_M	Pull-up
U3	MODE7/FD6	Pull-down
T4	FALE/PC0	Open
W1	MODE3/FD3	Must be used
V3	MODE5/FD5	Must be used



BGA ball no.	Pin Name	Treatment of Unused Pins*4
Y1	TXD2/PA2	Open
W2	MODE2/FD2	Must be used
V4	MODE4/FD4	Must be used
U4	MODE8/FD7	Must be used
W3	MODE1/FD1	Must be used
Y2	RXD2/PA1	Pull-up
AA1	SCK2/PA0	Pull-up
AB3	SDA	Open
AB2	SCL	Open
Y4	RXD1/AUDATA2	Pull-up
W5	WDTOVF/IRQ1/AUDCK/DACK1	Open
AA3	MODE0/FD0	Must be used
Y5	RXD0/AUDATA0	Pull-up
AA4	TXD1/AUDATA3	Open
W6	TDO	Open*2
AA5	TXD0/AUDATA1	Open
AB4	SCK1/FR/B	Pull-up
Y6	TMS	Open*2
W7	TRST	Connect to ground or PRESET*2*3
AA6	TDI	Open*2
AB5	SCK0/AUDSYNC/FCLE	Pull-up
AB6	TCK	Open*2
Y8	LCD_VEP_WC/DR5/PH0	Open
AA8	LCD_FLM/VSYN/SPS/EX_VSYN/BT_VSYN	Open
AB8	LCD_CL1/HSYN/SPL/EX_HSYN/BT_HSYN	Open
AA9	LCD_M_DISP/DE_C/DE_H/BT_DE_C	Open
Y9	LCD_VCP_WC/DR4/PH1	Open
AB9	LCD_CLK/DCLKIN	Pull-up
Y10	LCD_DATA15/DR3/PG7	Open
AA10	LCD_DATA14/DR2/PG6	Open

BGA ball no.	Pin Name	Treatment of Unused Pins* <sup>4</sup>
AB10	LCD_DATA13/DR1/PG5	Open
Y11	LCD_DATA12/DR0/PG4	Open
AA11	LCD_DATA11/DG5/PG3	Open
AB11	LCD_DATA10/DG4/PG2	Open
Y12	LCD_DATA9/DG3/PG1	Open
AA12	LCD_DATA8/DG2/PG0	Open
AB12	LCD_DATA7/DG1/BT_DATA7/PI4	Open
Y13	LCD_DATA6/DG0/BT_DATA6/PI3	Open
AA13	LCD_DATA5/DB5/BT_DATA5/PI2	Open
AB13	LCD_DATA4/DB4/BT_DATA4/PI1	Open
Y14	LCD_DATA3/DB3/BT_DATA3	Open
AA14	LCD_DATA2/DB2/BT_DATA2	Open
AB14	LCD_DATA1/DB1/BT_DATA1	Open
AA15	LCD_DATA0/DB0/BT_DATA0	Open
Y15	LCD_CL2/DE_V/PH3	Open
AB15	LCD_DON/DCLKOUT/PH2	Open
Y16	PI0/COM/CDE	Pull-up
AB17	RDY	Pull-down* <sup>1</sup>
AA17	NMI	Pull-up
AA18	BACK	Open
Y17	RD	Open
W17	CS3	Open
Y19	BREQ	Pull-up
AB21	EXTAL	Must be used
AB22	XTAL	Open
Y20	BS	Open
Y22	PRESET	Must be used
W18	CS0	Must be used
W21	ASEBRKAK/BRKACK/TCLK/PC1	Open* <sup>2</sup>
V19	A25/PB7/DREQ0/RTS0	Open
W22	A24/PB6/DACK0/CTS0	Open
U19	A17	Open

BGA ball no.	Pin Name	Treatment of Unused Pins <sup>*4</sup>
V21	A23/PB5/DTEND0/RTS1	Open
V20	A21/PB3	Open
U22	A20/PB2	Open
V22	A22/PB4/CTS1	Open
U21	A19/PB1	Open
U20	A18/PB0	Open
T19	D15	Open
T20	D14	Open
T21	D1	Must be used
T22	D0	Must be used
R19	D13	Open
R20	D12	Open
R21	D3	Must be used
R22	D2	Must be used
P19	D11	Open
P20	D10	Open
P21	D5	Must be used
P22	D4	Must be used
N19	D9	Open
N22	D6	Must be used
N21	D7	Must be used
N20	D8	Open
M22	DQMLL	Open
M21	DQMUL	Open
M19	DQMUU	Open
L22	D16	Open
M20	DQMLU	Open
L21	D17	Open
K22	D18	Open
K21	D19	Open
L19	D31	Open
L20	D30	Open

BGA ball no.	Pin Name	Treatment of Unused Pins*4
J22	D20	Open
J21	D21	Open
H22	D22	Open
K20	D28	Open
K19	D29	Open
G22	A15	Open
H21	D23	Open
J20	D26	Open
F22	A13	Open
G21	A16	Open
J19	D27	Open
H20	D24	Open
E22	A10	Open
F21	A14	Open
H19	D25	Open
D22	A4	Open
G20	A11	Open
E21	A5	Open
D21	R/W	Open
F20	A8	Open
G19	A12	Open
C22	CKE	Open
C21	RAS	Open
B22	CLKOUT	Open
F19	A9	Open
E20	A6	Open
E19	A7	Open
D20	CAS	Open
A21	CS1	Open
B20	CS2	Open
C19	A0	Open
D18	D47/IDECS0	Open

BGA ball no.	Pin Name	Treatment of Unused Pins*4
A20	A3	Open
B19	A1	Open
D17	D45/IODACK	Open
C18	D46/IDECS1	Open
B18	D33/PF6	Open
A19	A2	Open
C17	D44/IDEINT	Open
D16	D43/IDEIORDY	Open
C16	D42/IDEIORD	Open
A18	D32/PF7	Open
C15	D40/IDEIOWR	Open
D15	D41/IODREQ	Open
B17	D35/IDEA0	Open
B16	D37/IDEA1	Open
B15	D39/IDED14	Open
A17	D34/PF5	Open
A16	D36/IDEA2	Open
D14	D63/IDED1	Open
A15	D38/IDED15	Open
C14	D62/IDED0	Open
B14	WE2/DQM64UL	Open
A14	WE0/DQM64LL	Open
C13	D60/IDED2	Open
B13	WE3/DQM64UU	Open
A13	WE1/DQM64LU	Open
D13	D61/IDED3	Open
A12	D48/IDED13	Open
D12	D59/IDED5	Open
C12	D58/IDED4	Open
B12	D49/IDED12	Open
B11	D51/IDED10	Open

BGA ball no.	Pin Name	Treatment of Unused Pins*4
A11	D50/IDED11	Open
C11	D56/IDED6	Open
A10	D52/IDED9	Open
B10	D53/IDED8	Open
D11	D57/IDED7	Open
C10	D54/IDERST	Open
D10	D55/DIRECTION	Open
A7	DM	Open
A6	DP	Open
B6	VBUS	Open
E8	REFRIN	Open

Notes: 1. This pin is pulled-up within the LSI after a power-on reset (initial state). Set the IPUP bit in BCR (MCU) to 1 to switch pulling-up of the RDY pin off.

Pulling-up of the  $\overline{\text{BREQ}}$  pin is also off when the IPUP bit is 1. So if the  $\overline{\text{BREQ}}$  pin is in use, use a pull-up resistance on the board to pull the  $\overline{\text{BREQ}}$  pin up.

2. When designing a board for use with an emulator, follow the directions for the emulator.
3. When not using emulator, the pin should be fixed to ground or connected to another pin which operates in the same manner as  $\overline{\text{PRESET}}$ . However, since the  $\overline{\text{TRST}}$  pin is pulled up within this LSI, a weak current flows when the pin is externally connected to ground pin. The value of the current is determined by a resistance of the pull-up MOS for the  $\overline{\text{TRST}}$  pin. Although this current does not affect the operation of this LSI, it consumes power unnecessarily.
4. Treatment of pins if unused refers to the pin state after a power-on reset.

## G. Type Name

**Table G.1 Type name of the products**

Model Name	Catalog Number	Operating temperature range	Solder Ball Composition	PKG Code	SDHI
R5S77640N300BG	R5S77640N300BG	−20 to +85°C	Lead free	PRBG0404GA-A	Not mounted
R5S77640P300BG	R5S77640P300BG	−40 to +85°C	Lead free	PRBG0404GA-A	Not mounted
R5S77641N300BG	R5S77641N300BG	−20 to +85°C	Lead free	PRBG0404GA-A	Mounted*
R5S77641P300BG	R5S77641P300BG	−40 to +85°C	Lead free	PRBG0404GA-A	Mounted*

Note \* This product mounts the SD host interface (SDHI). As regards the SD host interface (SDHI) information, a nondisclosure agreement needs to be entered into before release. Ask our sales representative details on this matter.

H. Package Dimensions

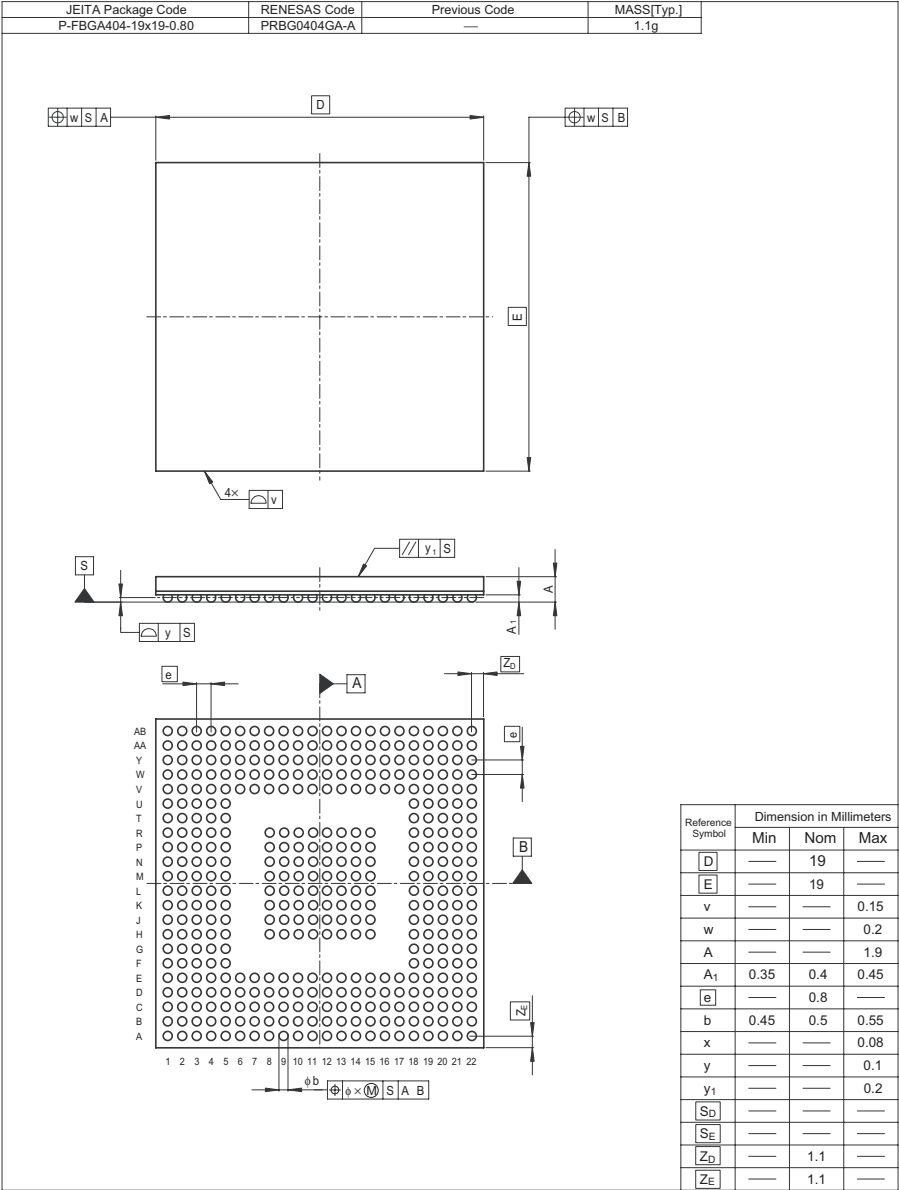


Figure H.1 Package Dimensions



# Index

## Numerics

10-Bit address format .....	576
7-Bit address format .....	575

## A

Absolute Maximum Ratings .....	1573
AC Characteristics .....	1581
AC Characteristics Measurement	
Conditions.....	1657
Address space identifier (ASID).....	157
Address translation .....	156
Addressing modes.....	53
AND/NAND flash memory controller (FLCTL) .....	1277
Arithmetic operation instructions .....	61
ASID.....	170
Auto-Reload Count Operation.....	483
Auto-Request mode .....	384

## B

Big endian.....	46
Block diagram.....	13
Branch instructions .....	65
Break detection and processing .....	551
Burst mode.....	395
Bus Timing .....	1586

## C

Cacheability bit.....	171
Caches.....	207
Clock pulse generator (CPG).....	245
Clock Timing.....	1581
Command access mode.....	1308
Control registers .....	32

Control Signal Timing .....	1584
Cycle-steal mode.....	393

## D

Data address error .....	116
Data TLB miss exception.....	110, 192
Data TLB multiple hit exception .....	192
Data TLB multiple-hit exception .....	109
Data TLB protection violation exception .....	113, 194
DC Characteristics .....	1575
Delay slot.....	51
Delayed branches.....	51
Direct memory access controller (DMAC).....	359
Dirty bit.....	172
Division by zero.....	144
DMAC Module Timing .....	1605
Double-precision floating-point extended registers.....	37
Double-precision floating-point registers.....	37
Dual address mode .....	391

## E

ECC code .....	1315
ECC error check.....	1315
Effective address.....	53
Electrical Characteristics.....	1573
Equation for getting SCBRR value .....	513
EtherC Module Signal Timing .....	1644
Exception flow .....	104
Exception handling .....	95
Exception/interrupt codes .....	102
Execution cycles .....	85

External request mode ..... 384

## F

Fixed mode ..... 388

Fixed-point transfer instructions ..... 59

Floating-point control instructions ..... 70

Floating-point double-precision  
instructions ..... 69

Floating-point graphics acceleration  
instructions ..... 70

Floating-point registers ..... 33, 37

Floating-point single-precision  
instructions ..... 68

FPU error ..... 144

FPU exception ..... 125

FPU exception handling ..... 145

FPU Exception sources ..... 144

## G

General FPU disable exception ..... 122

General FPU disable exceptions and  
slot FPU disable exceptions ..... 144

General illegal instruction exception ..... 120

General interrupt request ..... 126

General purpose I/O (GPIO) ..... 1337

General registers ..... 32

Geometric operation instructions ..... 146

## H

H-UDI reset ..... 108

H-UDI-Related Pin Timing ..... 1648

## I

I/O Port Timing ..... 1642

I<sup>2</sup>C bus data format ..... 574

I<sup>2</sup>C bus interface (IIC) ..... 555

Inexact exception ..... 144

Initial page write exception ..... 112, 195

Input Capture Function ..... 484

Instruction address error ..... 118

Instruction execution state ..... 47

Instruction fetch cycle break ..... 1462

Instruction set ..... 51

Instruction TLB miss exception ..... 111, 190

Instruction TLB multiple hit  
exception ..... 109, 189

Instruction TLB protection violation  
exception ..... 115, 191

Intermittent mode ..... 394

Interrupt controller (INTC) ..... 409

Interrupt response time ..... 465

Invalid operation ..... 144

IPG settings ..... 745

IRQ interrupts ..... 455

Issue rates ..... 85

ITLB ..... 173

ITLB address array ..... 198

ITLB data array ..... 199

## L

LCD controller (LCDC) ..... 1001

LCD module power-supply states ..... 1048

Little endian ..... 46

Load-store architecture ..... 51

Logic operation instructions ..... 63

## M

Magic packet ..... 744

Memory management unit ..... 149

Memory-mapped registers ..... 45

MII registers ..... 741, 742

Module standby mode ..... 1422

Multiple interrupts ..... 464

Multiple virtual memory mode .....	156
------------------------------------	-----

## N

NMI (nonmaskable interrupt) .....	126
NMI interrupt.....	455
Notes on display-off mode (LCDC stopped) .....	1049

## O

On-chip module interrupts .....	456
On-Chip peripheral module request mode .....	386
Operand access cycle break .....	1464
Operation in asynchronous mode .....	530
Operation in clocked synchronous mode .....	541
Output Addition Circuit.....	1657
Overflow.....	144

## P

P0, P3, and U0 areas .....	153
P1 area .....	154
P2 area .....	154
P4 area .....	154
Page size bits .....	171
Pair single-precision data transfer instructions .....	147
Physical address space.....	155
Pin arrangement.....	14
Pipelining.....	71
Power-Down mode .....	1411
Power-down state .....	47
Power-on reset .....	108
Power-on/Power-off Sequence .....	1574
Power-supply control sequences.....	1044
PPN.....	171

Pre-execution user break/post-execution user break .....	124
Privileged mode .....	32
Processing modes.....	32
Programming model.....	31
Protection key data.....	171

## R

Receive data sampling timing and receive margin (asynchronous mode) .....	551
Receive descriptor.....	791
Registers	
APR.....	728
BEMPENB.....	849
BEMPSTS.....	868
BRDYENB .....	845
BRDYSTS .....	864
BUSWAIT .....	815
CAMR0.....	1454
CAMR1.....	1454
CAR0 .....	1453
CAR1 .....	1453
CBCR.....	1460
CBR0 .....	1445
CBR1 .....	1445
CCMFR.....	1459
CCR .....	212
CDCR.....	718
CDMR1.....	1457
CDR1 .....	1456
CEFCR.....	721
CETR1 .....	1458
CFIFO .....	827
CFIFOCTR .....	837
CFIFOSEL .....	830
CHCR.....	370
CPUOPM .....	1659
CRR0 .....	1451
CRR1 .....	1451

D0FBCFG.....	826	FLINTDMACR .....	1296
D0FIFO .....	827	FLTRCR .....	1305
D0FIFOCTR.....	837	FPSCR .....	42, 139
D0FIFOSEL .....	830	FPUL.....	142
D1FBCFG.....	826	FRECR.....	722
D1FIFO .....	827	FRMNUM.....	869
D1FIFOCTR.....	837	FRQCR .....	251
D1FIFOSEL .....	830	GBR .....	41
DAR.....	367	GECMR .....	734
DARB.....	368	ICCCR .....	569
DBR.....	41	ICMAR .....	569
DCPCFG.....	879	ICMCR .....	564
DCPCTR.....	881	ICMIER .....	568
DCPMAXP .....	880	ICMSR.....	566
DEVADDn .....	930	ICR0.....	419
DMAOR0 .....	378	ICR1.....	421
DMARS.....	381	ICRXD.....	571
DVSTCTR.....	818	ICSAR.....	563
ECMR .....	705	ICSCR.....	558
ECSIPR .....	711	ICSIER.....	562
ECSR .....	709	ICSSR .....	559
EDMR .....	753	ICTXD .....	571
EDRRR.....	755	INT2A0.....	432
EDTRR.....	754	INT2A01 .....	434
EESIPR.....	763	INT2A1 .....	436
EESR .....	758	INT2A11 .....	438
EXPEVT .....	97	INT2B .....	448
FCETR.....	781	INT2GPIC.....	453
FDR .....	772	INT2MSKCR.....	444
FLADR.....	1291	INT2MSKCR1 .....	446
FLADR2.....	1293	INT2MSKR .....	440
FLBSYCNT.....	1302	INT2MSKR1 .....	442
FLBSYTMR.....	1301	INT2PRI .....	430
FLCMCDR.....	1290	INTENB0.....	841
FLCMDCR.....	1287	INTENB1 .....	843
FLCMNCR.....	1284	INTEVT .....	98
FLDATAR.....	1295	INTMSK0.....	424
FLDTCNTR .....	1294	INTMSKCLR0 .....	425
FLDTFIFO .....	1303	INTPRI .....	422
FLECFIFO.....	1304	INTREQ.....	423

INTSTS0.....	852	PFTCR.....	732, 733
INTSTS1.....	858	PIPEBUF.....	899
IOSR.....	785	PIPECFG.....	892
IRMCR.....	168	PIPEMAXP.....	902
LCCR.....	719, 720	PIPEnCTR.....	906
LDACLNR.....	1024	PIPEnTRE.....	926
LDCNTR.....	1032	PIPEnTRN.....	928
LDDFR.....	1012	PIPEPERI.....	904
LDHCNR.....	1019	PIPESEL.....	891
LDHSYNR.....	1020	PIR.....	712
LDICKR.....	1007	PLLCR.....	253
LDINTR.....	1025	PR.....	41
LDLAOR.....	1016	PRR.....	1664
LDLIRNR.....	1036	PSR.....	716
LDMTR.....	1009	PTEH.....	159
LDPALCR.....	1017	PTEL.....	160
LDPMMR.....	1028	PVR.....	1663
LDPR.....	1018	QACR0.....	214
LDPSPR.....	1030	QACR1.....	215
LDSARL.....	1015	RAMCR.....	216, 239
LDSARU.....	1014	RBWAR.....	777
LDUINTLNR.....	1035	RDFAR.....	778
LDUINTR.....	1033	RDLAR.....	757
LDVDLNR.....	1021	RFCR.....	725
LDVSYNR.....	1023	RFLR.....	715
LDVTLNR.....	1022	RFOCR.....	776
MACH.....	41	RMCR.....	773
MACL.....	41	RMFCR.....	769
MAFCR.....	726	RPADIR.....	783
MAHR.....	713	SAR.....	366
MALR.....	714	SARB.....	367
MMUCR.....	162	SCBRR.....	513
MPR.....	729	SCEMR.....	526
MSTPCR0.....	1415	SCFCR.....	518
MSTPCR1.....	1419	SCFDR.....	521
NMIFCR.....	426	SCFRDR.....	496
NRDYENB.....	847	SCFSR.....	505
NRDYSTS.....	866	SCFTDR.....	497
PASCR.....	167	SCLSR.....	525
PC.....	41	SCRSR.....	496

SCSCR.....	501	TSFRCR.....	723
SCSMR.....	498	TSTR.....	475
SCSPTR.....	522	TTB.....	161
SCTSR.....	497	UFRMNUM.....	872
SDBSR.....	1485	USBADDR.....	873
SDINT.....	1484	USBINDX.....	877
SDIR.....	1483	USBLENG.....	878
SGR.....	41	USBREQ.....	874
SOFCFG.....	851	USBVAL.....	876
SPC.....	41	USERIMASK.....	428
SR.....	39	VBR.....	41
SRCCTRL.....	1325	VDC2CLKCR.....	254
SRCID.....	1320	Relative priorities.....	102
SRCIDCTRL.....	1322	Reset state.....	47
SRCOD.....	1321	Rounding.....	143
SRCODCTRL.....	1323	Round-robin mode.....	388
SRCSTAT.....	1328		
SSR.....	41	<b>S</b>	
STBCR.....	1414	Sampling rate converter (SRC).....	1317
SYSCFG.....	811	SCIF interrupt sources.....	549
SYSSTS.....	816	SCIF Module Timing.....	1608
TBRAR.....	779	SDHI Module Timing.....	1641
TCNT.....	477	Sector access mode.....	1311
TCOR.....	477	Sending a break signal.....	551
TCPR2.....	480	Sequential break.....	1465
TCR.....	368, 478	Serial communication interface with	
TCRB.....	369	FIFO (SCIF).....	489
TDFAR.....	780	Serial Sound Interface (SSI).....	619
TDLAR.....	756	Setting the display resolution.....	1044
TEA.....	162	Share status bit.....	171
TESTMODE.....	823	Shift instructions.....	64
TFTR.....	770	Sign-extended.....	46
TFUCR.....	775	Single virtual memory mode.....	156
TLFRCR.....	724	Single-precision floating-point	
TOCR.....	474	extended.....	37
TPAUSER.....	730, 731	Single-precision floating-point	
TRA.....	96	extended register matrix.....	38
TRIMD.....	784	Single-precision floating-point	
TROCR.....	717	registers.....	37
TRSCER.....	766		

Single-precision floating-point vector registers.....	37
Sleep mode .....	1420
Slot FPU disable exception.....	123
Slot illegal instruction exception .....	121
Software standby mode.....	1421
SSI Module Timing .....	1609
STIF ModuleSignal Timing.....	1652
System control instructions.....	65
System registers .....	33
System registers related to FPU.....	33

## T

T bit .....	52
TAP control .....	1506
TCNT Count Timing .....	483
Timer Unit (TMU).....	469
Transceiver Timing.....	1640
Transmit descriptor.....	786
Types of exceptions .....	102

## U

Unconditional trap .....	119
Underflow .....	144
USB 2.0 host/function module (USB) ....	801
User break controller.....	1441
User break operation.....	1461
User debugging interface .....	1477
User mode .....	32
UTLB .....	170
UTLB address array .....	202
UTLB data array .....	203

## V

Validity bit .....	171
Vector addresses .....	102
Video display controller (VDC2).....	1221
Virtual address space .....	152
VPN .....	170

## W

Watchdog Timer and Reset.....	1425
Watchdog Timer Timing.....	1605
Write-through bit .....	172





---

# **Renesas 32-Bit RISC Microcomputer Hardware Manual SH7764 Group**

Publication Date: Rev.1.00, Nov. 22, 2007  
Published by: Sales Strategic Planning Div.  
Renesas Technology Corp.  
Edited by: Customer Support Department  
Global Strategic Communication Div.  
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

---



## RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

### **Renesas Technology America, Inc.**

450 Holger Way, San Jose, CA 95134-1368, U.S.A  
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

### **Renesas Technology Europe Limited**

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

### **Renesas Technology (Shanghai) Co., Ltd.**

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120  
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

### **Renesas Technology Hong Kong Ltd.**

7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong  
Tel: <852> 2265-6688, Fax: <852> 2730-6071

### **Renesas Technology Taiwan Co., Ltd.**

10th Floor, No.99, Fushing North Road, Taipei, Taiwan  
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

### **Renesas Technology Singapore Pte. Ltd.**

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: <65> 6213-0200, Fax: <65> 6278-8001

### **Renesas Technology Korea Co., Ltd.**

Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea  
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

### **Renesas Technology Malaysia Sdn. Bhd**

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: <603> 7955-9390, Fax: <603> 7955-9510



# SH7764 Group Hardware Manual



Renesas Technology Corp.

2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan