To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

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

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

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

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

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

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

User's Manual

32

SH7712

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperHTM RISC engine Family / SH7700 Series

SH7712 HD6417712

Renesas Electronics
www.renesas.com

Rev.1.00 2005.12

a third party.

2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any party's rights, originating in the use of any product data, diagrams, charts, programs, algorithm circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, program algorithms represents information on products at the time of publication of these materials, and subject to change by Renesas Technology Corp. without notice due to product improvements other reasons. It is therefore recommended that customers contact Renesas Technology Co an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a to system before making a final decision on the applicability of the information and products. Re Technology Corp. assumes no responsibility for any damage, liability or other loss resulting fr information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a de system that is used under circumstances in which human life is potentially at stake. Please c Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor w considering the use of a product contained herein for any specific purposes, such as apparat systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they be exported under a license from the Japanese government and cannot be imported into a co other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the produc contained therein.

RENESAS

are in their open states, intermediate levels are induced by noise in the vicinity, a 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 throughou chip and a low level is input on the reset pin. During the period where the states a undefined, the register settings and the output state of each pin are also undefined your system so that it does not malfunction because of processing while it is in th undefined state. For those products which have a reset function, reset the LSI imr 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 reg may have been be allocated to these addresses. Do not access these registers; the operation is not guaranteed if they are accessed.

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

The configuration of the functional description of each module differs according to th 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. Eac includes notes in relation to the descriptions given, and usage notes are given, as required final part of each section.

7. List of Registers
8. Electrical Characteristics
9. Appendix
10. Main Revisions and Additions in this Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier vers This does not include all of the revised contents. For details, see the actual locations in th manual.

11. Index

characteristics of this LSI to the above users.
Refer to the SH-3/SH-3E/SH3-DSP Programming Manual for a detailed de
of the instruction set.

Notes on reading this manual:

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

Product Classifications and Abbreviations

Basic Classification	Product Code
SH7712	HD6417712

- In order to understand the overall functions of the chip

  Read the manual according to the contents. This manual can be roughly categorized in
  on the CPU, system control functions, peripheral functions and electrical characteristi

- In order to understand the details of the CPU's functions

  Read the SH-3/SH-3E/SH3-DSP Programming Manual.

Signal notation: An overbar is added to a low-active signal: xxxx

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

SH7712 manuals:

Document Title	Document
SH7712 Hardware Manual	This manua
SH-3/SH-3E/SH3-DSP Programming Manual	ADE-602-0

User's manuals for development tools:

Document Title	Document
SuperH™ RISC engine C/C++ Compiler,Assembler,Optimizing Linkage Editor Compiler Package V.9.00 User's Manual	REJ10B015
SuperH™ RISC engine High-performance Embedded Workshop 3 Users Manual	REJ10B002
SuperH RISC engine High-Performance Embedded Workshop 3 Tutorial	REJ10B002

Application note:

Document Title	Document
SuperH RISC engine C/C++ Compiler Package Application Note	REJ05B046

FIFO	First-in first-out
H-UDI	User debugging interface
INTC	Interrupt controller
JTAG	Joint test action group
LSB	Least significant bit
MMU	Memory management unit
MSB	Most significant bit
PFC	Pin function controller
RISC	Reduced instruction set computer
RTC	Realtime clock
SCIF	Serial communication interface with FIFO
SIOF	Serial I/O with FIFO
TLB	Translation lookaside buffer
TMU	Timer unit
UART	Universal asynchronous receiver/transmitter
UBC	User break controller
WDT	Watchdog timer

Rev. 1.00 Dec. 27, 2005 P

Section 3 DSP Operating Unit

Section 12 Bus State Controller (BSC)

Section 13 Direct Memory Access Controller (DMAC)

Appendix

Section 3 DSP Operating Unit

The LSI has a large capacity (32-kbyte) cache memory, 16-kbyte on-chip X/Y memory, interrupt controller for system configuration to enable flexible system design. It supports speed data transfer using an on-chip direct memory access controller (DMAC). Its exter memory access support provides direct connection to various types of memory.

The strong on-chip power saving function reduces power consumption even during high operation.

1.1 Features

The features of this LSI are shown below.

CPU:

- Original Renesas-Technology SuperH architecture
- Compatible with SH-1, SH-2, and SH-3 at object code level
- 32-bit internal data bus
- Various groups of built-in registers
  General registers: Sixteen 32-bit registers (including eight 32-bit bank registers)
  Control registers: Five 32-bit registers
  System registers: Four 32-bit registers
- Supports RISC-type instruction set
  Instruction length: 16-bit fixed length for improved code efficiency
  Load/store architecture
  Delayed branch instructions
  Instruction set based on C language
- Instruction execution time: one instruction/cycle for basic instructions
- Logical address space: 4 Gbytes

Six 32-bit data registers
Two 40-bit data registers
- Supports extended harvard architecture for DSP data bus
  Two data buses
  One instruction bus
- Maximum four parallel operations
  ALU, multiply, and two load/store
- Two addressing units to generate addresses for two memory access
- Supports DSP data addressing modes
  Increment and indexing (with or without modulo addressing)
- Zero-overhead repeat loop control
- Conditional execution instructions
- Supports user DSP mode and privileged DSP mode

Memory management unit (MMU):

- 4 Gbytes of address space, 256 address spaces (8-bit ASID)
- Page unit sharing
- Supports multiple page sizes
  1 kbyte or 4 kbytes
- Supports 128-entry, 4-way set associative TLB
- Supports software selection of replacement way and random-replacement algorithms
- Contents of TLB are directly accessible by address mapping

Maximum two 16-bit accesses from the DSP
8-/16-/32-bit access from the DMAC or E-DMAC

- A total of 16 kbytes memory (8-kbyte RAM each for X- and Y-memory)

Interrupt controller (INTC):

- Supports seven external interrupt pins (NMI, IRQ5 to IRQ0)
- Supports fifteen level interrupt pins ($\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$)
- Supports one interrupt request output pin ($\overline{\text{IRQOUT}}$)
- On-chip peripheral interrupt: Priority level is independently selected for each modul
- Supports software vector mode
- Selection of falling/rising/high/low

User break controller (UBC):

- Address, data value, access type, and data size are available for setting as break cond
- Supports the sequential break function
- Two break channels

On-Chip Oscillation Circuits:

- Clock source selectable between an external supply (EXTAL or CKIO) and crystal r
  The internal clock and peripheral clock can be adjusted by setting the PLL circuit an
  ratio.
- Three types of clocks generated:
  CPU clock (I clock): 200 MHz (max)
  Bus clock (B clock): 66 MHz (max)
  Peripheral clock (P clock): 33 MHz (max)

- Physical address space is divided into eight areas: area 0, areas 2 to 4, each a maximu Mbytes, and areas 5A, 5B, 6A, 6B; each a maximum of 32 Mbytes.
- The following features are settable for each area.

  Bus size (8, 16 or 32 bits): The supported bus size differs for each area.

  Number of access wait cycles: Numbers of wait-state cycles during reading and writi independently selectable for some areas.

  Setting of idle wait cycles: For the same area or different area.

  Specifying the memory to be connected to each area enables direct connection to SRA SRAM with byte selection, burst ROM (synchronization/asynchronous), SDRAM and PCMCIA.
- Outputs chip select signal ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A/B}$, and $\overline{CS6A/B}$) for correspondi (The CS assert/negate timing can be selected by software.)

Direct memory access controller (DMAC):

- Six channels. Two of these channels (ch0 and ch1) support external requests.
- Supports burst mode and cycle-stealing mode

Timer unit (TMU):

- 3-channel auto reload 32-bit on-chip timer
- 4 types of counter input clocks can be selected

Realtime clock (RTC)*[1]:

- On-chip clock, calendar, and alarm
- On-chip 32 kHz crystal oscillator with 1/256-second resolution (interrupt cycle)

- 64 bytes each for transmit/receive FIFO
- 8-/16-/16-bit stereo-audio input/output supported
- Two channels (SIOF0 and SIOF1)
- DMA transfer
- Frame synchronous signal

Ethernet controller (EtherC):

- MAC (Media Access Control)
- Data frame assembly/disassembly (frame format conforming to IEEE802.3u)
- CSMA/CD link management (collision prevention and collision processing)
- CRC processing
- Full-duplex transmit/receive support
- Detects short frames and long frames
- Conforms to MII (Media Independent Interface) standard *[2]
- Conversion from 8-bit stream data in MAC layer to MII nibble (4-bit) stream
- Station management (STA function)
- 10/100 Mbps transfer rate adjustable
- WOL (Wake-On-LAN) signal output with Magic Packet*[3] detection
- CAM sense signal input

Ethernet Controller Direct Memory Access Controller (E-DMAC):

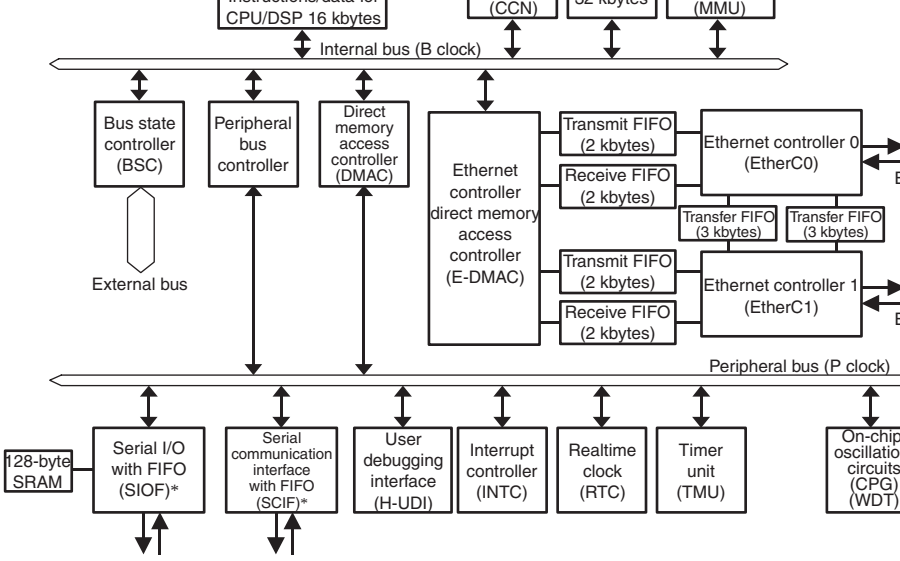
- EtherC — Transfer between external and internal memories
- 16-byte burst transfer
- Single address transfer
- Chain block transfer

supplies even if only RTC operates.

2. +5 V I/O is not supported.
3. Magic Packet is the registered trademark of Advanced Micro Devices Inc.

Product Lineup:

Abbreviation	Power supply voltage		Maximum operating frequency	Type name	Packa
	I/O	Internal			
SH7712	3.3 V ± 0.3 V	1.5 V ± 0.1 V	200 MHz	HD6417712BP/BPV	256-pi (BP-25
				HD6417712F/FV	256-pi (FP-25



Note: * SCIF and SIOF have two channels respectively.

Figure 1.1 Block Diagram

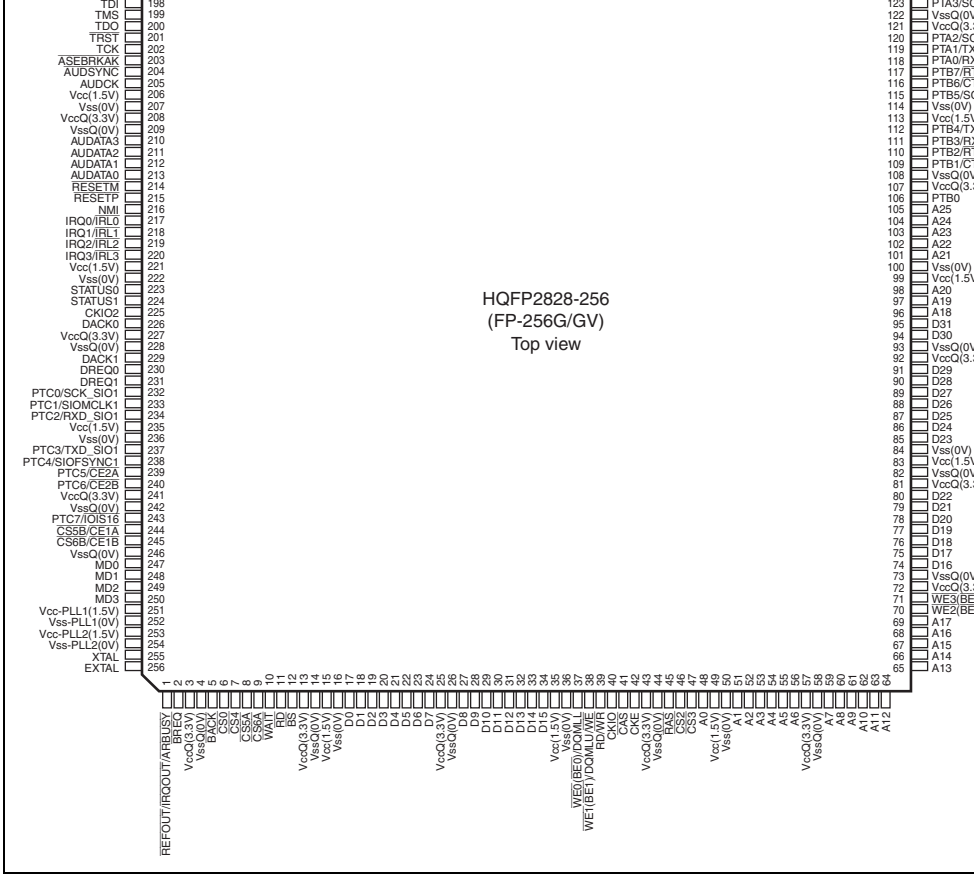
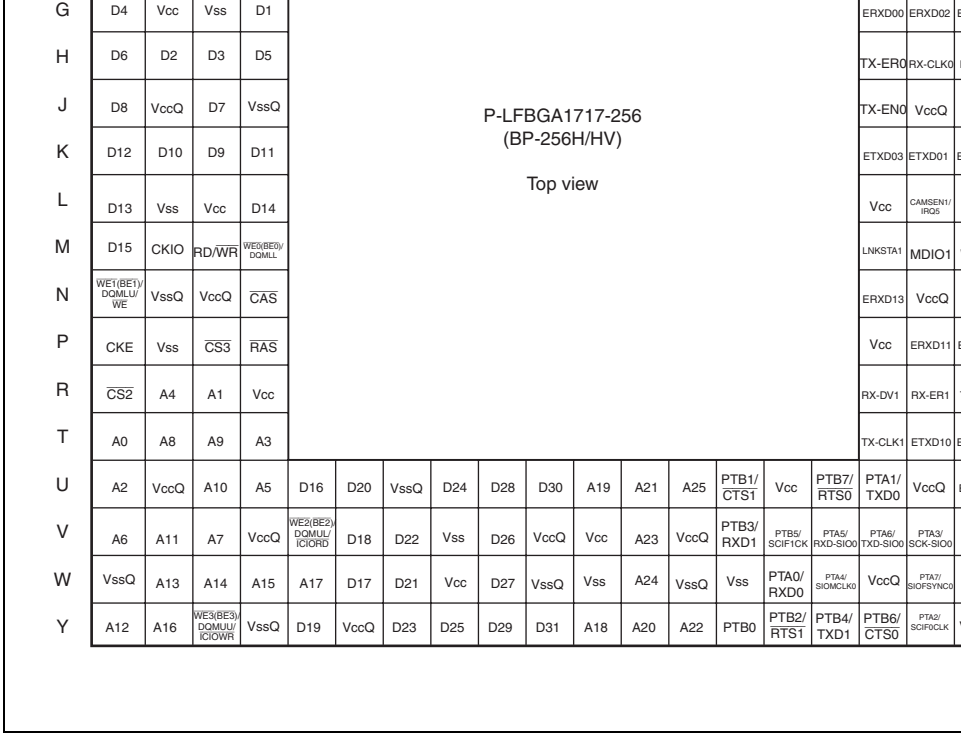


Figure 1.2 Pin Assignment (HQFP2828-256(FP-256G/GV))



G	D4	Vcc	Vss	D1												ERXD00	ERXD02
H	D6	D2	D3	D5												TX-ER0	RX-CLK0
J	D8	VccQ	D7	VssQ												TX-EN0	VccQ
K	D12	D10	D9	D11												ETXD03	ETXD01
L	D13	Vss	Vcc	D14												Vcc	CAMSEN1/ IRQ5
M	D15	CKIO	RD/$\overline{WR}$	WE0(BE0)/ DQMLL												LNKSTA1	MDIO1
N	WE1(BE1)/ DQMLU/ $\overline{WE}$	VssQ	VccQ	$\overline{CAS}$												ERXD13	VccQ
P	CKE	Vss	$\overline{CS3}$	$\overline{RAS}$												Vcc	ERXD11
R	$\overline{CS2}$	A4	A1	Vcc												RX-DV1	RX-ER1
T	A0	A8	A9	A3												TX-CLK1	ETXD10
U	A2	VccQ	A10	A5	D16	D20	VssQ	D24	D28	D30	A19	A21	A25	PTB1/ $\overline{CTS1}$	Vcc	PTB7/ $\overline{RTS0}$	PTA1/ TXD0	VccQ
V	A6	A11	A7	VccQ	WE2(BE2)/ DQMUL/ ICIORD	D18	D22	Vss	D26	VccQ	Vcc	A23	VccQ	PTB3/ RXD1	PTB5/ SCIF1CK	PTA5/ RXD-SIO0	PTA6/ TXD-SIO0	PTA3/ SCK-SIO0
W	VssQ	A13	A14	A15	A17	D17	D21	Vcc	D27	VssQ	Vss	A24	VssQ	Vss	PTA0/ RXD0	PTA4/ SIOMCLK0	VccQ	PTA7/ SIOFSYNC0
Y	A12	A16	WE3(BE3)/ DQMUU/ ICIOWR	VssQ	D19	VccQ	D23	D25	D29	D31	A18	A20	A22	PTB0	PTB2/ $\overline{RTS1}$	PTB4/ TXD1	PTB6/ $\overline{CTS0}$	PTA2/ SCIF0CLK

Figure 1.3 Pin Assignment (P-LFBGA1717-256(BP-256H/HV))

6	E3	CS0	O	Chip select 0
7	C1	$\overline{\text{CS4}}$	O	Chip select 4
8	D3	$\overline{\text{CS5A}}$	O	Chip select 5 A
9	D1	$\overline{\text{CS6A}}$	O	Chip select 6 A
10	E4	$\overline{\text{WAIT}}$	I	Hardware wait request
11	F2	$\overline{\text{RD}}$	O	Read strobe
12	F3	$\overline{\text{BS}}$	O	Bus cycle start signal
13	E1	VccQ		I/O power supply (3.3 V)
14	F4	VssQ		I/O power supply (0 V)
15	G2	Vcc		Internal power supply (1.5 V)
16	G3	Vss		Internal power supply (0 V)
17	F1	D0	IO	Data bus
18	G4	D1	IO	Data bus
19	H2	D2	IO	Data bus
20	H3	D3	IO	Data bus
21	G1	D4	IO	Data bus
22	H4	D5	IO	Data bus
23	H1	D6	IO	Data bus
24	J3	D7	IO	Data bus
25	J2	VccQ		I/O power supply (3.3 V)
26	J4	VssQ		I/O power supply (0 V)
27	J1	D8	IO	Data bus
28	K3	D9	IO	Data bus
29	K2	D10	IO	Data bus

37	M4	WE0(BE0)/DQMLL	O/O	D7 to D0-select signal/DQM (
38	N1	$\overline{WE1}$($\overline{BE1}$)/DQMLU/ $\overline{WE}$	O/O/O	D15 to D8-select signal/DQM (SDRAM)/PCMCIA write cycl
39	M3	RD/$\overline{WR}$	O	Read/write
40	M2	CKIO	IO	System clock I/O
41	N4	$\overline{CAS}$	O	CAS (SDRAM)
42	P1	CKE	O	CK enable (SDRAM)
43	N3	VccQ		I/O power supply (3.3 V)
44	N2	VssQ		I/O power supply (0 V)
45	P4	$\overline{RAS}$	O	RAS (SDRAM)
46	R1	$\overline{CS2}$	O	Chip select 2
47	P3	$\overline{CS3}$	O	Chip select 3
48	T1	A0	O	Address bus
49	R4	Vcc		Internal power supply (1.5 V)
50	P2	Vss		Internal power supply (0 V)
51	R3	A1	O	Address bus
52	U1	A2	O	Address bus
53	T4	A3	O	Address bus
54	R2	A4	O	Address bus
55	U4	A5	O	Address bus
56	V1	A6	O	Address bus
57	U2	VccQ		I/O power supply (3.3 V)
58	W1	VssQ		I/O power supply (0 V)
59	V3	A7	O	Address bus

67	W4	A15	O	Address bus
68	Y2	A16	O	Address bus
69	W5	A17	O	Address bus
70	V5	$\overline{WE2}$($\overline{BE2}$)/DQMUL/ $\overline{ICIORD}$	O/O/O	D23 to D16-select signal/DQM (SDRAM)/PCMCIA I/O read
71	Y3	$\overline{WE3}$($\overline{BE3}$)/DQMUU/ $\overline{ICIOWR}$	O/O/O	D31 to D24-select signal/DQM (SDRAM)/PCMCIA I/O write
72	V4	VccQ		I/O power supply (3.3 V)
73	Y4	VssQ		I/O power supply (0 V)
74	U5	D16	IO	Data bus
75	W6	D17	IO	Data bus
76	V6	D18	IO	Data bus
77	Y5	D19	IO	Data bus
78	U6	D20	IO	Data bus
79	W7	D21	IO	Data bus
80	V7	D22	IO	Data bus
81	Y6	VccQ		I/O power supply (3.3 V)
82	U7	VssQ		I/O power supply (0 V)
83	W8	Vcc		Internal power supply (1.5 V)
84	V8	Vss		Internal power supply (0 V)
85	Y7	D23	IO	Data bus
86	U8	D24	IO	Data bus
87	Y8	D25	IO	Data bus
88	V9	D26	IO	Data bus

96	Y11	A18	O	Address bus
97	U11	A19	O	Address bus
98	Y12	A20	O	Address bus
99	V11	Vcc		Internal power supply (1.5 V)
100	W11	Vss		Internal power supply (0 V)
101	U12	A21	O	Address bus
102	Y13	A22	O	Address bus
103	V12	A23	O	Address bus
104	W12	A24	O	Address bus
105	U13	A25	O	Address bus
106	Y14	PTB0	IO	I/O port B
107	V13	VccQ		I/O power supply (3.3 V)
108	W13	VssQ		I/O power supply (0 V)
109	U14	PTB1/$\overline{\text{CTS1}}$	IO/I	I/O port B/SCIF1 transmit cle
110	Y15	PTB2/$\overline{\text{RTS1}}$	IO/O	I/O port B/SCIF1 transmit req
111	V14	PTB3/RXD1	IO/I	I/O port B/SCIF1 receive data
112	Y16	PTB4/TXD1	IO/O	I/O port B/SCIF1 transmit dat
113	U15	Vcc		Internal power supply (1.5 V)
114	W14	Vss		Internal power supply (0 V)
115	V15	PTB5/SCIF1CK	IO/IO	I/O port B/SCIF1 serial clock
116	Y17	PTB6/$\overline{\text{CTS0}}$	IO/I	I/O port B/SCIF0 transmit cle
117	U16	PTB7/$\overline{\text{RTS0}}$	IO/O	I/O port B/SCIF0 transmit req
118	W15	PTA0/RXD0	IO/I	I/O port A/SCIF0 receive data
119	U17	PTA1/TXD0	IO/O	I/O port A/SCIF0 transmit dat

127	W18	PTA7/SIOFSYNC0	IO/IO	I/O port A/SIOF0 frame sync
128	Y20	VccQ		I/O power supply (3.3 V)
129	W19	CRS1	I	MAC1 carrier detection
130	V19	COL1	I	MAC1 collision detection
131	U19	ETXD13	O	MAC1 transmit data 3
132	W20	ETXD12	O	MAC1 transmit data 2
133	T19	ETXD11	O	MAC1 transmit data 1
134	T18	ETXD10	O	MAC1 transmit data 0
135	V20	TX-EN1	O	MAC1 transmit enable
136	U18	VccQ		I/O power supply (3.3 V)
137	U20	VssQ		I/O power supply (0 V)
138	T17	TX-CLK1	I	MAC1 transmit clock
139	R19	TX-ER1	O	MAC1 transmit error
140	R18	RX-ER1	I	MAC1 receive error
141	T20	RX-CLK1	I	MAC1 receive clock
142	R17	RX-DV1	I	MAC1 receive data valid
143	P19	ERXD10	I	MAC1 receive data 0
144	P18	ERXD11	I	MAC1 receive data 1
145	R20	ERXD12	I	MAC1 receive data 2
146	P17	Vcc		Internal power supply (1.5 V)
147	N19	Vss		Internal power supply (0 V)
148	N18	VccQ		I/O power supply (3.3 V)
149	P20	VssQ		I/O power supply (0 V)

156	L18	CAMSENT/IRQ3	I/I	MAC1 CAM input/external int request
157	L19	CRS0	I	MAC0 carrier detection
158	L17	Vcc		Internal power supply (1.5 V)
159	L20	Vss		Internal power supply (0 V)
160	K20	COL0	I	MAC0 collision detection
161	K17	ETXD03	O	MAC0 transmit data 3
162	J20	ETXD02	O	MAC0 transmit data 2
163	K18	ETXD01	O	MAC0 transmit data 1
164	K19	ETXD00	O	MAC0 transmit data 0
165	J17	TX-EN0	O	MAC0 transmit enable
166	H20	TX-CLK0	I	MAC0 transmit clock
167	J18	VccQ		I/O power supply (3.3 V)
168	J19	VssQ		I/O power supply (0 V)
169	H17	TX-ER0	O	MAC0 transmit error
170	G20	RX-ER0	I	MAC0 receive error
171	H18	RX-CLK0	I	MAC0 receive clock
172	H19	RX-DV0	I	MAC0 receive data valid
173	G17	ERXD00	I	MAC0 receive data 0
174	F20	ERXD01	I	MAC0 receive data 1
175	G18	ERXD02	I	MAC0 receive data 2
176	E20	Vcc		Internal power supply (1.5 V)
177	F17	Vss		Internal power supply (0 V)
178	G19	ERXD03	I	MAC0 receive data 3

185	D19	VccQ		I/O power supply (3.3 V)
186	B20	VssQ		I/O power supply (0 V)
187	C18	VssQ		I/O power supply (0 V)
188	E19	MD4	I	Specifies area 0 bus width
189	E18	MD5	I	Endian select
190	D18	VccQ		I/O power supply (3.3 V)
191	C19	VssQ		I/O power supply (0 V)
192	A20	VccQ		I/O power supply (3.3 V)
193	B19	VccQ-RTC		RTC oscillator power supply (3
194	B18	XTAL2	O	On-chip RTC crystal oscillator
195	B17	EXTAL2	I	On-chip RTC crystal oscillator
196	A19	VssQ-RTC		RTC oscillator power supply (0
197	B16	$\overline{\text{ASEMD0}}$	I	ASE mode
198	C16	TDI	I	Test data input
199	A18	TMS	I	Test mode select
200	C17	TDO	O	Test data output
201	A17	$\overline{\text{TRST}}$	I	Test reset
202	D16	TCK	I	Test clock
203	B15	$\overline{\text{ASEBRKAK}}$	O	ASE break acknowledge
204	C15	$\overline{\text{AUDSYNC}}$	O	AUD synchronous
205	A16	AUDCK	O	AUD clock
206	D15	Vcc		Internal power supply (1.5 V)
207	B14	Vss		Internal power supply (0 V)

215	A13	RESETP	I	Power-on reset request
216	C12	NMI	I	Non-maskable interrupt reque
217	B12	IRQ0/$\overline{\text{IRL0}}$	I	External interrupt request
218	D12	IRQ1/$\overline{\text{IRL1}}$	I	External interrupt request
219	A12	IRQ2/$\overline{\text{IRL2}}$	I	External interrupt request
220	C11	IRQ3/$\overline{\text{IRL3}}$	I	External interrupt request
221	B11	Vcc		Internal power supply (1.5 V)
222	D11	Vss		Internal power supply (0 V)
223	A11	STATUS0	O	Processor status
224	A10	STATUS1	O	Processor status
225	D10	CKIO2	O	System clock output
226	A9	DACK0	O	DMA acknowledge 0
227	C10	VccQ		I/O power supply (3.3 V)
228	B10	VssQ		I/O power supply (0 V)
229	D9	DACK1	O	DMA acknowledge 1
230	A8	DREQ0	I	DMA request 0
231	C9	DREQ1	I	DMA request 1
232	B9	PTC0/SCK_SIO1	IO/IO	I/O port C/SIOF1 communica
233	D8	PTC1/SIOMCLK1	IO/I	I/O port C/SIOF1 clock input
234	A7	PTC2/RXD_SIO1	IO/I	I/O port C/SIOF1 receive data
235	C8	Vcc		Internal power supply (1.5 V)
236	B8	Vss		Internal power supply (0 V)
237	D7	PTC3/TXD_SIO1	IO/O	I/O port C/SIOF1 transmit dat
238	A6	PTC4/SIOFSYNC1	IO/IO	I/O port C/SIOF1 frame sync

245	B5	CS0B/CE1B	O/O	Chip select 0B/area 0 PCMCIA enable
246	B6	VssQ		I/O power supply (0 V)
247	D4	MD0	I	Clock mode select
248	A3	MD1	I	Clock mode select
249	B4	MD2	I	Clock mode select
250	A2	MD3	I	Area 0 bus width
251	C3	Vcc-PLL1		PLL1 power supply (1.5 V)
252	B5	Vss-PLL1		PLL1 power supply (0 V)
253	C5	Vcc-PLL2		PLL2 power supply (1.5 V)
254	C4	Vss-PLL2		PLL2 power supply (0 V)
255	B3	XTAL	O	Clock oscillator pin
256	A1	EXTAL	I	External clock/crystal oscillator

Notes: 1. VccQ-RTC must be supplied even if the realtime clock (RTC) is not used.
2. RTC in this LSI does not operate even if VccQ-RTC is turned on. The crystal o circuit for RTC operates with VccQ-RTC. The control circuit and the RTC coun operate with Vcc (common to the internal circuit). Therefore, all power supplies than VccQ-RTC should always be turned on even if only RTC operates.
3. Vcc-PLL1/Vcc-PLL2 must be supplied even if the on-chip CPG is not used.
4. VccQ (3.3 V), Vcc (1.5 V), VssQ, and Vss must be connected to the system po supply (for uninterrupted supply).

	Vss	—	Ground	Ground pin. Connect all to the system power sup There will be no operatio pins are open.
	VccQ	—	Power supply	Power supply for I/O pins all VccQ pins to the syst supply. There will be no if any pins are open.
	VssQ	—	Ground	Ground pin. Connect all to the system power sup There will be no operatio pins are open.
Clock	Vcc-PLL1	I	PLL1 power supply	Power supply for the on- oscillator.
	Vss-PLL1	I	PLL1 ground	Ground pin for the on-ch oscillator.
	Vcc-PLL2	I	PLL2 power supply	Power supply for the on- oscillator.
	Vss-PLL2	I	PLL2 ground	Ground pin for the on-ch oscillator.
	EXTAL	I	External clock	For connection to a cryst resonator. This pin can b used for external clock in examples of crystal reso connection and external input, see section 11, Or Oscillation Circuits.

Operating mode control	MD5 to MD0	I	Mode set	These pins set the operat mode. Do not change valu these pins during operatio MD2 to MD0 set the clock MD4 and MD3 set the bus mode of area 0, and MD5 endian.
System control	$\overline{\text{RESETP}}$	I	Power-on reset	When low, the system ent power-on reset state.
	$\overline{\text{RESETM}}$	I	Manual reset	When low, the system ent manual reset state.
	STATUS1 STATUS0	O	Status output	Indicates that this LSI is in software standby mode, r sleep.
	$\overline{\text{BREQ}}$	I	Bus request	Low when an external dev requests the release of th mastership.
	$\overline{\text{BACK}}$	O	Bus request acknowledge	Indicates that the bus mas has been released to an e device. Reception of the $\overline{\text{B}}$ signal informs the device has output the $\overline{\text{BREQ}}$ sign has acquired the bus mas

Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D31 to D0	I/O	Data bus	32-bit bidirectional bus.
Bus control	$\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A}$, $\overline{CS6A}$, $\overline{CS5B}$/$\overline{CE1A}$, $\overline{CS6B}$/$\overline{CE1B}$, $\overline{CE2A}$, $\overline{CE2B}$	O	Chip select 0, 2 to 4, 5A, 5B, 6A, 6B PCMCIA card select	Chip-select signals for e memory or devices. PCMCIA card select sign PCMCIA is used.
	$\overline{RD}$	O	Read	Indicates reading of data external devices.
	RD/$\overline{WR}$	O	Read/write	Read/write signal
	$\overline{BS}$	O	Bus start	Bus-cycle start signal
	$\overline{WE3}$($\overline{BE3}$)/ $\overline{ICIOWR}$	O	Byte write	Indicates that bits 31 to 2 data in the external mem device are being written. strobe signal when PCM used.
	$\overline{WE2}$($\overline{BE2}$)/ $\overline{ICIORD}$	O	Byte write	Indicates that bits 23 to 1 data in the external mem device are being written. strobe signal when PCM used.
	$\overline{WE1}$($\overline{BE1}$)/ $\overline{WE}$	O	Byte write	Indicates that bits 15 to 8 data in the external mem device are being written. write strobe signal when is used.

			selection	
	DQMUU	O	DQM	Selects D31 to D24 during a the SDRAM.
	DQMUL	O	DQM	Selects D23 to D16 during a the SDRAM.
	DQMLU	O	DQM	Selects D15 to D8 during ac the SDRAM.
	DQMLL	O	DQM	Selects D7 to D0 during acc the SDRAM.
	$\overline{\text{REFOUT}}$	O	Refresh request output	Outputs the refresh request master mode or bus release
	$\overline{\text{WAIT}}$	I	Wait	Inserts a wait cycle into the cycles during access to the space.
Direct memory access controller (DMAC)	DREQ0, DREQ1	I	DMA-transfer request	Input pin for external reques DMA transfer.
	DACK0, DACK1	O	DMA-transfer request accept	Output pin for request acce in response to external requ DMA transfer.
	TEND0, TEND1	O	DMA-transfer end output	Output pin for DMA transfer signal.

debugger (AUD)	AUDATA0			mode.
	AUDCK	O	AUD clock	Synchronous-clock outp AUD trace mode.
	$\overline{\text{AUDSYNC}}$	O	AUD synchronous signal	Data start-position ackn signal output pin in AUD mode.
E10A interface	$\overline{\text{ASEBRKAK}}$	O	Break mode acknowledge	Indicates that the E10A has entered its break m For the connection with see the SH7712 E10A E User's Manual (tentative
	$\overline{\text{ASEMD0}}$	I	ASE mode	Sets the ASE mode.
Realtime clock (RTC)	VccQ-RTC	I	RTC oscillator power supply	Power supply pin for the RTC
	VssQ-RTC	I	RTC oscillator ground	Ground pin for the on-ch
	EXTAL2	I	RTC external clock	Clock input pin for the o RTC clock (32.768 MHz connection example, ref section 15, Realtime Clo
	XTAL2	O	RTC crystal	Clock output pin for the RTC clock (32.768 MHz connection example, ref section 15, Realtime Clo

			(EtherC).
ETXD03 to ETXD00	O	MAC0 transmit data	4-bit transmit data pins. F connection example, refe section 18, Ethernet Cont (EtherC).
TX-EN1, TX-EN0	O	MAC1/0 transmit enable	These pins indicate that t data is ready on ETXD13 ETXD10 and ETXD03 to For a connection example section 18, Ethernet Cont (EtherC).
TX-CLK1, TX-CLK0	I	MAC1/0 transmit clock	Timing reference pins (cl TX-EN1/0, TX-ER1/0, ET ETXD10 and ETXD03 to For a connection example section 18, Ethernet Cont (EtherC).
TX-ER1, TX-ER0	O	MAC1/0 transmit error	These pins notify an erro transmission to the PHY- a connection example, re section 18, Ethernet Cont (EtherC).
RX-ER1, RX-ER0	I	MAC1/0 receive error	These pins notify an erro data reception. For a con example, refer to section Ethernet Controller (Ethe

			Ethernet Controller (Eth
ERXD13 to ERXD10	I	MAC1 receive data	4-bit receive data pins. F connection example, ref section 18, Ethernet Co (EtherC).
ERXD03 to ERXD00	I	MAC0 receive data	4-bit receive data pins. F connection example, ref section 18, Ethernet Co (EtherC).
MDC1, MDC0	O	MAC1/0 management data clock	Reference clock pins for information transfer via I a connection example, r section 18, Ethernet Co (EtherC).
MDIO1, MDIO0	I/O	MAC1/0 management data I/O	Bidirectional pins for exc management informatio connection example, ref section 18, Ethernet Co (EtherC).
WOL1, WOL0	O	MAC1/0 Wake-On-LAN	These pins indicate that Packet has been receive
LNKSTA1, LNKSTA0	I	MAC1/0 link status	Link state input pins fron PHY-LSI
EXOUT1, EXOUT0	O	MAC1/0 general-purpose external output	External output pins
CAMSEN1, CAMSEN0	I	MAC1/0 CAM input	CAM interface pins inpu

	RXD0		data	
	TXD1, TXD0	O	SCIF1/0 transmit data	Transmit data pins
	SCIF1CK, SCIF0CK	I/O	SCIF1/0 serial clock	Clock I/O pins
Serial I/O with FIFO (SIOF1/0)	SCK_SIO1, SCK_SIO0	I/O	SIOF1/0 communication clock	Transmit/receive commu clock I/O pins
	SIOMCLK1, SIOMCLK 0	I	SIOF1/0 clock input	Master-clock input pins
	RXD_SIO1, RXD_SIO0	I	SIOF1/0 receive data	Receive data pins
	TXD_SIO1, TXD_SIO0	O	SIOF1/0 transmit data	Transmit data pins
	SIOFSYNC1, SIOFSYNC0	I/O	SIOF1/0 Frame synchronous signal	Transmit/receive frame-synchronous-signal I/O p
I/O port	PTA7 to PTA0	I/O	General purpose I/O port A	8-bit general-purpose I/O
	PTB7 to PTB0	I/O	General purpose I/O port B	8-bit general-purpose I/O
	PTC7 to PTC0	I/O	General purpose I/O port C	8-bit general-purpose I/O

Reset State: In the reset state, the CPU is reset. The LSI supports two types of resets: po reset and manual reset. For details on resets, refer to section 4, Exception Handling.

In power-on reset, the registers and internal statuses of all LSI on-chip modules are initi manual reset, the register contents of a part of the LSI on-chip modules, such as the bus controller (BSC), are retained. For details, refer to section 23, List of Registers. The CPU statuses and registers are initialized both in power-on reset and manual reset. After initia the program branches to address H'A0000000 to pass control to the reset processing pro executed.

Exception Handling State: In the exception handling state, the CPU processing flow is temporarily by a general exception or interrupt exception processing. The program coun and status register (SR) are saved in the save program counter (SPC) and save status reg (SSR), respectively. The program branches to an address obtained by adding a vector of vector base register (VBR) and passes control to the exception processing program defi user to be executed. For details on reset, refer to section 4, Exception Handling.

Program Execution State: The CPU executes programs sequentially.

Low-Power Consumption State: The CPU stops operation to reduce power consumpti low-power consumption state can be entered by executing the SLEEP instruction. For d the low-power consumption state, refer to section 10, Power-Down Modes.

Figure 2.1 shows a status transition diagram.

resources from the user program. To change the processing mode from user to privileged transition to exception handling state is required*.

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

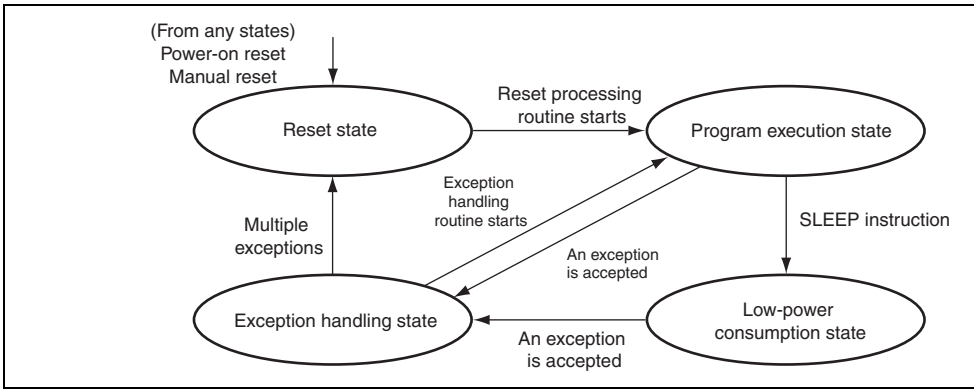


Figure 2.1 Processing State Transitions

areas are handled as address translatable areas.

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

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

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

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

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

P4 Area: The P4 area is defined as a control area which is non-cacheable and non-addre
translatable. This area can be accessed only in privileged mode. A part of the LSI's on-c
assigned to this area.

H'C0000000 to H'DFFFFFFF	P3	Privileged mode	0.5-Gbyte physical space, cacheable, translatable
H'E0000000 to H'FFFFFFFF	P4	Privileged mode	0.5-Gbyte control space, non-cacheab

2.2.2 External Memory Space

The LSI uses 29 bits of the 32-bit logical address to access external memory. In this case, Gbyte of external memory space can be accessed. The external memory space is manage units. Different types of memory can be connected to each area, as shown in figure 2.2. F details, please refer to section 12, Bus State Controller (BSC).

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

Normally, the upper three bits of the 32-bit logical address are masked and the lower 29 b used for external memory addresses.*[2] For example, address H'00000100 in the P0 area, a H'80000100 in the P1 area, address H'A0000100 in the P2 area, and address H'C000010 area of the logical address space are mapped into address H'00000100 of area 0 in the ex memory space. The P4 area in the logical address space is not mapped into the external m address. If an address in the P4 area is accessed, an external memory cannot be accessed.

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

2. If the address translation unit is enabled, arbitrary mapping in page units can b specified. For details, refer to section 5, Memory Management Unit (MMU).

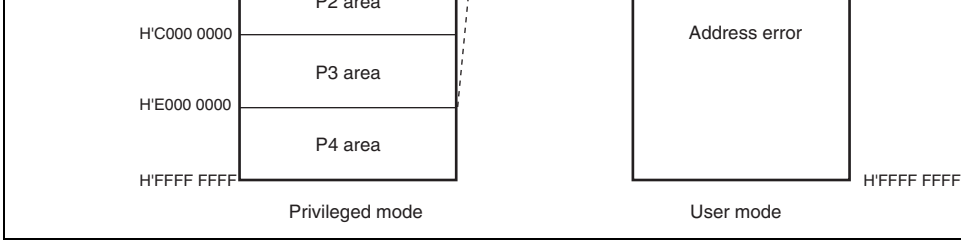


Figure 2.2 Logical Address to External Memory Space Mapping

and procedure register (PR) as system registers. These registers can be accessed regardles
processing mode.

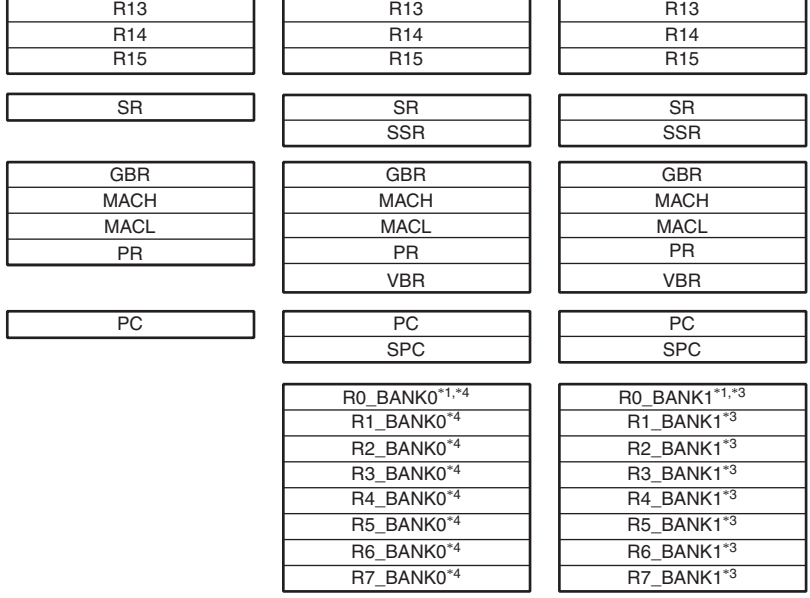
Program Counter: The program counter stores the value obtained by adding 4 to the cu
instruction address.

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

Table 2.2 shows the register values after reset. Figure 2.3 shows the register configuratio
process mode.

	= undefined
GBR, SSR, SPC	Undefined
VBR	H'00000000

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



(a) User mode register configuration

(b) Privileged mode register configuration (RB = 1)

(c) Privileged mode register configuration (RB = 0)

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

Figure 2.3 Register Configuration in Each Processing Mode

R7_BANK0 and R8 to R15 are accessed as general registers R0 to R15. The R0_BANK
R7_BANK1 registers in bank 1 cannot be accessed.

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

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

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

R11
R12
R13
R14
R15

Figure 2.4 General Registers

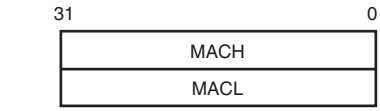
2.3.2 System Registers

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

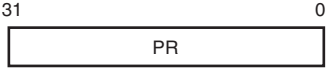
Multiply and Accumulate Registers (MACH/MACL): The multiply and accumulate re
(MACH/MACL) store the results of multiplication and accumulation instructions or mult
instructions. The MACH/MACL registers also store addition values for the multiplication
accumulations. After reset, these registers are undefined.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

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

Multiply and accumulate high and low registers (MACH/MACL)



Procedure register (PR)



Program counter (PC)

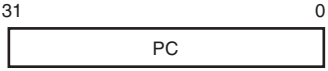


Figure 2.5 System Registers and Program Counter

Bit	Bit Name	Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value shou always be 0.
30	MD	1	R/W	Processing Mode Indicates the CPU processing mode. 0: User mode 1: Privileged mode The MD bit is set to 1 in reset or exception handl
29	RB	1	R/W	Register Bank The general registers R0 to R7 are banked regist RB bit selects a bank used in the privileged mode 0: Selects bank 0 registers. In this case, R0_BAN R7_BANK0 and R8 to R15 are used as genera registers. R0_BANK1 to R7_BANK1 can be accessed by LDC or STR instruction. 1: Selects bank 1 registers. In this case, R0_BAN R7_BANK1 and R8 to R15 are used as genera registers. R0_BANK0 to R7_BANK0 can be accessed by LDC or STR instruction. The RB bit is set to 1 in reset or exception handli

				always be 0.
9	M	—	R/W	M Bit
8	Q	—	R/W	Q Bit These bits are used by the DIV0S, DIV0U, and D instructions. These bits can be changed even in mode by using the DIV0S, DIV0U, and DIV1 inst These bits are undefined at reset. These bits do change in an exception handling state.
7 to 4	I3 to I0	All 1	R/W	Interrupt Mask Indicates the interrupt mask level. These bits do change even if an interrupt occurs. At reset, thes initialized to B'1111. These bits are not affected exception handling state.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
1	S	—	R/W	Saturation Mode Specifies the saturation mode for multiply instruc multiply and accumulate instructions. This bit ca specified by the SETS and CLRS instructions in mode. At reset, this bit is undefined. This bit is not affec exception handling state.

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

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

Global Base Register (GBR): The global base register (GBR) is referenced as a base reg
GBR indirect addressing mode. At reset, the GBR initial value is undefined.

Vector Base Register (VBR): The global base register (GBR) can be accessed only in pr
mode. If a transition from reset state to exception handling state occurs, this register is ref
as a base address. For details, refer to section 4, Exception Handling. At reset, the VBR is
initialized as H'00000000.

Figure 2.6 shows the control register configuration.

Figure 2.6 Control Register Configuration

2.4.2 Memory Data Formats

Memory data formats are classified into byte, word, and longword. Memory can be acces
byte, word, and longword. When the memory operand is only a byte (8 bits) or a word (1
is sign-extended into a longword when loaded into a register.

An address error will occur if word data starting from an address other than 2n or longwo
starting from an address other than 4n is accessed. In such cases, the data accessed canno
guaranteed.

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

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

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

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

The little endian mode can also be specified as data format. Either big-endian or little-en mode can be selected according to the external pin (MD5) at a power-on reset. When M at reset, the processor operates in big-endian mode. When MD5 is high at reset, the proc operates in little-endian mode. The endian mode cannot be modified dynamically.

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

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

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

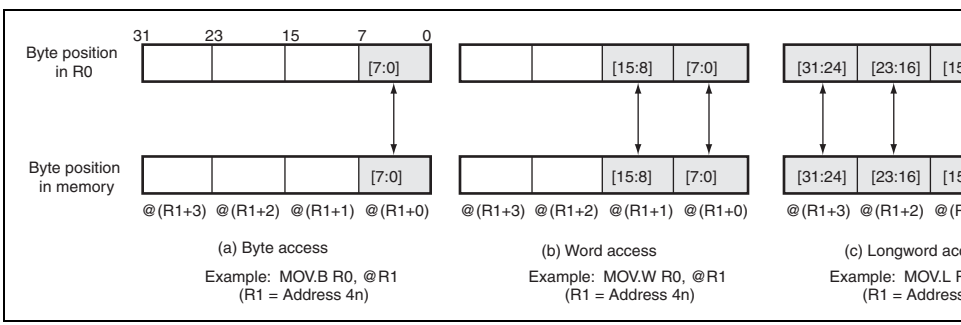


Figure 2.8 Data Format on Memory (Little Endian Mode)

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

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

Delayed Branching: Unconditional branch instructions are executed as delayed branches
delayed branch instruction, the branch is made after execution of the instruction (called th
instruction) immediately following the delayed branch instruction. This minimizes disrup
the pipeline when a branch is made.

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

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

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

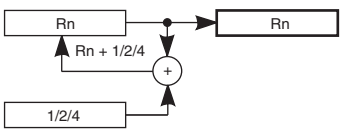
```
Example:  ADD      #1, R0  ; The T bit cannot be modified by the ADD inst
          CMP/EQ   #0, R0  ; The T bit is set to 1 if R0 is 0.
          BT       Target  ; Branch to TARGET if the T bit is set to 1 (R
```

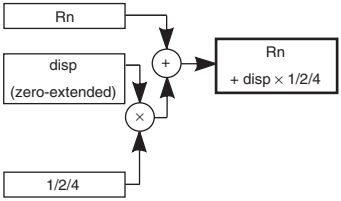
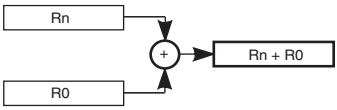
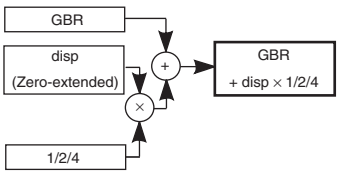
16-Bit/32-Bit Displacement: When data is referenced with a 16- or 32-bit displacement displacement value is placed in a table in memory beforehand. Using the method wherel longword immediate data is loaded when an instruction is executed, this value is transfe register and the data is referenced using indexed register indirect addressing mode.

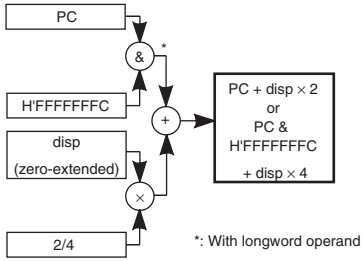
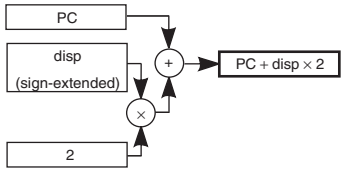
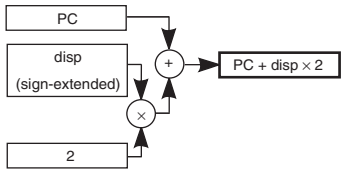
2.5.2 CPU Instruction Addressing Modes

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

Table 2.3 Addressing Modes and Effective Addresses for CPU Instructions

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

Register indirect with displacement	@(disp:4, Rn)	Effective address is register Rn contents with 4-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Byte: Rn + disp Word: Rn + disp Longword: Rn +
Indexed register indirect	@(R0, Rn)	Effective address is sum of register Rn and R0 contents. 	Rn + R0
GBR indirect with displacement	@(disp:8, GBR)	Effective address is register GBR contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Byte: GBR + dis Word: GBR + di Longword: GBR + 4

		size. With a longword operand, the lower 2 bits of PC are masked.  	disp × 4
PC-relative	disp:8	Effective address is PC with 8-bit displacement disp added after being sign-extended and multiplied by 2.  	PC + disp
	disp:12	Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2  	PC + disp

#imm:8	8-bit immediate data imm of TRAPA instruction is zero-extended and multiplied by 4.	—

Note: For addressing modes with displacement (disp) as shown below, the assembler de in this manual indicates the value before it is scaled (x 1, x2, or x4) according to th operand size to clarify the LSI operation. For details on assembler description, refe description rules in each assembler.

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

2.5.3 CPU Instruction Formats

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

xxxx: Instruction code

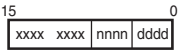
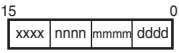
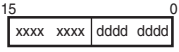
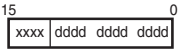
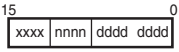
mmmm: Source register

nnnn: Destination register

iiii: Immediate data

dddd: Displacement

	system register	direct		
	Control register or system register	nnnn: pre-decrement register indirect	STC.L	SR,@
m type  	mmmm: register direct	Control register or system register	LDC	Rm,S
	mmmm: post-increment register indirect	Control register or system register	LDC.L	@Rm
	mmmm: register indirect	—	JMP	@Rm
	PC-relative using Rm	—	BRAF	Rm
nm type  	mmmm: register direct	nnnn: register direct	ADD	Rm,R
	mmmm: register indirect	nnnn: register indirect	MOV.L	Rm,@
	mmmm: post-increment register indirect (multiply-and-accumulate operation) nnnn: * post-increment register indirect (multiply-and-accumulate operation)	MACH, MACL	MAC.W	@Rm-

		register indirect with displacement		
nd4 type 	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B	R0,@(d
nmd type 	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L	Rm,@(d
	mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L	@(disp,
d type 	dddddddd: GBR indirect with displacement	R0 (register direct)	MOV.L	@(disp,
	R0 (register direct)	dddddddd: GBR indirect with displacement	MOV.L	R0,@(di
	dddddddd: PC-relative with displacement	R0 (register direct)	MOVA	@(disp,P
	dddddddd: PC-relative	—	BF	label
d12 type 	ddddddddddddd: PC-relative	—	BRA	label (label=di
nd8 type 	dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L	@(disp,

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

Type	Instruction	Op Code	Function	Inst
Data transfer instructions	5	MOV	Data transfer	39
			Immediate data transfer	
			Peripheral module data transfer	
			Structure data transfer	
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Upper/lower swap	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operation instructions	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Signed division initialization	
		DIV0U	Unsigned division initialization	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	

		SUB	Binary subtraction	
		SUBC	Binary subtraction with carry	
		SUBV	Binary subtraction with underflow	
Logic operation instructions	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit setting	
		TST	Logical AND and T bit setting	
		XOR	Exclusive logical OR	
Shift instructions	12	ROTL	1-bit left shift	16
		ROTR	1-bit right shift	
		ROTCL	1-bit left shift with T bit	
		ROTCR	1-bit right shift with T bit	
		SHAL	Arithmetic 1-bit left shift	
		SHAR	Arithmetic 1-bit right shift	
		SHLL	Logical 1-bit left shift	
		SHLLn	Logical n-bit left shift	
		SHLR	Logical 1-bit right shift	
		SHLRn	Logical n-bit right shift	
		SHAD	Arithmetic dynamic shift	
		SHLD	Logical dynamic shift	

		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control instructions	15	CLRT	T bit clear	75
		CLRMAC	MAC register clear	
		CLRS	S bit clear	
		LDC	Load into control register	
		LDS	Load into system register	
		LDTLB	PTEH/PTEL load into TLB	
		NOP	No operation	
		PREF	Data prefetch to cache	
		RTE	Return from exception handling	
		SETS	S bit setting	
		SETT	T bit setting	
		SLEEP	Transition to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	68			188

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

Rn:	Destination register	iiii:	Immediate data
imm:	Immediate data	dddd:	Displacement*2
disp:	Displacement		

|: Logical OR of each bit

^: Exclusive logical OR of each bit

~: Logical NOT of each bit

<<n: n-bit left shift

>>n: n-bit right shift

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

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

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

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

MOV.W	Rm,@Rn	0010nnnnmmmm0001	Rm→(Rn)	—	1
MOV.L	Rm,@Rn	0010nnnnmmmm0010	Rm→(Rn)	—	1
MOV.B	@Rm,Rn	0110nnnnmmmm0000	(Rm)→Sign extension→Rn	—	1
MOV.W	@Rm,Rn	0110nnnnmmmm0001	(Rm)→Sign extension→Rn	—	1
MOV.L	@Rm,Rn	0110nnnnmmmm0010	(Rm)→Rn	—	1
MOV.B	Rm,@–Rn	0010nnnnmmmm0100	Rn–1→Rn, Rm→(Rn)	—	1
MOV.W	Rm,@–Rn	0010nnnnmmmm0101	Rn–2→Rn, Rm→(Rn)	—	1
MOV.L	Rm,@–Rn	0010nnnnmmmm0110	Rn–4→Rn, Rm→(Rn)	—	1
MOV.B	@Rm+,Rn	0110nnnnmmmm0100	(Rm)→Sign extension→Rn, Rm+1→Rm	—	1
MOV.W	@Rm+,Rn	0110nnnnmmmm0101	(Rm)→Sign extension→Rn, Rm+2→Rm	—	1
MOV.L	@Rm+,Rn	0110nnnnmmmm0110	(Rm)→Rn, Rm+4→Rm	—	1
MOV.B	R0,@(disp,Rn)	10000000nnnndddd	R0→(disp+Rn)	—	1
MOV.W	R0,@(disp,Rn)	10000001nnnndddd	R0→(disp x 2+Rn)	—	1
MOV.L	Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm→(disp x 4+Rn)	—	1
MOV.B	@(disp,Rm),R0	10000100mmmmdddd	(disp+Rm)→Sign extension→R0	—	1
MOV.W	@(disp,Rm),R0	10000101mmmmdddd	(disp x 2+Rm)→Sign extension→R0	—	1
MOV.L	@(disp,Rm),Rn	0101nnnnmmmmdddd	(disp x 4+Rm)→Rn	—	1
MOV.B	Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm→(R0+Rn)	—	1
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm→(R0+Rn)	—	1
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm→(R0+Rn)	—	1

MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp+GBR)→Sign extension→R0	—	1
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp x 2+GBR)→Sign extension→R0	—	1
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp x 4+GBR)→R0	—	1
MOVA	@(disp,PC),R0	11000111dddddddd	disp x 4+PC→R0	—	1
MOVT	Rn	0000nnnn00101001	T→Rn	—	1
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm→Swap lowest two bytes→Rn	—	1
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm→Swap two consecutive words→Rn	—	1
XTRCT	Rm,Rn	0010nnnnmmmm1101	Rm: Middle 32 bits of Rn →Rn	—	1

CMP/EQ	Rm,Rn	0011nnnnmmmm0000	If Rn = Rm, 1 → T	—	1
CMP/HS	Rm,Rn	0011nnnnmmmm0010	If Rn ≥ Rm with unsigned data, 1 → T	—	1
CMP/GE	Rm,Rn	0011nnnnmmmm0011	If Rn ≥ Rm with signed data, 1 → T	—	1
CMP/HI	Rm,Rn	0011nnnnmmmm0110	If Rn > Rm with unsigned data, 1 → T	—	1
CMP/GT	Rm,Rn	0011nnnnmmmm0111	If Rn > Rm with signed data, 1 → T	—	1
CMP/PL	Rn	0100nnnn00010101	If Rn ≥ 0, 1 → T	—	1
CMP/PZ	Rn	0100nnnn00010001	If Rn > 0, 1 → T	—	1
CMP/STR	Rm,Rn	0010nnnnmmmm1100	If Rn and Rm have an equivalent byte, 1 → T	—	1
DIV1	Rm,Rn	0011nnnnmmmm0100	Single-step division (Rn/Rm)	—	1
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	—	1
DIV0U		0000000000011001	0 → M/Q/T	—	1
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	—	2 (to 5)*
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	—	2 (to 5)*

			→ Rn		
MAC.L	@Rm+, @Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC,Rn + 4 → Rn, Rm + 4 → Rm, 32 × 32 + 64 → 64 bits	—	2 (to 5)*
MAC.W	@Rm+, @Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC,Rn + 2 → Rn, Rm + 2 → Rm, 16 × 16 + 64 → 64 bits	—	2 (to 5)*
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL, 32 × 32 → 32 bits	—	2 (to 5)*
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL, 16 × 16 → 32 bits	—	1(to 3)*
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL, 16 × 16 → 32 bits	—	1(to 3)*
NEG	Rm,Rn	0110nnnnmmmm1011	0–Rm→Rn	—	1
NEGC	Rm,Rn	0110nnnnmmmm1010	0–Rm–T→Rn, Borrow→T	—	1
SUB	Rm,Rn	0011nnnnmmmm1000	Rn–Rm→Rn	—	1
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn–Rm–T→Rn, Borrow →T	—	1
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn–Rm→Rn, Underflow→T	—	1

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

OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	—	1
OR.B	#imm,@(R0, GBR)	11001111iiiiiiii	(R0+GBR) | imm → (R0+GBR)	—	3
TAS.B	@Rn	0100nnnn00011011	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)	—	4
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm; if the result is 0, 1 → T	—	1
TST	#imm,R0	11001000iiiiiiii	R0 & imm; if the result is 0, 1 → T	—	1
TST.B	#imm,@(R0, GBR)	11001100iiiiiiii	(R0 + GBR) & imm; if the result is 0, 1 → T	—	3
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	—	1
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	—	1
XOR.B	#imm,@(R0, GBR)	11001110iiiiiiii	(R0+GBR) ^ imm → (R0+GBR)	—	3

SHAL	Rn	0100nnnn00100000	T←Rn←0	—	1
SHAR	Rn	0100nnnn00100001	MSB→Rn→T	—	1
SHLD	Rm, Rn	0100nnnnmmmm1101	Rm ≥ 0: Rn << Rm → Rn Rm < 0: Rn >> Rm → [0 → Rn]	—	1
SHLL	Rn	0100nnnn00000000	T←Rn←0	—	1
SHLR	Rn	0100nnnn00000001	0→Rn→T	—	1
SHLL2	Rn	0100nnnn00001000	Rn<<2 → Rn	—	1
SHLR2	Rn	0100nnnn00001001	Rn>>2 → Rn	—	1
SHLL8	Rn	0100nnnn00011000	Rn<<8 → Rn	—	1
SHLR8	Rn	0100nnnn00011001	Rn>>8 → Rn	—	1
SHLL16	Rn	0100nnnn00101000	Rn<<16 → Rn	—	1
SHLR16	Rn	0100nnnn00101001	Rn>>16 → Rn	—	1

BT/S	label	10001101dddddddd	Delayed branch, if T = 1, disp × 2 + PC → PC; if T = 0, nop	—	2/1*
BRA	label	1010ddddddddddd	Delayed branch, disp × 2 + PC → PC	—	2
BRAF	Rm	0000mmmm00100011	Delayed branch,Rm + PC → PC	—	2
BSR	label	1011ddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	—	2
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	—	2
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	—	2
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	—	2
RTS		0000000000001011	Delayed branch, PR → PC	—	2

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

LDC	Rm,SSR	0100mmmm00111110	Rm→SSR	√	4
LDC	Rm,SPC	0100mmmm01001110	Rm→SPC	√	4
LDC	Rm,R0_BANK	0100mmmm10001110	Rm→R0_BANK	√	4
LDC	Rm,R1_BANK	0100mmmm10011110	Rm→R1_BANK	√	4
LDC	Rm,R2_BANK	0100mmmm10101110	Rm→R2_BANK	√	4
LDC	Rm,R3_BANK	0100mmmm10111110	Rm→R3_BANK	√	4
LDC	Rm,R4_BANK	0100mmmm11001110	Rm→R4_BANK	√	4
LDC	Rm,R5_BANK	0100mmmm11011110	Rm→R5_BANK	√	4
LDC	Rm,R6_BANK	0100mmmm11101110	Rm→R6_BANK	√	4
LDC	Rm,R7_BANK	0100mmmm11111110	Rm→R7_BANK	√	4
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm)→SR, Rm+4→Rm	√	8
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm)→GBR, Rm+4→Rm	—	4
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm)→VBR, Rm+4→Rm	√	4
LDC.L	@Rm+,SSR	0100mmmm00110111	(Rm)→SSR,Rm+4→Rm	√	4
LDC.L	@Rm+,SPC	0100mmmm01000111	(Rm)→SPC,Rm+4→Rm	√	4
LDC.L	@Rm+, R0_BANK	0100mmmm10000111	(Rm)→R0_BANK,Rm+4→Rm	√	4
LDC.L	@Rm+, R1_BANK	0100mmmm10010111	(Rm)→R1_BANK,Rm+4→Rm	√	4
LDC.L	@Rm+, R2_BANK	0100mmmm10100111	(Rm)→R2_BANK,Rm+4→Rm	√	4
LDC.L	@Rm+, R3_BANK	0100mmmm10110111	(Rm)→R3_BANK, Rm+4→Rm	√	4

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

STC.L	SPC,@–Rn	0100nnnn10000011	Rn–4→Rn, SPC→(Rn)	√	1
STC.L	R0_BANK,@–Rn	0100nnnn10000011	Rn–4→Rn, R0_BANK→(Rn)	√	1
STC.L	R1_BANK,@–Rn	0100nnnn10010011	Rn–4→Rn, R1_BANK→(Rn)	√	1
STC.L	R2_BANK,@–Rn	0100nnnn10100011	Rn–4→Rn, R2_BANK→(Rn)	√	1
STC.L	R3_BANK,@–Rn	0100nnnn10110011	Rn–4→Rn, R3_BANK→(Rn)	√	1
STC.L	R4_BANK,@–Rn	0100nnnn11000011	Rn–4→Rn, R4_BANK→(Rn)	√	1
STC.L	R5_BANK,@–Rn	0100nnnn11010011	Rn–4→Rn, R5_BANK→(Rn)	√	1
STC.L	R6_BANK,@–Rn	0100nnnn11100011	Rn–4→Rn, R6_BANK→(Rn)	√	1
STC.L	R7_BANK,@–Rn	0100nnnn11110011	Rn–4→Rn, R7_BANK→(Rn)	√	1
STS	MACH,Rn	0000nnnn00001010	MACH→Rn	—	1
STS	MACL,Rn	0000nnnn00011010	MACL→Rn	—	1
STS	PR,Rn	0000nnnn00101010	PR→Rn	—	1
STS.L	MACH,@–Rn	0100nnnn00000010	Rn–4→Rn, MACH→(Rn)	—	1
STS.L	MACL,@–Rn	0100nnnn00010010	Rn–4→Rn, MACL→(Rn)	—	1
STS.L	PR,@–Rn	0100nnnn00100010	Rn–4→Rn, PR→(Rn)	—	1
TRAPA	#imm	11000011iiiiiiii	Unconditional trap exception occurs*[2]	—	8

Notes: 1. Number of states before the chip enters the sleep state.
2. For details, refer to section 4, Exception Handling.

0000	Rn	00MD	0010	STC SR, Rn	STC GBR, Rn	STC VBR, Rn	STC SSR
0000	Rn	01MD	0010	STC SPC, Rn			
0000	Rn	10MD	0010	STC R0_BANK, Rn	STC R1_BANK, Rn	STC R2_BANK, Rn	STC R3_
0000	Rn	11MD	0010	STC R4_BANK, Rn	STC R5_BANK, Rn	STC R6_BANK, Rn	STC R7_
0000	Rm	00MD	0011	BSRF Rm		BRAF Rm	
0000	Rm	10MD	0011	PREF @Rm			
0000	Rn	Rm	01MD	MOV.B Rm, @(R0, Rn)	MOV.W Rm, @(R0, Rn)	MOV.L Rm, @(R0, Rn)	MUL.L R
0000	0000	00MD	1000	CLRT	SETT	CLRMAC	LDTLB
0000	0000	01MD	1000	CLRS	SETS		
0000	0000	Fx	1001	NOP	DIV0U		
0000	0000	Fx	1010				
0000	0000	Fx	1011	RTS	SLEEP	RTE	
0000	Rn	Fx	1000				
0000	Rn	Fx	1001			MOVT Rn	
0000	Rn	Fx	1010	STS MACH, Rn	STS MACL, Rn	STS PR, Rn	
0000	Rn	Fx	1011				
0000	Rn	Rm	11MD	MOV. B @(R0, Rm), Rn	MOV.W @(R0, Rm), Rn	MOV.L @(R0, Rm), Rn	MAC.L @Rm+,@
0001	Rn	Rm	disp	MOV.L Rm, @(disp:4, Rn)			
0010	Rn	Rm	00MD	MOV.B Rm, @Rn	MOV.W Rm, @Rn	MOV.L Rm, @Rn	

0011	Rn	Rm	11MD	ADD Rm, Rn	DMULS.L Rm,Rn	ADDC Rm, Rn		ADDV
0100	Rn	Fx	0000	SHLL Rn	DT Rn	SHAL Rn		
0100	Rn	Fx	0001	SHLR Rn	CMP/PZ Rn	SHAR Rn		
0100	Rn	Fx	0010	STS.L MACH, @–Rn	STS.L MACL, @–Rn	STS.L PR, @–Rn		
0100	Rn	00MD	0011	STC.L SR, @–Rn	STC.L GBR, @–Rn	STC.L VBR, @–Rn		STC.L Rn
0100	Rn	01MD	0011	STC.L SPC, @–Rn				
0100	Rn	10MD	0011	STC.L R0_BANK, @–Rn	STC.L R1_BANK, @–Rn	STC.L R2_BANK, @–Rn		STC.L R3_BAI
0100	Rn	11MD	0011	STC.L R4_BANK, @–Rn	STC.L R5_BANK, @–Rn	STC.L R6_BANK, @–Rn		STC.L R7_BAI
0100	Rn	Fx	0100	ROTL Rn		ROTCL Rn		
0100	Rn	Fx	0101	ROTR Rn	CMP/PL Rn	ROTCR Rn		
0100	Rm	Fx	0110	LDS.L @Rm+, MACH	LDS.L @Rm+, MACL	LDS.L @Rm+, PR		
0100	Rm	00MD	0111	LDC.L @Rm+, SR	LDC.L @Rm+, GBR	LDC.L @Rm+, VBR		LDC.L SSR
0100	Rm	01MD	0111	LDC.L @Rm+, SPC				
0100	Rm	10MD	0111	LDC.L @Rm+, R0_BANK	LDC.L @Rm+, R1_BANK	LDC.L @Rm+, R2_BANK		LDC.L @Rm+,
0100	Rm	11MD	0111	LDC.L @Rm+, R4_BANK	LDC.L @Rm+, R5_BANK	LDC.L @Rm+, R6_BANK		LDC.L @Rm+,
0100	Rn	Fx	1000	SHLL2 Rn	SHLL8 Rn	SHLL16 Rn		

0100	Rm	10MD	1110	LDC Rm, R0_BANK	LDC Rm, R1_BANK	LDC Rm, R2_BANK	LDC Rm R3_BAN
0100	Rm	11MD	1110	LDC Rm, R4_BANK	LDC Rm, R5_BANK	LDC Rm, R6_BANK	LDC Rm R7_BAN
0100	Rn	Rm	1111	MAC.W @Rm+, @Rn+			
0101	Rn	Rm	disp	MOV.L @ (disp:4, Rm), Rn			
0110	Rn	Rm	00MD	MOV.B @Rm, Rn	MOV.W @Rm, Rn	MOV.L @Rm, Rn	MOV
0110	Rn	Rm	01MD	MOV.B @Rm+, Rn	MOV.W @Rm+, Rn	MOV.L @Rm+, Rn	NOT
0110	Rn	Rm	10MD	SWAP.B Rm, Rn	SWAP.WRm, Rn	NEGC Rm, Rn	NEG
0110	Rn	Rm	11MD	EXTU.B Rm, Rn	EXTU.W Rm, Rn	EXTS.B Rm, Rn	EXTS.W
0111	Rn	imm		ADD # imm : 8, Rn			
1000	00MD	Rn	disp	MOV. B R0, @(disp: 4, Rn)	MOV. W R0, @(disp: 4, Rn)		
1000	01MD	Rm	disp	MOV.B @(disp:4, Rm), R0	MOV.W @(disp: 4, Rm), R0		
1000	10MD	imm/disp		CMP/EQ #imm:8, R0	BT disp: 8		BF dis
1000	11MD	imm/disp			BT/S disp: 8		BF/S dis
1001	Rn	disp		MOV.W @ (disp : 8, PC), Rn			
1010	disp			BRA disp: 12			
1011	disp			BSR disp: 12			

			#imm: 8, @(R0, GBR)	#imm: 8, @(R0, GBR)	#imm: 8, @(R0, GBR)	#imm: 8 GBR)
1101	Rn	disp	MOV.L @(disp: 8, PC), Rn			
1110	Rn	imm	MOV #imm:8, Rn			
1111	**********					

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

- Repeat loop control instructions and repeat loop control register access instructions a
  Looped programs can be executed efficiently by using the zero-overhead repeat cont
  For details, refer to section 3.3, CPU Extended Instructions.
- Modulo addressing control instructions and control register access instructions are ad
  Function allows access to data with a circular structure. For details, refer to section 3
  Data Transfer Instructions.
- DSP unit register access instructions are added. Some of the DSP unit registers can b
  the same way as the CPU system registers. For details, refer to section 3.4, DSP Data
  Instructions.

**Data Transfer Instructions for Data Transfers between DSP Unit Registers and On
X/Y memory:** Data transfer instructions for data transfers between the DSP unit register
chip X/Y memory are called double-data transfer instructions. Instruction codes for thes
transfer instructions are 16 bit codes as well as CPU instruction codes. These data transf
instructions perform data transfers between the DSP unit and on-chip X/Y memory that
connected to the DSP unit. These data transfer instructions can be described in combinat
other DSP unit operation instructions. For details, refer to section 3.4, DSP Data Transfe
Instructions.

**Data Transfer Instructions for Data Transfers between DSP Unit Registers and All
Address Spaces:** Data transfer instructions for data transfers between DSP unit register
logical address spaces are called single-data transfer instructions. Instruction codes for th
transfer instructions are 16 bit codes as well as CPU instruction codes. These data transf
instructions performs data transfers between the DSP unit registers and all logical addres
For details, refer to section 3.4, DSP Data Transfer Instructions.

DSP Unit Operation Instructions: DSP unit operation instructions are called DSP data
instructions. These instructions are provided to execute digital signal processing operati
speed using the DSP. Instruction codes for these instructions are 32 bits. The DSP data c

accessed in word units.

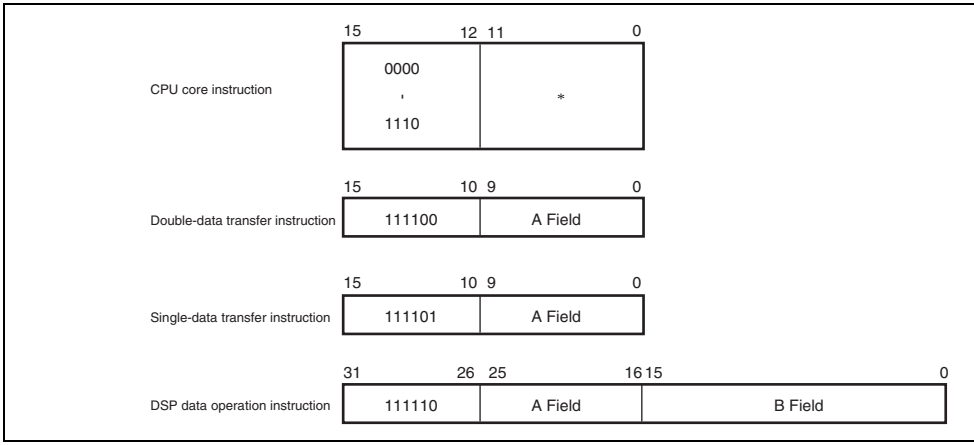


Figure 3.1 DSP Instruction Format

MD	DSP	Processing Mode	of Privileged Instruction Execution	DSP Extended Functions
0	0	User mode	Prohibited	Invalid
0	1	User DSP mode	Prohibited	Valid
1	0	Privileged mode	Allowed	Invalid
1	1	Privileged DSP mode	Allowed	Valid

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

3.2.2 DSP Mode Memory Map

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

Table 3.1 Logical Address Space

Address Range	Name	Protection	Description
H'A5000000 to H'A5FFFFFF	P2/Uxy	Privileged or DSP	16-Mbyte physical address space, no cacheable, non-address translatable Can be accessed in privileged mode privileged DSP mode, and user DSP

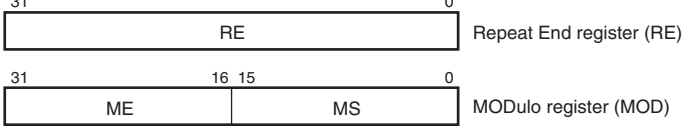


Figure 3.2 CPU Registers in DSP Mode

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	—	—	For details, refer to section 2, CPU.
27 to 16	RC11 to RC0	All 0	R/W	Repeat Counter Holds the number of repeat times in order to perfor control, and can be modified in privileged mode, pri DSP mode, or user DSP mode. At reset, this bit is i to 0. This bit is not affected in the exception handlin
15 to 13	—	—	—	For details, refer to section 2, CPU.
12	DSP	0	R/W	DSP Bit Enables or disables the DSP extended functions. If is set to 1, the DSP extended functions are enabled can be modified in privileged mode or privileged DS This bit cannot be modified in user DSP mode. At r bit is initialized to 0. This bit is not affected in the ex handling state.

				modified in privileged mode, privileged DSP mode DSP mode. At reset, these bits are initialized to 0. bits are not affected in the exception handling stat
1 to 0	—	—	—	For details, refer to section 2, CPU.

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

Repeat Start Register (RS): The repeat start register (RS) holds the start address of a lo module that is controlled by the repeat function. This register can be accessed in DSP m reset, the initial value of this register is undefined. This register is not affected in the exc handling state.

Repeat End Register (RE): The repeat end register (RE) holds the end address of a loo module that is controlled by the repeat function. This register can be accessed in DSP m reset, this register is initialized to 0. This register is not affected in the exception handlin

Modulo Register (MOD): The modulo register stores the modulo end address and mod address for modulo addressing in upper and lower 16 bits. The upper and lower 16 bits o modulo register are referred to as the ME register and MS register, respectively. This reg be accessed in DSP mode. At reset, the initial value of this register is undefined. This re not affected in the exception handling state.

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

- In user mode, if the LCD and STC instructions are used for the RS, an illegal instruc exception occurs.

MD	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1
RB	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1
BL	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1
RC [11:0]	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: OK	SETRC instruction	00000
DSP	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		0
DMY	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: OK		0
DMX	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: OK		0
Q	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x
M	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x
I[3:0]	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1111
RF[1:0]	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: OK	SETRC instruction	x
S	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x
T	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x

[Legend]

S: STC instruction

L: LDC instruction

OK: STC/LDC operation is enabled.

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

NG: Previous value is retained. No change.

x: Undefined

register (DSR). Figure 3.3 shows the DSP register configuration. These are 32-bit width
with the exception of registers A0 and A1. Registers A0 and A1 include 8 guard bits (fie
and A1G), giving them a total width of 40 bits. The DSR register stores the DSP data op
result (zero, negative, others). The DSP register has a DC bit whose function is similar t
of the CPU register. For details on DSR bits, refer to section 3.5, DSP Data Operation In

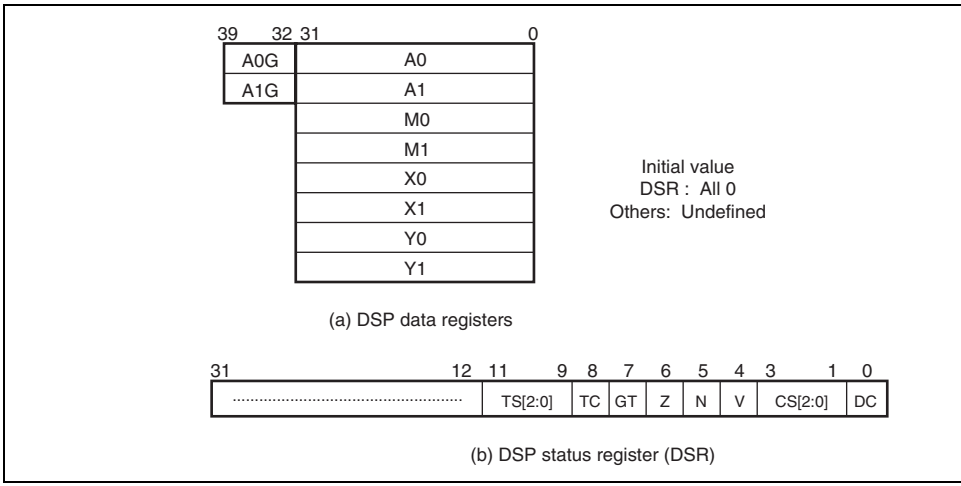


Figure 3.3 DSP Register Configuration

Example 1. Repeat loop consisting of 4 or more instructions

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

In the above program example, instructions from the RptStart address (instr1 instruction)
RptEnd address (instrN instruction) are repeated four times. These repeated instructions i
program are called repeat loop. The start and end instructions of the repeat loop are calle
repeat start instruction and repeat end instruction, respectively. The CPU sequentially exe
instructions and starts repeat loop control if the CPU detects the completion of a specific
instruction. This specific instruction is called the repeat detection instruction. In a repeat
consisting of 4 or more instructions, an instruction three instructions prior to the repeat en
instruction is regarded as the repeat detection instruction. In a repeat loop consisting of 4
instructions, the same instruction is regarded as the RptStart instruction and RptDtct instr

instructions, the RS register specifies the repeat start instruction address. In a repeat consisting of three or less instructions, a specific address is specified in the RS. This described later.

- Repeat counter (RC[11:0] bits of the SR)

  The repeat counter is specifies the number of repetitions by the SETRC instruction. I repeat loop execution, the RC holds the remaining number of repetitions.

- Repeat flags (RF[1:0] bits of the SR)

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

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

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

```
                                start instruction is regarded as
                                repeat detection instruction.
RptStart: instr1            ; [Repeat start instruction]
          Instr2            ;
RptEnd:   instr3            ; [Repeat end instruction]
```

- Example 3: Repeat loop consisting of two instructions

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

```
                                    start instruction is regarded as
                                    repeat detection instruction.
RptStart:
RptEnd:   instr1              ; [Repeat start instruction] ==
                                    [Repeat end instruction]
```

In repeat loops consisting of three instructions, two instructions and one instruction, spe addresses are specified in the RS register. RE – RS is calculated during SETRC instructi execution, and the number of instructions included in the repeat loop is determined acco the result. A value of 0, –2,and –4 in the result correspond to 3 instructions, two instruct one instruction, respectively.

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

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

RptEnd3: Address of the instruction three instructions prior to the repeat end instru

Repeat Control Instructions and Repeat Control Macros: To describe a repeat loop, t
and RE registers must be specified appropriately by the LDRS and LDRS instructions an
number of repetitions must be specified by the SERTC instruction. An 8-bit immediate da
general register can be used as an operand of the SETRC instruction. To specify the RC a
greater than 256, use SETRC Rm type instructions.

Table 3.4 Repeat Control Instructions

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

The RS and RE registers must be specified appropriately according to the rules shown in
The SH assembler supports control macros (REPEAT) as shown in table 3.5 to solve prol

RptEnd as repeat end instruction, and the [11:0] bits of Rm as number of repetitions. This macro is extended to three instructions: LDRS, LDRE, and SETRC which are converted correctly.

Using the repeat macros shown in table 3.5, examples 1 to 4 shown above can be simpli examples 5 to 8 as shown below.

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

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

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

```
          REPEAT RptStart, RptEnd, #4
          instr0              ;
RptStart: instr1              ; [Repeat start instruction]
          Instr2              ;
RptEnd:   instr3              ; [Repeat end instruction]
```

```
            instr0              ;
RptStart:
RptEnd:     instr1              ; [Repeat start instruction] ==
                                  [Repeat end instruction]
```

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

Table 3.6 DSP Mode Extended System Control Instructions

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

BRA, BSR, BT, BF, BT/F, BF/S, BSRF, RTS, BRAF, RTE, JSR, JMP, TRAPA

— Repeat control instructions

SETRC, LDRS, LDRE

— Load instructions for SR, RS, and RE registers

LDC Rn,SR, LDC @Rn+,SR, LDC Rn,RE, LDC @Rn+,RE, LDC Rn,RS, LDC

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

3. Instructions prohibited during repeat loop (In a repeat loop consisting of four or mor instructions)

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

— Repeat control instructions

SETRC, LDRS, LDRE

— Load instructions for SR, RS, and RE registers

LDC Rn,SR, LDC @Rn+,SR, LDC Rn,RE, LDC @Rn+,RE, LDC Rn,RS, LDC

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

can instruction must be placed before a repeat detection instruction.

Here, a branch includes a return from an exception processing routine. If an exception wh
return address is placed in an instruction following the repeat detection instruction occurs
repeat control cannot be returned correctly. Accordingly, an exception acceptance is restr
from the repeat detection instruction to the repeat end instruction. Exceptions such as inte
that can be retained by the CPU are retained. For exceptions that cannot be retained by th
transition to an exception occurs but a program cannot be returned to the previous execut
correctly. For details, refer to section 4, Exception Handling.

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

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

5. Branch from a repeat detection instruction

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

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

— PC relative addressing instructions

MOVA @(disp, PC), Rn

MOV.W @(disp, PC), Rn

MOV.L @(disp, PC), Rn

(Including the case when the MOV #imm,Rn is extended to MOV.W @(disp, P
MOV.L @(disp, PC), Rn)

Table 3.7 PC Value during Repeat Control (When RC[11:0] ≥ 2)

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

Note: In table 3.7, the following labels are used.

RptDtct: An address of the repeat detection instruction

RptDtct1: An address of the instruction one instruction following the repeat sta
instruction (In a repeat loop consisting of one to three instructions, I
a repeat start instruction)

RptDtct2: An address of the instruction two instruction following the repeat sta
instruction

RptDtct3: An address of the instruction three instruction following the repeat s
instruction

— If RC == 0, the repeat control function is not initiated even if a repeat detection in is executed. The repeat loop is executed once as normal instructions and a control passed to a repeat start instruction even if a repeat end instruction is executed.

3.3.2 Extended Repeat Control Instructions

In the repeat control function described in section 3.3.1, Repeat Control Instructions, ther some restrictions. To reduce these restrictions, this LSI supports the extended repeat instr to extend the repeat control function. These extended repeat control instructions were not supported in the conventional SH-DSP. To keep compatibility with the conventional SH- the conventional repeat control instructions called compatible repeat control instructions.

Program Examples Using the Extended Repeat Control Instructions: Examples of re programs using the extended repeat control instructions are shown below.

```
            ... ...             ;
            ... ...             ;
            instr(N-3)          ;
            instr(N-2)          ;
            instr(N-1)          ;
RptEnd:     instrN              ; [Repeat end instruction]
```

- Example 2: Repeat loop consisting of three instructions

```
            LDRS RptStart       ; Sets repeat start instruction ad
                                  to the RS register
            LDRE RptEnd         ; Sets repeat end instruction addr
                                  to the RE register
            LDRC #4             ; Sets the number of repetitions (
                                  the RC[11:0] bits of the SR regi
            instr0              ; At least one instruction is requ
                                  from LDRC instruction to [Repeat
                                  instruction]
RptStart:   instr1              ; [Repeat start instruction]
            instr2              ;
RptEnd:     instr3              ; [Repeat end instruction]
```

```
RptEnd:   instr2              ; [Repeat end instruction]
```

- Example 4: Repeat loop consisting of one instructions

```
          LDRS RptStart       ; Sets repeat start instruction add
                                to the RS register
          LDRE RptEnd         ; Sets repeat end instruction addre
                                to the RE register
          LDRC #4             ; Sets the number of repetitions (4
                                the RC[11:0] bits of the SR regis
          instr0              ; At least one instruction is requi
                                from LDRC instruction to [Repeat
Rptstart:                       instruction] RptStart:
RptEnd:   instr1              ; [Repeat start instruction]=
                                [Repeat end instruction]
```

In extended repeat control instructions, a repeat start instruction address and a repeat end instruction address are stored in the RS register and RE register, respectively, regardless o number of repeat instructions. In addition, the extended repeat control can be performed b the LDRC instruction instead of the SETRC instruction. During the extended repeat contr repeat loop can be recognized by executing a repeat end instruction. Therefore, there is no restriction on branches or exceptions.

Extended Repeat Control Instructions: To describe the extended repeat loop, the repea and end addresses must be specified to the RS and RE registers by the LDRS and LDRE instructions, respectively. For the LDRS and LDRE instructions of the extended repeat co instructions, the LDRS and LDRE instructions of the compatible repeat control instructio used. The number of repetitions are specified by the LDRC instruction. An 8-bit immedia the general register values can be used as an operand of the LDRC instruction. If 256 or g value is specified to the RC, use the LDRC Rm type instructions.

	RC[11:0] can be specified as 0 to 255. During extended repeat control, bit 0 of the RE register is set to 1.	
LDRC Rm	Sets the[11:0] bits of the Rm register to the RC[11:0] bits of the SR register and sets the information related to the number of repetitions to the RF[1:0] bits of the SR. RC[11:0] can be specified as 0 to 4095. During extended repeat control, bit 0 of the RE register is set to 1.	1

By executing the LDRC instruction, the CPU performs the extended repeat control funct
indicate that the CPU is being in extended repeat control, bit 0 of the RE register is set t
executing the LDRC instruction. To change the RE register value by a process such as a
exception handling, bit 0 of the RE register must be saved and restored correctly. By sav
restoring the RC[11:0] bits, DSP bit, and RF[1:0] bits of the SR register, RE register, an
register correctly, a control is returned to the extended repeat function correctly after pro
such as exception handling.

Restrictions on Extended Repeat Loop Control

1. Extended repeat control instruction assignment
   The LDRC instruction must be executed after executing the LDRS and LDRE instru
   addition, note that at least one instruction is required between the LDRC instruction
   repeat start instruction.
2. Illegal instruction one or more instructions following the repeat detection instruction
   If one of the following instructions is executed as a repeat end instruction, an illegal
   exception occurs.
   — Branch instructions

occurs, a control is passed to a branch destination.

3. Repeat counter and repeat control

   The CPU always execute a program with comparing the repeat end register (RE) and
   4) (current instruction address). If the (PC – 4) [31:1] matches the RE [31:1] while bit
   register is set to 1 and RC [11:0] of SR register is not 0, the extended repeat control fu
   initiated.

   — If RC ≥ 2, a control is passed to a repeat start instruction after a repeat end instruct
   been executed. The RC is decremented by 1 at the completion of the repeat end in

   — If RC == 1, the RC is decremented to 0 at the completion of the repeat end instruc
   a control is passed to the subsequent instruction.

   — If RC == 0, the repeat control function is not initiated even if a repeat detection in
   is executed. The repeat loop is executed once as normal instructions and a control
   passed to a repeat start instruction even if a repeat end instruction is executed.

described in combination with the DSP operation instructions to execute data transfe
operation in parallel,

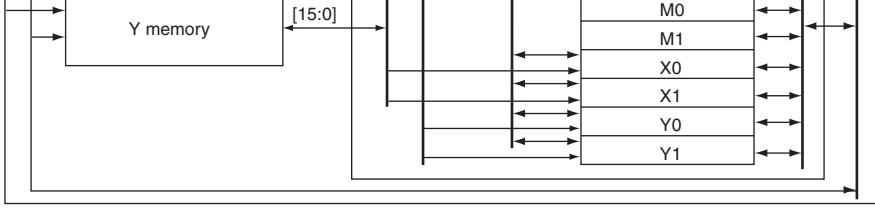
2. Single data transfer instructions

   The DSP unit is also connected to the L bus that is used by the CPU. The DSP regist
   than the DSR can access any logical addresses generated by the CPU. In this case, th
   data transfer instructions are used. The single data transfer instructions cannot be use
   combination with the DSP operation instructions and can access only one data item a

3. System control instructions

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

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



[Legend]
XAB: X bus (address)
XDB: X bus (data)
YAB: Y bus (address)
YDB: Y bus (data)
LAB: L bus (address)
LDB: L bus (data)
CDB: C bus (data)

Figure 3.4 DSP Registers and Bus Connections

Double Data Transfer Instructions (MOVX.W, MOVY.W, MOVX.L, MOVY.L): W
double data transfer group instructions, X memory and Y memory can be accessed in par

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

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

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

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

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

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

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

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

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

- Single data transfer instructions

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

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

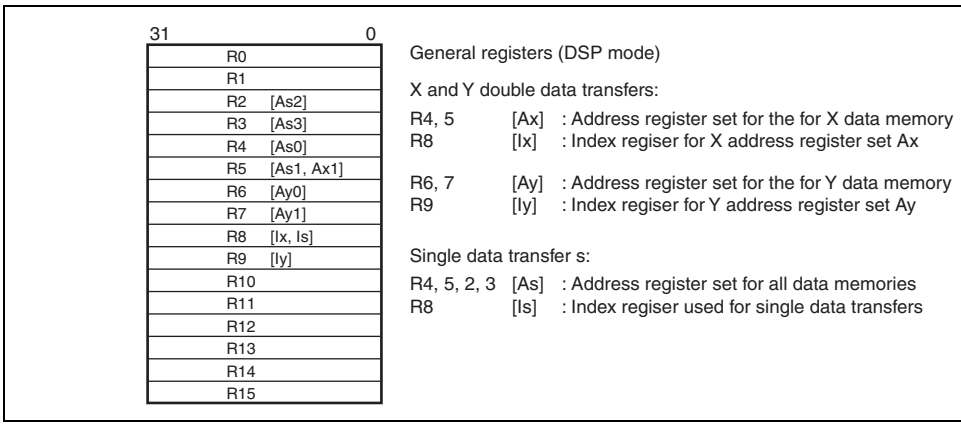


Figure 3.5 General Registers (DSP Mode)

Ay0: .REG (R6)

Ay1: .REG (R7)

Iy: .REG (R9)

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

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

As2: .REG (R2)

As3: .REG (R3)

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

	Ay: R6, R7	
Index register	Ix: R8, Iy: R9	Is: R8
Addressing	Nop/Inc (+2)/index addition: post-increment	Nop/Inc (+2, +4)/index ad post-increment
Addressing	—	Dec (–2, –4): pre-decreme
Modulo addressing	Possible	Not possible
Data bus	XDB, YDB	LDB
Data length	16 bits (word)	16/32 bits (word/longword
Bus conflict	No	Yes
Memory	X/Y data memory	Entire memory space
Source register	Dx, Dy: A0, A1	Ds: A0/A1, M0/M1, X0/X1 A0G, A1G
Destination register	Dx: X0/X1 Dy: Y0/Y1	Ds: A0/A1, M0/M1, X0/X1 A0G, A1G

Addressing Mode for Double Data Transfer Instructions: The double data transfer in supports the following three addressing modes.

- Non-update address register addressing

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

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

  The Ax and Ay registers are address pointers. After a data transfer, the value of the I register is added to each (post-increment). The double data transfer instructions do n

Single Data Addressing: The following four kinds of addressing can be used with single transfer instructions.

- Non-update address register addressing

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

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

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

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

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

3.4.3 Modulo Addressing

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

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

The MOD register is provided to set the start and end addresses of the modulo address are upper and lower words of the MOD register store modulo start address (MS) and modulo

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

Table 3.11 Modulo Addressing Control Instructions

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

```
LDC R10,SR              ; Specify SR.DMX=1,
                          SR.DMY=0, and X modulo addressing
MOV.L #H'A5007000,R14
MOVX.W @R4+,X0          ; R4: H'A5007000→ H'A5007002
MOVX.W @R4+,X0          ; R4: H'A5007002→ H'A5007004
MOVX.W @R4+,X0          ; R4: H'A5007004→ H'A5007000
                          (Matches to ME and MS is set)
MOVX.W @R4+,X0          ; R4: H'A5007000→ H'A5007002
MOVX.W @R4+,X0          ; R4: H'A5007000→ H'A5007002
```

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

When the addressing type of the X/Y data transfer instruction is no-update, the X/Y data
instruction is not returned to MS even if they match ME.

When the addressing type of the X/Y data transfer instruction is addition index register ad
the address pointed way not match the address pointer ME and exceed it. In this case, the
pointer value does not become the modulo start address.

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

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

3.4.5 Instruction Formats of Double and Single Transfer Instructions

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

Y memory data transfer	NOPY	1	1	1	1	0	0		0		0		0					
	MOVY.W @Ay,Dy								Ay		Dy		0					
	MOVY.W @Ay+,Dy																	
	MOVY.W @Ay+Iy,Dy																	
	MOVY.W Da,@Ay										Da		1					
	MOVY.W Da,@Ay+																	
	MOVY.W Da,@Ay+Iy																	

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

	MOVS.W Ds,@As+Is	7:A0	1	1
	MOVS.L @-As,Ds	8:X0	0	0
	MOVS.L @As,Ds	9:X1	0	1
	MOVS.L @As+,Ds	A:Y0	1	0
	MOVS.L @As+Is,Ds	B:Y1	1	1
	MOVS.L Ds,@-As	C:M0	0	0
	MOVS.L Ds,@As	D:A1G	0	1
	MOVS.L Ds,@As+	E:M1	1	0
	MOVS.L Ds,@As+Is	F:A0G	1	1

Note: * Codes reserved for system use.

DSP fixed-point data operation uses a DSP register other than A0 or A1 as the source reg
sign-extends the source value to bits 39 to 32. When it uses one of these registers as the
destination register, bits 39 to 32 of the result are discarded.

The second kind of operation is an X or Y data transfer operation, MOVX.W, MOVY.W.
operation accesses the X and Y memories through the 16-bit X and Y data buses (figure 3
register to be loaded or stored by this operation always comprises the upper 16 bits (bits 3
X0 or X1 can be the destination of an X memory load and Y0 or Y1 can be the destinatio
memory load, but no other register can be the destination register in this operation. When
read into the upper 16 bits of a register (bits 31 to 16), the lower 16 bits of the register (bi
0) are automatically cleared. A0 and A1 can be stored in the X or Y memory by this oper
no other registers can be stored.

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

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

		Integer, PDMSB	Sign-extended	24-bit result	Clea
		Logical, PSHL	Cleared	16-bit result	Clea
	Data transfer	MOVS.W	Sign-extended	16-bit data	Clea
		MOVS.L	Sign-extended	32-bit data	
A0G, A1G	Data transfer	MOVS.W	Data	No update	
		MOVS.L	Data	No update	
X0, X1 Y0, Y1 M0, M1	DSP operation	Fixed-point, PSHA, PMULS		32-bit result	
		Integer, logical, PDMSB, PSHL		16-bit result	Clea
	Data transfer	MOVX/Y.W, MOVS.W		16-bit result	Clea
		MOVS.L		32-bit data	

A0G, A1G	Data transfer	MOVS.W	Data		
		MOVS.L	Data		
X0, X1 Y0, Y1 M0, M1	DSP operation	Fixed-point, PDMSB, PSHA	Sign*	32-bit data	
		Integer	Sign*	16-bit data	
		Logical, PSHL, PMULS		16-bit data	
	Data transfer	MOVS.W		16-bit data	
		MOVS.L		32-bit data	

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

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

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

				000: Carry/borrow mode 001: Negative value mode 010: Zero mode 011: Overflow mode 100: Signed greater mode 101: Signed greater than or equal to mode 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)
8	TC	0	R/W	TC Bit 0: The T bit of the SR register is not affecte DSP instruction. 1: The T bit of the SR register changes acc the TS bit of the DSR register while the D instruction is executed. Note, however, th does not change during conditional DSP execution.
7	GT	0	R/W	Signed Greater Bit Indicates that the operation result is positive 0), or that operand 1 is greater than operan 1: Operation result is positive, or operand 1 than operand 2
6	Z	0	R/W	Zero Bit Indicates that the operation result is zero (0 operand 1 is equal to operand 2 1: Operation result is zero (0), or operands

3 to 1	CS2 to CS0	All 0	R/W	DC Bit Status Selection Designate the mode for selecting the operati status to be set in the DC bit 000: Carry/borrow mode 001: Negative value mode 010: Zero mode 011: Overflow mode 100: Signed greater mode 101: Signed greater than or equal to mode 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)
0	DC	0	R/W	DSP Status Bit Sets the status of the operation result in the designated by the CS bits 0: Designated mode status has not occurred 1: Designated mode status has occurred Indicates the operation result by carry or bor regardless of the CS bit status after the PAD PSUBC instruction has been executed.

3.3.2 DSP Operation Instruction Set

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

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

	DCF	ALUop. Sy, Dz
Unconditional single data operation instructions		ALUop. Sx, Sy, Dz
		ALUop. Sx, Dz
		ALUop. Sy, Dz
		MLTop. Se, Sf, Dg

Table 3.18 Correspondence between DSP Instruction Operands and Registers

	ALU/Shift Operations				Multiply Operatio		
Register	**Sx**	**Sy**	**Dz**	**Du**	**Se**	**Sf**	**D**
A0	Yes		Yes	Yes			Y
A1	Yes		Yes	Yes	Yes	Yes	Y
M0		Yes	Yes				Y
M1		Yes	Yes				Y
X0	Yes		Yes	Yes	Yes	Yes	
X1	Yes		Yes		Yes		
Y0		Yes	Yes	Yes	Yes	Yes	
Y1		Yes	Yes			Yes	

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

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

				DC bit. When an ALU or shift (PSHL) logical operation is execute MSB of the result, excluding the guard bits, is copied into bit.
0	1	0	Zero value mode	The DC bit is set if the result of an ALU or shift operation zeros, and is cleared otherwise.
0	1	1	Overflow mode	The DC bit is set if the result of an ALU or shift (PSHA) a operation exceeds the destination register range, excludi guard bits, and is cleared otherwise. When an ALU or shift (PSHL) logical operation is execute DC bit is always cleared.
1	0	0	Signed greater-than mode	This mode is similar to signed greater-or-equal mode, bu cleared if the result is all-zeros. DC = ~{(negative value ^ over-range) | zero value}; In case of arithmetic operation DC = 0; In case of logical operation
1	0	1	Signed greater-or-equal mode	If the result of an ALU or shift (PSHA) arithmetic operatio exceeds the destination register range, including the gua (over-range), the definition is the same as in negative val mode. If the result is not over-range, the definition is the of that in negative value mode. When an ALU or shift (PSHL) logical operation is execute DC bit is always cleared. DC = ~(negative value ^ over-range); In case of arithmetic operation DC = 0 ; In case of logical operation
1	1	0	Reserved (setting prohibited)	
1	1	1	Reserved (setting prohibited)	

```
                R4=H'00008000, R6=H'00005000, R9=H'00000004
                (R4)=H'1111, (R6)=H'2222
After execution: X0=H'11110000, Y0=H'55555555, A0=H'008888888
                R4=H'00008002, R6=H'00005004, R9=H'00000004
                (R4)=H'1111, (R6)=H'3456

When condition is False

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

Figure 3.7 Examples of Conditional Operations and Data Transfer Instruct

- Assignment of NOPX and NOPY Instruction Codes

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

			10110001000001
PADD X0,Y0,A0			11111000000000
			10110001000001
	MOVX.W @R4+,X0	MOVY.W @R6+R9,Y0	11110000000010
	MOVX.W @R4+,X0	NOPY	11110000000010
	MOVS.W @R4+,X0		11110100100010
	NOPX	MOVY.W @R6+R9,Y0	11110000000000
		MOVY.W @R6+R9,Y0	11110000000000
	NOPX	NOPY	11110000000000
NOP			00000000000010

3.5.3 DSP-Type Data Formats

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

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

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

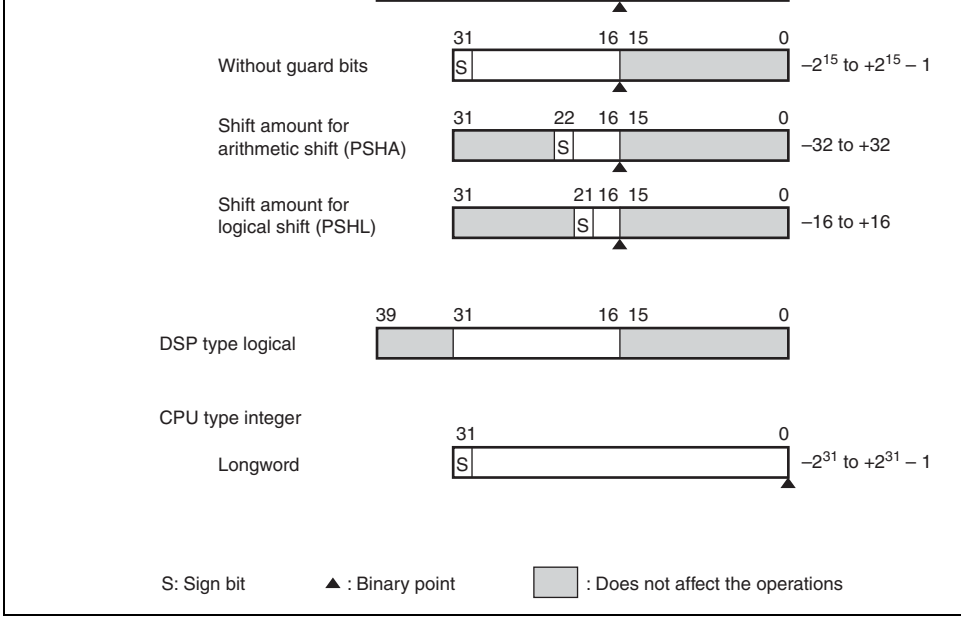


Figure 3.8 Data Formats

The shift amount for the arithmetic shift (PSHA) instruction has a 7-bit field that can rep values from –64 to +63, but –32 to +32 are valid numbers for the instruction. Also the sh amount for a logical shift operation has a 6-bit field, but –16 to +16 are valid numbers fo instruction. The results when an invalid shift amount is specified cannot be guaranteed.

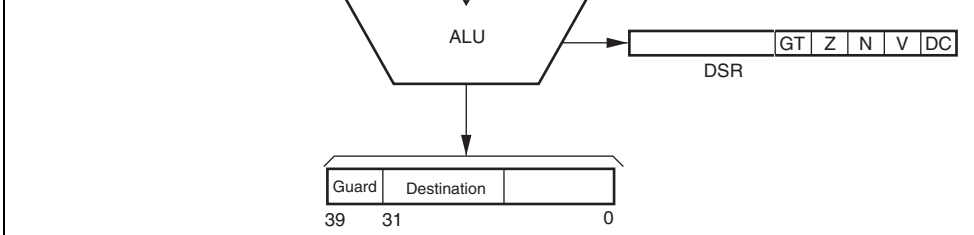


Figure 3.9 ALU Fixed-Point Arithmetic Operation Flow

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

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

PABS	Absolute	Sx	All 0	Dz
		All 0	Sy	Dz
PNEG	Negation	Sx	All 0	Dz
		All 0	Sy	Dz
PCLR	Clear	All 0	All 0	Dz

Table 3.22 Correspondence between Operands and Registers

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

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

Figure 3.10 Operation Sequence Example

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

Carry or Borrow Mode: CS[2:0] = 000: The DC bit indicates that carry or borrow is ge
from the most significant bit of the operation result, except the guard-bit parts. Some exa
shown in figure 3.11. This mode is the default condition. When the input data is negative
PABS or PNEG instruction, carry is generated.

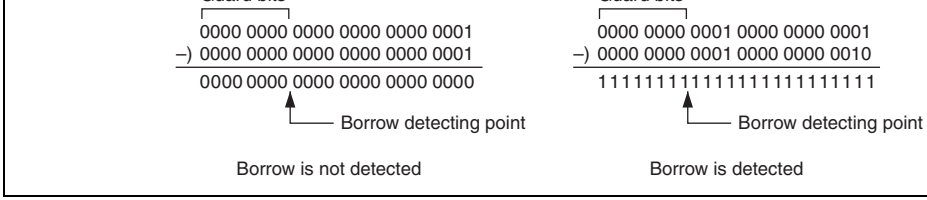


Figure 3.11 DC Bit Generation Examples in Carry or Borrow Mode

Negative Value Mode: CS[2:0] = 001: The DC flag indicates the same value as the MS operation result. When the result is a negative number, the DC bit shows 1. When it is a number, the DC bit shows 0. The ALU always executes 40-bit arithmetic operation, so t to detect whether positive or negative is always got from the MSB of the operation resul regardless of the destination operand. Some examples are shown in figure 3.12.

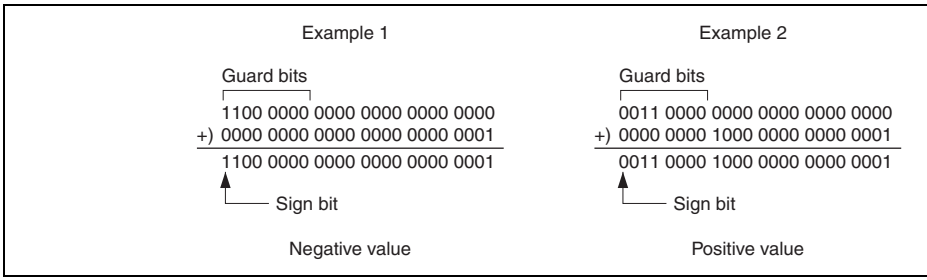


Figure 3.12 DC Bit Generation Examples in Negative Value Mode

Zero Value Mode: CS[2:0] = 010: The DC flag indicates whether the operation result i When the result is 0, the DC bit shows 1. When it is not 0, the DC bit shows 0.

Overflow Mode: CS[2:0] = 011: The DC bit indicates whether or not overflow occurs i result. When an operation yields a result beyond the range of the destination register, ex

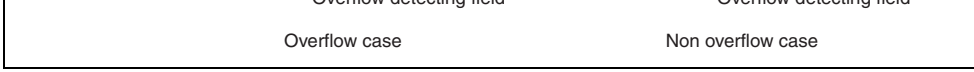


Figure 3.13 DC Bit Generation Examples in Overflow Mode

Signed Greater Than Mode: CS[2:0] = 100: The DC bit indicates whether or not the so data (signed) is greater than the source 2 data (signed) as the result of compare operation This mode is similar to the Negative Value Mode described before, because the result of compare operation is a positive value if the source 1 data is greater than the source 2 data However, the signed bit of the result shows a negative value if the compare operation yie result beyond the range of the destination operand, including the guard-bit parts (called “ range”), even though the source 1 data is greater than the source 2 data. The DC bit is upd concerning this type of special case in this condition mode. The equation below shows th definition of getting this condition:

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

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

Signed Greater Than or Equal Mode: CS[2:0] = 101: The DC bit indicates whether th 1 data (signed) is greater than or equal to the source 2 data (signed) as the result of compa operation PCMP. This mode is similar to the Signed Greater Than Mode described before equal case is also included in this mode. The equation below shows the definition of getti condition:

DC = ~ (Negative ^ Over-range)

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

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

3.5.5 ALU Integer Operations

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

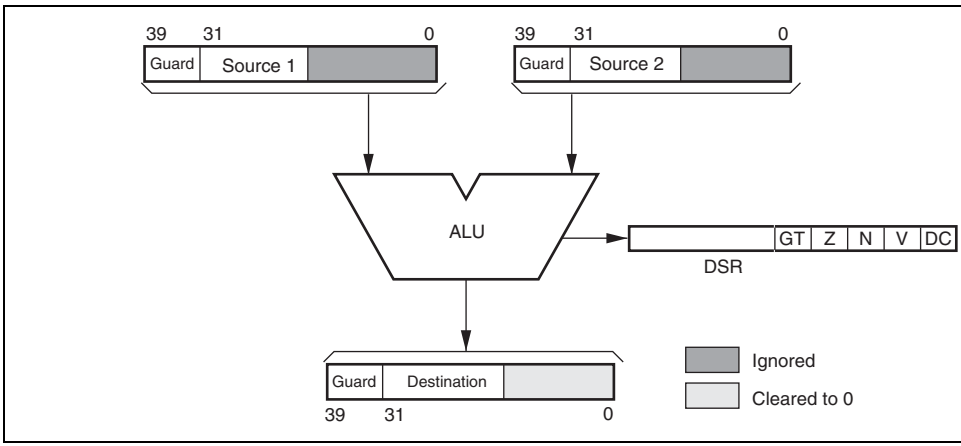


Figure 3.14 ALU Integer Arithmetic Operation Flow

word excluding the guard bits of the operation result are input into the destination r

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

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

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

- Overflow Protection

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

same stage as the MA stage in which memory access is performed.

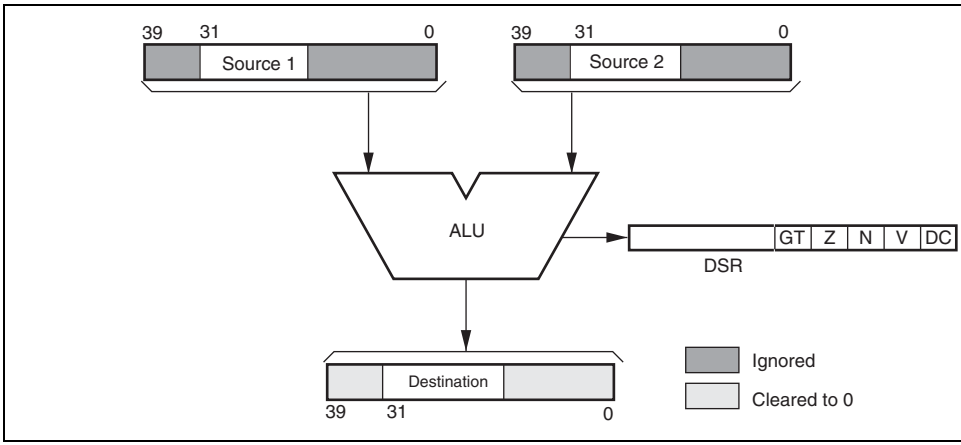


Figure 3.15 ALU Logical Operation Flow

Table 3.24 Variation of ALU Logical Operations

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

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

Signed Greater Than Mode: CS[2:0] = 100: The DC bit is always cleared.

Signed Greater Than or Equal Mode: CS[2:0] = 101: The DC bit is always cleared.

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

3.5.7 Fixed-Point Multiply Operation

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

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

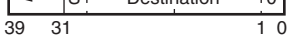


Figure 3.16 Fixed-Point Multiply Operation Flow

Table 3.25 Variation of Fixed-Point Multiply Operation

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

Table 3.26 Correspondence between Operands and Registers

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

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

The multiply operation of the DSP unit side is not integer but fixed-point arithmetic. So, words of each multiplier and multiplicand are input into a MAC unit as shown in figure the SH's standard multiply operations, the lower words of both source operands are inpu MAC unit. The operation result is also different from the SH's case. The SH's multiply o

does not mean (+1.0). If the S bit is 1, overflow is prevented and the result is H'00 FFFF.

3.5.8 Shift Operations

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

Table 3.27 Variation of Shift Operations

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

–32 <= Imm1 <= +32, –16 <= Imm2

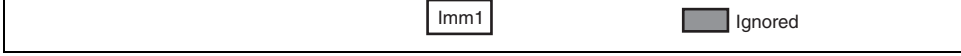


Figure 3.17 Arithmetic Shift Operation Flow

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

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

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

1. Carry or Borrow Mode: CS[2:0] = 000
   The DC bit indicates the last shifted out data as the operation result.
2. Negative Value Mode: CS[2:0] = 001

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

- Overflow Protection

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

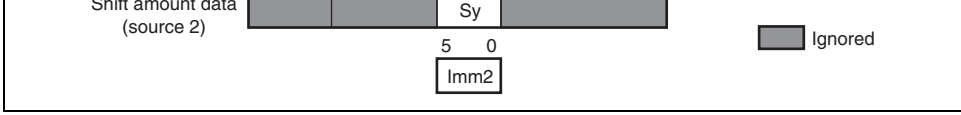


Figure 3.18 Logical Shift Operation Flow

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

Every time a logical shift operation is executed, the DC, N, Z, V, and GT bits in DSR ar updated in accordance with the operation result. In case of a conditional operation, they updated even though the specified condition is true and the operation is executed. In cas unconditional operation, they are always updated in accordance with the operation resul definition of the DC bit is selected by the CS[2:0] (condition selection) bits in DSR. The result is:

1. Carry or Borrow Mode: CS[2:0] = 000
   The DC bit indicates the last shifted out data as the operation result.
2. Negative Value Mode: CS[2:0] = 001
   Bit 31 of the operation result is loaded into the DC bit.
3. Zero Value Mode: CS[2:0] = 010

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

3.5.9 Most Significant Bit Detection Operation

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

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

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

Every time a PDMSB operation is executed, the DC, N, Z, V, and GT bits in DSR are bas
updated in accordance with the operation result. In case of a conditional operation, they a
updated, even though the specified condition is true, and the operation is executed. In cas
unconditional operation, they are always updated with the operation result.

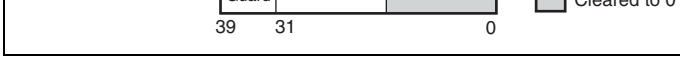


Figure 3.19 PDMSB Operation Flow

The definition of the DC bit is selected by the CS0–CS2 (condition selection) bits in DS DC bit result is

Carry or Borrow Mode: CS[2:0] = 000: The DC bit is always cleared.

Negative Value Mode: CS[2:0] = 001: The DC bit is set when the operation result is a value, and cleared when the operation result is zero or a positive value.

Zero Value Mode: CS[2:0] = 010: The DC bit is set when the operation result is zero; it is cleared.

Overflow Mode: CS[2:0] = 011: The DC bit is always cleared.

Signed Greater Than Mode: CS[2:0] = 100: The DC bit is set when the operation resu positive value; otherwise it is cleared.

Signed Greater Than or Equal Mode: CS[2:0] = 101: The DC bit is set when the oper result is zero or a positive value; otherwise it is cleared.

0	0	…	0	0	0	0	0	0	…	0	1	*	*	All 0	All 0	0	1	1	1	0
		:							:									:		
0	0	…	0	0	0	0	0	1	…	*	*	*	*	All 0	All 0	0	0	0	0	1
0	0	…	0	0	0	0	1	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0
0	0	…	0	0	0	1	*	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0
0	0	…	0	0	1	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1
0	0	…	0	1	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1
		:							:									:		
0	1	…	*	*	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	0	0
1	0	…	*	*	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	0	0
																		:		
1	1	…	1	0	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1
1	1	…	1	1	0	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1
1	1	…	1	1	1	0	*	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0
1	1	…	1	1	1	1	0	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0
1	1	…	1	1	1	1	1	0	…	*	*	*	*	All 0	All 0	0	0	0	0	1
		:							:									:		
1	1	…	1	1	1	1	1	1	…	1	0	*	*	All 0	All 0	0	1	1	1	0
1	1	…	1	1	1	1	1	1	…	1	1	0	*	All 0	All 0	0	1	1	1	0
1	1	…	1	1	1	1	1	1	…	1	1	1	0	All 0	All 0	0	1	1	1	1
1	1	…	1	1	1	1	1	1	…	1	1	1	1	All 0	All 0	0	1	1	1	1

Note: * means don't care.

Overflow mode part above. The GT bit always indicates the same state as the DC bit set greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

3.5.10 Rounding Operation

The DSP unit provides the function that rounds from 32 bits to 16 bits. In case of provid bit parts, it rounds from 40 bits to 24 bits. When a round instruction is executed, H'0000 added to the source operand data and then, the lower word is cleared. Figure 3.20 shows rounding operation flow and figure 3.21 shows the operation definition. Table 3.30 shov variation of this type of operation. The correspondence between each operand and regist same as ALU fixed-point operations as shown in table 3.21.

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

Every time rounding operation is executed, the DC, N, Z, V, and GT bits in DSR are bas updated in accordance with the operation result. In case of a conditional operation, they updated, even though the specified condition is true, and the operation is executed. In ca unconditional operation, they are always updated with the operation results. The definiti DC bit is selected by the CS0–CS2 (condition selection) bits in DSR. The result of these code bits is the same as the ALU-fixed point arithmetic operations.

Figure 3.20 Rounding Operation Flow

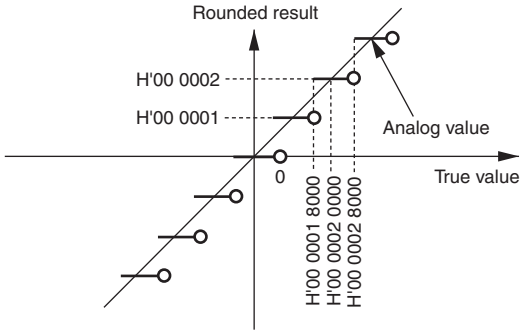


Figure 3.21 Definition of Rounding Operation

Table 3.30 Variation of Rounding Operation

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

- Overflow Protection

  The S bit in SR is effective for any rounding operations in the DSP unit. See section 3 Overflow Protection, for details.

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

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

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

Table 3.32 Definition of Overflow Protection for Integer Arithmetic Operations

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

Note: * means don't care.

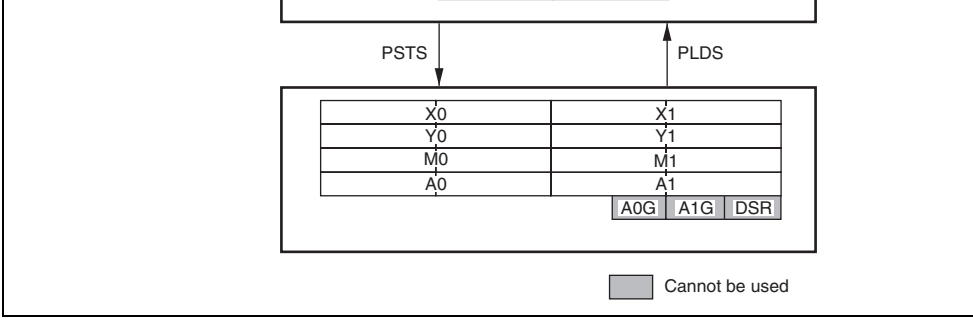


Figure 3.22 Local Data Move Instruction Flow

Table 3.33 Variation of Local Data Move Operations

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

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

Registers	A1			*1		*2	*1	*1	*2	*1
	M0				*1				*1	
	M1				*1				*1	
	X0	*2		*1		*2	*1	*1		*1
	X1	*2		*1			*1			*1
	Y0		*2		*1	*2	*1	*1		
	Y1		*2		*1			*1		

Notes: 1. Registers available for operands

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

There are three cases of operand conflict problems.

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

In these cases above, the result is not guaranteed.

LDRC @(disp,PC)	10001100dddddddd	(disp x 2 + PC) →RC	1	—
LDRE @(disp,PC)	10001110dddddddd	(disp x 2 + PC) →RE	1	—
STC MOD,Rn	0000nnnn01010010	MOD→Rn	1	—
STC RS,Rn	0000nnnn01100010	RS→Rn	1	—
STC RE,Rn	0000nnnn01110010	RE→Rn	1	—
STS DSR,Rn	0000nnnn01101010	DSR→Rn	1	—
STS A0,Rn	0000nnnn01111010	A0→Rn	1	—
STS X0,Rn	0000nnnn10001010	X0→Rn	1	—
STS X1,Rn	0000nnnn10011010	X1→Rn	1	—
STS Y0,Rn	0000nnnn10101010	Y0→Rn	1	—
STS Y1,Rn	0000nnnn10111010	Y1→Rn	1	—
STS.L DSR,@-Rn	0100nnnn01100010	Rn-4→Rn, DSR→(Rn)	1	—
STS.L A0,@-Rn	0100nnnn01110010	Rn-4→Rn, A0→(Rn)	1	—
STS.L X0,@-Rn	0100nnnn10000010	Rn-4→Rn, X0→(Rn)	1	—
STS.L X1,@-Rn	0100nnnn10010010	Rn-4→Rn, X1→(Rn)	1	—
STS.L Y0,@-Rn	0100nnnn10100010	Rn-4→Rn, Y0→(Rn)	1	—
STS.L Y1,@-Rn	0100nnnn10110010	Rn-4→Rn, Y1→(Rn)	1	—
STC.L MOD,@-Rn	0100nnnn01010011	Rn-4→Rn, MOD→(Rn)	1	—
STC.L RS,@-Rn	0100nnnn01100011	Rn-4→Rn, RS→(Rn)	1	—
STC.L RE,@-Rn	0100nnnn01110011	Rn-4→Rn, RE→(Rn)	1	—
LDS.L @Rn + ,DSR	0100nnnn01100110	(Rn) →DSR, Rn + 4→Rn	1	—
LDS.L @Rn + ,A0	0100nnnn01110110	(Rn) →A0, Rn + 4→Rn	1	—
LDS.L @Rn + ,X0	0100nnnn10000110	(Rn) →X0, Rn + 4→Rn	1	—
LDS.L @Rn + ,X1	0100nnnn10010110	(Rn) →X1, Rn + 4→Rn	1	—

LDS Rn,X1	0100nnnn10011010	Rn→X1	1	—
LDS Rn,Y0	0100nnnn10101010	Rn→Y0	1	—
LDS Rn,Y1	0100nnnn10111010	Rn→Y1	1	—
LDC Rn,MOD	0100nnnn01011110	Rn→MOD	4	—
LDC Rn,RS	0100nnnn01101110	Rn→RS	4	—
LDC Rn,RE	0100nnnn01111110	Rn→RE	4	—

			Ax + 2 → Ax	
	MOVX.W @Ax + Ix,Dx	111100A*D*0*11**	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + Ix → Ax	1
	MOVX.W Da,@Ax	111100A*D*1*01**	MSW of Da → (Ax)	1
	MOVX.W Da,@Ax +	111100A*D*1*10**	MSW of Da → (Ax), Ax + 2 → Ax	1
	MOVX.W Da,@Ax + Ix	111100A*D*1*11**	MSW of Da → (Ax), Ax + Ix → Ax	1
Y memory data transfer	NOPY	111100*0*0*0**00	Y memory no access	1
	MOVY.W @Ay,Dy	111100*A*D*0**01	(Ay) → MSW of Dy, 0 → LSW of Dy	1
	MOVY.W @Ay + ,Dy	111100*A*D*0**10	(Ay) → MSW of Dy, 0 → LSW of Dy, Ay + 2 → Ay	1
	MOVY.W @Ay + Iy,Dy	111100*A*D*0**11	(Ay) → MSW of Dy, 0 →LSW of Dy, Ay + Iy →Ay	1
	MOVY.W Da,@Ay	111100*A*D*1**01	MSW of Da → (Ay)	1
	MOVY.W Da,@Ay +	111100*A*D*1**10	MSW of Da → (Ay), Ay + 2 → Ay	1
	MOVY.W Da,@Ay + Iy	111100*A*D*1**11	MSW of Da → (Ay), Ay + Iy → Ay	1

MOVS.W @As + ,Ds	111101AADDDD1000	(As) → MSW of Ds, 0 → LSW of Ds, As + 2 → As	1	—
MOVS.W @As + Ix,Ds	111101AADDDD1100	(As) → MSW of Ds, 0 → LSW of Ds, As + Ix → As	1	—
MOVS.W Ds,@-As	111101AADDDD0001	As-2 → As, MSW of Ds → (As)	1	—
MOVS.W Ds,@As	111101AADDDD0101	MSW of Ds → (As)	1	—
MOVS.W Ds,@As +	111101AADDDD1001	MSW of Ds → (As), As + 2 →As	1	—
MOVS.W Ds,@As + Ix	111101AADDDD1101	MSW of Ds → (As), As + Ix → As	1	—
MOVS.L @-As,Ds	111101AADDDD0010	As-4 → As, (As) → Ds	1	—
MOVS.L @As,Ds	111101AADDDD0110	(As) → Ds	1	—
MOVS.L @As + ,Ds	111101AADDDD1010	(As) → Ds, As + 4 → As	1	—
MOVS.L @As + Ix,Ds	111101AADDDD1110	(As) → Ds, As + Ix → As	1	—
MOVS.L Ds,@-As	111101AADDDD0011	As-4 → As, Ds → (As)	1	—
MOVS.L Ds,@As	111101AADDDD0111	Ds → (As)	1	—
MOVS.L Ds,@As +	111101AADDDD1011	Ds → (As), As + 4 → As	1	—
MOVS.L Ds,@As + Ix	111101AADDDD1111	Ds → (As), As + Ix → As	1	—

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

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

	R7 (Ay1)			Yes			
	R8 (Ix)	Yes					
	R9 (Iy)				Yes		
DSP register	A0						Yes
	A1						Yes
	M0						
	M1						
	X0		Yes				
	X1		Yes				
	Y0					Yes	
	Y1					Yes	
	A0G						
	A1G						

Note: Yes: The register which can be set.

PMULS Se,Sf,Dg	0111eeffxxyygguu		
PSUB Sx,Sy,Du PMULS Se,Sf,Dg	111110********** 0110eeffxxyygguu	Sx-Sy ->Du Se*Sf ->Dg (Signed)	1
PADD Sx,Sy,Dz	111110********** 10110001xxyyzzzz	Sx + Sy ->Dz	1
DCT PADD Sx,Sy,Dz	111110********** 10110010xxyyzzzz	If DC=1, Sx + Sy ->Dz If DC=0, nop	1
DCF PADD Sx,Sy,Dz	111110********** 10110011xxyyzzzz	If DC=0, Sx + Sy ->Dz If DC=1, nop	1
PSUB Sx,Sy,Dz	111110********** 10100001xxyyzzzz	Sx-Sy ->Dz	1
DCT PSUB Sx,Sy,Dz	111110********** 10100010xxyyzzzz	If DC=1, Sx-Sy ->Dz If DC=0, nop	1
DCF PSUB Sx,Sy,Dz	111110********** 10100011xxyyzzzz	If DC=0, Sx-Sy ->Dz If DC=1, nop	1
PSHA Sx,Sy,Dz	111110********** 10010001xxyyzzzz	If Sy>=0, Sx<<Sy ->Dz (arithmetic shift) If Sy<0, Sx>>Sy ->Dz	1
DCT PSHA Sx,Sy,Dz	111110********** 10010010xxyyzzzz	If DC=1 & Sy>=0, Sx<<Sy ->Dz (arithmetic shift) If DC=1 & Sy<0, Sx>>Sy ->Dz If DC=0, nop	1
DCF PSHA Sx,Sy,Dz	111110********** 10010011xxyyzzzz	If DC=0 & Sy>=0, Sx<<Sy ->Dz (arithmetic shift) If DC=0 & Sy<0, Sx>>Sy ->Dz If DC=1, nop	1

PCOPY Sy,Dz	111110********** 1111100100yyzzzz	Sy ->Dz	1
DCT PCOPY Sx,Dz	111110********** 11011010xx00zzzz	If DC=1, Sx ->Dz If DC=0, nop	1
DCT PCOPY Sy,Dz	111110********** 1111101000yyzzzz	If DC=1, Sy ->Dz If DC=0, nop	1
DCF PCOPY Sx,Dz	111110********** 11011011xx00zzzz	If DC=0, Sx ->Dz If DC=1, nop	1
DCF PCOPY Sy,Dz	111110********** 1111101100yyzzzz	If DC=0, Sy ->Dz If DC=1, nop	1
PDMSB Sx,Dz	111110********** 10011101xx00zzzz	Sx ->Dz normalization count shift value	1
PDMSB Sy,Dz	111110********** 1011110100yyzzzz	Sy ->Dz normalization count shift value	1
DCT PDMSB Sx,Dz	111110********** 10011110xx00zzzz	If DC=1, normalization count shift value Sx ->Dz If DC=0, nop	1
DCT PDMSB Sy,Dz	111110********** 1011111000yyzzzz	If DC=1, normalization count shift value Sy ->Dz If DC=0, nop	1
DCF PDMSB Sx,Dz	111110********** 10011111xx00zzzz	If DC=0, normalization count shift value Sx ->Dz If DC=1, nop	1
DCF PDMSB Sy,Dz	111110********** 1011111100yyzzzz	If DC=0, normalization count shift value Sy ->Dz If DC=1, nop	1

DCF PINC Sx,Dz	111110********** 10011011xx00zzzz	If DC=0, MSW of Sx + 1 ->Dz If DC=1, nop	1
DCF PINC Sy,Dz	111110********** 1011101100yyzzzz	If DC=0, MSW of Sy+ 1 ->Dz If DC=1, nop	1
PNEG Sx,Dz	111110********** 11001001xx00zzzz	0-Sx ->Dz	1
PNEG Sy,Dz	111110********** 1110100100yyzzzz	0-Sy ->Dz	1
DCT PNEG Sx,Dz	111110********** 11001010xx00zzzz	If DC=1, 0-Sx ->Dz If DC=0, nop	1
DCT PNEG Sy,Dz	111110********** 1110101000yyzzzz	If DC=1, 0-Sy ->Dz If DC=0, nop	1
DCF PNEG Sx,Dz	111110********** 11001011xx00zzzz	If DC=0, 0-Sx ->Dz If DC=1, nop	1
DCF PNEG Sy,Dz	111110********** 1110101100yyzzzz	If DC=0, 0-Sy ->Dz If DC=1, nop	1
POR Sx,Sy,Dz	111110********** 10110101xxyyzzzz	Sx | Sy ->Dz	1
DCT POR Sx,Sy,Dz	111110********** 10110110xxyyzzzz	If DC=1, Sx | Sy ->Dz If DC=0, nop	1
DCF POR Sx,Sy,Dz	111110********** 10110111xxyyzzzz	If DC=0, Sx | Sy ->Dz If DC=1, nop	1

DCT PXOR Sx,Sy,Dz	111110********** 10100110xxyyzzzz	If DC=1, Sx ^ Sy ->Dz If DC=0, nop	1
DCF PXOR Sx,Sy,Dz	111110********** 10100111xxyyzzzz	If DC=0, Sx ^ Sy ->Dz If DC=1, nop	1
PDEC Sx,Dz	111110********** 10001001xx00zzzz	Sx [39:16]-1 ->Dz	1
DCT PDEC Sx,Dz	111110********** 10001010xx00zzzz	If DC=1, Sx [39:16]-1 ->Dz If DC=0, nop	1
DCF PDEC Sx,Dz	111110********** 10001011xx00zzzz	If DC=0, Sx [39:16]-1 ->Dz If DC=1, nop	1
PDEC Sy,Dz	111110********** 1010100100yyzzzz	Sy [31:16]-1 ->Dz	1
DCT PDEC Sy,Dz	111110********** 1010101000yyzzzz	If DC=1, Sy [31:16]-1 ->Dz If DC=0, nop	1
DCF PDEC Sy,Dz	111110********** 1010101100yyzzzz	If DC=0, Sy [31:16]-1 ->Dz If DC=1, nop	1
PCLR Dz	111110********** 1000110100000zzzz	h'00000000 ->Dz	1
DCT PCLR Dz	111110********** 1000111000000zzzz	If DC=1, h'00000000 ->Dz If DC=0, nop	1
DCF PCLR Dz	111110********** 1000111100000zzzz	If DC=0, h'00000000 ->Dz If DC=1, nop	1

DCF PSTS MACH,Dz	111110********** 1100111110000zzzz	If DC=0, MACH ->Dz	1
PSTS MACL,Dz	111110********** 1101110010000zzzz	MACL ->Dz	1
DCT PSTS MACL,Dz	111110********** 1101111100000zzzz	If DC=1, MACL ->Dz	1
DCF PSTS MACL,Dz	111110********** 1101111110000zzzz	If DC=0, MACL ->Dz	1
PLDS Dz,MACH	111110********** 1110011010000zzzz	Dz ->MACH	1
DCT PLDS Dz,MACH	111110********** 1110011100000zzzz	If DC=1, Dz ->MACH	1
DCF PLDS Dz,MACH	111110********** 1110011110000zzzz	If DC=0, Dz ->MACH	1
PLDS Dz,MACL	111110********** 1111111010000zzzz	Dz ->MACL	1
DCT PLDS Dz,MACL	111110********** 1111111100000zzzz	If DC=1, Dz ->MACL	1
DCF PLDS Dz,MACL	111110********** 1111111110000zzzz	If DC=0, Dz ->MACL	1
PADDC Sx,Sy,Dz	111110********** 10110000xxyyzzzz	Sx + Sy + DC ->Dz Carry ->DC	1

PRND Sx,Dz	111110********** 10011000xx00zzzz	Sx + h'00008000 ->Dz h'0000 ->LSW of Dz	1
PRND Sy,Dz	111110********** 1011100000yyzzzz	Sy + h'00008000 ->Dz h'0000 ->LSW of Dz	1

Note: * See table 3.19.

0000	Rn	Fx	0001				
0000	Rn	00MD	0010	STC SR, Rn	STC GBR, Rn	STC VBR, Rn	STC SSR
0000	Rn	01MD	0010	STC SPC, Rn	STC MOD, Rn	STC RS, Rn	STC RE,
0000	Rn	10MD	0010	STC R0_BANK, Rn	STC R1_BANK, Rn	STC R2_BANK, Rn	STC R3_
0000	Rn	11MD	0010	STC R4_BANK, Rn	STC R5_BANK, Rn	STC R6_BANK, Rn	STC R7_
0000	Rm	00MD	0011	BSRF Rm		BRAF Rm	
0000	Rm	10MD	0011	PREF @Rm			
0000	Rn	Rm	01MD	MOV.B Rm, @(R0, Rn)	MOV.W Rm, @(R0, Rn)	MOV.L Rm, @(R0, Rn)	MUL.L R
0000	0000	00MD	1000	CLRT	SETT	CLRMAC	LDTLB
0000	0000	01MD	1000	CLRS	SETS		
0000	0000	10MD	1000				
0000	0000	11MD	1000				
0000	0000	Fx	1001	NOP	DIV0U		
0000	0000	Fx	1010				
0000	0000	Fx	1011	RTS	SLEEP	RTE	
0000	Rn	Fx	1000				
0000	Rn	Fx	1001			MOVT Rn	
0000	Rn	00MD	1010	STS MACH, Rn	STS MACL, Rn	STS PR, Rn	
0000	Rn	01MD	1010			STS DSR, Rn	STS A0,
0000	Rn	10MD	1010	STS X0, Rn	STS X1, Rn	STS Y0, Rn	STS Y1,
0000	Rn	Fx	1011				

0011	Rn	Rm	00MD	CMP/EQ Rm, Rn		CMP/HS Rm, Rn	CMP/GE R
0011	Rn	Rm	01MD	DIV1 Rm, Rn	DMULU.L Rm,Rn	CMP/HI Rm, Rn	CMP/GT R
0011	Rn	Rm	10MD	SUB Rm, Rn		SUBC Rm, Rn	SUBV Rm
0011	Rn	Rm	11MD	ADD Rm, Rn	DMULS.L Rm,Rn	ADDC Rm, Rn	ADDV Rm
0100	Rn	Fx	0000	SHLL Rn	DT Rn	SHAL Rn	
0100	Rn	Fx	0001	SHLR Rn	CMP/PZ Rn	SHAR Rn	
0100	Rn	Fx	0010	STS.L MACH, @–Rn	STS.L MACL, @–Rn	STS.L PR, @–Rn	
0100	Rn	00MD	0011	STC.L SR, @–Rn	STC.L GBR, @–Rn	STC.L VBR, @–Rn	STC.L SS
0100	Rn	01MD	0011	STC.L SPC, @–Rn	STC.L MOD, @–Rn	STC.L RS, @–Rn	STC.L R
0100	Rn	10MD	0011	STC.L R0_BANK, @–Rn	STC.L R1_BANK, @–Rn	STC.L R2_BANK, @–Rn	STC.L R3_BANK,
0100	Rn	11MD	0011	STC.L R4_BANK, @–Rn	STC.L R5_BANK, @–Rn	STC.L R6_BANK, @–Rn	STC.L R7_BANK,
0100	Rn	Fx	0100	ROTL Rn	SETRC Rn	ROTCL Rn	
0100	Rn	Fx	0101	ROTR Rn	CMP/PL Rn	ROTCR Rn	
0100	Rm	00MD	0110	LDS.L @Rm+, MACH	LDS.L @Rm+, MACL	LDS.L @Rm+, PR	
0100	Rm	01MD	0110			LDS.L @Rm+, DSR	LDS.L @R
0100	Rm	10MD	0110	LDS.L @Rm+, X0	LDS.L @Rm+, X1	LDS.L @Rm+, Y0	LDS.L @R

0100	Rm	00MD	1010	LDS Rm, MACH	LDS Rm, MACL	LDS Rm, PR	
0100	Rm	01MD	1010			LDS Rm, DSR	LDS Rm
0100	Rm	10MD	1010	LDS Rm, X0	LDS Rm, X1	LDS Rm, Y0	LDS Rm
0100	Rm/ Rn	Fx	1011	JSR @Rm	TAS.B @Rn	JMP @Rm	
0100	Rn	Rm	1100	SHAD Rm, Rn			
0100	Rn	Rm	1101	SHLD Rm, Rn			
0100	Rm	00MD	1110	LDC Rm, SR	LDC Rm, GBR	LDC Rm, VBR	LDC Rm
0100	Rm	01MD	1110	LDC Rm, SPC	LDC Rm, MOD	LDC Rm, RS	LDC Rm
0100	Rm	10MD	1110	LDC Rm, R0_BANK	LDC Rm, R1_BANK	LDC Rm, R2_BANK	LDC Rm
0100	Rm	11MD	1110	LDC Rm, R4_BANK	LDC Rm, R5_BANK	LDC Rm, R6_BANK	LDC Rm
0100	Rn	Rm	1111	MAC.W @Rm+, Rn+			
0101	Rn	Rm	disp	MOV.L @(disp:4, Rm), Rn			
0110	Rn	Rm	00MD	MOV.B @Rm, Rn	MOV.W @Rm, Rn	MOV.L @Rm, Rn	MOV Rn
0110	Rn	Rm	01MD	MOV.B @Rm+, Rn	MOV.W @Rm+, Rn	MOV.L @Rm+, Rn	NOT Rn
0110	Rn	Rm	10MD	SWAP.B Rm, Rn	SWAP.W Rm, Rn	NEGC Rm, Rn	NEG Rn
0110	Rn	Rm	11MD	EXTU.B Rm, Rn	EXTU.W Rm, Rn	EXTS.B Rm, Rn	EXTS.W
0111	Rn	imm		ADD #imm : 8, Rn			
1000	00MD	Rn	disp	MOV.B R0, @(disp: 4, Rn)	MOV.W R0, @(disp: 4, Rn)	SETRC #imm	
		imm					

1100	00MD	imm/disp	MOV.B R0, @(disp: 8, GBR)	MOV.W R0, @(disp: 8, GBR)	MOV.L R0, @(disp: 8, GBR)	TRAP A
1100	01MD	disp	MOV.B @(disp: 8, GBR), R0	MOV.W @(disp: 8, GBR), R0	MOV.L @(disp: 8, GBR), R0	MOVA @(disp: 8,
1100	10MD	imm	TST #imm: 8, R0	AND #imm: 8, R0	XOR #imm: 8, R0	OR #imm
1100	11MD	imm	TST.B #imm: 8, @(R0, GBR)	AND.B #imm: 8, @(R0, GBR)	XOR.B #imm: 8, @(R0, GBR)	OR.B #imm: 8, @(
1101	Rn	disp	MOV.L @(disp: 8, PC), Rn			
1110	Rn	imm	MOV #imm:8, Rn			
1111	00**	********	MOVX.W, MOVY.W Double data transfer instruction			
1111	01**	********	MOVS.W, MOVS.L Single data transfer instruction			
1111	10**	********	MOVX.W, MOVY.W Double data transfer instruction, with DSP parallel operation ins (32-bit instruction)			
1111	11**	********				

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

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

support the above functions are called exception handling. This LSI has two types of exc
general exceptions and interrupts. The user can execute the required processing by assig
exception handling routines corresponding to the required exception processing and then
the source program.

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

4.1 Register Descriptions

There are five registers for exception handling. A register with an undefined initial value
be initialized by the software. Refer to section 23, List of Registers, for the addresses an
sizes of these registers.

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

Figure 4.1 Register Bit Configuration

4.1.1 TRAPA Exception Register (TRA)

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

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

				should always be 0.
11 to 0	EXPEVT	*	R/W	12-bit Exception Code

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

4.1.3 Interrupt Event Register (INTEVT)

INTEVT is assigned to address H′FFFFFFD8 and consists of the exception code or the i
priority code. Whether the occurrence of an interrupt sets the exception code or the inter
priority code depends on the interrupt sources. (See section 8.3.5, Interrupt Exception H
and Priority, for details.) These exception codes and interrupt priority codes are automat
specified by the hardware when an exception occurs. Only bits 11 to 0 in INTEVT can b
written using the software.

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

				should always be 0.
11 to 0	INTEVT2	—	R	12-bit Exception Code

4.1.5 Exception Address Register (TEA)

TEA is assigned to address H′FFFFFFFC and the logical address for an exception occurre
stored in this register when an exception related to memory accesses occurs. TEA can be
using the software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TEA	0	R/W	Logical address for Exception Occurren

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

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

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

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

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

A vector address for general exceptions is determined by adding a vector offset to a vecto address. The vector offset for general exceptions other than the TLB error exception is H′00000100. The vector offset for interrupts is H′00000600. The vector base address is lo into the vector base register (VBR) using the software. The vector base address should re the P1 or P2 fixed physical address space.

4.2.3 Exception Codes

The exception codes are written to bits 11 to 0 in EXPEVT (for reset or general exceptior INTEVT2 (for interrupt requests) to identify each specific exception event. See section 8, Controller (INTC), for details on the exception codes for interrupt requests. Table 4.1 list exception codes for resets and general exceptions.

4.2.4 Exception Request and BL Bit (Multiple Exception Prevention)

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

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

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

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

Re-Execution Type and Processing-Completion Type Exceptions: All exceptions are
into two types: a re-execution type and a processing-completion type. If a re-execution t
exception is accepted, the current instruction executed when the exception is accepted is
terminated and the instruction address is saved to the SPC. After returning from the exce
processing, program execution resumes from the instruction where the exception was ac
a processing-completion type exception, the current instruction executed when the excep
accepted is completed, the next instruction address is saved to the SPC, and then the exc
processing is executed.

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

re-execution type)

4. An exception caused by an instruction decode (General illegal instruction exceptions illegal instruction exceptions: re-execution type, unconditional trap: processing-comp type)
5. An exception related to data access (CPU address error and MMU related exceptions execution type)
6. Unconditional trap (processing-completion type)
7. A user break other than one before instruction execution (processing-completion typ
8. DMA address error (processing-completion type)

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

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

	*5 TLB miss (instruction access) *4	2	1-1	Reset	H'040	H
	TLB invalid (instruction access)*4	2	1-2	Reset	H'040	H
	TLB protection violation (instruction access)*4	2	1-3	Reset	H'0A0	H
	Illegal general instruction exception	2	2	Reset	H'180	H
	Illegal slot instruction exception	2	2	Reset	H'1A0	H
Completed	Unconditional trap (TRAPA instruction)	2	4	Reset	H'160	H
Re-executed	CPU address error (data read/write)*4	2	3	Reset	H'0E0/ H'100	H
	*5 TLB miss (data read/write)*4	2	3-1	Reset	H'040/ H'060	H
	TLB invalid (data read/write)*4	2	3-2	Reset	H'040/ H'060	H
	TLB protection violation (data read/write)*4	2	3-3	Reset	H'0A0/ H'0C0	H
	Initial page write (data write)*4	2	3-4	Reset	H'080	H
Completed	User breakpoint (After instruction execution, address)	2	5	Ignored	H'1E0	H

are not requested.

2. For details on priorities in multiple interrupt sources, refer to section 8, Interrup Controller (INTC).
3. If an interrupt is accepted, the interrupt source register (EXPEVT) is not chang interrupt source code is specified in interrupt source register 2 (EXPEVT2). Fo refer to section 8, Interrupt Controller (INTC).
4. If one of these exceptions occurs in a specific part of the repeat loop, a specifi and vector offset are specified.
5. These exception codes are valid when the MMU is used.

Conditions

Power-on reset is request

- Operations

  Set EXPEVT to H'000, initialize the CPU and on-chip peripheral modules, and branc reset vector H′A0000000. For details, refer to the register descriptions in the relevan

  Be sure to perform power-On Reset at the time of a power supply injection.

Manual Reset:

- Conditions

  Manual reset is request

- Operations

  Set EXPEVT to H'020, initialize the CPU and on-chip peripheral modules, and branc reset vector H′A0000000. For details, refer to the register descriptions in the relevan

H-UDI Reset:

- Conditions

  An H-UDI reset command is input (see section 22.4.4 H-UDI Reset.)

- Operations

  EXPEVT is set to H'000, vector base register (VBR) and status register (SR) are init and branched to the reset vector (H'A0000000). VBR is cleared to H'00000000 by initialization. In SR, the MD, RB, and BL bits are set to 1, the DSP bit is cleared to 0 interrupt mask bits (I3 to I0) are set to B'1111. Then, the CPU and on-chip peripher are initialized. For details, see the Register Description in each section.

mode

- Types
  Instruction synchronous, re-execution type
- Save address
  Instruction fetch: An instruction address to be fetched when an exception occurred
  Data access: An instruction address where an exception occurs (a delayed branch inst
  address if an instruction is assigned to a delay slot)
- Exception code
  An exception occurred during read: H′0E0
  An exception occurred during write: H′100
- Remarks
  The logical address (32 bits) that caused the exception is set in TEA.

Illegal general instruction exception:

- Conditions
  — When undefined code not in a delay slot is decoded
  Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, B

Note: For details on undefined code, refer to table 2.12 in section 2.6.2, Operation Code
When an undefined code other than H′FC00 to H′FFFF is decoded, operation can
guaranteed.

  — When a privileged instruction not in a delay slot is decoded in user mode
  Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that acces
  with LDC/STC are not privileged instructions.

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

- Remarks

  The exception is a processing-completion type, so PC of the instruction after the TRA instruction is saved to SPC. The 8-bit immediate value in the TRAPA instruction is qu and set in TRA[9:2].

User break point trap:

- Conditions

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

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

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

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

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

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

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

  H′1E0
- Remarks

  For details on the user break controller, refer to section 9, User Break Controller.

branch instruction destination address if an instruction is assigned to a delay slot)

- Exception code
  H′5C0
- Remarks
  An exception occurs when a DMA transfer is executed while an illegal instruction ac
  described above is specified in the DMAC. Since the DMA transfer is performed
  asynchronously with the CPU instruction operation, an exception is also requested
  asynchronously with the instruction execution. For details on the DMAC, refer to se
  Direct Memory Access Controller (DMAC).

4.3.3 General Exceptions (MMU Exceptions)

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

A vector offset for a TLB miss exception is defined as H′00000400 to simplify exceptio
determination. For details on MMU exception operations, refer to section 5, Memory M
Unit (MMU).

TLB miss exception:

- Conditions
  Comparison of TLB addresses shows no address match.
- Types
  Instruction synchronous, re-execution type

up TLB miss processing, the offset differs from other exceptions.

TLB invalid exception:

- Conditions
  Comparison of TLB addresses shows address match but V = 0.
- Types
  Instruction synchronous, re-execution type
- Save address
  Instruction fetch: An instruction address to be fetched when an exception occurred
  Data access: An instruction address where an exception occurs (a delayed branch inst
  address if an instruction is assigned to a delay slot)
- Exception code
  An exception occurred during read: H′040
  An exception occurred during write: H′060
- Remarks
  The logical address (32 bits) that caused the exception is set in TEA and the MMU re
  are updated.

TLB protection exception:

- Conditions
  When a hit access violates the TLB protection information (PR bits).
- Types
  Instruction synchronous, re-execution type

Initial page write exception:

- Conditions
  A hit occurred to the TLB for a data write access, but D = 0.
- Types
  Instruction synchronous, re-execution type
- Save address
  Instruction fetch: An instruction address to be fetched when an exception occurred
  Data access: An instruction address where an exception occurs (a delayed branch ins
  address if an instruction is assigned to a delay slot)
- Exception code
  H′080
- Remarks
  The logical address (32 bits) that caused the exception is set in TEA and the MMU r
  are updated.

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

4.4.2 CPU Address Error

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

4.4.3 Exception in Repeat Control Period

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

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

3. An interrupt, break or DMA address error request is retained while SR.RC[11
4. A specific exception code is specified while SR.RC[11:0] ≥1.

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

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

```
RptEnd:   RptDtct3                ; [C2][Repeat end instruction]
          InstrNext               ; [A]
```

- Example 3: Repeat loop consisting of two instructions

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

```
RptEnd:   RptDecF                 ; [C1][Repeat start
                                      instruction]== [Repeat e
                                      instruction]
          InstrNext               ; [A]
```

Instruction where Exception Occurs	Number of Instructions in Repeat Loop			
	1	2	3	4 or Grea
RptDtct	RptDtct	RptDtct	RptDtct	RptDtct
RptDtct1	RptDtct1	RptDtct1	RptDtct1	RptDtct1
RptDtct2	—	RptDtct1	RptDtct1	RS-4
RptDtct3	—	—	RptDtct1	RS-2

Note: The following labels are used here.

RptDtct: Repeat detection instruction address

RptDtct1: An instruction address one instruction following the repeat detection inst

RptDtct2: An instruction address two instruction following the repeat detection inst

RptDtct3: An instruction address three instruction following the repeat detection in

RS: Repeat start instruction address

If a re-execution type exception is accepted at an instruction in the hatched areas a return address to be saved in the SPC is incorrect. If RC[11:0] is 1 or 0, a correct r address is saved in the SPC.

Illegal Instruction Exception in Repeat Control Period: If one of the following instruc executed at the address following RptDtct1, a general illegal instruction exception occurs details on an address to be saved in the SPC, refer to the description in section 4.4.3, Exc Repeat Control Period.

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

- Interrupt, DMA address error

  An exception request is not accepted and retained at instructions [B] and [C]. If an in
  indicates as [A] is executed at the next time, an exception request is accepted.* As sh
  program examples 1 to 4, any interrupt or DMA address error cannot be accepted in
  loop consisting of four instructions or less.

Note: * An interrupt request or a DMA address error exception request is retained in
  interrupt controller (INTC) and the direct memory access controller (DMAC)
  CPU can accept a request.

- User break before instruction execution

  A user break before instruction execution is accepted at instruction [B], and an addre
  instruction [B] is saved in the SPC. This exception cannot be accepted at instruction
  the exception request is retained until an instruction [A] or [B] is executed at the nex
  Then, the exception request is accepted before an instruction [A] or [B] is executed.
  case, an address of instruction [A] or [B] is saved in the SPC.

- User break after instruction execution

  A user break after instruction execution cannot be accepted at instructions [B] and [C
  exception request is retained until an instruction [A] or [B] is executed at the next tim
  the exception request is accepted before an instruction [A] or [B] is executed. In this
  address of instruction [A] or [B] is saved in the SPC.

Table 4.4 Exception Acceptance in Repeat Loop

Exception Type	Instruction [B]	Instruction [C]
Interrupt	Not accepted	Not accepted
DMA address error	Not accepted	Not accepted
User break before instruction execution	Accepted	Not accepted
User break after instruction execution	Not accepted	Not accepted

repeat control period.

Note: In a repeat loop consisting of one to three instructions, some restrictions apply to detection instructions and all the remaining instructions. In a repeat loop consisti or more instructions, restrictions apply to only the four instructions that include a end instruction. The restriction occurs when SR.RC[11:0] ≥ 1.

Table 4.5 Instruction Where a Specific Exception Occurs when Memory Access E Occurs in Repeat Control (SR.RC[11:0] ≥ 1)

Instruction where Exception Occurs	Number of Instructions in Repeat Loop			
	1	2	3	4 or Grea
RptDtct				
RptDtct1	Instruction/data access	Instruction/data access	Instruction/data access	Instructio access
RptDtct2	—	Instruction/data access	Instruction/data access	Instructio access
RptDtct3	—	—	Instruction/data access	Instructio access

Note: The following labels are used here.
RptDtct: Repeat detection instruction
RptDtct1: An instruction of one instruction following the repeat detection instructio
RptDtct2: An instruction of two instruction following the repeat detection instruc
RptDtct3: An instruction of three instruction following the repeat detection instru

control period.

Note: In a repeat loop consisting of one to three instructions, some restrictions apply to detection instructions and all the remaining instructions. In a repeat loop consist or more instructions, restrictions apply to only the four instructions that include end instruction. The restriction occurs when SR.RC[11:0] ≥ 1.

4.5 Usage Notes

1. An instruction assigned at a delay slot of the RTE instruction is executed after the c the SSR is restored into the SR. An acceptance of an exception related to instructior determined according to the SR before restore. An acceptance of other exceptions is determined by processing mode of the SR after restore, and BL bit value. A process completion type exception is accepted before an instruction at the RTE branch desti address is executed. However, note that the correct operation cannot be guaranteed execution type exception occurs.
2. In an instruction assigned at a delay slot of the RTE instruction, a user break cannot accepted.
3. If the MD and BL bits of the SR register are changed by the LDC instruction, an ex accepted according to the changed SR value from the next instruction.* A processin completion type exception is accepted after the next instruction is executed. An inte DMA address error in re-execution type exceptions are accepted before the next ins executed.

Note: * If an LDC instruction is executed for the SR, the following instructions are re and an instruction fetch exception is accepted according to the modified SR v

The MMU is a feature designed to make efficient use of physical memory. As shown in
if a process is smaller in size than the physical memory, the entire process can be mappe
physical memory. However, if the process increases in size to the extent that it no longe
physical memory, it becomes necessary to partition the process and to map those parts r
execution onto memory as occasion demands (figure 5.1 (1)). Having the process itself c
this mapping onto physical memory would impose a large burden on the process. To lig
burden, the idea of virtual memory was born as a means of performing en bloc mapping
physical memory (figure 5.1 (2)). In a virtual memory system, substantially more virtual
than physical memory is provided, and the process is mapped onto this virtual memory.
process only has to consider operation in virtual memory. Mapping from virtual memory
physical memory is handled by the MMU. The MMU is normally controlled by the oper
system, switching physical memory to allow the virtual memory required by a process t
mapped onto physical memory in a smooth fashion. Switching of physical memory is pe
via secondary storage, etc.

The virtual memory system that came into being in this way is particularly effective in a
sharing system (TSS) in which a number of processes are running simultaneously (figur
If processes running in a TSS had to take mapping onto virtual memory into considerati
running, it would not be possible to increase efficiency. Virtual memory is thus used to r
load on the individual processes and so improve efficiency (figure 5.1 (4)). In the virtual
system, virtual memory is allocated to each process. The task of the MMU is to perform
mapping of these virtual memory areas onto physical memory. It also has a memory pro
feature that prevents one process from inadvertently accessing another process's physica

When address translation from virtual memory to physical memory is performed using t
it may occur that the relevant translation information is not recorded in the MMU, with t
that one process may inadvertently access the virtual memory allocated to another proce
case, the MMU will generate an exception, change the physical memory mapping, and r
new address translation information.

address translation. With the paging method, the unit of translation is a fixed-size address (usually of 1 to 64 kbytes) called a page.

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

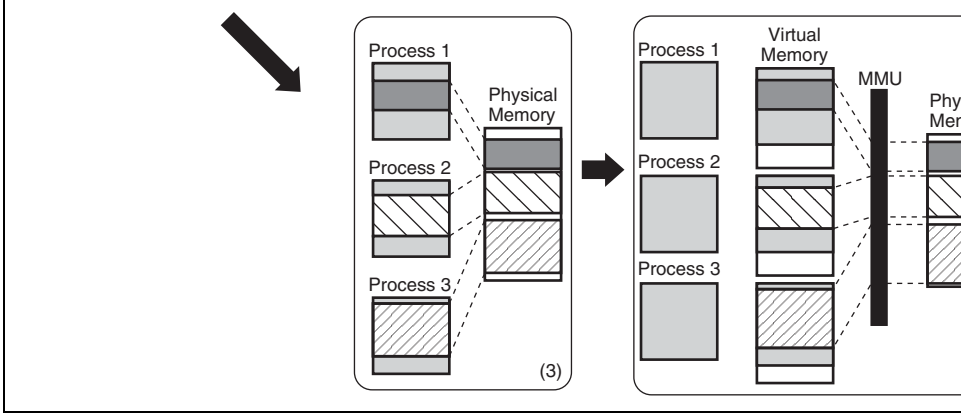


Figure 5.1 MMU Functions

5.1.1 MMU of This LSI

Virtual Address Space: This LSI supports a 32-bit virtual address space that enables ac
4-Gbyte address space. As shown in figures 5.2 and 5.3, the virtual address space is divi
several areas. In privileged mode, a 4-Gbyte space comprising areas P0 to P4 are access
user mode, a 2-Gbyte space of U0 area is accessible, and a 16-Mbyte space of Uxy area
accessible if the DSP bit of the SR register is set to 1. Access to any area (excluding the
and Uxy area) in user mode will result in an address error.

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

in area 1 in the physical address space via the TLB, the C bit of the corresponding pag
be cleared to 0.

2. P1 Area

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

3. P2 Area

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

4. P4 Area

   The P4 area is mapped to the on-chip I/O of this LSI. This area cannot be accessed via
   cache and cannot be address-translated by the TLB. Figure 5.4 shows the configuratio
   P4 area.

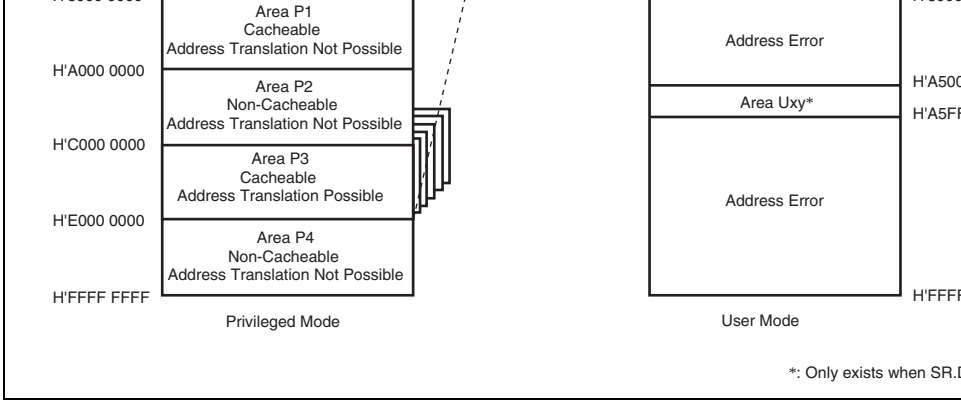


Figure 5.2 Virtual Address Space (MMUCR.AT = 1)

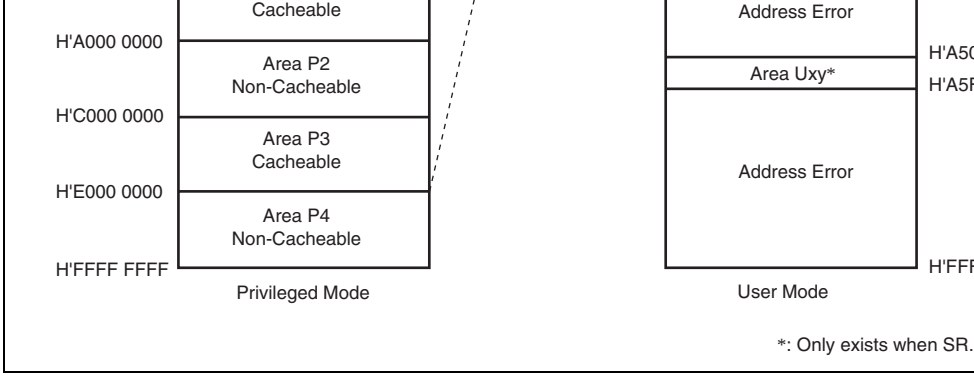


Figure 5.3 Virtual Address Space (MMUCR.AT = 0)

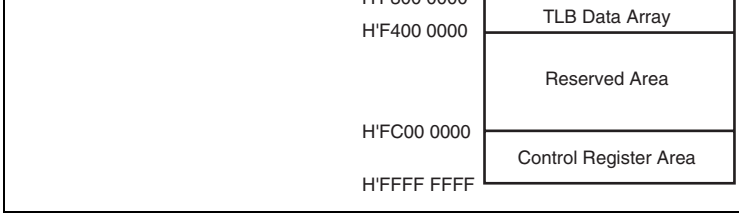


Figure 5.4 P4 Area

The area from H'F000 0000 to H'F0FF FFFF is for direct access to the cache address arr
more information, see section 6.4, Memory-Mapped Cache.

The area from H'F100 0000 to H'F1FF FFFF is for direct access to the cache data array.
information, see section 6.4, Memory-Mapped Cache.

The area from H'F200 0000 to H'F2FF FFFF is for direct access to the TLB address arra
more information, see section 5.6, Memory-Mapped TLB.

The area from H'F300 0000 to H'F3FF FFFF is for direct access to the TLB data array. I
information, see section 5.6, Memory-Mapped TLB.

The area from H'FC00 0000 to H'FFFF FFFF is reserved for registers of the on-chip per
modules. For more information, see section 23, List of Registers.

5. Uxy Area

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

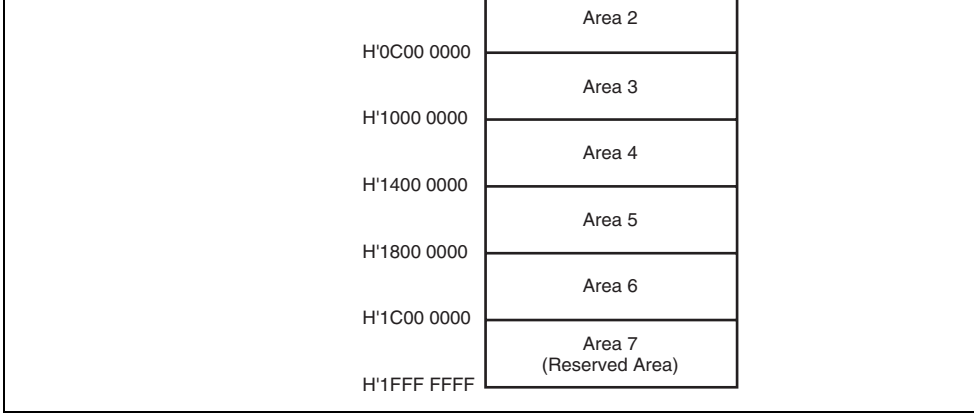


Figure 5.5 External Memory Space

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

If the virtual address is not registered in the TLB, a TLB miss exception occurs and proce will shift to the TLB miss handler. In the TLB miss handler, the TLB address translation external memory is searched and the corresponding physical address and the page control information are registered in the TLB. After returning from the handler, the instruction th the TLB miss is re-executed. When the MMU is enabled, address translation information

Single Virtual Memory Mode and Multiple Virtual Memory Mode: There are two v
memory modes: single virtual memory mode and multiple virtual memory mode. In sing
memory mode, multiple processes run in parallel using the virtual address space exclusi
the physical address corresponding to a given virtual address is specified uniquely. In m
virtual memory mode, multiple processes run in parallel sharing the virtual address spac
given virtual address may be translated into different physical addresses depending on th
By the value set to the MMU control register (MMUCR), either single or multiple virtua
selected.

In terms of operation, the only difference between single virtual memory mode and mult
virtual memory mode is in the TLB address comparison method (see section 5.3.3, TLB
Comparison).

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

In single virtual memory mode, the ASID is used to provide memory protection for proc
running simultaneously and using the virtual address space exclusively (see section 5.3.
Address Comparison).

5.2 Register Descriptions

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

The page table entry register high (PTEH) register residing at address H'FFFFFFF0, whic
consists of a virtual page number (VPN) and ASID. The VPN set is the VPN of the virtua
at which the exception is generated in case of an MMU exception or address error except
When the page size is 4 kbytes, the VPN is the upper 20 bits of the virtual address, but in
the upper 22 bits of the virtual address are set. The VPN can also be modified by software
ASID, software sets the number of the currently executing process. The VPN and ASID a
recorded in the TLB by the LDTLB instruction.

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

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

				These bits are always read as 0. The write should always be 0.
28 to 10	PPN	—	R	Number of Physical Page
9	—	0	R/W	Page management information
8	V	—		For more details, see section 5.3, TLB Fun
7	—	0		
6, 5	PR	—		
4	SZ	—		
3	C	—		
2	D	—		
1	SH	—		
0	—	0		

5.2.3 Translation Table Base Register (TTB)

The translation table base register (TTB) residing at address H'FFFFFFF8, which points base address of the current page table. The hardware does not set any value in TTB auto TTB is available to software for general purposes. The initial value is undefined.

5.2.4 MMU Control Register (MMUCR)

The MMU control register (MMUCR) residing at address H'FFFFFFE0, which makes th settings described in figure 5.3. Any program that modifies MMUCR should reside in th area.

				should always be 0.
5, 4	RC	All 0	R/W	Random counter A 2-bit random counter that is automatically by hardware according to the following rule event of an MMU exception. When a TLB miss exception occurs, all of T way corresponding to the virtual address at the exception occurred are checked. If all w valid, 1 is added to RC; if there is one or mo invalid way, they are set by priority from wa the order way 0, way 1, way 2, way 3. In the of an MMU exception other than a TLB miss exception, the way which caused the excep set in RC.
3	—	0	R	Reserved These bits are always read as 0. The write should always be 0.
2	TF	0	R/W	TLB flush Write 1 to flush the TLB (clear all valid bits TLB to 0). When they are read, 0 is always
1	IX	0	R/W	Index mode 0: VPN bits 16 to 12 are used as the TLB in number. 1: The value obtained by EX-ORing ASID b in PTEH and VPN bits 16 to 12 is used a TLB index number.

5.3.1 Configuration of the TLB

The TLB caches address translation table information located in the external memory. T
translation table stores the logical page number and the corresponding physical number,
address space identifier, and the control information for the page, which is the unit of ad
translation. Figure 5.6 shows the overall TLB configuration. The TLB is 4-way set assoc
with 128 entries. There are 32 entries for each way. Figure 5.7 shows the configuration o
addresses and TLB entries.

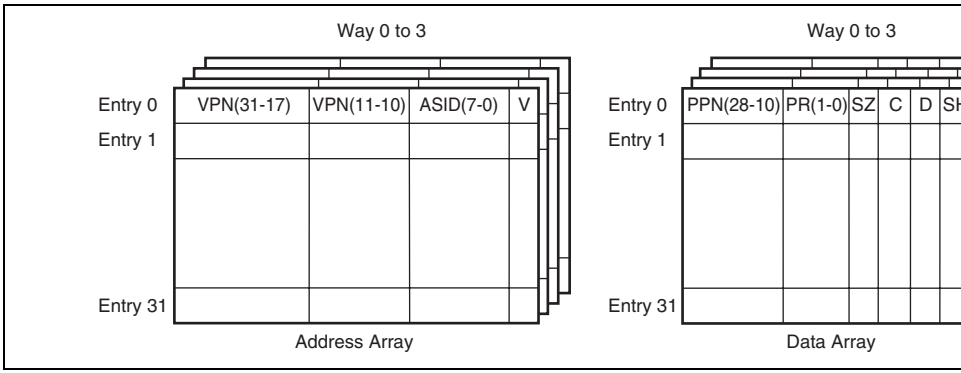


Figure 5.6 Overall Configuration of the TLB

page. Since VPN bits 16 to 12 are used as the index number, they are not stored in the TLB entry. Attention must be paid to the synonym problem (see section 5.4.4, Avoiding Synonym Problems).

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

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

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

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

PPN: Physical page number
Upper 22 bits of physical address. PPN bits 11 to10 are not used in case of a 4-kbyte page.

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

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

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

Figure 5.7 Virtual Address and TLB Structure

run simultaneously in the same virtual address space (multiple virtual memory) and a sp
entry is selected by indexing of each process. In single virtual memory mode (MMUCR
IX bit should be set to 0. Figures 5.8 and 5.9 show the indexing schemes.

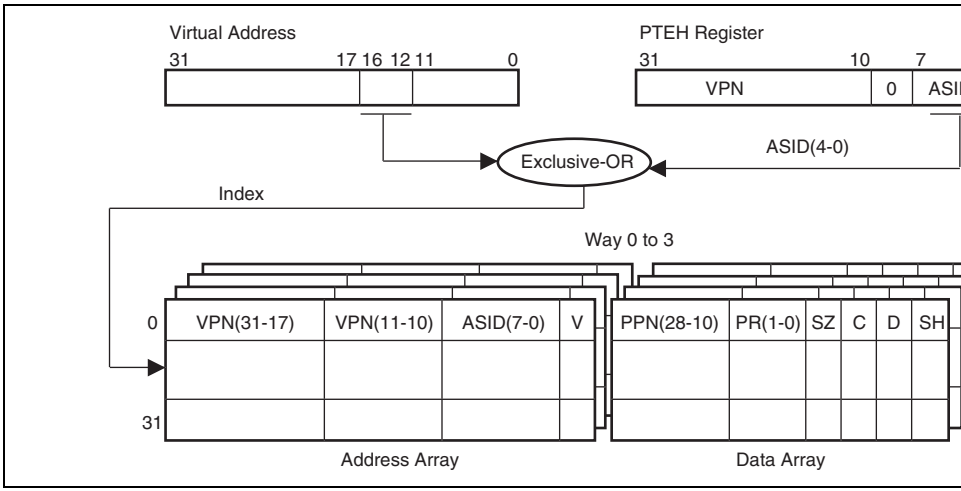


Figure 5.8 TLB Indexing (IX = 1)

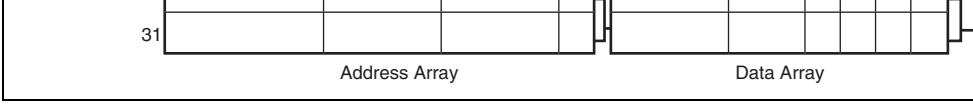


Figure 5.9 TLB Indexing (IX = 0)

5.3.3 TLB Address Comparison

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

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

1. If there are two identical TLB entries with the same VPN and a setting is made such t
   TLB hit is made only by a process with ASID = H'FF when one is in the shared state
   and the other in the non-shared state (SH = 0), then if the ASID in PTEH is set to H'F
   a possibility of simultaneous TLB hits in both these ways.
2. If several entries which have different ASID with the same VPN are registered in sing
   memory mode, there is the possibility of simultaneous TLB hits in more than one way
   accessing the corresponding page in privileged mode. Several entries with the same V
   not be registered in single virtual memory mode.
3. There is the possibility of simultaneous TLB hits in more than one way. These hits ma
   depending on the contents of ASID in PTEH when a page to which SH is set 1 is regis
   the TLB in index mode (MMUCR.IX = 1). Therefore a page to which SH is set 1 mus

are compared. ASIDs are compared when there is no sharing between processes (SH = 0
when there is sharing (SH = 1).

When single virtual memory is supported (MMUCR.SV = 1) and privileged mode is eng
(SR.MD = 1), all process resources can be accessed. This means that ASIDs are not com
when single virtual memory is supported and privileged mode is engaged. The objects o
comparison are shown in figure 5.10.

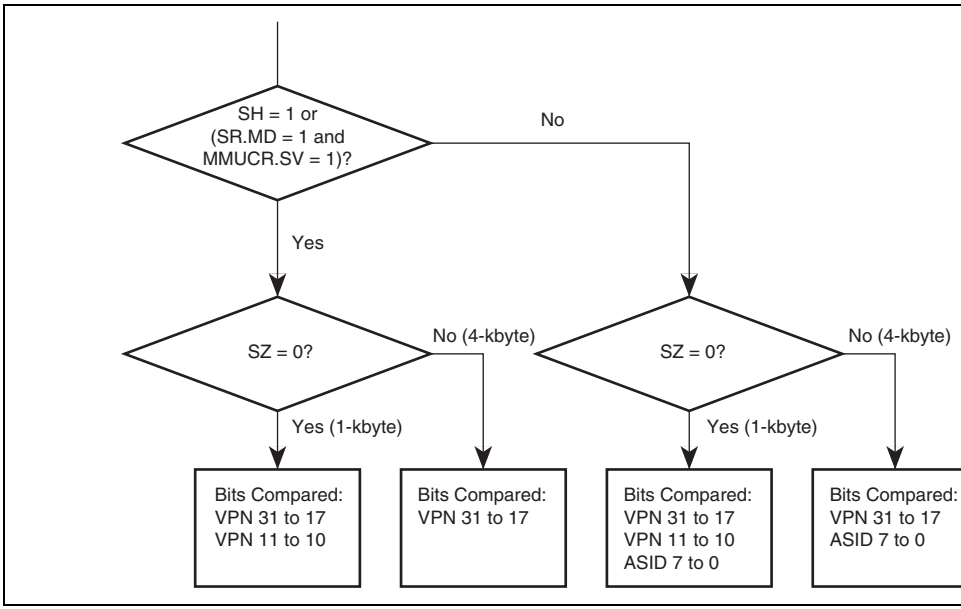


Figure 5.10 Objects of Address Comparison

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

Access states designated by the D, C, and PR bits are shown in table 5.1.

PR bit	00	Permitted	TLB protection violation exception	TLB protection violation exception	TLB prot violation exceptio
	01	Permitted	Permitted	TLB protection violation exception	TLB prot violation exceptio
	10	Permitted	TLB protection violation exception	Permitted	TLB prot violation exceptio
	11	Permitted	Permitted	Permitted	Permitte

details of the determination method and the hardware processing, see section 5.5, MM
Exceptions.

5.4.2 MMU Software Management

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

1. MMU register setting

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

   TLB entry recording can be done in two ways by using the LDTLB instruction, or by
   directly to the memory-mapped TLB. For TLB entry deletion and reading, the memor
   allocation TLB can be accessed. See section 5.4.3, MMU Instruction (LDTLB), for de
   the LDTLB instruction, and section 5.6, Memory-Mapped TLB, for details of the mem
   mapped TLB.
3. MMU exception processing

   When an MMU exception is generated, it is handled on the basis of information set fr
   hardware side. See section 5.5, MMU Exceptions, for details.

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

Figure 5.11 shows the case where the IX bit in MMUCR is 0.

When an MMU exception occurs, the virtual page number of the virtual address that cau
exception is set in PTEH by hardware. The way is set in the RC bit of MMUCR for each
exception according to the rules (see section 5.2.4, MMU Control Register (MMUCR)).
Consequently, if the LDTLB instruction is issued after setting only PTEL in the MMU e
processing routine, TLB entry recording is possible. Any TLB entry can be updated by s
rewriting of PTEH and the RC bits in MMUCR.

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

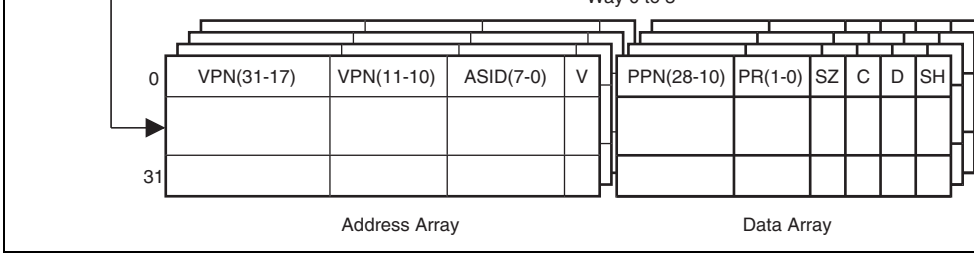


Figure 5.11 Operation of LDTLB Instruction

5.4.4 Avoiding Synonym Problems

When a 1- or 4-kbyte page is recorded in a TLB entry, a synonym problem may arise. If a
of virtual addresses are mapped onto a single physical address, the same physical address
be recorded in a number of cache entries, and it will not be possible to guarantee data con
The reason that this problem occurs is explained below with reference to figure 5.12. The
relationship between bit n of the virtual address and cache size is shown in the following

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

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

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

1. When address translation information whereby a number of 1-kbyte page TLB entrie translated into the same physical address is recorded in the TLB, ensure that the VPN are the same.
2. When address translation information whereby a number of 4-kbyte page TLB entrie translated into the same physical address is recorded in the TLB, ensure that the VPN the same.
3. Do not use the same physical addresses for address translation information of differe sizes.

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

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

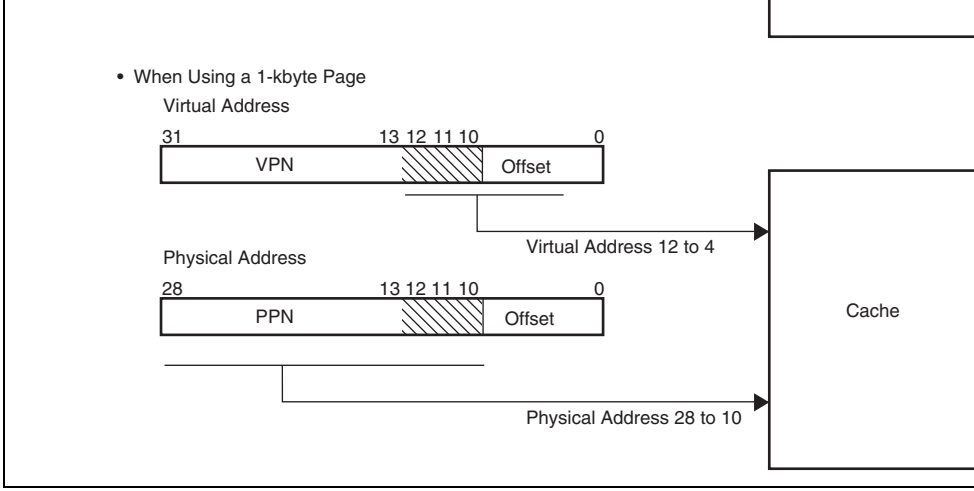


Figure 5.12 Synonym Problem (32-kbyte Cache)

compared and no match is found. TLB miss exception processing includes both hardwar
software operations.

Hardware Operations: In a TLB miss, this hardware executes a set of prescribed opera
follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH re
2. The virtual address causing the exception is written to the TEA register.
3. Either exception code H'040 for a load access, or H'060 for a store access, is written
   EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurr
   written to the save program counter (SPC). If the exception occurred in a delay slot,
   value indicating the address of the related delayed branch instruction is written to the
5. The contents of the status register (SR) at the time of the exception are written to the
   status register (SSR).
6. The MD bit in SR is set to 1 to place the privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The RB bit in SR is set to 1.
9. The RC field in the MMU control register (MMUCR) is incremented by 1 when all e
   indexed are valid. When some entries indexed are invalid, the smallest way number
   set in RC.
10. Execution branches to the address obtained by adding the value of the VBR contents
    H'00000400 to invoke the user-written TLB miss exception handler.

Software (TLB Miss Handler) Operations: The software searches the page tables in e
memory and allocates the required page table entry. Upon retrieving the required page ta
software must execute the following operations:

A TLB protection violation exception results when the virtual address and the address arr
selected TLB entry are compared and a valid entry is found to match, but the type of acce
permitted by the access rights specified in the PR field. TLB protection violation exceptio
processing includes both hardware and software operations.

Hardware Operations: In a TLB protection violation exception, this hardware executes
prescribed operations, as follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH reg
2. The virtual address causing the exception is written to the TEA register.
3. Either exception code H'0A0 for a load access, or H'0C0 for a store access, is written
   EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurre
   written into SPC (if the exception occurred in a delay slot, the PC value indicating the
   of the related delayed branch instruction is written into SPC).
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1 to place the privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The RB bit in SR is set to 1.
9. The way that generated the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR contents a
    H'00000100 to invoke the TLB protection violation exception handler.

Software (TLB Protection Violation Handler) Operations: Software resolves the TLB
protection violation and issues the RTE (return from exception handler) instruction to ter
the handler and return to the instruction stream. Issue the RTE instruction after issuing tw
instructions from the LDTLB instruction.

3. Either exception code H'040 for a load access, or H'060 for a store access, is written
   EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurr
   written to the SPC. If the exception occurred in a delay slot, the PC value indicating
   address of the delayed branch instruction is written to the SPC.
5. The contents of SR at the time of the exception are written into SSR.
6. The mode (MD) bit in SR is set to 1 to place the privileged mode.
7. The block (BL) bit in SR is set to 1 to mask any further exception requests.
8. The RB bit in SR is set to 1.
9. The way number causing the exception is written to RC in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR contents
    H'00000100, and the TLB protection violation exception handler starts.

Software (TLB Invalid Exception Handler) Operations: The software searches the pa
in external memory and assigns the required page table entry. Upon retrieving the requir
table entry, software must execute the following operations:

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

2. The virtual address causing the exception is written to the TEA register.
3. Exception code H'080 is written to the EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurred written to the SPC. If the exception occurred in a delay slot, the PC value indicating th address of the related delayed branch instruction is written to the SPC.
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1 to place the privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The RB bit in SR is set to 1.
9. The way that caused the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR contents a H'00000100 to invoke the user-written initial page write exception handler.

Software (Initial Page Write Handler) Operations: The software must execute the foll operations:

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

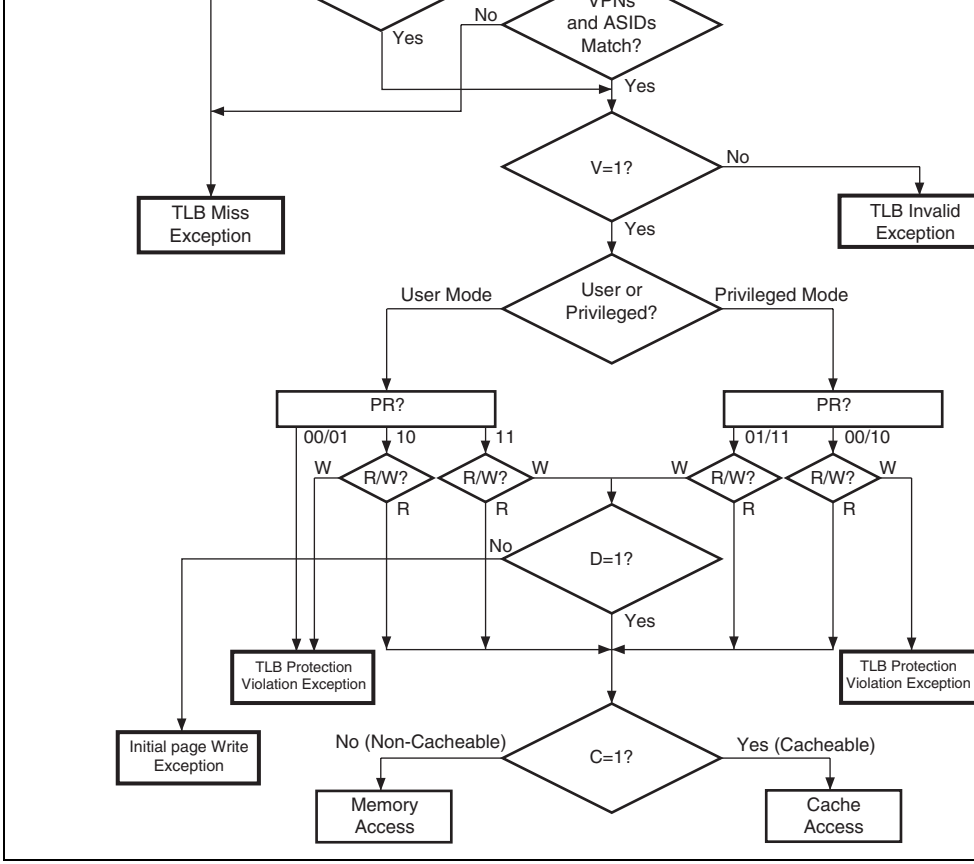


Figure 5.13 MMU Exception Generation Flowchart

5.6.1 Address Array

The address array is assigned to H'F2000000 to H'F2FFFFFF. To access an address arra
bit address field (for read/write operations) and 32-bit data field (for write operations) m
specified. The address field specifies information for selecting the entry to be accessed;
field specifies the VPN, V bit and ASID to be written to the address array (figure 5.14 (1

In the address field, specify the entry address for selecting the entry (bits 16 to 12), W fo
selecting the way (bits 9 to 8) and H'F2 to indicate address array access (bits 31 to 24). T
in MMUCR indicates whether an EX-OR is taken of the entry address and ASID.

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

1. Address array read

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

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

5.6.2 Data Array

The data array is assigned to H'F3000000 to H'F3FFFFFF. To access a data array, the 32
address field (for read/write operations), and 32-bit data field (for write operations) mus
specified. The address section specifies information for selecting the entry to be accesse
section specifies the longword data to be written to the data array (figure 5.14 (2)).

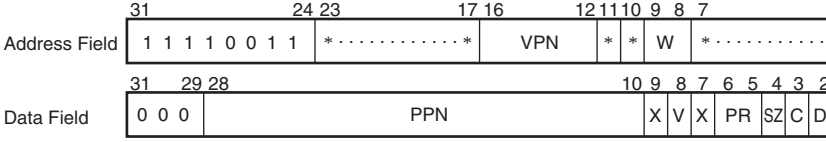
In the address section, specify the entry address for selecting the entry (bits 16 to 12), W
selecting the way (bits 9 to 8), and H'F3 to indicate data array access (bits 31 to 24). The
MMUCR indicates whether an EX-OR is taken of the entry address and ASID.

• Write Access

VPN: Virtual Page Number
V: Valid Bit
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)
ASID: Address Space Identifier
*: Don't Care Bit

(2) TLB Data Array Access

• Read/Write Access



PPN: Physical Page Number
PR: Protection Key Field
C: Cacheable Bit
SH: Share Status Bit
VPN: Virtual Page Number
X: 0 for Read, Don't Care Bit for Write
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)
V: Valid Bit
SZ: Page-Size Bit
D: Dirty bit
*: Don't Care Bit

Figure 5.14 Specifying Address and Data for Memory-Mapped TLB Acces

Reading the Data of a Specific Entry: This example reads the data section of a specifi entry. The bit order indicated in the data field in figure 5.14 (2) is read. R0 specifies the and the data section of a selected entry is read to R1.

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

5.7 Usage Note

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

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

- Write system: Write-back/write-through is selectable for spaces P0, P1, P3, and U0
- Replacement method: Least-recently used (LRU) algorithm

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

6.1.1 Cache Structure

The cache mixes instructions and data and uses a 4-way set associative system. It is com
four ways (banks), and each of which is divided into an address section and a data sectio

Each of the address and data sections is divided into 512 entries. The entry data is called
Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 8 kbytes (16 by
entries) in the cache as a whole (4 ways). The cache capacity is 32 kbytes as a whole. Fi
shows the cache structure.

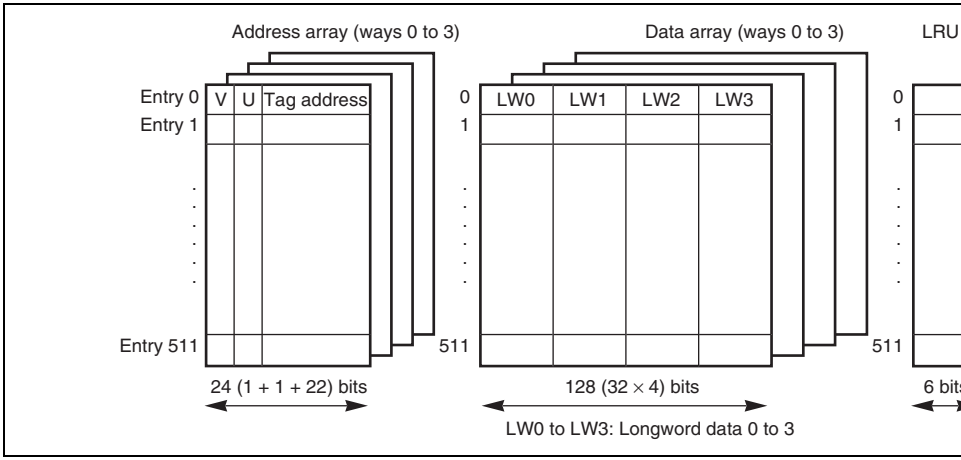


Figure 6.1 Cache Structure

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

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

Six LRU bits indicate the way to be replaced, when a cache miss occurs. Table 6.1 shows
relationship between the LRU bits and the way to be replaced when the cache locking me
is disabled. (For the relationship when the cache locking mechanism is enabled, refer to s
6.2.2, Cache Control Register 2 (CCR2).) If a bit pattern other than those listed in table 6
in the LRU bits by software, the cache will not function correctly. When modifying the L
by software, set one of the patterns listed in table 6.1.

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

Table 6.1 LRU and Way Replacement (when Cache Locking Mechanism is Disab

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

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

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

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

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

Table 6.2 shows the relationship between the settings of bits and the way that is to be repl when the cache is missed by a prefetch instruction.

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

In cache lock mode, bits W3LOCK and W2LOCK restrict the way that is to be replaced, instructions other than the prefetch instruction are issued. Table 6.3 shows the relationshi between the settings of bits in CCR2 and the way that is to be replaced when the cache is by instructions other than the prefetch instruction.

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

				Value.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
9	W3LOAD	0	R/W	Way 3 Load (W3LOAD)
8	W3LOCK	0	R/W	Way 3 Lock (W3LOCK) When the cache is missed by a prefetch in while in cache lock mode and when bits W and W3LOCK in CCR2 are set to 1, the da always loaded into way 3. Under any othe condition, the prefetched data is loaded in to which LRU points.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
1	W2LOAD	0	R/W	Way 2 Load (W2LOAD)
0	W2LOCK	0	R/W	Way 2 Lock (W2LOCK) When the cache is missed by a prefetch in while in cache lock mode and when bits W and W2LOCK in CCR2 are set to 1, the da always loaded into way 2. Under any othe condition, the prefetched data is loaded in to which LRU points.

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

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

Table 6.3 Way Replacement when Instructions other than the PREF Instruction I Cache

DSP Bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replaced
0	*	*	*	*	Determined by LRU (tal
1	*	0	*	0	Determined by LRU (tal
1	*	0	*	1	Determined by LRU (tal
1	*	1	*	0	Determined by LRU (tal
1	*	1	*	1	Determined by LRU (tal

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

Table 6.4 LRU and Way Replacement (when W2LOCK = 1 and W3LOCK =0)

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

010100, 010110, 011110, 011111	
100000, 100001, 101001, 101011, 110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

6.2.3 Cache Control Register 3 (CCR3)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
23 to 16	CSIZE7 to CSIZE0	H'01	R/W	Cache Size Specify the cache size as shown below. 0000 0001: 16-kbyte cache 0000 0010: 32-kbyte cache Settings other than above are prohibited.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.

address of that entry is read. The virtual address (bits 31 to 10) of the access to memory a
physical address (tag address) read from the address array are compared. The address com
uses all four ways. When the comparison shows a match and the selected entry is valid (V
cache hit occurs. When the comparison does not show a match or the selected entry is not
= 0), a cache miss occurs. Figure 6.2 shows a hit on way 1.

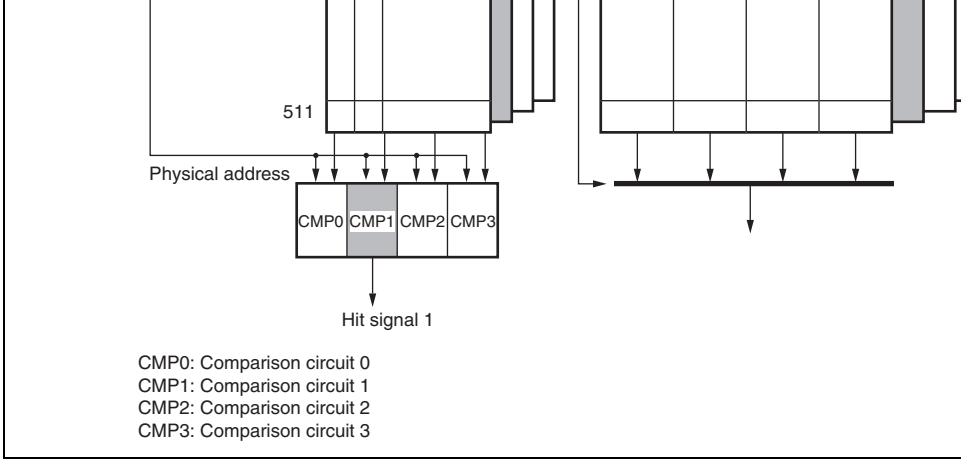


Figure 6.2 Cache Search Scheme

6.3.2 Read Access

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

Read Miss: An external bus cycle starts and the entry is updated. The way to be replace
in table 6.3. Entries are updated in 16-byte units. When the desired instruction or data th
the miss is loaded from external memory to the cache, the instruction or data is transferr
CPU in parallel with being loaded to the cache. When it is loaded to the cache, the U bit
to 0 and the V bit is set to 1 to indicate that the hit way is the most recently hit way. Wh
bit for the entry which is to be replaced by entry updating in write-back mode is 1, the c
update cycle starts after the entry is transferred to the write-back buffer. After the cache
its update cycle, the write-back buffer writes the entry back to the memory. Transfer is i
units.

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

Write Miss: In write-back mode, an external write cycle starts when a write miss occurs,
entry is updated. The way to be replaced is shown in table 6.3. When the U bit of the entr
is to be replaced by entry updating is 1, the cache-update cycle starts after the entry has b
transferred to the write-back buffer. Data is written to the cache and the U bit and the V b
to 1. The LRU is updated to indicate that the replaced way is the most recently updated w
the cache has completed its update cycle, the write-back buffer writes the entry back to th
memory. Transfer is in 16-byte units. In write-through mode, no write to cache occurs in
miss; the write is only to the external memory.

6.3.5 Write-Back Buffer

When the U bit of the entry to be replaced in write-back mode is 1, the entry must be writ
to the external memory. To increase performance, the entry to be replaced is first transfer
write-back buffer and fetching of new entries to the cache takes priority over writing back
external memory. After the fetching of new entries to the cache completes, the write-back
writes the entry back to the external memory. During the write-back cycles, the cache can
accessed. The write-back buffer can hold one line of cache data (16 bytes) and its physica
address. Figure 6.3 shows the configuration of the write-back buffer.

the memory-mapped cache to make the data invalid and written back, as required. Mem
shared by this LSI's CPU and DMAC should also be handled in this way.

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

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

In the data field, specify the tag address, LRU bits, U bit, and V bit. Figures 6.4 and 6.5 s
address and data formats. The following three operations are available in the address arra

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

Address-Array Write (non-Associative Operation): Write the tag address, LRU bits, U
V bit, specified by the data field of the write instruction, to the entry that corresponds to t
address and way as specified by the address field of the write instruction. Ensure that the
associative bit (A bit) in the address field is set to 0. When writing to a cache line for whi
bit = 1 and the V bit =1, write the contents of the cache line back to memory, then write t
address, LRU bits, U bit, and V bit specified by the data field of the write instruction. Wh
written to the V bit, 0 must also be written to the U bit for that entry.

Address-Array Write (Associative Operation): When writing with the associative bit (
the address = 1, the addresses in the four ways for the entry specified by the address field
write instruction are compared with the tag address that is specified by the data field of th
instruction. If the MMU is enabled in this case, a logical address specified by data is trans
into a physical address via the TLB before comparison. Write the U bit and the V bit spec
the data field of the write instruction to the entry of the way that has a hit. However, the t

specifies the longword data to be written to the data array.

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

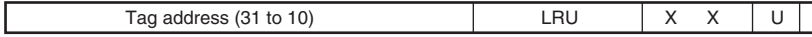
Since access size of the data array is fixed at longword, bits 1 and 0 of the address field set to 00.

Figures 6.4 and 6.5 show the address and data formats.

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

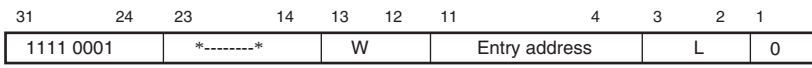
Data-Array Read: Read the data specified by L of the address filed, from the entry that corresponds to the entry address and the way that is specified by the address filed.

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

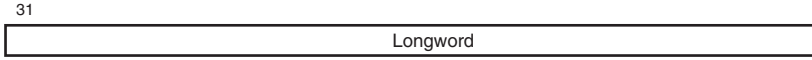


(2) Data array access (both read and write accesses)

(a) Address specification



(b) Data specification



*: Don't care bit
X: 0 for read, don't care for write

Figure 6.4 Specifying Address and Data for Memory-Mapped Cache Acces (16 kbytes mode)

0	0	0	Tag address (28 to 10)	LRU	X X	U

(2) Data array access (both read and write accesses)

(a) Address specification

31 24	23 15	14 13	12 4	3 2	1
1111 0001	*--------*	W	Entry address	L	0

(b) Data specification

31
Longword

*: Don't care bit
X: 0 for read, don't care for write

Figure 6.5 Specifying Address and Data for Memory-Mapped Cache Acce (32 kbytes mode)

```
; R1=H'F0000088; address array access, entry=B'00001000, W=1
;
MOV.L R0,@R1
```

Reading the Data of a Specific Entry: To read the data field of a specific entry is enable
memory-mapped cache access. The longword indicated in the data field of the data array
6.4 or 6.5 is read into the register. In the example shown below, R0 specifies the address
shows what is read.

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

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

In the logical address space, this memory is located in the addresses shown in table 7
addresses are included in space P2 (when SR.MD = 1) or Uxy (when SR.MD = 0 an
= 1) according to the CPU operating mode.

Table 7.1 X/Y Memory Logical Addresses

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

On the other hand, this memory is located in a part of area 1 in the physical address spac
this memory is accessed from the physical address space, addresses in which the upper t
are 0 in addresses shown in table 7.1 are used. In the X-bus and Y-bus address spaces, a
which the upper 16 bits are ignored in addresses of X memory and Y memory shown in
are used.

- Ports

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

the I bus after the logical addresses are converted to be the physical addresses using the M
long as a conflict on the page does not occur, access via the L bus is performed in one cyc
Several cycles are necessary for accessing via the I bus. According to the CPU operating
access from the CPU is as follows:

Privileged mode and privileged DSP mode (SR. MD = 1): The X/Y memory can be ac
the CPU directly from space P2. The MMU can be used to map the logical addresses in sp
and P3 to this memory.

User DSP mode (SR.MD = 0 and SR.DSP = 1): The X/Y memory can be accessed by th
directly from space Uxy. The MMU can be used to map the logical addresses in space U0
memory.

User mode (SR.MD = 0 and SR.DSP = 0): The MMU can be used to map the logical ad
in space U0 to this memory.

7.2.2 Access from DSP

Methods for accessing from the DSP differ according to instructions.

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

In the case of a single data transfer instruction, methods for accessing from the DSP are d
via the L bus from the logical addresses, and via the I bus after the logical addresses are c
to be the physical addresses using the MMU. As long as a conflict on the page does not o
access via the L bus is performed in one cycle. Several cycles are necessary for accessing
bus. According to the CPU operating mode, access from the CPU is as follows:

physical address bus. Addresses in which the upper three bits are 0 in addresses shown i
must be used.

7.3 Usage Notes

7.3.1 Page Conflict

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

7.3.2 Bus Conflict

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

7.3.3 MMU and Cache Settings

When the X/Y memory is accessed via the I bus using the cache from the CPU and DSP
operation cannot be guaranteed. If the X/Y memory is accessed while the cache is enabl
(CCR1.CE = 1), it is advisable to access the X/Y memory via the L bus from space P2 o
the X/Y memory is accessed from space P0, P3, or U0, it is advisable to access the X/Y

CCR1.CE	MMUCR.AT	P0, U0	P1	P2, Uxy	P3
0	0	B	B	A	B
0	1	B	B	A	B
1	0	X	X	A	X
1	1	C	X	A	C

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

7.3.4 Sleep Mode

In sleep mode, I bus master modules such as the DMAC and E-DMAC cannot access the memory.

7.3.5 Address Error

If writing that may causes an address error is performed on the X/Y memory, the contents X/Y memory are not guaranteed.

- 16 levels of interrupt priority can be set

  By setting the interrupt-priority registers, the priorities of on-chip peripheral module interrupts can be selected from 16 levels for individual request sources.
- NMI noise canceler function

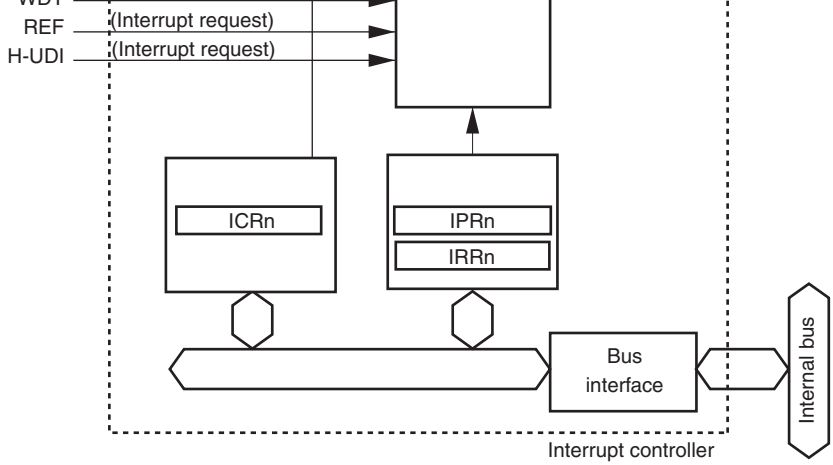
  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interru exception service routine, the pin state can be checked, enabling it to be used as a no canceler.
- IRQ interrupts can be set

  Detection of low level, rising edge, falling edge, or high level
- Interrupt request signal can be externally output ($\overline{\text{IRQOUT}}$ pin)

  The bus mastership can be requested by notifying the external bus master that the ex interrupt and on-chip peripheral module interrupt requests have been generated.

8.1.1 Block Diagram

Figure 8.1 shows a block diagram of the INTC.



[Legend]

DMAC:	Direct memory access controller
SCIF0/1:	Serial communication interface with FIFO
E-DMAC:	Direct memory access controller for ethernet controller
SIOF0/1:	Serial I/O with FIFO
TMU:	Timer unit
RTC:	Realtime clock unit
WDT:	Watchdog timer
REF:	Refresh request in bus state controller
H-UDI:	User-debugging interface
ICRn:	Interrupt control registers 0, 1
IPRn:	Interrupt priority registers A to I
IRRn:	Interrupt request registers 0 to 5, 7 and 8
SR:	Status register

Figure 8.1 Block Diagram of INTC

	IRL3 to IRL0*[1]		
Bus mastership request output pin*[2]	$\overline{\text{IRQOUT}}$	Output	Notifies that an interrupt r has generated

Notes: 1. The $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins and the IRQ3 to IRQ0 cannot be used simultaneously these pins are multiplexed with the IRQ3 to IRQ0 pins.

2. When the NMI or H-UDI interrupt requests are generated and the response ti CPU is short, this pin may not be asserted.

8.3 Interrupt Sources

There are four types of interrupt sources: NMI, IRQ, IRL, and on-chip peripheral modul interrupt has a priority level (0 to 16), with 1 the lowest and 16 the highest. Priority leve an interrupt, so the interrupt request is ignored.

8.3.1 NMI Interrupt

The NMI interrupt has the highest priority level of 16. When the BLMSK bit in the inter control register 1 (ICR1) is set to 1 or the BL bit in the status register (SR) is 0 and the N ICR1 is 0, NMI interrupts are accepted. NMI interrupts are edge-detected. In sleep or sta mode, the interrupt is accepted regardless of the BL setting. The NMI edge select bit (N the interrupt control register 0 (ICR0) is used to select either rising or falling edge detec

When using edge-input detection for NMI interrupts, a pulse width of at least two Pϕ cy (peripheral clock) is necessary. NMI interrupt exception handling does not affect the int mask level bits (I3 to I0) in the status register (SR). When the BL bit is 1 and the BLMS ICR1 is set to 1, only the NMI interrupt is accepted.

It is possible to wake the chip up from sleep mode or standby mode with the NMI interr

Edge input interrupt detection requires input of a pulse width of more than two cycles on basis.

When using level-sensing for IRQ interrupts, the pin levels must be retained until the CPU samples the pins. Therefore, the interrupt source must be cleared by the interrupt handler.

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by IRQ interr handling. IRQ interrupts can be used for recovering from standby when the corresponding interrupt level is higher than that of bits I3 to I0 in SR. (However, only when the RTC is u recovering from standby by using the clock for the RTC is enabled.)

8.3.3 IRL Interrupts

IRL interrupts are input by the $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins as level. The priority level is the higher that is indicated by the $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins. When the values of the $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins are (B′0000), the highest level interrupt request (interrupt priority level 15) is indicated. Whe values of the pins are 15 (B′1111), no interrupt is requested (interrupt priority level 0). Fi shows an example of connection for an IRL interrupt. Table 8.3 lists the $\overline{\text{IRL}}$ pins and int level.

IRL interrupts are included with a noise canceler function and detected when the sampled each peripheral module clock keep the same value for 2 cycles. This prevents sampling er in $\overline{\text{IRL}}$ pin changing. In standby mode, a noise canceler is handled by the RTC clock (32 because the peripheral module clocks are halted. Therefore, when the RTC is not used, re from standby by IRL interrupts cannot be executed in standby mode.

The priority level of IRL interrupts should be kept until an interrupt is accepted and its ha started. However, changing to higher level is enabled.

Figure 8.2 Example of IRL Interrupt Connection

8.3.4 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following 14 modules:

- Direct memory access controller (DMAC)
- Serial communication interface with FIFO (SCIF0 and SCIF1)
- Direct memory access controller for ethernet controller (E-DMAC) (includes an EtherC interrupt)
- Serial I/O with FIFO (SIOF0 and SIOF1)
- Timer unit (TMU0 to TMU2)
- Realtime clock (RTC)
- Watchdog timer (WDT)
- Bus state controller (BSC)
- User-debugging interface (H-UDI)

Not every interrupt source is assigned a different interrupt vector. Sources are reflected i
interrupt event registers (INTEVT and INTEVT2). It is easy to identify sources by using
of INTEVT or INTEVT2 as a branch offset.

A priority level (from 0 to 15) can be set for each module except the H-UDI by writing t
interrupt priority registers A, B, and E to I (IPRA, IPRB, and IPRE to IPRI). The priorit
the H-UDI interrupt is 15 (fixed).

order of interrupt priority.

Each interrupt source is assigned a unique code by INTEVT or INTEVT2. The start addr
interrupt service routine is common for each interrupt source. This is why, for instance, t
of INTEVT or INTEVT2 is used as an offset at the start of the interrupt service routine a
branched to in order to identify the interrupt source.

IRQ interrupt and on-chip peripheral module interrupt priorities can be set freely betwee
for each module by setting interrupt priority registers A to I (IPRA to IPRI). A reset assig
priority level 0 to IRQ and on-chip peripheral module interrupts.

If the same priority level is assigned to two or more interrupt sources and interrupts from
sources occur simultaneously, their priority order is the default priority order indicated at
in tables 8.2 and 8.3.

Table 8.2 Interrupt Exception Handling Sources and Priority (IRQ Mode)

Interrupt Source		Interrupt Code*[1]	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit
NMI		H′1C0*[2]	16	—	—
H-UDI		H′5E0*[2]	15	—	—
IRQ	IRQ0	H′600*[3]	0 to 15 (0)	IPRC (3 to 0)	—
	IRQ1	H′620*[3]	0 to 15 (0)	IPRC (7 to 4)	—
	IRQ2	H′640*[3]	0 to 15 (0)	IPRC (11 to 8)	—
	IRQ3	H′660*[3]	0 to 15 (0)	IPRC (15 to 12)	—
	IRQ4	H′680*[3]	0 to 15 (0)	IPRD (3 to 0)	—
	IRQ5	H′6A0*[3]	0 to 15 (0)	IPRD (7 to 4)	—

	TXI0	H′8E0*[3]			Low
SCIF1	ERI1	H′900*[3]	0 to 15 (0)	IPRE (7 to 4)	High
	RXI1	H′920*[3]			↕
	BRI1	H′940*[3]			
	TXI1	H′960*[3]			Low
DMAC (2)	DEI4	H′B80*[3]	0 to 15 (0)	IPRF (11 to 8)	High
	DEI5	H′BA0*[3]			Low
E-DMAC	EINT0	H′C00*[3]	0 to 15 (0)	IPRG (15 to 12)	—
	EINT1	H′C20*[3]	0 to 15 (0)	IPRG (11 to 8)	—
	EINT2	H′C40*[3]	0 to 15 (0)	IPRG (7 to 4)	—
SIOF0	ERI0	H′E00*[3]	0 to 15 (0)	IPRH (3 to 0)	High
	TXI0	H′E20*[3]			↕
	RXI0	H′E40*[3]			
	CCI0	H′E60*[3]			Low
SIOF1	ERI1	H′E80*[3]	0 to 15 (0)	IPRI (7 to 4)	High
	TXI1	H′EA0*[3]			↕
	RXI1	H′EC0*[3]			
	CCI1	H′EE0*[3]			Low
TMU0	TUNI0	H′400*[2]	0 to 15 (0)	IPRA (15 to 12)	—
TMU1	TUNI1	H′420*[2]	0 to 15 (0)	IPRA (11 to 8)	—
TMU2	TUNI2	H′440*[2]	0 to 15 (0)	IPRA (7 to 4)	—

3. The code that indicates the interrupt level (H′200 to H′3C0) is set in INTEVT. F
on correspondence between the interrupt level and INTEVT, see table 8.4.

	IRL[3:0] = B′0011	H′260*[3]	12	—	—
	IRL[3:0] = B′0100	H′280*[3]	11	—	—
	IRL[3:0] = B′0101	H′2A0*[3]	10	—	—
	IRL[3:0] = B′0110	H′2C0*[3]	9	—	—
	IRL[3:0] = B′0111	H′2E0*[3]	8	—	—
	IRL[3:0] = B′1000	H′300*[3]	7	—	—
	IRL[3:0] = B′1001	H′320*[3]	6	—	—
	IRL[3:0] = B′1010	H′340*[3]	5	—	—
	IRL[3:0] = B′1011	H′360*[3]	4	—	—
	IRL[3:0] = B′1100	H′380*[3]	3	—	—
	IRL[3:0] = B′1101	H′3A0*[3]	2	—	—
	IRL[3:0] = B′1110	H′3C0*[3]	1	—	—
IRQ	IRQ4	H′680*[3]	0 to 15 (0)	IPRD (3 to 0)	—
	IRQ5	H′6A0*[3]	0 to 15 (0)	IPRD (7 to 4)	—

	TXI0	H′8E0*[3]			Low
SCIF1	ERI1	H′900*[3]	0 to 15 (0)	IPRE (7 to 4)	High
	RXI1	H′920*[3]			↕
	BRI1	H′940*[3]			
	TXI1	H′960*[3]			Low
DMAC (2)	DEI4	H′B80*[3]	0 to 15 (0)	IPRF (11 to 8)	High
	DEI5	H′BA0*[3]			Low
E-DMAC	EINT0	H′C00*[3]	0 to 15 (0)	IPRG (15 to 12)	—
	EINT1	H′C20*[3]	0 to 15 (0)	IPRG (11 to 8)	—
	EINT2	H′C40*[3]	0 to 15 (0)	IPRG (7 to 4)	—
SIOF0	ERI0	H′E00*[3]	0 to 15 (0)	IPRH (3 to 0)	High
	TXI0	H′E20*[3]			↕
	RXI0	H′E40*[3]			
	CCI0	H′E60*[3]			Low
SIOF1	ERI1	H′E80*[3]	0 to 15 (0)	IPRI (7 to 4)	High
	TXI1	H′EA0*[3]			↕
	RXI1	H′EC0*[3]			
	CCI1	H′EE0*[3]			Low
TMU0	TUNI0	H′400*[2]	0 to 15 (0)	IPRA (15 to 12)	—
TMU1	TUNI1	H′420*[2]	0 to 15 (0)	IPRA (11 to 8)	—
TMU2	TUNI2	H′440*[2]	0 to 15 (0)	IPRA (7 to 4)	—

3. The code that indicates the interrupt level (H′200 to H′3C0) is set in INTEVT. on correspondence between the interrupt level and INTEVT, see table 8.4.

Table 8.4 Interrupt Level and INTEVT Code

Interrupt Level	INTEVT Code
15	H′200
14	H′220
13	H′240
12	H′260
11	H′280
10	H′2A0
9	H′2C0
8	H′2E0
7	H′300
6	H′320
5	H′340
4	H′360
3	H′380
2	H′3A0
1	H′3C0

- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register I (IPRI)
- Interrupt request register 0 (IRR0)
- Interrupt request register 1 (IRR1)
- Interrupt request register 2 (IRR2)
- Interrupt request register 3 (IRR3)
- Interrupt request register 4 (IRR4)
- Interrupt request register 5 (IRR5)
- Interrupt request register 7 (IRR7)
- Interrupt request register 8 (IRR8)

8.4.1 Interrupt Priority Registers A to I (IPRA to IPRI)

IPRA to IPRI are 16-bit readable/writable registers in which priority levels from 0 to 15 a
on-chip peripheral module and IRQ interrupts. These registers are initialized to H'0000 by
power-on reset or manual reset, but are not initialized in standby mode.

7	—	0	R/W
6	—	0	R/W
5	—	0	R/W
4	—	0	R/W
3	—	0	R/W
2	—	0	R/W
1	—	0	R/W
0	—	0	R/W

Table 8.5 Interrupt Sources and IPRA to IPRI

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to
IPRA	TMU0	TMU1	TMU2	RTC
IPRB	WDT	REF	Reserved*	Reserve
IPRC	IRQ3	IRQ2	IRQ1	IRQ0
IPRD	Reserved*	Reserved*	IRQ5	IRQ4
IPRE	DMAC (1)	SCIF0	SCIF1	Reserve
IPRF	Reserved*	DMAC (2)	Reserved*	Reserve
IPRG	E-DMAC (1)	E-DMAC (2)	E-DMAC (3)	Reserve
IPRH	Reserved*	Reserved*	Reserved*	SIOF0
IPRI	Reserved*	Reserved*	SIOF1	Reserve

Note: * Always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	0/1*	R	NMI Input Level Sets the level of the signal input at the N This bit can be read from to determine t pin level. This bit cannot be modified. 0 : NMI input level is low 1 : NMI input level is high
14	—	0	R	Reserved
13	—	0	R	These bits are always read as 0. The w should always be 0.
12	—	0	R	
11	—	0	R	
10	—	0	R	
9	—	0	R	
8	NMIE	0	R/W	NMI Edge Select Selects whether the falling or rising edg interrupt request signal at the NMI pin is detected. 0 : Interrupt request is detected on fallin of NMI pin input 1 : Interrupt request is detected on risin NMI pin input

Note: When NMI input is high, 0 when NMI input is low.

8.4.3 Interrupt Control Register 1 (ICR1)

ICR1 is a 16-bit register that specifies the detection mode for external interrupt input pin IRQ5 individually: rising edge, falling edge, high level, or low level. This register is init H′4000 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	MAI	0	R/W	All Interrupt Mask When this bit is set to 1, all interrupt re are masked while low level is input to t pin. In standby mode, an NMI interrupt masked. 0: When the NMI pin is low, all interrup are not masked 1: When the NMI pin is low, all interrup are masked
14	IRQLVL	1	R/W	Interrupt Request Level Detection Enables or disables the use of pins IR IRQ0 as four independent interrupt pin IRQ4 and IRQ5 pins are not affected. 0 : Use of pins IRQ3 to IRQ0 as four independent interrupt pins enabled 1 : Use of pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ as encode interrupt pins

				This bit is always read as 0. The write v should always be 0.
11	IRQ51S	0	R/W	IRQn Sense Select
10	IRQ50S	0	R/W	These bits select whether interrupt requ signals corresponding to pins IRQ5 to I detected by a rising edge, falling edge, level, or low level.
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	
5	IRQ21S	0	R/W	
4	IRQ20S	0	R/W	
3	IRQ11S	0	R/W	
2	IRQ10S	0	R/W	
1	IRQ01S	0	R/W	
0	IRQ00S	0	R/W	

Bit 2n+1	Bit 2n	
IRQn1S	IRQn0S	
0	0	Interrupt request detected on fallin edge of IRQn inp
0	1	Interrupt request detected on risin of IRQn input
1	0	Interrupt request detected on low l IRQn input
1	1	Interrupt request detected on high of IRQn input

Legend n=0 to 5

5	IRQ5R	0	R/W	IRQn Interrupt Request
4	IRQ4R	0	R/W	Indicates whether there is interrupt req
3	IRQ3R	0	R/W	to the IRQn pin. When edge-detection
2	IRQ2R	0	R/W	set for IRQn, an interrupt request is cle
1	IRQ1R	0	R/W	writing 0 to the IRQnR bit after reading
0	IRQ0R	0	R/W	1.
				When level-detection mode is set for I
				interrupt request is set/cleared by only
				to the IRQn pin.
				IRQnR
				0: No interrupt request input to IRQn p
				1: Interrupt request input to IRQn pin
				Legend: n = 0 to 5

8.4.5 Interrupt Request Register 1 (IRR1)

IRR1 is an 8-bit register that indicates whether interrupt requests from the DMAC and t
are generated. This register is initialized to H'00 by a power-on reset or manual reset, bu
initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TXI0R	0	R	TXI0 Interrupt Request
				Indicates whether the TXI0 (SCIF0) int request is generated.
				0: TXI0 interrupt request is not genera
				1: TXI0 interrupt request is generated

				1: RXI0 interrupt request is generated
4	ERI0R	0	R	ERI0 Interrupt Request Indicates whether the ERI0 (SCIF0) inte request is generated. 0: ERI0 interrupt request is not generate 1: ERI0 interrupt request is generated
3	DEI3R	0	R	DEI3 Interrupt Request Indicates whether the DEI3 (DMAC) inte request is generated. 0: DEI3 interrupt request is not generate 1: DEI3 interrupt request is generated
2	DEI2R	0	R	DEI2 Interrupt Request Indicates whether the DEI2 (DMAC) inte request is generated. 0: DEI2 interrupt request is not generate 1: DEI2 interrupt request is generated
1	DEI1R	0	R	DEI1 Interrupt Request Indicates whether the DEI1 (DMAC) inte request is generated. 0: DEI1 interrupt request is not generate 1: DEI1 interrupt request is generated
0	DEI0R	0	R	DEI0 Interrupt Request Indicates whether the DEI0 (DMAC) inte request is generated. 0: DEI0 interrupt request is not generate 1: DEI0 interrupt request is generated

				Indicates whether the TXI1 (SCIF1) int request is generated. 0: TXI1 interrupt request is not genera 1: TXI1 interrupt request is generated
2	BRI1R	0	R	BRI1 Interrupt Request Indicates whether the BRI1 (SCIF1) in request is generated. 0: BRI1 interrupt request is not genera 1: BRI1 interrupt request is generated
1	RXI1R	0	R	RXI1 Interrupt Request Indicates whether the RXI1 (SCIF1) in request is generated. 0: RXI1 interrupt request is not genera 1: RXI1 interrupt request is generated
0	ERI1R	0	R	ERI1 Interrupt Request Indicates whether the ERI1 (SCIF1) in request is generated. 0: ERI1 interrupt request is not genera 1: ERI1 interrupt request is generated

2	CUIR	0	R	CUI Interrupt Request Indicates whether the CUI (RTC) interru request is generated. 0: CUI interrupt request is not generate 1: CUI interrupt request is generated
1	PRIR	0	R	PRI Interrupt Request Indicates whether the PRI (RTC) interru request is generated. 0: PRI interrupt request is not generate 1: PRI interrupt request is generated
0	ATIR	0	R	ATI Interrupt Request Indicates whether the ATI (RTC) interru request is generated. 0: ATI interrupt request is not generate 1: ATI interrupt request is generated

6	TUNI2R	0	R	TUNI2 Interrupt Request Indicates whether the TUNI2 (TMU2) i request is generated. 0: TUNI2 interrupt request is not gener 1: TUNI2 interrupt request is generate
5	TUNI1R	0	R	TUNI1Interrupt Request Indicates whether the TUNI1 (TMU1) i request is generated. 0: TUNI1 interrupt request is not gener 1: TUNI1 interrupt request is generate
4	TUNI0R	0	R	TUNI0 Interrupt Request Indicates whether the TUNI0 (TMU0) i request is generated. 0: TUNI0 interrupt request is not gener 1: TUNI0 interrupt request is generate
3	ITIR	0	R	ITI Interrupt Request Indicates whether the ITI (WDT) interr request is generated. 0: ITI interrupt request is not generate 1: ITI interrupt request is generated
2	—	0	R	Reserved
1	—	0	R	These bits are always read as 0. The v should always be 0.

IRR5 is an 8-bit register that indicates whether interrupt requests from the DMAC and E
are generated. This register is initialized to H'00 by a power-on reset or manual reset, but
initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The w should always be 0.
5	DEI5R	0	R	DEI5 Interrupt Request Indicates whether the DEI5 (DMAC) inte request is generated. 0: DEI5 interrupt request is not generate 1: DEI5 interrupt request is generated
4	DEI4R	0	R	DEI4 Interrupt Request Indicates whether the DEI4 (DMAC) inte request is generated. 0: DEI4 interrupt request is not generate 1: DEI4 interrupt request is generated
3	—	0	R	Reserved This bit is always read as 0. The write v should always be 0.

				1: EINT1 interrupt request is generated
0	EINT0R	0	R	EINT0 Interrupt Request Indicates whether the EINT0 (E-DMAC request is generated. 0: EINT0 interrupt request is not gener 1: EINT0 interrupt request is generated

8.4.10 Interrupt Request Register 7 (IRR7)

IRR7 is an 8-bit register that indicates whether an interrupt request from the SIOF0 is ge This register is initialized to H'00 by a power-on reset or manual reset, but is not initializ standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CCI0R	0	R	CCI0 Interrupt Request Indicates whether the CCI0 (SIOF0) in request is generated. 0: CCI0 interrupt request is not genera 1: CCI0 interrupt request is generated
6	RXI0R	0	R	RXI0 Interrupt Request Indicates whether the RXI0 (SIOF0) in request is generated. 0: RXI0 interrupt request is not genera 1: RXI0 interrupt request is generated

				1: ERI0 interrupt request is generated
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The w should always be 0.

8.4.11 Interrupt Request Register 8 (IRR8)

IRR8 is an 8-bit register that indicates whether interrupt requests from the SIOF1 are gen This register is initialized to H'00 by a power-on reset or manual reset, but is not initialize standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CCI1R	0	R	CCI1 Interrupt Request Indicates whether the CCI1 (SIOF1) inte request is generated. 0: CCI1 interrupt request is not generat 1: CCI1 interrupt request is generated
6	RXI1R	0	R	RXI1 Interrupt Request Indicates whether the RXI1 (SIOF1) inte request is generated. 0: RXI1 interrupt request is not generate 1: RXI1 interrupt request is generated

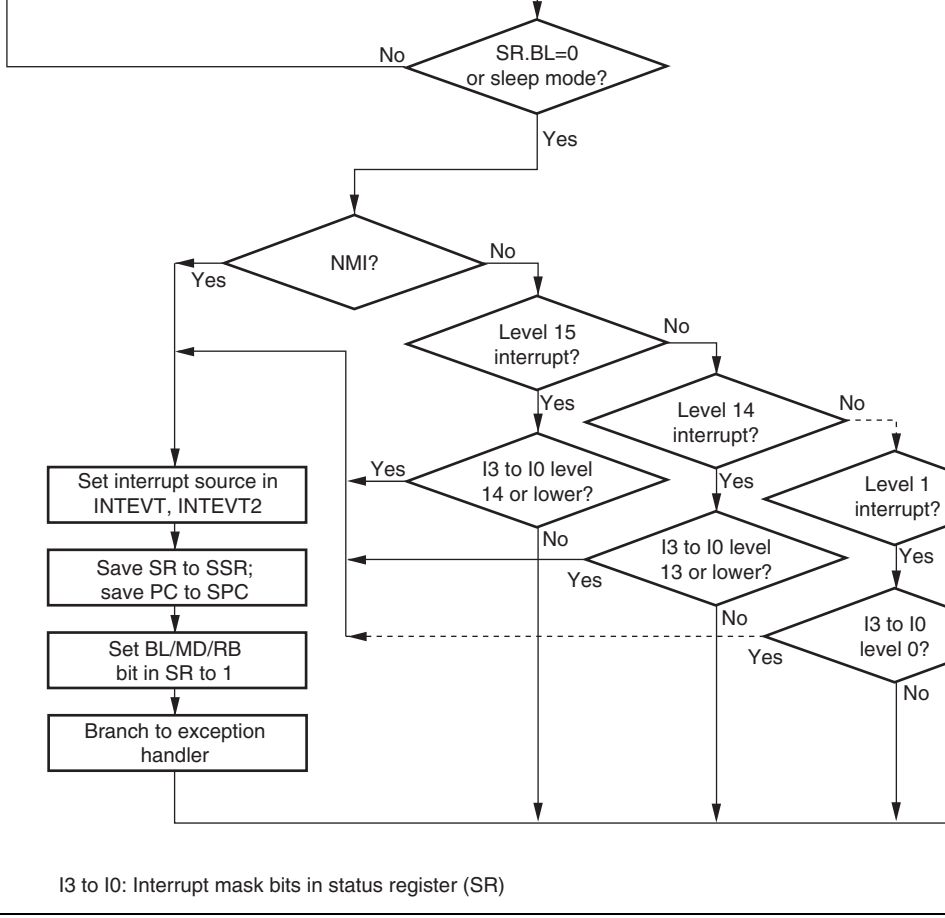
				1: ERI1 interrupt request is generated
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The v should always be 0.

priority interrupts are held pending. If two of these interrupts have the same priority le multiple interrupts occur within a single module, the interrupt with the highest priority selected, according to tables 8.2 and 8.3, Interrupt Exception Handling Sources and P

3. The priority level of the interrupt selected by the interrupt controller is compared with interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the request prio is higher than the level in bits I3 to I0, the interrupt controller accepts the interrupt an an interrupt request signal to the CPU.
4. Detection timing: The INTC operates, and notifies the CPU of interrupt requests, in synchronization with the peripheral clock (Pϕ). The CPU receives an interrupt at a bre instructions.
5. The interrupt source code is set in the interrupt event registers (INTEVT and INTEVT
6. The status register (SR) and program counter (PC) are saved to SSR and SPC, respect
7. The block bit (BL), mode bit (MD), and register bank bit (RB) in SR are set to 1.
8. The CPU jumps to the start address of the interrupt handler (the sum of the value set i vector base register (VBR) and H'00000600). This jump is not a delayed branch. The handler may branch with INTEVT or INTEVT2 value as its offset in order to identify interrupt source. This enables it to branch to the handling routine for the individual in source.

Notes: 1. The interrupt mask bits (I3 to I0) in the status register (SR) are not changed by acceptance of an interrupt in this LSI.

2. The interrupt source flag should be cleared in the interrupt handler. To ensure interrupt request that should have been cleared is not inadvertently accepted a the interrupt source flag after it has been cleared, and then clear the BL bit or an RTE instruction.



I3 to I0: Interrupt mask bits in status register (SR)

Figure 8.3 Interrupt Operation Flowchart

5. Handle the interrupt.
6. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the one handled can be accepted after clearing the BL bit in step 4. See figure 8.3 on a sample int operation flowchart.

The UBC has the following features:

1. The following break comparison conditions can be set.

   Number of break channels: two channels (channels A and B)

   User break can be requested as either the independent or sequential condition on cha
   and B (sequential break setting: channel A and then channel B match with break con
   but not in the same bus cycle).

- Address

  Compares 40 bits configured of the ASID and addresses 32 bits: the ASID can be se
  either all-bit comparison or all-bit mask. Comparison bits are maskable in 1-bit units
  mask addresses at lower 12 bits (4-k page), lower 10 bits (1-k page), or any size of p

  One of the four address buses (logic address bus (LAB), internal address bus (IAB),

  X-memory address bus (XAB), and Y-memory address bus (YAB)) can be selected.

- Data

  Only on channel B, 32-bit maskable.

  One of the four data buses (L-bus data (LDB), I-bus data (IDB), X-memory data bus
  and Y-memory data bus (YDB)) can be selected.

- Bus cycle

  Instruction fetch or data access

- Read/write

- Operand size

  Byte, word, and longword

2. A user-designed user-break condition exception processing routine can be run.

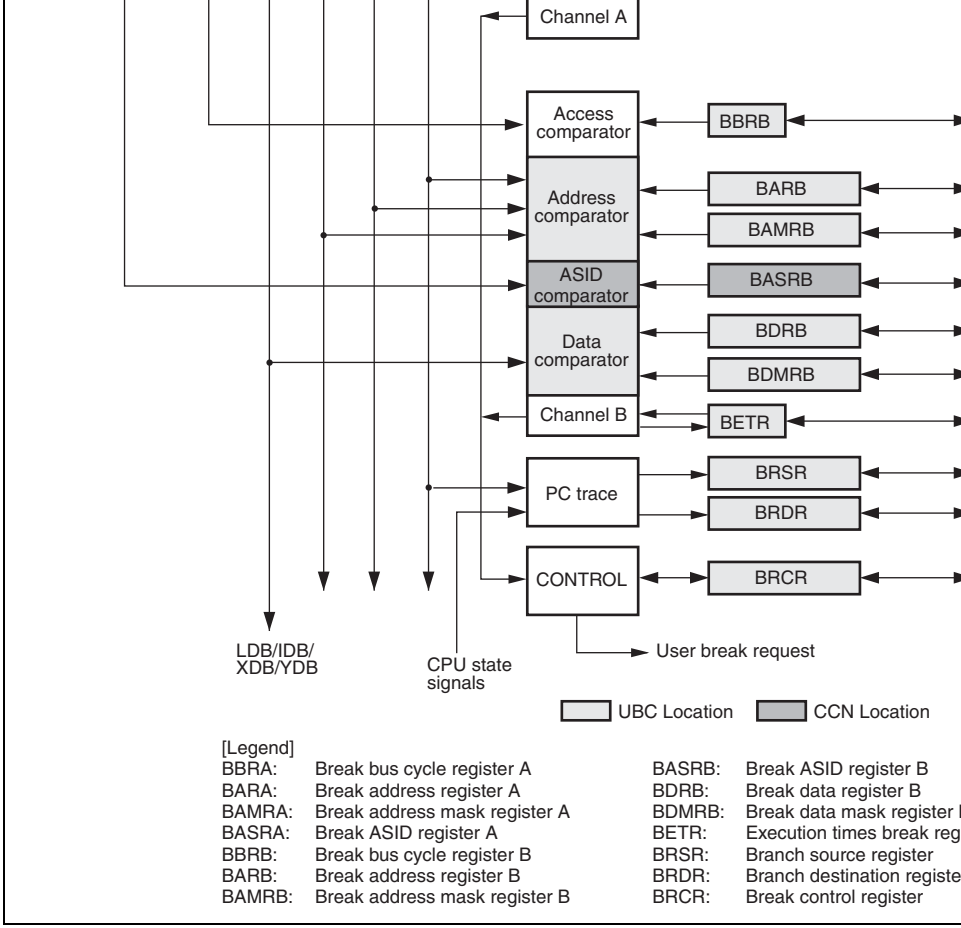


Figure 9.1 Block Diagram of User Break Controller

- Break bus cycle register B (BBRB)
- Break data register B (BDRB)
- Break data mask register B (BDMRB)
- Break control register (BRCR)
- Execution times break register (BETR)
- Branch source register (BRSR)
- Branch destination register (BRDR)
- Break ASID register A (BASRA)
- Break ASID register B (BASRB)

9.2.1 Break Address Register A (BARA)

BARA is a 32-bit readable/writable register. BARA specifies the address used as a break in channel A.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAA31 to BAA 0	All 0	R/W	Break Address A Store the address on the LAB or IAB specifying conditions of channel A.

				0: Break address bit BAAn of channel A is inc the break condition 1: Break address bit BAAn of channel A is ma is not included in the break condition Note: n = 31 to 0

9.2.3 Break Bus Cycle Register A (BBRA)

BBRA is a 16-bit readable/writable register, which specifies (1) L bus cycle or I bus cyc instruction fetch or data access, (3) read or write, and (4) operand size in the break cond channel A.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
7	CDA1	0	R/W	L Bus Cycle/I Bus Cycle Select A
6	CDA0	0	R/W	Select the L bus cycle or I bus cycle as the bu the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the L bus cycle 10: The break condition is the I bus cycle 11: The break condition is the L bus cycle

3	RWA1	0	R/W	Read/Write Select A
2	RWA0	0	R/W	Select the read cycle or write cycle as the bus the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or wr
1	SZA1	0	R/W	Operand Size Select A
0	SZA0	0	R/W	Select the operand size of the bus cycle for th channel A break condition. 00: The break condition does not include oper 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

If the I bus or L bus is selected in BBRB, an IAB address is set in BAB31 to BAB0.

If the X memory is selected in BBRB, the values to 1 in XAB are set in BAB31 to BAB17. In this c values in BAB16 to BAB0 are arbitrary.

If the Y memory is selected in BBRB, the values to 1 in YAB are set in BAB15 to BAB1. In this ca values in BAB31 to BAB16, and BABO are arbitr

Table 9.1 Specifying Break Address Register

Bus Selection in BBRB	BAB31 to BAB17	BAB16	BAB15 to BAB1	BA
L bus	LAB31 to LAB0			
I bus	IAB31 to IAB0			
X bus	XAB15 to XAB1	Don't care	Don't care	Do
Y bus	Don't care	Don't care	YAB15 to YAB1	Do

				break condition 1: Break address BABn of channel B is masked not included in the break condition Note: n = 31 to 0

9.2.6 Break Data Register B (BDRB)

BDRB is a 32-bit readable/writable register. The control bits CDB1, CDB0, XYE, and X BBRB select one of the four data buses for break condition B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDB31 to BDB0	All 0	R/W	Break Data Bit B Stores data which specifies a break condition i B. If the I bus is selected in BBRB, the break data is set in BDB31 to BDB0. If the L bus is selected in BBRB, the break data is set in BDB31 to BDB0. If the X memory is selected in BBRB, the break bits 15 to 0 in XDB is set in BDB31 to BDB16. case, the values in BDB15 to BDB0 are arbitra If the Y memory is selected in BBRB, the break bits 15 to 0 in YDB are set in BDB15 to BDB0. case, the values in BDB31 to BDB16 are arbitra

3. Set the data in bits 31 to 16 when including the value of the data bus as an L break condition for the MOVS.W @-As,Ds, MOVS.W @As,Ds, MOVS.W @A MOVS.W @As+Ix,Ds instruction.

9.2.7 Break Data Mask Register B (BDMRB)

BDMRB is a 32-bit readable/writable register. BDMRB specifies bits masked in the bre specified by BDRB.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDMB31 to BDMB 0	All 0	R/W	Break Data Mask B Specifies bits masked in the break data of cha specified by BDRB (BDB31 to BDB0). 0: Break data BDBn of channel B is included break condition 1: Break data BDBn of channel B is masked a included in the break condition Note: n = 31 to 0

Notes: 1. Specify an operand size when including the value of the data bus in the break
2. When the byte size is selected as a break condition, the same byte data mu bits 15 to 8 and 7 to 0 in BDRB as the break mask data in BDMRB.
3. Set the mask data in bits 31 to 16 when including the value of the data bus a bus break condition for the MOVS.W @-As,Ds, MOVS.W @As,Ds, MOVS.W @As+,Ds, or MOVS.W @As+Ix,Ds instruction.

9	XYE	0	R/W	Selects the X memory bus or Y memory bus as channel B break condition. Note that this bit se enabled only when the L bus is selected with th and CDB0 bits. Selection between the X memo and Y memory bus is done by the XYS bit. 0: Selects L bus for the channel B break condit 1: Selects X/Y memory bus for the channel B b condition
8	XYS	0	R/W	Selects the X bus or the Y bus as the bus of th channel B break condition. 0: Selects the X bus for the channel B break co 1: Selects the Y bus for the channel B break co
7	CDB1	0	R/W	L Bus Cycle/I Bus Cycle Select B
6	CDB0	0	R/W	Select the L bus cycle or I bus cycle as the bus the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the L bus cycle 10: The break condition is the I bus cycle 11: The break condition is the L bus cycle

3	RWB1	0	R/W	Read/Write Select B
2	RWB0	0	R/W	Select the read cycle or write cycle as the bus the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or w
1	SZB1	0	R/W	Operand Size Select B
0	SZB0	0	R/W	Select the operand size of the bus cycle for th B break condition. 00: The break condition does not include ope 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

5. Enable PC trace.
6. Enable ASID check.

BRCR is a 32-bit readable/writable register that has break conditions match flags and bits setting a variety of break conditions.

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
21	BASMA	0	R/W	Break ASID Mask A Specifies whether bits in channel A break ASID7 ASID0 (BASA7 to BASA0) which are set in BAS masked or not. 0: All BASRA bits are included in the break cond and the ASID is checked 1: All BASRA bits are not included in the break conditions and the ASID is not checked
20	BASMB	0	R/W	Break ASID Mask B Specifies whether bits in channel B break ASID7 ASID0 (BASB7 to BASB0) which are set in BAS masked or not. 0: All BASRB bits are included in the break cond and the ASID is checked 1: All BASRB bits are not included in the break c and the ASID is not checked

				match 1: The L bus cycle condition for channel A mat
14	SCMFCB	0	R/W	L Bus Cycle Condition Match Flag B When the L bus cycle condition in the break co set for channel B is satisfied, this flag is set to cleared to 0). In order to clear this flag, write 0 bit. 0: The L bus cycle condition for channel B doe match 1: The L bus cycle condition for channel B mat
13	SCMFDA	0	R/W	I Bus Cycle Condition Match Flag A When the I bus cycle condition in the break cor set for channel A is satisfied, this flag is set to cleared to 0). In order to clear this flag, write 0 bit. 0: The I bus cycle condition for channel A does match 1: The I bus cycle condition for channel A matc
12	SCMFDB	0	R/W	I Bus Cycle Condition Match Flag B When the I bus cycle condition in the break cor set for channel B is satisfied, this flag is set to cleared to 0). In order to clear this flag, write 0 bit. 0: The I bus cycle condition for channel B does 1: The I bus cycle condition for channel B matc

				1: PC break of channel A is set after instruction execution
9	—	0	R	Reserved
8	—	0	R	These bits are always read as 0. The write valu always be 0.
7	DBEB	0	R/W	Data Break Enable B Selects whether or not the data bus condition is included in the break condition of channel B. 0: No data bus condition is included in the cond channel B 1: The data bus condition is included in the con channel B
6	PCBB	0	R/W	PC Break Select B Selects the break timing of the instruction fetch channel B as before or after instruction executi 0: PC break of channel B is set before instructi execution 1: PC break of channel B is set after instruction execution
5	—	0	R	Reserved
4	—	0	R	These bits are always read as 0. The write valu always be 0.

		0	R	These bits are always read as 0. The write val always be 0.
0	ETBE	0	R/W	Number of Execution Times Break Enable Enables the execution-times break condition o channel B. If this bit is 1 (break enable), a user issued when the number of break conditions m with the number of execution times that is spec BETR. 0: The execution-times break condition is disab channel B 1: The execution-times break condition is enab channel B

				These bits are always read as 0. The write value always be 0.
11 to 0	BET11 to BET0	All 0	R/W	Number of Execution Times

Note: When the instruction fetch cycle is specified as the break condition of the channel break condition is triggerd by the following instructions, the BETR is decrimented b following value (not by one).

Instruction	Decrement value
RTE	4
DMULS.L Rm,Rn	2
DMULU.L Rm,Rn	2
MAC.L @Rm+,@Rn	2
MAC.W @Rm+,@Rn	2
MUL.L Rm,Rn	3
AND.B #imm,@(R0,GBR)	3
OR.B #imm,@(R0,GBR)	3
TAS.B @Rn	3
TST.B #imm,@(R0,GBR)	3
XOR.B #imm,@(R0,GBR)	3
LDC Rm,SR	4
LDC Rm,GBR	4
LDC Rm,VBR	4
LDC Rm,SSR	4
LDC Rm,SPC	4
LDC Rm,R0_BANK	4
LDC Rm,R1_BANK	4
LDC Rm,R2_BANK	4
LDC Rm,R3_BANK	4
LDC Rm,R4_BANK	4
LDC Rm,R5_BANK	4
LDC Rm,R6_BANK	4
LDC Rm,R7_BANK	4

Instruction	Decremen
LDC.L @Rm+,SR	6
LDC.L @Rm+,GBR	4
LDC.L @Rm+,VBR	4
LDC.L @Rm+,SSR	4
LDC.L @Rm+,SPC	4
LDC.L @Rm+,R0_BANK	4
LDC.L @Rm+,R1_BANK	4
LDC.L @Rm+,R2_BANK	4
LDC.L @Rm+,R3_BANK	4
LDC.L @Rm+,R4_BANK	4
LDC.L @Rm+,R5_BANK	4
LDC.L @Rm+,R6_BANK	4
LDC.L @Rm+,R7_BANK	4
LDC.L @Rn+,MOD	4
LDC.L @Rn+,RS	4
LDC.L @Rn+,RE	4
LDC Rn,MOD	4
LDC Rn,RS	4
LDC Rn,RE	4
BSR label	4
BSRF Rm	2
JSR @Rm	2

				Indicates whether the branch source addres stored. When a branch source address is fe flag is set to 1. This flag is cleared to 0 by re from BRSR. 0: The value of BRSR register is invalid 1: The value of BRSR register is valid
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
27 to 0	BSA27 to BSA0	—	R	Branch Source Address Store bits 27 to 0 of the branch source addr

				Indicates whether a branch destination addres stored. When a branch destination address is this flag is set to 1. This flag is cleared to 0 by BRDR. 0: The value of BRDR register is invalid 1: The value of BRDR register is valid
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
27 to 0	BDA27 to BDA0	—	R	Branch Destination Address Store bits 27 to 0 of the branch destination ad

9.2.13 Break ASID Register A (BASRA)

BASRA is an 8-bit readable/writable register that specifies ASID which becomes the brea condition for channel A. BASRA is in CCN.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BASA7 to BASA0	—	R/W	Break ASID A Store ASID (bits 7 to 0) which is the break conditi channel A.

9.3 Operation

9.3.1 Flow of the User Break Operation

The flow from setting of break conditions to user break exception processing is describe

1. The break addresses and corresponding ASID are set in the break address registers (
   BARB) and break ASID registers (BASRA or BASRB in CCN). The masked addres
   in the break address mask registers (BAMRA or BAMRB). The break data is set in t
   data register (BDRB). The masked data is set in the break data mask register (BDMR
   bus break conditions are set in the break bus cycle registers (BBRA or BBRB). Thre
   of BBRA or BBRB (L bus cycle/I bus cycle select, instruction fetch/data access sele
   read/write select) are each set. No user break will be generated if even one of these g
   set with 00. The respective conditions are set in the bits of the break control register
   Make sure to set all registers related to breaks before setting BBRA or BBRB.
2. When the break conditions are satisfied, the UBC sends a user break request to the C
   sets the L bus condition match flag (SCMFCA or SCMFCB) and the I bus condition
   flag (SCMFDA or SCMFDB) for the appropriate channel. When the X/Y memory b
   specified for channel B, SCMFCB is used for the condition match flag.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCMF
   be used to check if the set conditions match or not. The matching of the conditions s
   but they are not reset. 0 must first be written to them before they can be used again.
4. There is a chance that the break set in channel A and the break set in channel B occu
   the same time. In this case, there will be only one break request to the CPU, but thes
   break channel match flags could be both set.
5. When selecting the I bus as the break condition, note the following:

— For instruction fetch cycles issued on the L bus by the CPU, even though their log
addresses are not to be cached, they are issued in longwords and their addresses ar
rounded to match longword boundaries.

— If a logical address issued on the L bus by the CPU is an address to be cached and
miss occurs, its bus cycle is issued as a cache fill cycle on the I bus. In this case, it
in longwords and its address is rounded to match longword boundaries. However
cache fill is not performed for a write miss in write through mode. In this case, the
cycle is issued with the data size specified on the L bus and its address is not roun
write back mode, a write back cycle may be issued in addition to a read fill cycle.
longword bus cycle whose address is rounded to match longword boundaries.

— I bus cycles (including read fill cycles) resulting from instruction fetches on the L
the CPU are defined as instruction fetch cycles on the I bus, while other bus cycles
defined as data access cycles.

— The DMAC and E-DMAC only issues data access cycles for I bus cycles.

— If a break condition is specified for the I bus, even when the condition matches in
cycle resulting from an instruction executed by the CPU, at which instruction the
to be accepted cannot be clearly defined.

6. While the block bit (BL) in the CPU status register (SR) is set to 1, no breaks can be a
However, condition determination will be carried out, and if the condition matches, th
corresponding condition match flag is set to 1.

fetched by overrun (instructions fetched at a branch or during an interrupt transition, be executed). When this kind of break is set for the delay slot of a delayed branch ins the break is generated prior to execution of the delayed branch instruction.

Note: If a branch does not occur at a delay condition branch instruction, the subsec instruction is not recognized as a delay slot.

3. When the condition is specified to be occurred after execution, the instruction set wi break condition is executed and then the break is generated prior to the execution of instruction. As with pre-execution breaks, this cannot be used with overrun fetch ins When this kind of break is set for a delayed branch instruction and its delay slot, a br generated until the first instruction at the branch destination.
4. When an instruction fetch cycle is set for channel B, the break data register B (BDRB ignored. Therefore, break data cannot be set for the break of the instruction fetch cyc
5. If the I bus is set for a break of an instruction fetch cycle, the condition is determined instruction fetch cycles on the I bus. For details, see 5 in section 9.3.1, Flow of the U Operation.

9.3.3 Break on Data Access Cycle

1. If the L bus is specified as a break condition for data access break, condition compar performed for the logical addresses (and data) accessed by the executed instructions, break occurs if the condition is satisfied. If the I bus is specified as a break condition comparison is performed for the physical addresses (and data) of the data access cyc issued on the I bus by all bus masters including the CPU, and a break occurs if the co satisfied. For details on the CPU bus cycles issued on the I bus, see 5 in section 9.3.1 the User Break Operation.
2. The relationship between the data access cycle address and the comparison condition operand size is listed in table 9.3.

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

3. When the data value is included in the break conditions on channel B:

   When the data value is included in the break conditions, either longword, word, or by
   specified as the operand size of the break bus cycle register B (BBRB). When data va
   included in break conditions, a break is generated when the address conditions and da
   conditions both match. To specify byte data for this case, set the same data in two byt
   15 to 8 and bits 7 to 0 of the break data register B (BDRB) and break data mask regist
   (BDMRB). When word or byte is set, bits 31 to 16 of BDRB and BDMRB are ignore
   word data in bits 31 to 16 in BDRB and BDMRB when including the value of the dat
   break condition for the MOVS.W @-As,Ds, MOVS.W @As,Ds, MOVS.W @As+,Ds
   MOVS.W @As+Ix,Ds instruction (bits 15 to 0 are ignored).

4. Access by a PREF instruction is handled as read access in longword units without acc
   Therefore, if including the value of the data bus when a PREF instruction is specified
   break condition, a break will not occur.

5. If the L bus is selected, a break occurs on ending execution of the instruction that mat
   break condition, and immediately before the next instruction is executed. However, w
   is also specified as the break condition, the break may occur on ending execution of th
   instruction following the instruction that matches the break condition. If the I bus is s
   the instruction at which the break will occur cannot be determined. When this kind of
   occurs at a delayed branch instruction or its delay slot, the break may not actually tak
   until the first instruction at the branch destination.

Y-memory address in the lower 16 bits. Specification of X/Y-memory data is the sa
BDRB and BDMRB.

3. The timing of a data access break for the X memory or Y memory bus to occur is the
data access break of the L bus. For details, see 5 in section 9.3.3, Break on Data Acc

9.3.5 Sequential Break

1. By setting the SEQ bit in BRCR to 1, the sequential break is issued when a channel
condition matches after a channel A break condition matches. A user break is not ge
even if a channel B break condition matches before a channel A break condition mat
When channels A and B conditions match at the same time, the sequential break is n
To clear the channel A condition match when a channel A condition match has occu
channel B condition match has not yet occurred in a sequential break specification, c
SEQ bit in BRCR to 0.
2. In sequential break specification, the L/I/X/Y bus can be selected and the execution
break condition can be also specified. For example, when the execution times break
is specified, the break condition is satisfied when a channel B condition matches wit
H'0001 after a channel A condition has matched.

instruction that matched the condition is not executed, and the break occurs before it.
when a delay slot instruction matches the condition, the address of the delayed branch
instruction is saved in the SPC.

2. When instruction fetch (after instruction execution) is specified as a break condition:

   The address of the instruction following the instruction that matched the break conditi
   saved in the SPC. The instruction that matches the condition is executed, and the brea
   before the next instruction is executed. However when a delayed branch instruction or
   slot matches the condition, these instructions are executed, and the branch destination
   is saved in the SPC.

3. When data access (address only) is specified as a break condition:

   The address of the instruction immediately after the instruction that matched the break
   condition is saved in the SPC. The instruction that matches the condition is executed,
   break occurs before the next instruction is executed. However when a delay slot instru
   matches the condition, the branch destination address is saved in the SPC.

4. When data access (address + data) is specified as a break condition:

   When a data value is added to the break conditions, the address of an instruction that i
   two instructions of the instruction that matched the break condition is saved in the SP
   which instruction the break occurs cannot be determined accurately.

   When a delay slot instruction matches the condition, the branch destination address is
   the SPC. If the instruction following the instruction that matches the break condition i
   branch instruction, the break may occur after the branch instruction or delay slot has f
   In this case, the branch destination address is saved in the SPC.

saved in BRDR.

When a repeat loop of the DSP extended function is used, control being transferred f
repeat end instruction to the repeat start instruction is not recognized as a branch, an
values are not stored in BRSR and BRDR.

3. BRSR and BRDR have eight pairs of queue structures. The top of queues is read firs
address stored in the PC trace register is read. BRSR and BRDR share the read point
BRSR and BRDR in order, the queue only shifts after BRDR is read. After switching
PCTE bit (in BRCR) off and on, the values in the queues are invalid.

9.3.8 Usage Examples

Break Condition Specified for L Bus Instruction Fetch Cycle:

- Register specifications

  BARA = H'00000404, BAMRA = H'00000000, BBRA = H'0054, BARB = H'00008
  BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'0000
  BRCR = H'00300400

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00000404, Address mask: H'00000000

  Bus cycle: L bus/instruction fetch (after instruction execution)/read (operand size is
  included in the condition)

  The ASID check is not included.

  <Channel B>

  Address: H'00008010, Address mask: H'00000006

  Data: H'00000000, Data mask: H'00000000

Specified conditions: Channel A/channel B sequential mode

<Channel A>

Address: H'00037226, Address mask: H'00000000, ASID = H'80

Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

<Channel B>

Address: H'0003722E, Address mask: H'00000000, ASID = H'70

Data: H'00000000, Data mask: H'00000000

Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

After an instruction with ASID = H’80 and address H'00037226 is executed, a user br
occurs before an instruction with ASID = H’70 and address H'0003722E is executed.

- Register specifications

  BARA = H'00027128, BAMRA = H'00000000, BBRA = H'005A, BARB = H'00031
  BAMRB = H'00000000, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'0000
  BRCR = H'00300000

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00027128, Address mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/write/word

  The ASID check is not included.

  <Channel B>

  Address: H'00031415, Address mask: H'00000000

  Data: H'00000000, Data mask: H'00000000

  The ASID check is not included.

  Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size i
  included in the condition)

Bus cycle: L bus/instruction fetch (before instruction execution)/write/word

<Channel B>

Address: H'0003722E, Address mask: H'00000000, ASID = H'70

Data: H'00000000, Data mask: H'00000000

Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

Since instruction fetch is not a write cycle on channel A, a sequential condition does match. Therefore, no user break occurs.

- Register specifications

  BARA = H'00000500, BAMRA = H'00000000, BBRA = H'0057, BARB = H'00001
  BAMRB = H'00000000, BBRB = H'0057, BDRB = H'00000000, BDMRB = H'0000
  BRCR = H'00300001, BETR = H'0005

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00000500, Address mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/longword

  The ASID check is not included.

  <Channel B>

  Address: H'00001000, Address mask: H'00000000

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/longword

  The number of execution-times break enable (5 times)

  The ASID check is not included.

  On channel A, a user break occurs before an instruction of address H'00000500 is ex
  On channel B, a user break occurs after the instruction of address H'00001000 are ex
  four times and before the fifth time.

Address: H'00008010, Address mask: H'00000006, ASID = H'70

Data: H'00000000, Data mask: H'00000000

Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size i included in the condition)

A user break occurs after an instruction with ASID = H’80 and addresses H'0000800 H'00008FFE is executed or before an instruction with ASID = H’70 and addresses H' to H'00008016 are executed.

Break Condition Specified for L Bus Data Access Cycle:

- Register specifications

  BARA = H'00123456, BAMRA = H'00000000, BBRA = H'0064, BARB = H'000AB BAMRB = H'000000FF, BBRB = H'006A, BDRB = H'0000A512, BDMRB = H'000 BRCR = H'00000080, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00123456, Address mask: H'00000000, ASID = H’80

  Bus cycle: L bus/data access/read (operand size is not included in the condition)

  <Channel B>

  Address: H'000ABCDE, Address mask: H'000000FF, ASID = H’70

  Data: H'0000A512, Data mask: H'00000000

  Bus cycle: L bus/data access/write/word

  On channel A, a user break occurs with longword read from ASID = H’80 and addres H'00123454, word read from address H'00123456, or byte read from address H'00123 channel B, a user break occurs when word H'A512 is written in ASID = H’70 and add H'000ABC00 to H'000ABCFE.

Y Address: H'0000F000, Address mask: H'FFFF0000

Data: H'00004567, Data mask: H'00000000

Bus cycle: Y bus/data access/write/word

The ASID check is not included.

On channel A, a user break occurs during word read from address H'01000000 in the space. On channel B, a user break occurs when word data H'4567 is written in addre H'0000F000 in the Y memory space. The X/Y-memory space is changed by a mode

Break Condition Specified for I Bus Data Access Cycle:

- Register specifications

  BARA = H'00314156, BAMRA = H'00000000, BBRA = H'0094, BARB = H'00055 BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00007878, BDMRB = H'000 BRCR = H'00000080, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00314156, Address mask: H'00000000, ASID = H'80

  Bus cycle: I bus/instruction fetch/read (operand size is not included in the condition

  <Channel B>

  Address: H'00055555, Address mask: H'00000000, ASID = H'70

  Data: H'00000078, Data mask: H'0000000F

  Bus cycle: I bus/data access/write/byte

  On channel A, a user break occurs when instruction fetch is performed for ASID = H address H'00314156 in the memory space.

  On channel B, a user break occurs when ASID = H'70 and byte data H'7* is written H'00055555 on the I bus.

match occurs in another bus cycle in sequential break setting. Therefore, no break occ
if a bus cycle, in which an A-channel match and a channel B match occur simultaneou
set.

4. When a user break and another exception occur at the same instruction, which has hig
priority is determined according to the priority levels defined in table 4.1 in section 4,
Exception Handling. If an exception with higher priority occurs, the user break is not
generated.
   - Pre-execution break has the highest priority.
   - When a post-execution break or data access break occurs simultaneously with a re
execution-type exception (including pre-execution break) that has higher priority,
execution-type exception is accepted, and the condition match flag is not set (see t
exception in the following note). The break will occur and the condition match fla
set only after the exception source of the re-execution-type exception has been cle
the exception handling routine and re-execution of the same instruction has ended
   - When a post-execution break or data access break occurs simultaneously with a
completion-type exception (TRAPA) that has higher priority, though a break does
occur, the condition match flag is set.
5. Note the following exception for the above note.

   If a post-execution break or data access break is satisfied by an instruction that genera
CPU address error (or TLB related exception) by data access, the CPU address error (
related exception) is given priority to the break. Note that the UBC condition match fl
in this case.

This LSI has the following power-down modes and function:

1. Sleep mode
2. Software standby mode
3. Module standby function

Table 10.1 shows the transition conditions for entering the modes from the program exe
state, as well as the CPU and peripheral module states in each mode and the procedures
canceling each mode.

Software Standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halt	Halt	Held	Held	Halt	Held	Self-refreshed	1. 2.
Module standby function	Set MSTP bit to 1 in STBCR, STBCR2, and STBCR3	Run	Run	Held	Held	Specified module halts	*2	Refreshed	1. 2.

Notes: 1. The RTC runs when the START bit in RCR2 is set to 1. For details, see section Realtime Clock (RTC).
2. Depends on the on-chip peripheral modules. For details, see section 1, Overvi Pin Function.

10.1.2 Reset

A reset is used at power-on or to re-execute from the initial state. This LSI supports two t reset: power-on reset and manual reset. In power-on reset, any processing to be currently is terminated and any events not executed are canceled to execute reset processing immed manual reset, processing required to maintain external memory contents is continued. Th following shows the conditions in which power-on reset or manual reset occurs.

- Power-on reset
  1. A low level signal is input to the $\overline{\text{RESETP}}$ pin.
  2. The WDT counter overflows if the WDT starts counting while the WT/$\overline{\text{IT}}$ and RS WTCSR are set to 1 and cleared to 0, respectively.
  3. An H-UDI reset occurs. (For details on the H-UDI reset, refer to section 22, User Debugging Interface (H-UDI).)

```
    NOP
    NOP
TESTCR2_SET
    NOP
    MOV.B   R0, @R1
    MOV.B   R0, @R1
    MOV.L   R0, @R2
    NOP
    NOP
    NOP
    MOV.L   @R2, R3
    CMP/EQ  R3, R0
    BF      TESTCR2_SET
    NOP
    NOP
```

			LH: Standby mode LL: Normal operation
Power-on reset	$\overline{\text{RESETP}}$	I	Inputting low level signal to this pin cause a t to power-on reset processing.
Manual reset	$\overline{\text{RESETM}}$	I	Inputting low level signal to this pin cause a t to manual reset processing.

Note: H and L indicate high and low levels, respectively. The STATUS1 and STATUS0 p
indicate the pin status in this order.

10.2 Register Descriptions

The following registers are used for the power-down modes. Refer to section 23, List of R
for the addresses and access size for these registers.

- Standby control register (STBCR)
- Standby control register 2 (STBCR2)
- Standby control register 3 (STBCR3)

10.2.1 Standby Control Register (STBCR)

STBCR is an 8-bit readable/writable register that specifies the state of the power-down m
register is initialized to H′00 at power-on reset but retains the previous value after manua

2	MSTP2	0	R/W	Module Stop Bit 2 When the MSTP2 bit is set to 1, the suppl clock to the TMU is halted. 0: TMU runs 1: Clock supply to TMU halted
1	MSTP1	0	R/W	Module Stop Bit 1 When the MSTP1 bit is set to 1, the suppl clock to the RTC is halted. 0: RTC runs 1: Clock supply to RTC halted
0	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.

				0: H-UDI runs 1: Clock supply to H-UDI halted
6	MSTP9	0	R/W	Module Stop Bit 9 When the MSTP9 bit is set to 1, the supply clock to the UBC is halted. 0: UBC runs 1: Clock supply to UBC halted
5	MSTP8	0	R/W	Module Stop Bit 8 When the MSTP8 bit is set to 1, the supply clock to the DMAC is halted. 0: DMAC runs 1: Clock supply to DMAC halted
4	MSTP7	0	R/W	Module Stop Bit 7 When the MSTP7 bit is set to 1, the supply clock to the DSP is halted. 0: DSP runs 1: Clock supply to DSP halted
3	MSTP6	0	R/W	Module Stop Bit 6 When the MSTP6 bit is set to 1, the supply clock to the TLB is halted. 0: TLB runs 1: Clock supply to TLB halted

				When the MSTP3 bit is set to 1, the suppl clock to the X/Y memory is halted. 0: The X/Y memory runs 1: Clock supply to the X/Y memory halted

10.2.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of the peripher in the power-down mode. This register is initialized to H′00 at power-on reset but retain previous value after manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
3	MSTP33	0	R/W	Module Stop Bit 33 When the MSTP33 bit is set to 1, the supp clock to the SIOF1 is halted. 0: SIOF1 runs 1: Clock supply to SIOF1 halted
2	MSTP32	0	R/W	Module Stop Bit 32 When the MSTP32 bit is set to 1, the supp clock to the SIOF0 is halted. 0: SIOF0 runs 1: Clock supply to SIOF0 halted

10.3 Operation

10.3.1 Sleep Mode

Transition to Sleep Mode: Executing the SLEEP instruction when the STBY bit in STB causes a transition from the program execution state to sleep mode. Although the CPU ha immediately after executing the SLEEP instruction, the contents of its internal registers re unchanged. The on-chip peripheral modules continue to run in sleep mode and the clock to be output to the CKIO pin. In sleep mode, a high signal and low signal are output from STATUS1 and STATUS0 pins, respectively.

Canceling Sleep Mode: Sleep mode is canceled by an interrupt (NMI, IRQ, IRL, or on-c peripheral module) or reset. Interrupts are accepted in sleep mode even when the BL bit i If necessary, save SPC and SSR to the stack before executing the SLEEP instruction.

- Canceling with an Interrupt

  When an NMI, IRQ, IRL, or on-chip peripheral module interrupt occurs, sleep mode i canceled and interrupt exception handling is executed. A code indicating the interrupt set in INTEVT and INTEVT2.
- Canceling with a Reset

  Sleep mode is canceled by a power-on reset or a manual reset.

Table 10.3 Register States in Software Standby Mode

Module	Registers Initialized	Registers Retaini
Interrupt Controller (INTC)	—	All registers
On-Chip Oscillation Circuits	—	All registers
User Break Controller (UBC)	—	All registers
Bus State Controller (BSC)	—	All registers
Timer Unit (TMU)	TSTR	Registers other tha
I/O ports	—	All registers
H-UDI	—	All registers
SCIF0/1	—	All registers
SIOF0/1	—	All registers
EtherC, E-DMAC	—	All registers
DMAC	—	All registers

- Canceling with an Interrupt

The on-chip WDT can be used for hot starts. When the chip detects an NMI, IRQ*[1], IRL* RTC*[1] interrupt, the clock will be supplied to the entire chip and software standby mode after the time set in the WDT's timer control/status register has elapsed. The STATUS1 a STATUS0 pins go low. Interrupt exception handling then begins and a code indicating th interrupt source is set in INTEVT and INTEVT2. After the branch to the interrupt handli routine, clear the STBY bit in STBCR. The WDT stops automatically. If the STBY bit is cleared, the WDT continues operation and a transition is made to software standby mode WTCNT reaches H'80. A manual reset is not accepted until the STBY bit is cleared to 0.

Interrupts are accepted in software standby mode even when the BL bit in SR is 1. If nece save SPC and SSR to the stack before executing the SLEEP instruction.

Immediately after an interrupt is detected, the phase of the CKIO pin clock output may be unstable, until the software standby mode is canceled.

Notes: 1. Only when the RTC is used, software standby mode can be canceled by an IR or RTC.
2. This standby mode can be canceled only by a power-on reset.

Figure 10.1 Canceling Standby Mode with STBCR.STBY

- Canceling with a Reset

Software standby mode is canceled by a reset (power-on or manual). Keep the $\overline{\text{RESETP}}$ $\overline{\text{RESETM}}$ pin low until the clock oscillation settles. The internal clock will continue to b to the CKIO pin.

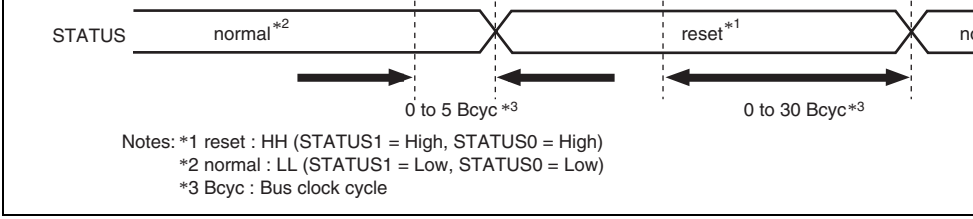
10.3.3 Module Standby Function

Transition to Module Standby Function: Setting each MSTP bit in the standby contro to 1 halts the supply of clocks to the corresponding on-chip peripheral modules. This fu be used to reduce the power consumption in normal or sleep mode. Before a transition is module should be disabled.

In the module standby state, the functions of the external pins of the on-chip peripheral change depending on the on-chip peripheral module. For details, see section 1, Overview Function. All of the register states are the same as those in standby mode. For details, se 10.3.

Canceling Module Standby Function: The module standby function can be canceled b the MSTP bits to 0, or by a power-on reset.

To cancel the module standby function by clearing the corresponding MSTP bit to 0, rea MSTP bit to check the MSTP bit was cleared correctly.



Notes: *1 reset : HH (STATUS1 = High, STATUS0 = High)
*2 normal : LL (STATUS1 = Low, STATUS0 = Low)
*3 Bcyc : Bus clock cycle

Figure 10.2 STATUS Output at Power-On Reset

Figure 10.3 STATUS Output at Manual Reset

Software Standby Mode:

- Software standby mode is canceled by an interrupt

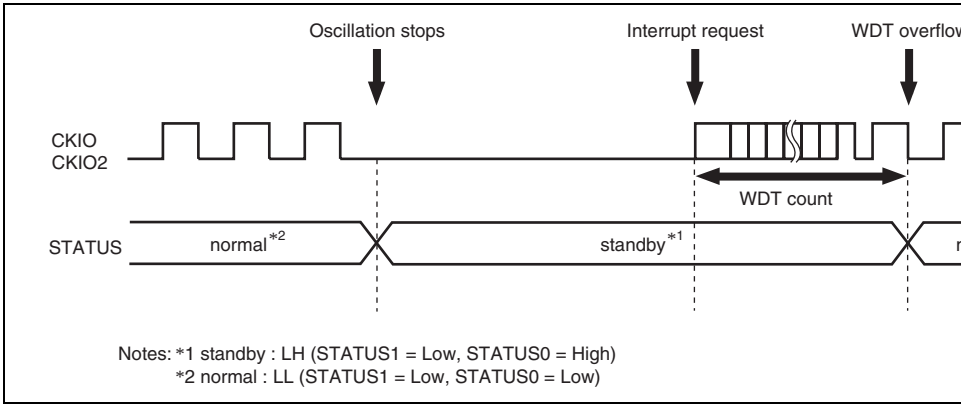


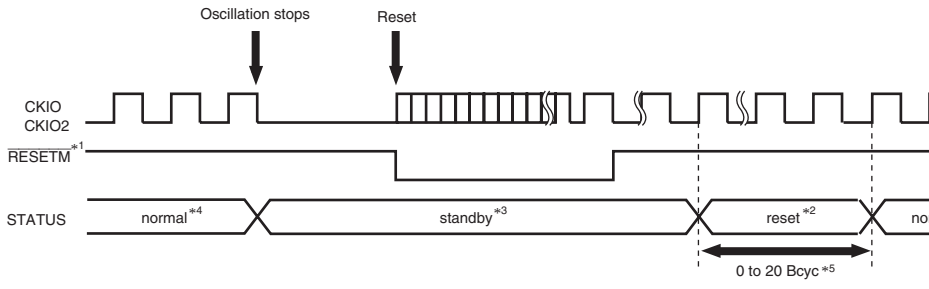
Figure 10.4 STATUS Output when Software Standby Mode is Canceled by Int

0 to 10 Bcyc *5 0 to 30 Bcyc *5

Notes: *1 If a standby mode is canceled by a power on reset, the WDT stops counting.
$\overline{\text{RESETP}}$ must be kept low for the PLL oscillation stabilization time.
*2 reset : HH (STATUS1 = High, STATUS0 = High)
*3 standby : LH (STATUS1 = Low, STATUS0 = High)
*4 normal : LL (STATUS1 = Low, STATUS0 = Low)
*5 Bcyc : Bus clock cycle

Figure 10.5 STATUS Output when Software Standby Mode is Canceled by Power-

- Software standby mode is canceled by a manual reset



Notes: *1 If a standby mode is canceled by a power on reset, the WDT stops counting.
$\overline{\text{RESETM}}$ must be kept low for the PLL oscillation stabilization time.
*2 reset : HH (STATUS1 = High, STATUS0 = High)
*3 standby : LH (STATUS1 = Low, STATUS0 = High)
*4 normal : LL (STATUS1 = Low, STATUS0 = Low)
*5 Bcyc : Bus clock cycle

Figure 10.6 STATUS Output when Software Standby Mode is Canceled by Manu

Figure 10.7 STATUS Output when Sleep Mode is Canceled by Interrup

- Sleep mode is canceled by a power-on reset

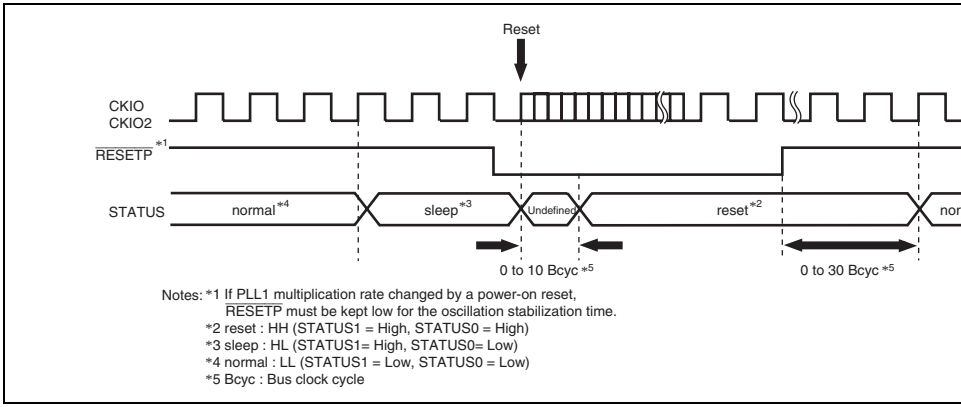


Figure 10.8 STATUS Output when Sleep Mode is Canceled by Power-on R

*4 normal:LL (STATUS1 = Low, STATUS0 = Low)
*5 Bcyc:Bus clock cycle

Figure 10.9 STATUS Output when Sleep Mode is Canceled by Manual Res

standby mode and temporary standbys, such as frequency changes. It can also be used as
ordinary watchdog timer or interval timer.

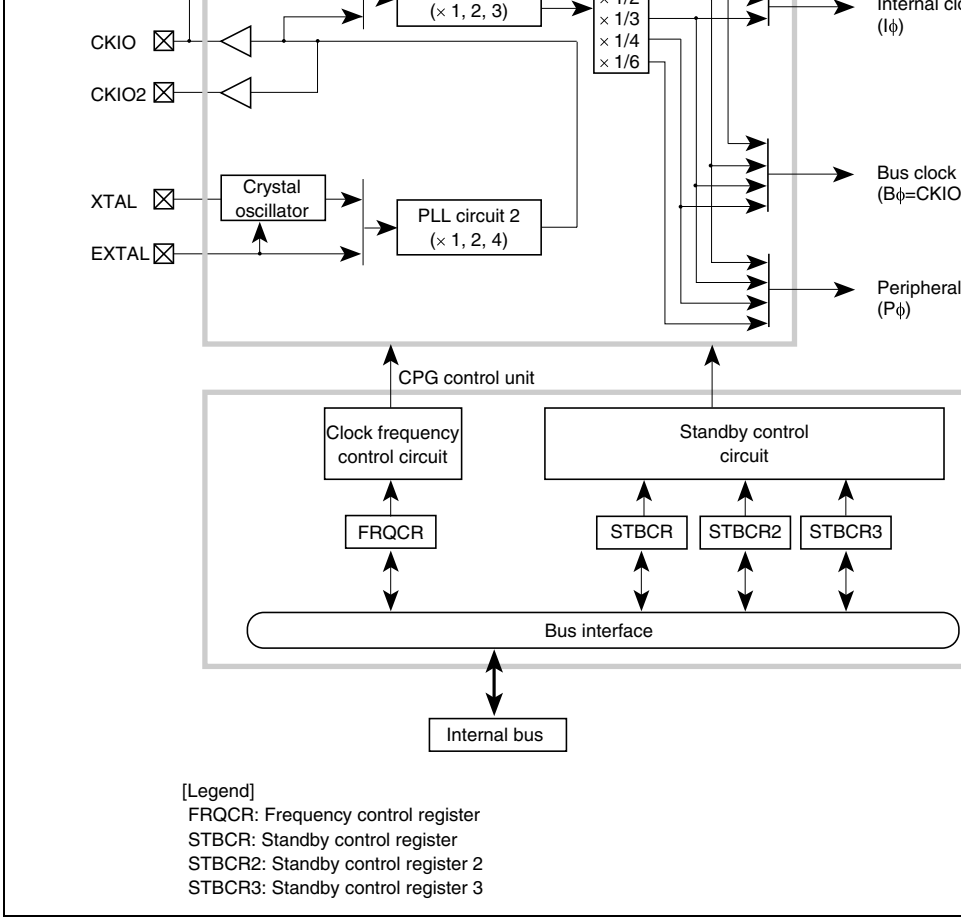
11.1.1 Features

The CPG has the following features:

- Seven clock modes: Selection of seven clock modes according to the frequency rang
used and direct connection of crystal resonator or external clock input.
- Three clocks generated independently: An internal clock (Iϕ) for the CPU and cache
peripheral clock (Pϕ) for the peripheral modules; and a bus clock (Bϕ = CKIO) for th
bus interface.
- Frequency change function: Internal and peripheral clock frequencies can be change
independently using the PLL circuit and divider circuit within the CPG. Frequencies
changed by software using frequency control register (FRQCR) settings.
- Power-down mode control: The clock can be stopped for sleep mode and standby mo
specific modules can be stopped using the module standby function.

The WDT has the following features:

- Can be used to ensure the clock settling time:
Use the WDT to cancel standby mode and the temporary standbys which occur when
frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode:
Internal resets occur after counter overflow.
Selection of power-on reset or manual reset.



[Legend]
FRQCR: Frequency control register
STBCR: Standby control register
STBCR2: Standby control register 2
STBCR3: Standby control register 3

Figure 11.1 Block Diagram of CPG

and EXTAL pins. This crystal oscillator operates according to the clock operating mo
setting.

4. Divider 1: Divider 1 generates a clock at the operating frequency used by the internal
peripheral clock. The operating frequency of the internal clock (Iϕ) can be 1, 1/2, or 1
the output frequency of PLL circuit 1, as long as it stays at or above the clock frequen
CKIO pin. The operating frequency of the peripheral clock (Pϕ) can be 1/2, 1/3, 1/4, o
times the output frequency of PLL circuit 1 within 8.34 MHz ≤ Pϕ ≤ 33.34 MHz. The
ratio is set in the frequency control register.
5. Clock Frequency Control Circuit: The clock frequency control circuit controls the clo
frequency using the MD pins and the frequency control register.
6. Standby Control Circuit: The standby control circuit controls the state of the on-chip o
and other modules during clock switching or in sleep or standby mode.
7. Frequency Control Register: The frequency control register has control bits assigned f
following functions: clock output/non-output from the CKIO pin, the frequency multi
ratio of PLL circuit 1, and the frequency division ratio of the internal clock and the pe
clock.
8. Standby Control Register: The standby control register has bits for controlling the pow
modes. See section 10, Power-Down Modes, for more information.

Crystal oscillator pins (clock input pins)	XTAL	O	Connects a crystal oscillator.
	EXTAL	I	Connects a crystal oscillator. Also used to inpu external clock.
Clock I/O pin	CKIO	IO	Inputs or outputs an external clock.
Clock output pin	CKIO2	O	Outputs an external clock.

Note: To prevent device malfunction, the value of the mode control pin is sampled only power-on reset.

11.3 Clock Operating Modes

Table 11.2 shows the relationship between the mode control pins (MD2 to MD0) combi and the clock modes. Table 11.3 shows the available combinations of the values of the c modes and frequency control register (FRQCR).

Table 11.2 Clock Operating Modes

	Pin Values			Clock I/O		PLL2	PLL1	CKI
Mode	MD2	MD1	MD0	Source	Output	On/Off	On/Off	Fre
0	0	0	0	EXTAL	CKIO	ON	ON	(EX
					CKIO2	(x 1)	(x 1, 2, 3)	
1	0	0	1	EXTAL	CKIO	ON	ON	(EX
					CKIO2	(x 4)	(x 1, 2, 3)	
2	0	1	0	Crystal resonator	CKIO	ON	ON	(Cry
					CKIO2	(x 4)	(x 1, 2, 3)	
4	1	0	0	Crystal resonator	CKIO	ON	ON	(Cry
					CKIO2	(x 1)	(x 1, 2, 3)	

Mode 0: An external clock is input from the EXTAL pin and undergoes waveform shaping
circuit 2 before being supplied inside this LSI. An input clock frequency of 33.3 M
66.67 MHz can be used, and the CKIO frequency range is 33.3 MHz to 66.67 M

Mode 1: An external clock is input from the EXTAL pin and its frequency is multiplied by 4
circuit 2 before being supplied inside this LSI, allowing a low-frequency external
be used. An input clock frequency of 10.00 MHz to 16.67 MHz can be used, and
CKIO frequency range is 40.00 MHz to 66.67 MHz.

Mode 2: The on-chip crystal oscillator operates, with the oscillation frequency being multi
by PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency e
clock to be used. A crystal oscillation frequency of 10.00 MHz to 16.67 MHz can
and the CKIO frequency range is 40.00 MHz to 66.67 MHz.

Mode 4: The on-chip crystal oscillator operates and undergoes waveform shaping by PLL
before being supplied inside this LSI. A crystal oscillation frequency of 33.34 MH
48.00 MHz can be used, and the CKIO frequency range is 33.34 MHz to 48.00 M

Mode 5: An external clock is input from the EXTAL pin and its frequency is multiplied by 2
circuit 2 before being supplied inside this LSI, allowing a low-frequency external
be used. An input clock frequency of 16.67 MHz to 33.34 MHz can be used, and
CKIO frequency range is 33.34 MHz to 66.67 MHz.

Mode 6: The on-chip crystal oscillator operates, with the oscillation frequency being multi
by PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency c
be used. A crystal oscillation frequency of 10.00 MHz to 16.67 MHz can be used
CKIO frequency range is 40.00 MHz to 66.67 MHz

Mode 7: In this mode, the CKIO pin is an input, an external clock is input to this pin, and
undergoes waveform shaping and also frequency multiplication according to the
by PLL circuit 1 before being supplied to this LSI. As PLL circuit 1 compensates
fluctuations in the CKIO pin load, this mode is suitable for connection of synchro
DRAM.

	1103	On (x2)	On (x1)	2:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz MHz
	1104	On (x2)	On (x1)	2:1:1/3	33.34 MHz to 66.67 MHz	33.34 MHz MHz
	1204	On (x3)	On (x1)	3:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz MHz
1, 2	1001	On (x1)	On (x4)	4:4:2	10.00 MHz to 16.67 MHz	40.00 MHz MHz
	1002	On (x1)	On (x4)	4:4:4/3	10.00 MHz to 16.67 MHz	40.00 MHz MHz
	1003	On (x1)	On (x4)	4:4:1	10.00 MHz to 16.67 MHz	40.00 MHz MHz
	1103	On (x2)	On (x4)	8:4:2	10.00 MHz to 16.67 MHz	40.00 MHz MHz
	1104	On (x2)	On (x4)	8:4:4/3	10.00 MHz to 16.67 MHz	40.00 MHz MHz
	1204	On (x3)	On (x4)	12:4:2	10.00 MHz to 16.67 MHz	40.00 MHz MHz
4	1001	On (x1)	On (x1)	1:1:1/2	33.34 MHz to 48.00 MHz	33.34 MHz MHz
	1002	On (x1)	On (x1)	1:1:1/3	33.34 MHz to 48.00 MHz	33.34 MHz MHz
	1003	On (x1)	On (x1)	1:1:1/4	33.34 MHz to 48.00 MHz	33.34 MHz MHz
	1103	On (x2)	On (x1)	2:1:1/2	33.34 MHz to 48.00 MHz	33.34 MHz MHz

	1003	On (x1)	On (x2)	2:2:1/2	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1103	On (x2)	On (x2)	4:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1104	On (x2)	On (x2)	4:2:2/3	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1204	On (x3)	On (x2)	6:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
6	1001	On (x1)	On (x2)	2:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1002	On (x1)	On (x2)	2:2:2/3	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1003	On (x1)	On (x2)	2:2:1/2	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1103	On (x2)	On (x2)	4:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1104	On (x2)	On (x2)	4:2:2/3	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1204	On (x3)	On (x2)	6:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
7	1001	On (x1)	Off	1:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1002	On (x1)	Off	1:1:1/3	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1003	On (x1)	Off	1:1:1/4	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz

1. Use the CKIO frequency within 33.34 MHz ≤ CKIO ≤ 66.67 MHz.
2. The input to divider 1 is the output of PLL circuit 1.
3. Use the internal clock frequency within 33.34 MHz ≤ Iϕ ≤ 200.00 MHz.
   The internal clock frequency is the product of the frequency of the CKIO pin, t
   frequency multiplication ratio of PLL circuit 1 selected by the STC bit in FRQC
   the division ratio selected by the IFC bit in FRQCR.
   Do not set the internal clock frequency lower than the CKIO pin frequency.
4. Use the peripheral clock frequency within 8.34 MHz ≤ Pϕ ≤ 33.34 MHz.
   The peripheral clock frequency is the product of the frequency of the CKIO pi
   frequency multiplication ratio of PLL circuit 1 selected by the STC bit in FRQC
   the division ratio selected by the PFC bit in FRQCR.
   Do not set the peripheral clock frequency higher than the frequency of the CK
5. × 1, × 2, or × 3 can be used as the multiplication ratio of PLL circuit 1. × 1, × 1
   can be selected as the division ratio of an internal clock. × 1/2, × 1/3, × 1/4, or
   × 1/6 can be selected as the division ratio of a peripheral clock. Set the rate in
6. The output frequency of PLL circuit 1 is the product of the CKIO frequency an
   multiplication ratio of PLL circuit 1. Use the output frequency under 200.00 M

and the frequency division ratio of the internal clock and the peripheral clock.

Only word access can be used on the FRQCR register. FRQCR is initialized to H'1003 by power-on reset, but retains its value in a manual reset and in standby mode.

The write values to bits 15 to 13, 11 to 10, 7 to 6, and 3 should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
12	CKOEN	1	R/W	Clock Output Enable CKOEN specifies whether a clock is output fr CKIO pin or the CKIO pin is placed in the lev state in the standby mode, CKIO pin is fixed during STATUS 1 = L, and STATUSO = H, w CKOEN is set to 0. Therefore, the malfunctio external circuit because of an unstable CKIO releasing the standby mode can be prevente CKIO pin becomes to input pin regardless of value of the CKOEN bit in clock operating mo 0: CKIO pin goes to low level state in standb 1: Clock is output from CKIO pin
11	—	0	R	Reserved
10	—	0	R	These bits are always read as 0. The write va should always be 0.

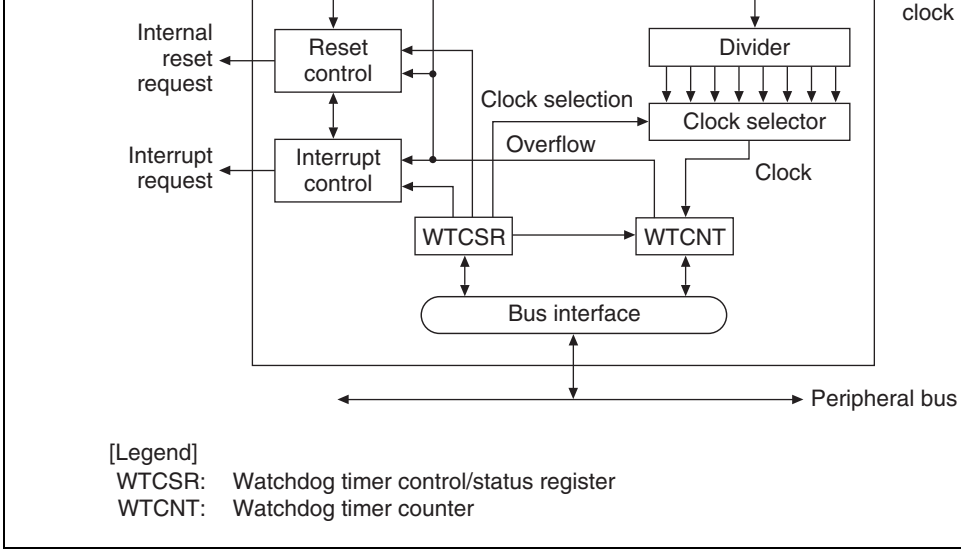
5	IFC1	0	R/W	Internal Clock Frequency Division Ratio
4	IFC0	0	R/W	These bits specify the frequency division rat internal clock (Iϕ) with respect to the output of PLL circuit 1. 00: × 1 time 01: × 1/2 time 10: × 1/3 time 11: Reserved (setting prohibited)
3	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
2	PFC2	0	R/W	Peripheral Clock Frequency Division Ratio
1	PFC1	1	R/W	These bits specify the division ratio of the p clock (Pϕ) frequency with respect to the out frequency of PLL circuit 1. 001: × 1/2 time 010: × 1/3 time 011: × 1/4 time 100: × 1/6 time Other than above: Reserved (setting prohib
0	PFC0	1	R/W	

1. In the initial state, the multiplication rate of PLL circuit 1 is 1.
2. Set a value that will become the specified oscillation settling time in the WDT and sto
   WDT. The following must be set:
   TME bit in WTCSR = 0: WDT stops
   CKS2 to CKS0 bits in WTCSR: Division ratio of WDT count clock
   WTCNT: Initial counter value
3. Set the desired value in the STC1 and STC0 bits. The division ratio can also be set in
   and IFC0 bits and PFC2 to PFC0 bits.
4. The processor pauses internally and the WDT starts incrementing. The internal and pe
   clocks both stop and the WDT is supplied with the clock. The clock will continue to b
   at the CKIO pin.
5. Supply of the clock that has been set begins at WDT count overflow, and the processo
   operating again. The WDT stops after it overflows.

11.5.2 Changing Division Ratio

The WDT will not count unless the multiplication rate is changed simultaneously.

1. In the initial state, IFC1 and IFC0 = 00 and PFC2 to PFC0 = 011.
2. Set the IFC1, IFC0, and PFC2 to PFC0 bits to the new division ratio. The values that
   are limited by the clock mode and the multiplication rate of PLL circuit 1. Note that if
   wrong value is set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.



[Legend]
WTCSR: Watchdog timer control/status register
WTCNT: Watchdog timer counter

Figure 11.2 Block Diagram of WDT

WTCNT is an 8-bit readable/writable counter. WTCNT increments on the selected clock. an overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interv mode. WTCNT is initialized to H'00 only by a power-on reset through the $\overline{\text{RESETP}}$ pin. word access to write to WTCNT, with H'5A in the upper byte. Use a byte access to read

Note: WTCNT differs from other registers in that it is more difficult to write to. See sec 11.7.3, Notes on Register Access, for details.

11.7.2 Watchdog Timer Control/Status Register (WTCSR)

WTCSR is an 8-bit readable/writable register composed of bits to select the clock used fo count, bits to select the timer mode, and overflow flags.

WTCSR is initialized to H'00 only by a power-on reset through the $\overline{\text{RESETP}}$ pin. When a overflow causes an internal reset, WTCSR retains its value. When used to count the clock time for canceling a standby, it retains its value after counter overflow.

Use a word access to write to WTCSR, with H'A5 in the upper byte. Use a byte access to WTCSR.

Note: WTCSR differs from other registers in that it is more difficult to write to. See sec 11.7.3, Notes on Register Access, for details.

				Selects whether to use the WDT as a watch or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: If WT/$\overline{IT}$ is modified when the WDT i the up-count may not be performed
5	RSTS	0	R/W	Reset Select Selects the type of reset when the WTCNT in watchdog timer mode. In interval timer m setting is ignored. 0: Power-on reset 1: Manual reset
4	WOVF	0	R/W	Watchdog Timer Overflow Indicates that the WTCNT has overflowed i watchdog timer mode. This bit is not set in i timer mode. 0: No overflow 1: WTCNT has overflowed in watchdog time
3	IOVF	0	R/W	Interval Timer Overflow Indicates that the WTCNT has overflowed i timer mode. This bit is not set in watchdog t mode. 0: No overflow 1: WTCNT has overflowed in interval timer

011	1/32	546 μs
100	1/64	1.09 ms
101	1/256	4.36 ms
110	1/1024	17.48 ms
111	1/4096	69.91 ms

Note: If bits CKS2 to CKS0 are modified wh
WDT is running, the up-count may nc
performed correctly. Ensure that thes
are modified only when the WDT is n
running.

11.7.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTC
more difficult to write to than other registers. The procedure for writing to these registers
below.

Writing to WTCNT and WTCSR: These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction.

When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the w
as shown in figure 11.3. When writing to WTCSR, set the upper byte to H'A5 and transfe
lower byte as the write data. This transfer procedure writes the lower byte data to WTCN
WTCSR.

Figure 11.3 Writing to WTCNT and WTCSR

11.8 Using WDT

11.8.1 Canceling Standbys

The WDT can be used to cancel standby mode with an interrupt such as an NMI. The pr
described below. (The WDT does not run when resets are used for canceling, so keep th
or $\overline{\text{RESETM}}$ pin low until the clock stabilizes.)

1. Before transitioning to standby mode, always clear the TME bit in WTCSR to 0. Wh
   TME bit is 1, an erroneous reset or interval timer interrupt may be generated when th
   overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the initial
   the counter in WTCNT. These values should ensure that the time till count overflow
   than the clock oscillation settling time.
3. Move to standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting the edge change of the NMI signal.
5. When the WDT count overflows, the CPG starts supplying the clock and the process
   resumes operation. The WOVF flag in WTCSR is not set at this time.
6. Since the WDT continues counting from H'00, clear the STBY bit in STBCR to 0 in
   interrupt processing program and this will stop the WDT. When the STBY bit remai
   LSI again enters the standby mode when the WDT has counted up to H'80. This stan
   can be canceled by power-on resets.

5. When the frequency control register (FRQCR) is written, the processor stop temporar
WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the proce
resumes operation. The WOVF flag in WTCSR is not set at this time.
5. The counter stops at the values H'00.
6. Before changing the WTCNT after the execution of the frequency change instruction,
confirm that the value of the WTCNT is H'00 by reading the WTCNT.

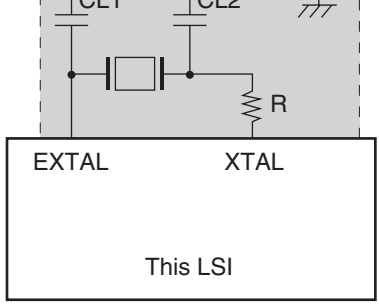
11.8.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in WTCSR to 1, set the reset type in the RSTS bit, set the type of co
clock in the CKS2 to CKS0 bits, and set the initial value of the counter in WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to p
the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and gener
type of reset specified by the RSTS bit. The counter then resumes counting.

11.8.4 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every ov
the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{\text{IT}}$ bit in WTCSR to 0, set the type of count clock in the CKS2 to CKS0
set the initial value of the counter in WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1 and an inte
timer interrupt request is sent to INTC. The counter then resumes counting.



Note: The values for CL1, CL2, and the damping resistance should be determined
consultation with the crystal manufacturer.

Figure 11.4 Points for Attention when Using Crystal Resonator

Bypass Capacitors: Insert a laminated ceramic capacitor as a bypass capacitor for each
and Vcc/VccQ pair. Mount the bypass capacitors to the power supply pins, and use com
with a frequency characteristic suitable for the operating frequency of the LSI, as well a
capacitance value.

Pin assignments of HQFP2828-256 (FP-256G/GV)

Vss/VssQ and Vcc/VccQ pair of digital circuitry
3 and 4, 13 and 14, 15 and 16, 25 and 26, 35 and 36, 43 and 44, 49 and 50, 57 and 58
73, 81 and 82, 83 and 84, 92 and 93, 99 and 100, 107 and 108, 113 and 114, 121 and
and 137, 146 and 147, 148 and 149, 158 and 159, 167 and 168, 176 and 177, 185 and
and 192, 206 and 207, 208 and 209, 221 and 222, 227 and 228, 235 and 236, 241 and

Vss/VssQ and Vcc/VccQ pair of the on-chip oscillator
193 and 196, 251 and 252, 253 and 254

When Using a PLL Oscillator Circuit: Keep the wiring from the PLL Vcc and PLL Vs connection pattern to the power supply pins short, and make the pattern width large, to m the inductance component.

Connect the EXTAL pin to VccQ or VssQ and make the XTAL pin open in clock mode 7

The analog power supply system of the PLL is sensitive to a noise. Therefore the system malfunction may occur by the intervention with other power supply. Do not supply the an power supply with the same resource as the digital power supply of Vcc and VccQ.

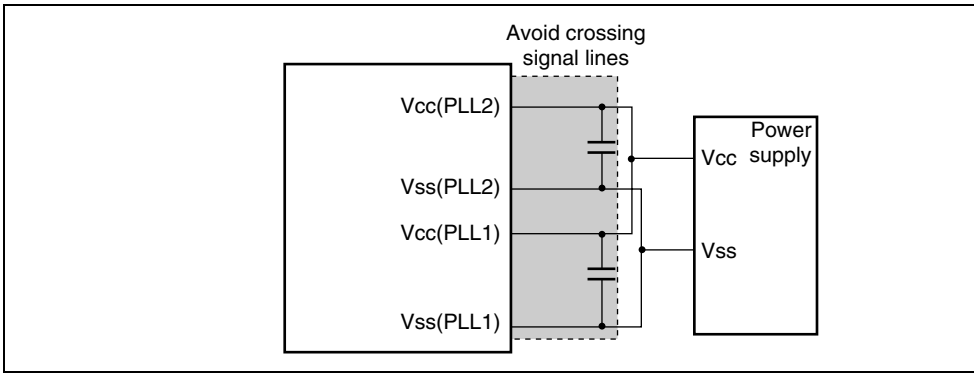


Figure 11.5 Points for Attention when Using PLL Oscillator Circuit

1. External address space

- A maximum 32 or 64 Mbytes for each of the eight areas, CS0, CS2 to CS4, CS5A, C CS6A and CS6B, totally 384 Mbytes (divided into eight areas).
- A maximum 64 Mbytes for each of the six areas, CS0, CS2 to CS4, CS5, and CS6, t total of 384 Mbytes (divided into six areas).
- Can specify the normal space interface, byte-selection SRAM, burst ROM (clock sy or asynchronous), SDRAM, PCMCIA for each address space.
- Can select the data bus width (8, 16, or 32 bits) for each address space.
- Controls the insertion of the wait state for each address space.
- Controls the insertion of the wait state for each read access and write access.
- Can set the independent idling cycle in the continuous access for five cases: read-wr same space/different space), read-read (in same space/different space), or the first cy write access.

2. Normal space interface

- Supports the interface that can directly connect to the SRAM.

3. Burst ROM (clock asynchronous) interface

- High-speed access to the ROM that has the page mode function.

4. SDRAM interface

- Can set the SDRAM in up to 2 areas.
- Multiplex output for row address/column address.
- Efficient access by single read/single write.
- High-speed access by bank-active mode.
- Supports an auto-refresh and self-refresh.

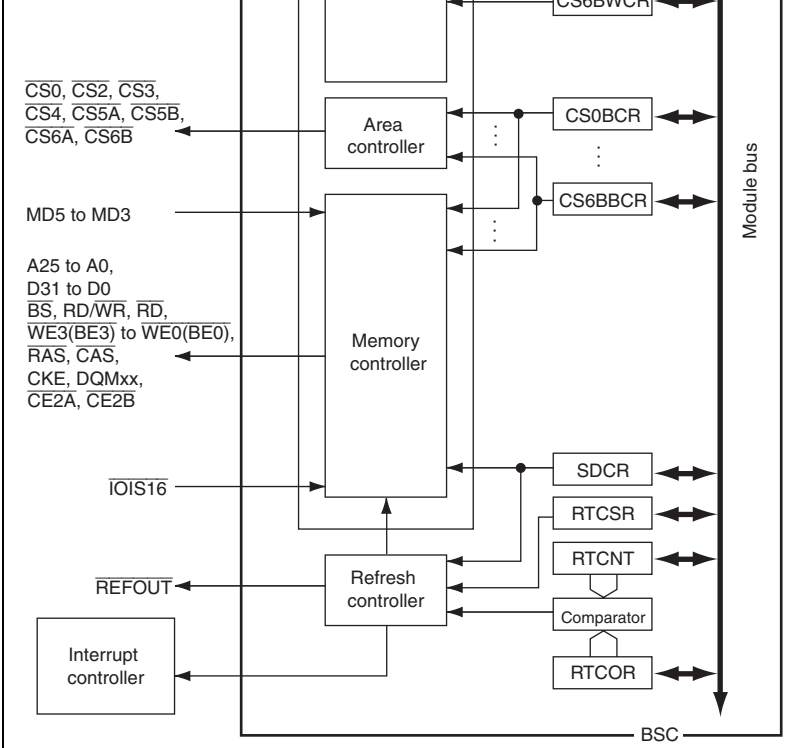
- Can connect directly to a burst ROM of the clock synchronous type.

8. Bus arbitration
- Shares all of the resources with other CPU and outputs the bus enable after receiving request from external devices.

9. Refresh function
- Supports the auto-refresh and self-refresh functions.
- Specifies the refresh interval using the refresh counter and clock selection.
- Can execute concentrated refresh by specifying the refresh counts (1, 2, 4, 6, or 8).

10. Interval timer using refresh counter
- Generates an interrupt request by a compare match.

The block diagram of the BSC is shown in figure 12.1.



[Legend]

CMNCR:	Common control register
CSnWCR:	CSn space wait control register (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B)
CSnBCR:	CSn space bus control register (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B)
SDCR:	SDRAM control register
RTCSR:	Refresh timer control/status register
RTCNT:	Refresh timer counter
RTCOR:	Refresh time constant register

Figure 12.1 Block Diagram of BSC

		synchronous/asynchronous), or PCMCIA is accessed. Asserte same timing as $\overline{CAS}$ in SDRAM access.
$\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$	O	Chip select
$\overline{CS5A}$	O	Chip select Active only for address map 1
$\overline{CS5B}$/$\overline{CE1A}$	O	Chip select Corresponds to PCMCIA card select signals D7 to D0 when th PCMCIA is used.
$\overline{CE2A}$	O	Corresponds to PCMCIA card select signals D15 to D8 when t PCMCIA is used.
$\overline{CS6A}$	O	Chip select Active only for address map 1
$\overline{CS6B}$/$\overline{CE1B}$	O	Chip select Corresponds to PCMCIA card select signals D7 to D0 when th PCMCIA is used.
$\overline{CE2B}$	O	Corresponds to PCMCIA card select signals D15 to D8 when t PCMCIA is used.
RD/$\overline{WR}$	O	Read/write Connects to $\overline{WE}$ pins when SDRAM or byte-selection SRAM is connected.
$\overline{RD}$	O	Read pulse signal (read data output enable signal) A strobe signal to indicate the memory read cycle when the PC used.

		Connected to the byte select signal when a byte-selection SF connected.
		Functions as the memory write strobe signal when the PCMC used.
$\overline{WE0(BE0)}$	O	Indicates that D7 to D0 are being written to.
		Connected to the byte select signal when a byte-selection SF connected.
$\overline{RAS}$	O	Connects to $\overline{RAS}$ pin when SDRAM is connected.
$\overline{CAS}$	O	Connects to $\overline{CAS}$ pin when SDRAM is connected.
CKE	O	Connects to CKE pin when SDRAM is connected.
$\overline{IOIS16}$	I	PCMCIA 16-bit I/O signal
		Valid only in little endian mode.
		Make it into low level at the time of big endian mode.
DQMUU	O	Connected to the DQMxx when the SDRAM is connected.
DQMUL		DQMUU: Selects D31 to D24
DQMLU		DQMUL: Selects D23 to D16
DQMLL		DQMLU: Selects D15 to D8
		DQMLL: Selects D7 to D0
$\overline{WAIT}$	I	External wait input
$\overline{BREQ}$	I	Bus request input
$\overline{BACK}$	O	Bus acknowledge output
MD5 to MD3	I	MD5: Selects data alignment (big endian or little endian)
		MD4 and MD3: Specify area 0 bus width (8/16/32 bits)
$\overline{REFOUT}$	O	Refresh request output when a bus is released

it outputs chip select signals ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A}$, $\overline{CS5B}$, $\overline{CS6A}$, and $\overline{CS6B}$) for eacl
$\overline{CS0}$ is asserted during area 0 access; $\overline{CS5A}$ is asserted during area 5A access when addre
is selected; and $\overline{CS5B}$ is asserted when address map 2 is selected.

12.3.2 Shadow Area

Areas 0, 2 to 4, 5A, 5B, 6A, and 6B are decoded by physical addresses A28 to A25, whic
correspond to areas 000 to 111. Address bits 31 to 29 are ignored. This means that the ran
area 0 addresses, for example, is H'00000000 to H'03FFFFFF, and its corresponding shad
is the address space in P1 to P3 areas obtained by adding to it H'20000000 × n (n = 1 to 6

The address range for area 7 is H'1C000000 to H'1FFFFFFF. The address space H'1C000
H'20000000 × n to H'1FFFFFFF + H'20000000 × n (n = 0 to 6) corresponding to the area
shadow space is reserved, so do not use it.

Area P4 (H'E0000000 to H'EFFFFFFF) is an I/O area and is assigned for internal register
addresses. Therefore, area P4 does not become shadow space.

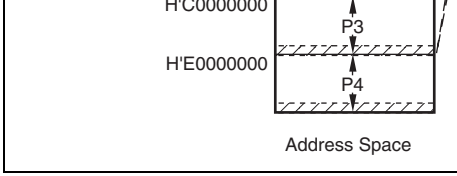


Figure 12.2 Address Space

		Burst ROM (Asynchronous)	
		Burst ROM (Synchronous)	
H′04000000 to H′07FFFFFF	Area 1	Internal I/O register area*[2]	64 Mbyte
H′08000000 to H′0BFFFFFF	Area 2	Normal memory*[3]	64 Mbyte
		Byte-selection SRAM	
		SDRAM	
H′0C000000 to H′0FFFFFFF	Area 3	Normal memory*[3]	64 Mbyte
		Byte-selection SRAM	
		SDRAM	
H′10000000 to H′13FFFFFF	Area 4	Normal memory*[3]	64 Mbyte
		Byte-selection SRAM	
		Burst ROM (Asynchronous)	
H′14000000 to H′15FFFFFF	Area 5A	Normal memory*[3]	32 Mbyte
H′16000000 to H′17FFFFFF	Area 5B	Normal memory*[3]	32 Mbyte
		Byte-selection SRAM	
H′18000000 to H′19FFFFFF	Area 6A	Normal memory*[3]	32 Mbyte
H′1A000000 to H′1BFFFFFF	Area 6B	Normal memory*[3]	32 Mbyte
		Byte-selection SRAM	
H′1C000000 to H′1FFFFFFF	Area 7	Reserved area*[1]	64 Mbyte

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.
2. Set the top three bits of the address to 101 to allocate in the P2 space.
3. Memory that has an interface such as SRAM.

H′0C000000 to H′0FFFFFFF	Area 3	Normal memory*[4] Byte-selection SRAM SDRAM	64 Mbyt
H′10000000 to H′13FFFFFF	Area 4	Normal memory*[4] Byte-selection SRAM Burst ROM (Asynchronous)	64 Mbyt
H′14000000 to H′17FFFFFF	Area 5*[2]	Normal memory*[4] Byte-selection SRAM PCMCIA	64 Mbyt
H′18000000 to H′1BFFFFFF	Area 6*[2]	Normal memory*[4] Byte-selection SRAM PCMCIA	64 Mbyt
H′1C000000 to H′1FFFFFFF	Area 7	Reserved area*[1]	64 Mbyt

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correc operation cannot be guaranteed.
2. For area 5, CS5BBCR and CS5BWCR are valid.
For area 6, CS6BBCR and CS6BWCR are valid.
3. Set the top three bits of the address to 101 to allocate in the P2 space.
4. Memory that has an interface such as SRAM.

0	0	Normal memory	Reserved (Setting prohibited)
	1		8 bits*
1	0		16 bits
	1		32 bits

Note: * The bus width must not be specified as eight bits if the burst ROM (clock synch interface is selected.

12.3.5 Data Alignment

This LSI supports the big endian and little endian methods of data alignment. The data al is specified using the external pin (MD5) at power-on reset as shown in table 12.5.

Table 12.5 Correspondence between External Pin (MD5) and Endians

MD5	Endian
0	Big endian
1	Little endian

- Bus control register for area 4 (CS4BCR)
- Bus control register for area 5A (CS5ABCR)
- Bus control register for area 5B (CS5BBCR)
- Bus control register for area 6A (CS6ABCR)
- Bus control register for area 6B (CS6BBCR)
- Wait control register for area 0 (CS0WCR)
- Wait control register for area 2 (CS2WCR)
- Wait control register for area 3 (CS3WCR)
- Wait control register for area 4 (CS4WCR)
- Wait control register for area 5A (CS5AWCR)
- Wait control register for area 5B (CS5BWCR)
- Wait control register for area 6A (CS6AWCR)
- Wait control register for area 6B (CS6BWCR)
- SDRAM control register (SDCR)
- Refresh timer control/status register (RTCSR)*[1]
- Refresh timer counter (RTCNT)*[1]
- Refresh time constant register (RTCOR)*[1]
- SDRAM mode register for area 2 (SDMR2)*[2]
- SDRAM mode register for area 3 (SDMR3)*[2]

Notes: 1. This register only accepts 32-bit writing to prevent incorrect writing. In this c
upper 16 bits of the data must be H'A55A. Otherwise, writing cannot be perf
reading, the upper 16 bits are read as H'0000.

2. The contents of this register are stored in SDRAM. When this register space
accessed, the corresponding register in SDRAM is written to. For details, see
description of Power-on Sequence in section 12.5.5, SDRAM Interface.

14	BSD	0	R/W	Bus Access Start Timing Specification After Bus Acknowledge Specifies the bus access start timing after the ext acknowledge signal is received. 0: Starts the external access at the same timing a address drive start after the bus acknowledge s received. 1: Starts the external access one cycle following t address drive start after the bus acknowledge s received.
13	—	0	R	Reserved This bit is always read as 0. The write value shou be 0.
12	MAP	0	R/W	Space Specification Selects the address map for the external address The address maps to be selected are shown in ta and 12.3. 0: Selects address map 1. 1: Selects address map 2.
11	BLOCK	0	R/W	Bus Lock Bit Specifies whether or not the $\overline{\text{BREQ}}$ signal is recei 0: Receives $\overline{\text{BREQ}}$. 1: Does not receive $\overline{\text{BREQ}}$.

				11: Reserved (Setting prohibited)
8	DMAIW2	0	R/W	Wait States between Access Cycles when DMA S
7	DMAIW1	0	R/W	Address is Transferred
6	DMAIW0	0	R/W	Specify the number of idle cycles to be inserted a access to an external device with DACK when DM address transfer is performed. The method of ins cycles depends on the contents of DMAIWA. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycled inserted 100: 6 idle cycled inserted 101: 8 idle cycle inserted 110: 10 idle cycles inserted 111: 12 idle cycled inserted
5	DMAIWA	0	R/W	Method of Inserting Wait States between Access when DMA Single Address is Transferred Specifies the method of inserting the idle cycles s by the DMAIW1 and DMAIW0 bits. Clearing this b make this LSI insert the idle cycles when another which includes this LSI, drives the data bus after external device with DACK drove it. When the ext device with DACK drives the data bus continuous cycles are not inserted. Setting this bit will make t insert the idle cycles even when the continuous a to an external device with DACK are performed.

				big endian. 1: The external pin for specifying endian (MD5) wa level on power-on reset. This LSI is being opera little endian.
2	CK2DRV	0	R/W	CKIO2 Drive Specifies whether the CKIO2 pin outputs a low lev or clock (Bϕ). 0: Outputs a low level signal 1: Outputs a clock (Bϕ)
1	HIZMEM	0	R/W	High-Z Memory Control Specifies the pin state in standby mode for A25 to $\overline{CSn}$, RD/$\overline{WR}$, $\overline{WEn}$ ($\overline{BEn}$)/DQMxx, and $\overline{RD}$. When released, these pins enter the high-impedance sta regardless of the setting of this bit. 0: High impedance in standby mode 1: Driven in standby mode

		standby mode and bus released.

Note: * The external pin (MD5) for specifying endian is sampled on power-on reset. W
endian is specified, this bit is read as 0 and when little endian is specified, this
read as 1.

12.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B

CSnBCR specifies the type of memory connected to each space, data-bus width of each
the number of wait cycles between access cycles.

Do not access external memory other than area 0 until the CSnBCR initialization is com

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value shou be 0.

				011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
27	IWRWD2	0	R/W	Idle Cycles for Another Space Read-Write
26	IWRWD1	1	R/W	Specify the number of idle cycles to be inserted aft
25	IWRWD0	1	R/W	access to a memory that is connected to the space target access cycle is a read-write one in which co accesses switch between different spaces. 000: No idle cycle inserted 001: 1 idle cycles inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

				100: 6 idle cycles inserted
				101: 8 idle cycles inserted
				110: 10 idle cycles inserted
				111: 12 idle cycles inserted
21	IWRRD2	0	R/W	Idle Cycles for Read-Read in Another Space
20	IWRRD1	1	R/W	Specify the number of idle cycles to be inserted a
19	IWRRD0	1	R/W	access to a memory that is connected to the spac target cycle is a read-read cycle of which continuo accesses switch between different space.
				000: No idle cycle inserted
				001: 1 idle cycles inserted
				010: 2 idle cycles inserted
				011: 4 idle cycles inserted
				100: 6 idle cycles inserted
				101: 8 idle cycles inserted
				110: 10 idle cycles inserted
				111: 12 idle cycles inserted

011: 4 idle cycles inserted
100: 6 idle cycles inserted
101: 8 idle cycles inserted
110: 10 idle cycles inserted
111: 12 idle cycles inserted

0110: Reserved (setting prohibited)

0111: Burst ROM (clock synchronous)*2

1000: Reserved (setting prohibited)

1001: Reserved (setting prohibited)

1010: Reserved (setting prohibited)

1011: Reserved (setting prohibited)

1100: Reserved (setting prohibited)

1101: Reserved (setting prohibited)

1110: Reserved (setting prohibited)

1111: Reserved (setting prohibited)

Note: Memory type for area 0 immediately after re
normal space. The normal space, burst ROM
asynchronous), or burst ROM (clock synchro
can be selected by these bits.

For details on memory type in each area, see table
and 12.3.

				11: 32-bit size Notes: 1. The data bus width for area 0 is specif the external pin. The BSZ1 and BSZ0 settings in CS0BCR are ignored. 2. If area 5 or area 6 is specified as PCM space, the bus width can be specified 8 bits or 16 bits. 3. If area 2 or area 3 is specified as SDR space, the bus width can be specified 16 bits or 32 bits.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value sl always be 0.

Notes: 1. CS0BCR samples the external pins (MD3 and MD4) that specify the bus width power-on reset.
2. The burst ROM (clock synchronous) must be accessed as a cacheable space.

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 s always be 0.
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{WEn}$ ($\overline{BEn}$) and RD/$\overline{WR}$ signal timin the byte-selection SRAM interface is used. 0: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal at the read/write and asserts the RD/$\overline{WR}$ signal during the write cycle. 1: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal during the read/ access cycle and asserts the RD/$\overline{WR}$ signal at timing.
19 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Asse
11	SW0	0	R/W	$\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion Specify the number of delay cycles from address assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is v The specification by this bit is valid even when the of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.

- CS2WCR, CS3WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) and RD/$\overline{\text{WR}}$ signal timin the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal at the read/write and asserts the RD/$\overline{\text{WR}}$ signal during the write cycle. 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/w access cycle and asserts the RD/$\overline{\text{WR}}$ signal at timing.
19 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.

				0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is val specification by this bit is valid even when the numb access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value sh always be 0.

				and asserts the RD/$\overline{WR}$ signal during the write cycle. 1: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal during the read/w access cycle and asserts the RD/$\overline{WR}$ signal at timing.
19	—	0	R	Reserved This bit is always read as 0. The write value shoul be 0.
18	WW2	0	R/W	Number of Write Access Wait Cycles
17	WW1	0	R/W	Specify the number of cycles that are necessary f
16	WW0	0	R/W	access. 000: The same cycles as WR3 to WR0 setting (re wait) 001: 0 cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.

10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessa
8	WR1	1	R/W	read/write access.
7	WR0	0	R/W	0000: 0 cycle
				0001: 1 cycle
				0010: 2 cycles
				0011: 3 cycles
				0100: 4 cycles
				0101: 5 cycles
				0110: 6 cycles
				0111: 8 cycles
				1000: 10 cycles
				1001: 12 cycles
				1010: 14 cycles
				1011: 18 cycles
				1100: 24 cycles
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification
				Specifies whether or not the external wait input is v specification by this bit is valid even when the num access wait cycles is 0.
				0: External wait is valid
				1: External wait is ignored

				10: 2.5 cycles 11: 3.5 cycles

- CS5AWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
18	WW2	0	R/W	Number of Write Access Wait Cycles
17	WW1	0	R/W	Specify the number of cycles that are necessary f
16	WW0	0	R/W	access. 000: The same cycles as WR3 to WR0 setting (re wait) 001: 0 cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.

9	WR2	0	R/W	Specify the number of wait cycles that are necessa
8	WR1	1	R/W	read/write access.
7	WR0	0	R/W	0000: 0 cycle
				0001: 1 cycle
				0010: 2 cycles
				0011: 3 cycles
				0100: 4 cycles
				0101: 5 cycles
				0110: 6 cycles
				0111: 8 cycles
				1000: 10 cycles
				1001: 12 cycles
				1010: 14 cycles
				1011: 18 cycles
				1100: 24 cycles
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification
				Specify whether or not the external wait input is va specification by this bit is valid even when the num access wait cycle is 0.
				0: External wait is valid
				1: External wait is ignored

				10: 2.5 cycles 11: 3.5 cycles

- CS5BWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) and RD/$\overline{\text{WR}}$ signal timin the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal at the read/write and asserts the RD/$\overline{\text{WR}}$ signal during the write cycle. 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/ access cycle and asserts the RD/$\overline{\text{WR}}$ signal at timing.
19	—	0	R	Reserved This bit is always read as 0. The write value shou be 0.

				101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Asse
11	SW0	0	R/W	$\overline{\text{RD}}$, $\overline{\text{WE}}$n ($\overline{\text{BE}}$n) Assertion Specify the number of delay cycles from address a assertion to $\overline{\text{RD}}$ and $\overline{\text{WE}}$n ($\overline{\text{BE}}$n) assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is v specification by this bit is valid even when the nu access wait cycles is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
1 0	HW1 HW0	0 0	R/W R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) ne Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessa
8	WR1	1	R/W	read/write access.
7	WR0	0	R/W	0000: 0 cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

1	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) neg
0	HW0	0	R/W	Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{W}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				or 16-burst access for an 8-bit bus width during 16 access. If this bit is set to 1, 2-burst access is perf four times when the bus width is 16 bits and 4-bur access is performed four times when the bus widt bits. To use a device that does not support 8-burst acc 16-burst access, set this bit to 1. 0: Enables 8-burst access for a 16-bit bus width a burst access for an 8-bit bus width. 1: Disables 8-burst access for a 16-bit bus width a burst access for an 8-bit bus width.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted b the second or later access cycles in burst access. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.

				0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is v specification by this bit is valid even when the nu access wait cycles is 0. 0: External wait is valid 1: External wait is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.

				access is performed four times when the bus widt bits. To use a device that does not support 8-burst acc 16-burst access, set this bit to 1. 0: Enables 8-burst access for a 16-bit bus width a burst access for an 8-bit bus width. 1: Disables 8-burst access for a 16-bit bus width a burst access for an 8-bit bus width.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted b the second or later access cycles in burst access. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.

				11: 3.5 cycles
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted i
8	W1	1	R/W	read/write access cycle.
7	W0	0	R/W	0000: 0 cycle
				0001: 1 cycle
				0010: 2 cycles
				0011: 3 cycles
				0100: 4 cycles
				0101: 5 cycles
				0110: 6 cycles
				0111: 8 cycles
				1000: 10 cycles
				1001: 12 cycles
				1010: 14 cycles
				1011: 18 cycles
				1100: 24 cycles
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification
				Specifies whether or not the external wait input is specification by this bit is valid even when the nu access wait cycles is 0.
				0: External wait is valid
				1: External wait is ignored

				01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

SDRAM*:

- CS2WCR

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 s always be 0.
10	—	1	R	Reserved These bits are always read as 1. The write value s always be 1.
9	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.
8	A2CL1	1	R/W	CAS Latency for Area 2
7	A2CL0	0	R/W	Specify the CAS latency for area 2. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles

15				These bits are always read as 0. The write value always be 0.
14	TRP1	0	R/W	Number of Wait Cycles Waiting Completion of Pr
13	TRP0	0	R/W	Specify the number of minimum wait cycles to be to wait the completion of precharge. The setting f and 3 is common. (1) From starting auto-charge to issuing the ACTV command for the same bank (2) From issuing the PRE/PALL command to issu ACTV command for the same bank (3) To transiting to power-down mode/deep powe mode (4) From issuing the PALL command at auto-refre issuing the REF command (5) From issuing the PALL command at self-refres issuing the SELF command 00: 0 cycle 01: 1 cycles 10: 2 cycles 11: 3 cycles
12	—	0	R	Reserved This bit is always read as 0. The write value shou be 0.

				11: 8 cycles
9	—	0	R	Reserved This bit is always read as 0. The write value shoul be 0.
8	A3CL1	1	R/W	CAS Latency for Area 3.
7	A3CL0	0	R/W	Specify the CAS latency for area 3. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.

				reception of the WRITA command to the start precharge in each SDRAM data sheet. Set this bit so that the number of cycles is not cycles specified by this bit. (2) This LSI is in bank active mode from issuing th command to issuing the PRE command, and t to different row address in the same bank is p 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
2	—	0	R	Reserved This bit is always read as 0. The write value shou be 0.

		00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles

Note: * If both areas 2 and 3 are specified as SDRAM, TRP1/0, TRCD0/1, TRWL1/0, a
TRC1/0 bit settings are common. If only one area is connected to the SDRAM,
area 3. In this case, specify area 2 as normal space or byte-selection SRAM.

				the PCMCIA interface is selected. SA1 0: Specifies memory card interface when A25 = 1 1: Specifies I/O card interface when A25 = 1 SA0 0: Specifies memory card interface when A25 = 0 1: Specifies I/O card interface when A25 = 0
19 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.

0110: 6.5 cycles
0111: 7.5 cycles
1000: 8.5 cycles
1001: 9.5 cycles
1010: 10.5 cycles
1011: 11.5 cycles
1100: 12.5 cycles
1101: 13.5 cycles
1110: 14.5 cycles
1111: 15.5 cycles

				0111: 26 cycles 1000: 30 cycles 1001: 33 cycles 1010: 36 cycles 1011: 38 cycles 1100: 52 cycles 1101: 60 cycles 1110: 64 cycles 1111: 80 cycles
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is v specification by this bit is valid even when the nu access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.

				0110: 6.5 cycles 0111: 7.5 cycles 1000: 8.5 cycles 1001: 9.5 cycles 1010: 10.5 cycles 1011: 11.5 cycles 1100: 12.5 cycles 1101: 13.5 cycles 1110: 14.5 cycles 1111: 15.5 cycles

Burst ROM (Clock Synchronous):

- CS0WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value sl always be 0.
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted be the second or later access cycles in burst access. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

				0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is v specification by this bit is valid even when the nu access wait cycles is 0. 0: External wait is valid 1: External wait is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.

20	A2ROW1	0	R/W	Number of Bits of Row Address for Area 2
19	A2ROW0	0	R/W	Specify the number of bits of row address for area 2 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
18	—	0	R	Reserved This bit is always read as 0. The write value should be 0.
17	A2COL1	0	R/W	Number of Bits of Column Address for Area 2
16	A2COL0	0	R/W	Specify the number of bits of column address for ar 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value sh always be 0.
13	DEEP	0	R/W	Deep Power-Down Mode This bit is valid for low-power SDRAM. If the RMOD set to 1 while this bit is set to 1, the deep power-do command is issued and the low-power SDRAM ent deep power-down mode. 0: Self-refresh mode 1: Deep power-down mode

				at the rising edge of CKIO. Read data from SDR latched at the rising edge of CKIO. 1: Command, address, and write data for SDRAM at the falling edge of CKIO. Read data from SDF latched at the falling edge of CKIO.
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refresh operation of th SDRAM is performed. 0: No refresh 1: Refresh
10	RMODE	0	R/W	Refresh Control Specifies whether to perform auto-refresh or self-r when the RFSH bit is 1. When the RFSH bit is 1 a is 1, self-refresh starts immediately. When the RFS and this bit is 0, auto-refresh starts according to th contents that are set in RTCSR, RTCNT, and RTC 0: Auto-refresh is performed 1: Self-refresh is performed
9	PDOWN	0	R/W	Power-Down Mode Specify whether SDRAM is put in power-down mo after the access to memory other than SDRAM is completed. This bit, when set to 1, drives the CKE and places SDRAM in power-down mode by using access to a memory other than SDRAM as a trigg 0: Does not place SDRAM in power-down mode a access to a memory other than SDRAM. 1: Places SDRAM in power-down mode after an a memory other than SDRAM.

				When both areas 2 and 3 are set to SDRAM auto-precharge mode.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
4	A3ROW1	0	R/W	Number of Bits of Row Address for Area 3
3	A3ROW0	0	R/W	Specify the number of bits of the row address for a 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
2	—	0	R	Reserved This bit is always read as 0. The write value shoul be 0.
1	A3COL1	0	R/W	Number of Bits of Column Address for Area 3
0	A3COL0	0	R/W	Specify the number of bits of the column address 3. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

				always be 0.
7	CMF	0	R/W	Compare Match Flag Indicates that a compare match occurs between t timer counter (RTCNT) and refresh time constant (RTCOR). This bit is set or cleared in the followin conditions. 0: Clearing condition: When 0 is written in CMF a reading out RTCSR during CMF = 1. 1: Setting condition: When the condition RTCNT = is satisfied.
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables a CMF interrupt request whe bit of RTCSR is set to 1. 0: Disables the CMF interrupt request 1: Enables the CMF interrupt request
5	CKS2	0	R/W	Clock Select
4	CKS1	0	R/W	Select the clock input to count-up the refresh time
3	CKS0	0	R/W	(RTCNT). 000: Stop the counting-up 001: Bϕ/4 010: Bϕ/16 011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/4096

011: 6 times

100: 8 times

101: Reserved (setting prohibited)

110: Reserved (setting prohibited)

111: Reserved (setting prohibited)

12.4.6 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS2 to CKS
RTCSR. When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT r
0 after counting up to 255. When the RTCNT is written, the upper 16 bits of the write dat
be H'A55A to cancel write protection.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
7 to 0	—	All 0	R/W	8-bit Counter

affects only interrupts and does not affect refresh requests. This makes it possible to cou
number of refresh requests during refresh by interrupts, and to specify the refresh and in
timer interrupts simultaneously. When the RTCOR is written, the upper 16 bits of the w
must be H'A55A to cancel write protection.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
7 to 0	—	All 0	R/W	8-bit Counter

Three data bus widths (8 bits, 16 bits, and 32 bits) are available for normal memory and t
selection SRAM. Two data bus widths (16 bits and 32 bits) are available for SDRAM. Tw
bus widths (8 bits and 16 bits) are available for PCMCIA interface. Data alignment is per
in accordance with the data bus width of the device and endian. This also means that whe
longword data is read from a byte-width device, the read operation must be done four tim
this LSI, data alignment and conversion of data length is performed automatically betwee
respective interfaces.

Tables 12.6 to 12.11 show the relationship between endian, device data width, and access

Table 12.6 32-Bit External Device/Big Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WE3}}$(BE3), DQMUU	$\overline{\text{WE2}}$(BE2), DQMUL	$\overline{\text{WE1}}$(BE1), DQMLU	$\overline{\text{W}}$ D
Byte access at 0	Data 7 to 0	—	—	—	Assert	—	—	–
Byte access at 1	—	Data 7 to 0	—	—	—	Assert	—	–
Byte access at 2	—	—	Data 7 to 0	—	—	—	Assert	–
Byte access at 3	—	—	—	Data 7 to 0	—	—	—	A
Word access at 0	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—	–
Word access at 2	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	A
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	A

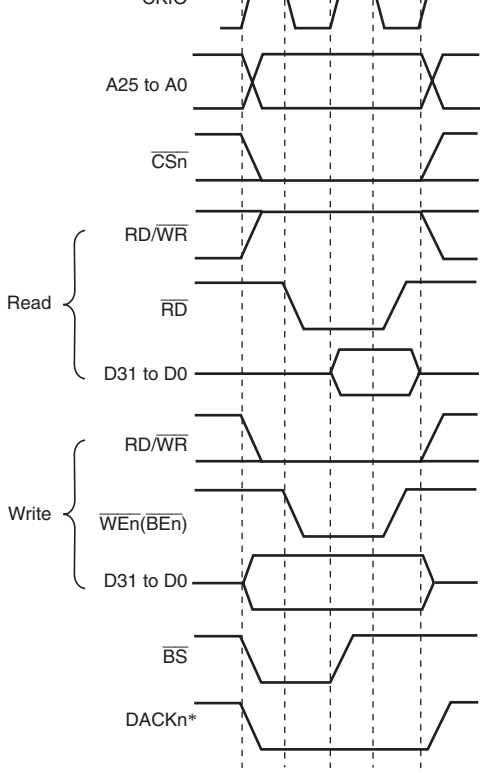
Byte access at 3		—	—	—	Data 7 to 0	—	—	—
Word access at 0		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert
Word access at 2		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert
Longword access at 0	1st time at 0	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert
	2nd time at 2	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert

Byte access at 3		—	—	—	Data 7 to 0	—	—	—	A
Word access at 0	1st time at 0	—	—	—	Data 15 to 8	—	—	—	A
	2nd time at 1	—	—	—	Data 7 to 0	—	—	—	A
Word access at 2	1st time at 2	—	—	—	Data 15 to 8	—	—	—	A
	2nd time at 3	—	—	—	Data 7 to 0	—	—	—	A
Longword access at 0	1st time at 0	—	—	—	Data 31 to 24	—	—	—	A
	2nd time at 1	—	—	—	Data 23 to 16	—	—	—	A
	3rd time at 2	—	—	—	Data 15 to 8	—	—	—	A
	4th time at 3	—	—	—	Data 7 to 0	—	—	—	A

Byte access at 3	Data 7 to 0	—	—	—	Assert	—	—	
Word access at 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	
Word access at 2	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—	
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	

Byte access at 3		—	—	Data 7 to 0	—	—	—	Assert	–
Word access at 0		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	A
Word access at 2		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	A
Longword access at 0	1st time at 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	A
	2nd time at 2	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert	A

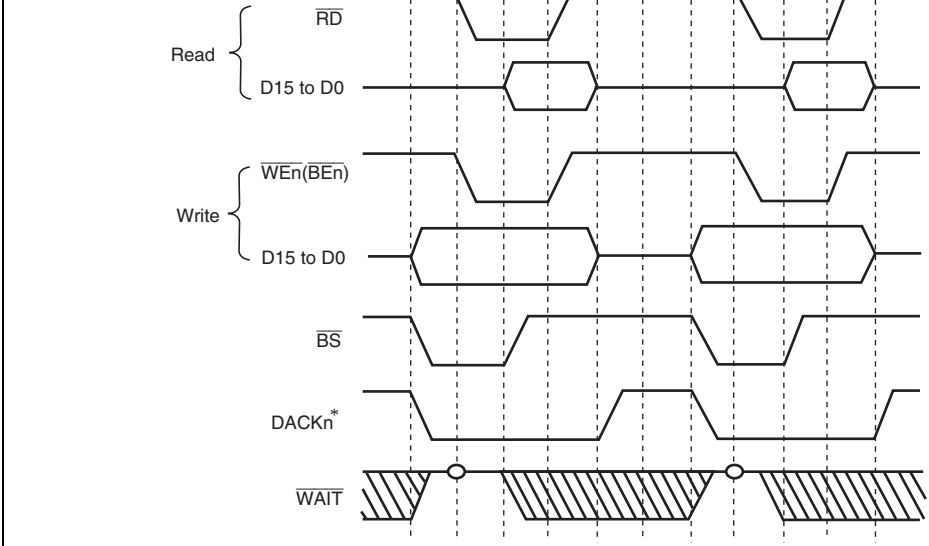
Byte access at 3		—	—	—	Data 7 to 0	—	—	—
Word access at 0	1st time at 0	—	—	—	Data 7 to 0	—	—	—
	2nd time at 1	—	—	—	Data 15 to 8	—	—	—
Word access at 2	1st time at 2	—	—	—	Data 7 to 0	—	—	—
	2nd time at 3	—	—	—	Data 15 to 8	—	—	—
Longword access at 0	1st time at 0	—	—	—	Data 7 to 0	—	—	—
	2nd time at 1	—	—	—	Data 15 to 8	—	—	—
	3rd time at 2	—	—	—	Data 23 to 16	—	—	—
	4th time at 3	—	—	—	Data 31 to 24	—	—	—



Note: * The waveform for DACKn is when active low is specified.

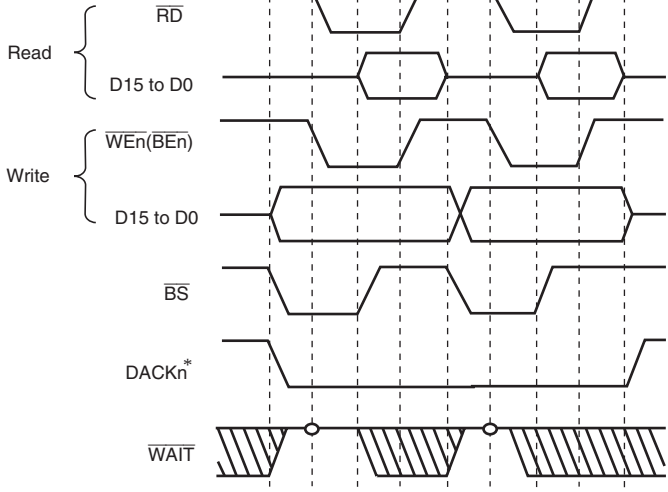
Figure 12.3 Normal Space Basic Access Timing (Access Wait 0)

There is no access size specification when reading. The correct access start address is out
least significant bit of the address, but since there is no access size specification, 32 bits a



Note: * The waveform for DACKn is when active low is specified.

Figure 12.4 Continuous Access for Normal Space 1, Bus Width = 16 bits, Longwor
CSnWCR.WM Bit = 0 (Access Wait = 0, Cycle Wait = 0)



Note: * The waveform for DACKn is when active low is specified.

Figure 12.5 Continuous Access for Normal Space 2, Bus Width = 16 bits, Longwo CSnWCR.WM Bit = 1 (Access Wait = 0, Cycle Wait = 0)

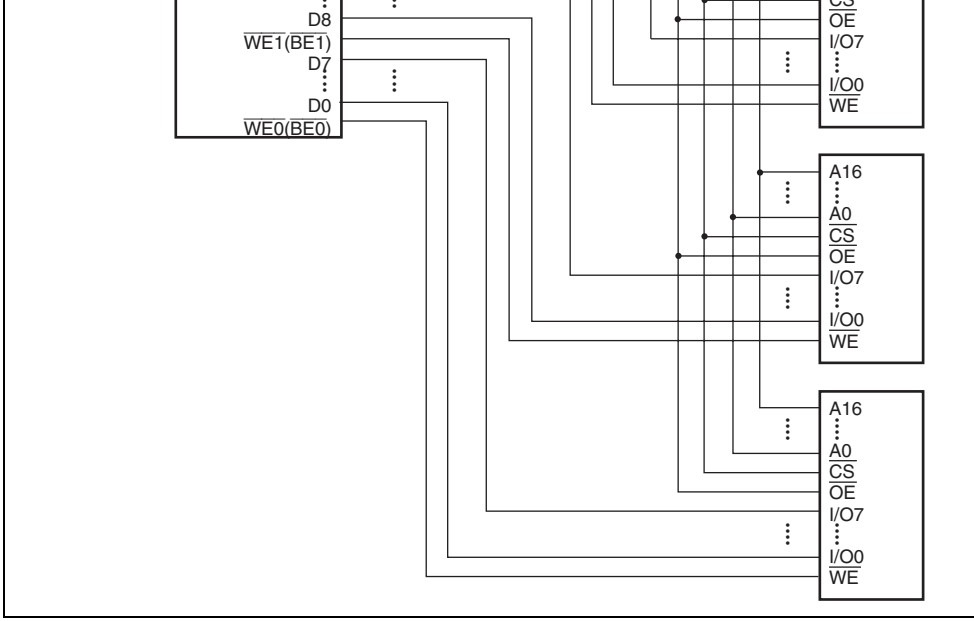


Figure 12.6 Example of 32-Bit Data-Width SRAM Connection

Figure 12.7 Example of 16-Bit Data-Width SRAM Connection

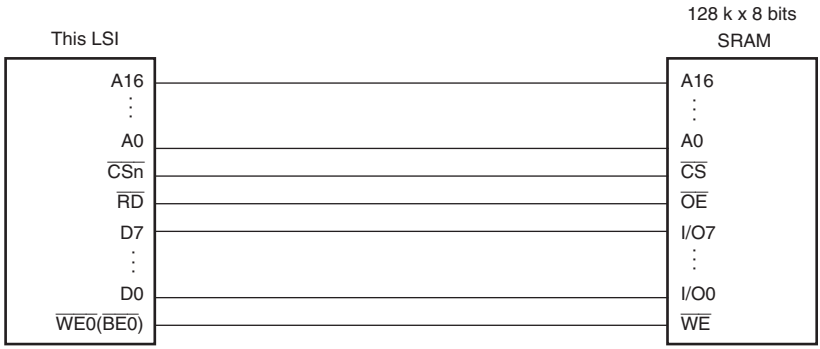
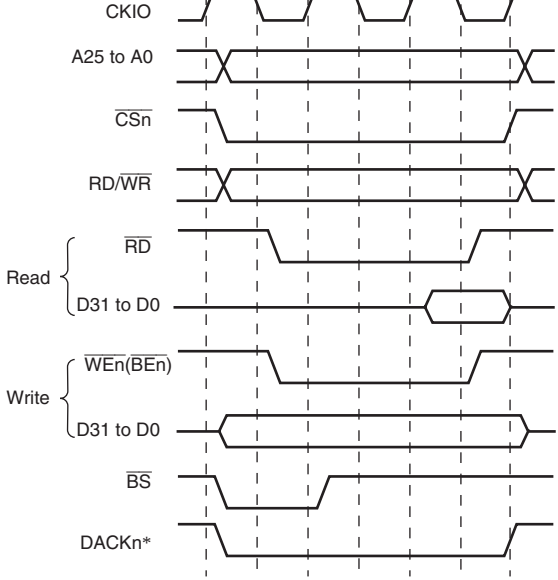


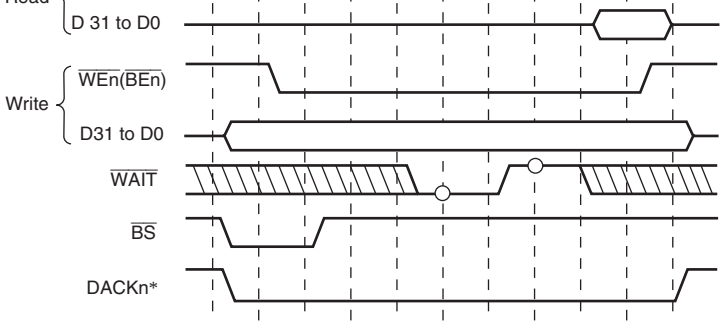
Figure 12.8 Example of 8-Bit Data-Width SRAM Connection



Note: * The waveform for DACKn is when active low is specified.

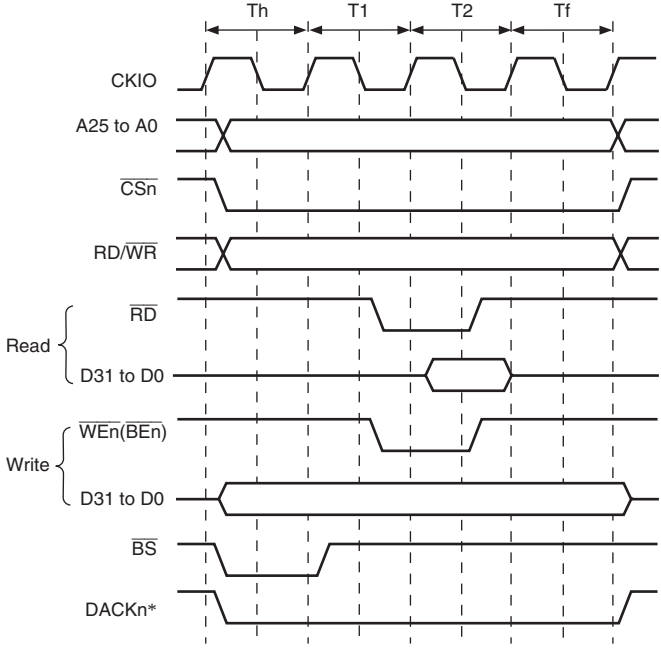
Figure 12.9 Wait Timing for Normal Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait input WAIT signal is als sampled. WAIT pin sampling is shown in figure 12.10. A 2-cycle wait is specified as a so wait. The WAIT signal is sampled on the falling edge of CKIO at the transition from the cycle to the T2 cycle.



Note: * The waveform for DACKn is when active low is specified.

Figure 12.10 Wait State Timing for Normal Space Access (Wait State Insertion b Signal)



Note: * The waveform for DACKn is when active low is specified.

Figure 12.11 $\overline{\text{CSn}}$ Assert Period Expansion

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are sup
the SDRAM operating mode.

Commands for SDRAM can be specified by $\overline{RAS}$, $\overline{CAS}$, RD/$\overline{WR}$, and specific address s
These commands are shown below.

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks precharge (PALL)
- Specified bank precharge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with precharge (READA)
- Write (WRIT)
- Write with precharge (WRITA)
- Write mode register (MRS)

The byte to be accessed is specified by DQMUU, DQMUL, DQMLU, and DQMLL. Re
writing is performed for a byte whose corresponding DQMxx is low. For details on the
relationship between DQMxx and the byte to be accessed, refer to section 12.5.1, Endian
Size and Data Alignment.

Figures 12.12 and 12.13 show examples of the connection of the SDRAM with the LSI.

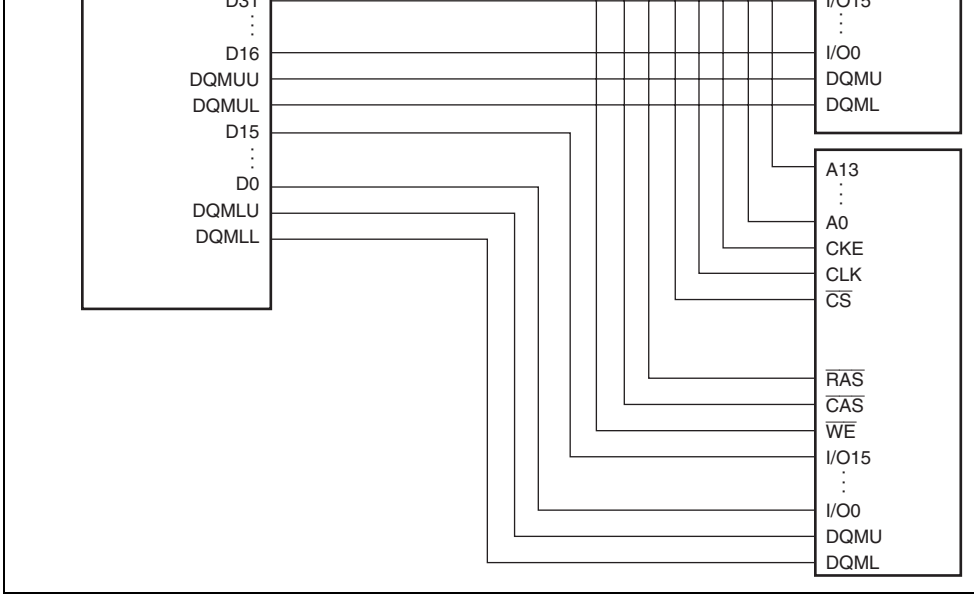


Figure 12.12 Example of 32-Bit Data-Width SDRAM Connection

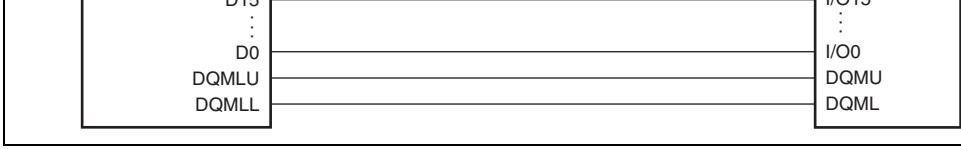


Figure 12.13 Example of 16-Bit Data-Width SDRAM Connection

Address Multiplexing: An address multiplexing is specified so that SDRAM can be co without external multiplexing circuitry according to the setting of bits BSZ[1:0]in CSnB AxROW[1:0] and AxCOL[1:0] in SDCR. Tables 12.12 to 12.17 show the relationship b settings of bits BSZ[1:0], AxROW[1:0], and AxCOL[1:0] and the bits output at the addr Do not specify those bits in the manner other than this table, otherwise the operation of t not guaranteed. A25 to A18 are not multiplexed and the original values of address are al output at these pins.

When the data bus width is 16 bits (BSZ[1:0] =B'10), A0 of SDRAM specifies a word a Therefore, connect this A0 pin of SDRAM to the A1 pin of the LSI; the A1 pin of SDRA A2 pin of the LSI, and so on. When the data bus width is 32 bits (BSZ[1:0] =B'11), the SDRAM specifies a longword address. Therefore, connect this A0 pin of SDRAM to the the LSI; the A1 pin of SDRAM to the A3 pin of the LSI, and so on.

A16	A24	A16		
A15	A23	A15		
A14	A22*[2]	A22*[2]	A12 (BA1)*[3]	Specifies bank
A13	A21*[2]	A21*[2]	A11 (BA0)	
A12	A20	L/H*[1]	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	

64-Mbit product (512 kwords x 32 bits x 4 banks, column 8 bits product): 1

16-Mbit product (512 kwords x 16 bits x 2 banks, column 8 bits product): 2

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accord
access mode.

2. Bank address specification

3. If the number of 16-Mbit SDRAM (512 kwords × 16 bits × 2 banks: pin with 8-
column) is two, the bank address specification is not required. Therefore, the
address should be not used.

A16	A23	A16		
A15	A23*[2]	A23*[2]	A13 (BA1)	Specifies bank
A14	A22*[2]	A22*[2]	A12 (BA0)	
A13	A21	A13	A11	Address
A12	A20	L/H*[1]	A10/AP	Specifies address/precha
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
128-Mbit product (1 Mword x 32 bits x 4 banks, column 8 bits product): 1				
64-Mbit product (1 Mword x 16 bits x 4 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accordi access mode.

2. Bank address specification

A16	A25	A16		
A15	A24*[2]	A24*[2]	A13 (BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A12 (BA0)	
A13	A22	A13	A11	Address
A12	A21	L/H*[1]	A10/AP	Specifies address/prech
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
256-Mbit product (2 Mwords x 32 bits x 4 banks, column 9 bits product): 1				
128-Mbit product (2 Mwords x 16 bits x 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accord access mode.

2. Bank address specification

A16	A26	A16		
A15	A25*[2]	A25*[2]	A13 (BA1)	Specifies bank
A14	A24*[2]	A24*[2]	A12 (BA0)	
A13	A23	A13	A11	Address
A12	A22	L/H*[1]	A10/AP	Specifies address/precha
A11	A21	A11	A9	Address
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	

Example of connected memory
512-Mbit product (4 Mwords x 32 bits x 4 banks, column 10 bits product): 1
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 10 bits product): 2

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accord access mode.
2. Bank address specification

A16	A25*[2]	A25*[2]	A14 (BA1)	Specifies bank
A15	A24*[2]	A24*[2]	A13 (BA0)	
A14	A23	A14	A12	Address
A13	A22	A13	A11	
A12	A21	L/H*[1]	A10/AP	Specifies address/precha
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
512-Mbit product (4 Mwords x 32 bits x 4 banks, column 9 bits product): 1				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accordi access mode.

2. Bank address specification

A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21	A21		
A12	A20*[2]	A20*[2]	A11 (BA0)	Specifies bank
A11	A19	L/H*[1]	A10/AP	Specifies address/precha
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
		Example of connected memory		
		16-Mbit product (512 kwords x 16 bits x 2 banks, column 8 bits product): 1		

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accord access mode.

2. Bank address specification

A16	A24	A16		
A15	A23	A15		
A14	A22*[2]	A22*[2]	A13 (BA1)	Specifies bank
A13	A21*[2]	A21*[2]	A12 (BA0)	
A12	A20	A12	A11	Address
A11	A19	L/H*[1]	A10/AP	Specifies address/precha
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (1 Mword x 16 bits x 4 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accordi access mode.

2. Bank address specification

A16	A25	A16		
A15	A24	A15		
A14	A23*[2]	A23*[2]	A13 (BA1)	Specifies bank
A13	A22*[2]	A22*[2]	A12 (BA0)	
A12	A21	A12	A11	Address
A11	A20	L/H*[1]	A10/AP	Specifies address/precha
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
128-Mbit product (2 Mwords x 16 bits x 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accord access mode.

2. Bank address specification

A16	A26	A16		
A15	A25	A15		
A14	A24*[2]	A24*[2]	A13 (BA1)	Specifies bank
A13	A23*[2]	A23*[2]	A12 (BA0)	
A12	A22	A12	A11	Address
A11	A21	L/H*[1]	A10/AP	Specifies address/precha
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accordi access mode.

2. Bank address specification

A16	A25	A16		
A15	A24*[2]	A24*[2]	A14 (BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20	L/H*[1]	A10/AP	Specifies address/precha
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accord access mode.

2. Bank address specification

A16	A26	A16		
A15	A25*[2]	A25*[2]	A14 (BA1)	Specifies bank
A14	A24*[2]	A24*[2]	A13 (BA0)	
A13	A23	A13	A12	Address
A12	A22	A12	A11	
A11	A21	L/H*[1]	A10/AP	Specifies address/precha
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
512-Mbit product (8 Mwords x 16 bits x 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high accordi access mode.

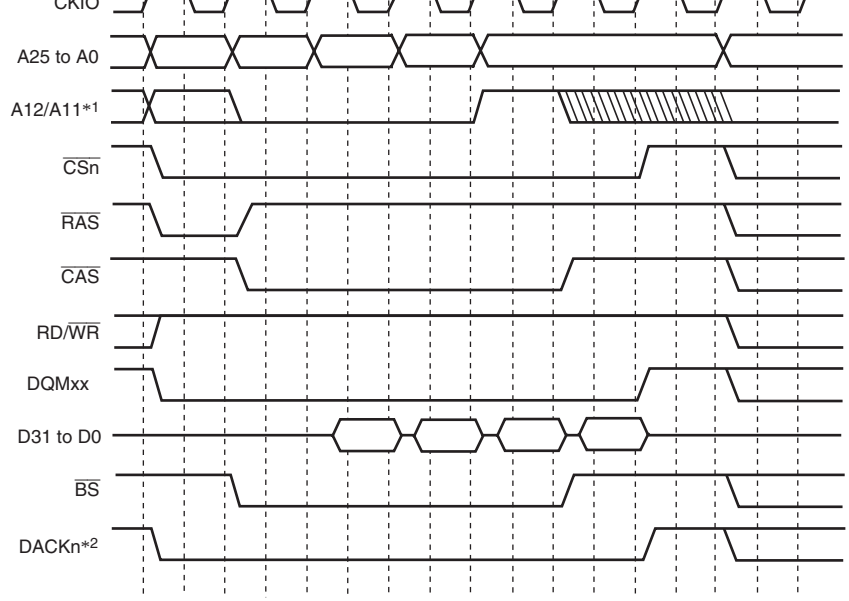
2. Bank address specification

Table 12.18 Relationship between Access Size and Number of Bursts

Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bytes	8
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	16 bytes	4

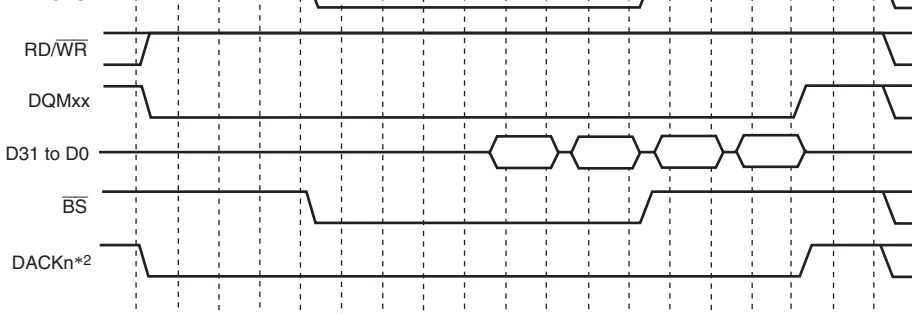
Figures 12.14 and 12.15 show a timing chart in burst read. In burst read, an ACTV comm output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, th command is issued in the Tc4 cycle, and the read data is received at the rising edge of th clock (CKIO) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion auto-precharge induced by the READA command in the SDRAM. In the Tap cycle, a ne command will not be issued to the same bank. However, access to another CS space or a bank in the same SDRAM space is enabled. The number of Tap cycles is specified by th and TRP0 bits in CS3WCR.

In this LSI, wait cycles can be inserted by specifying each bit in CSnWCR to connect th in variable frequencies. Figure 12.15 shows an example in which wait cycles are inserte number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cyc the READA command is output can be specified using the TRCD1 and TRCD0 bits in C If the TRCD1 and TRCD0 bits specify one cycle or more, a Trw cycle where the NOP c is issued is inserted between the Tr cycle and Tc1 cycle. The number of cycles from the



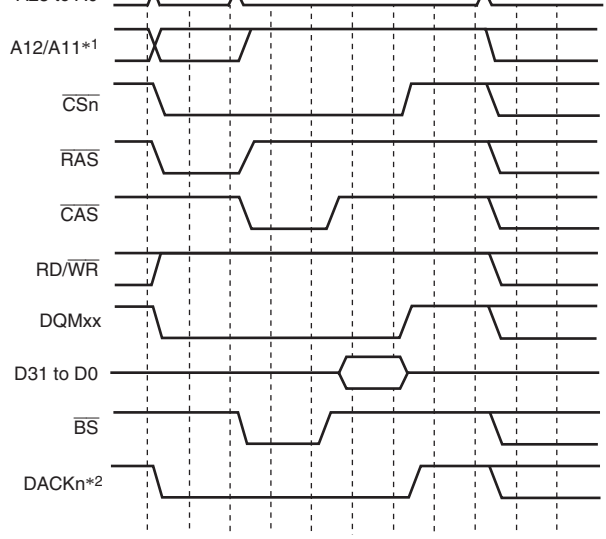
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.14 Burst Read Basic Timing (Auto Precharge)



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

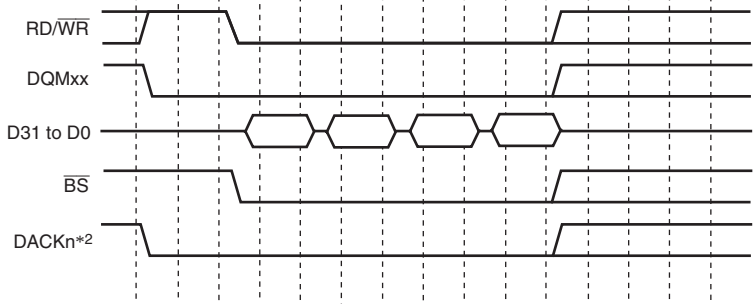
Figure 12.15 Burst Read Wait Specification Timing (Auto Precharge)



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

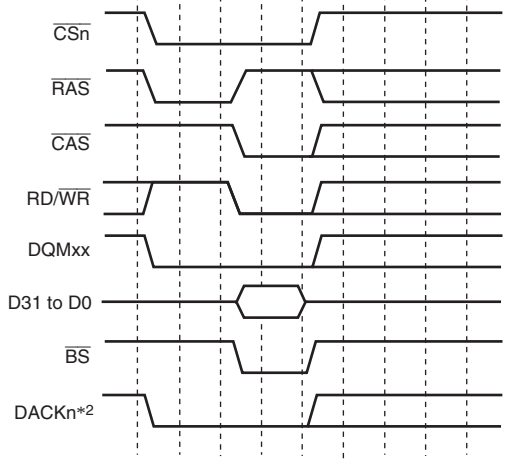
Figure 12.16 Basic Timing for Single Read (Auto Precharge)

Figure 12.17 shows a timing chart for burst writes. In burst write, an ACTV command is
the Tr cycle, the WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and the WR
command is issued to execute an auto-precharge in the Tc4 cycle. In the write cycle, the
is output simultaneously with the write command. After the write command with the aut
precharge is output, the Trw1 cycle that waits for the auto-precharge initiation is followe
Tap cycle that waits for completion of the auto-precharge induced by the WRITA comm
SDRAM. In the Tap cycle, a new command will not be issued to the same bank. Howev
to another CS space or another bank in the same SDRAM space is enabled. The number
cycles is specified by the TRWL1 and TRWL0 bits in CS3WCR. The number of Tap cy
specified by the TRP1 and TRP0 bits in CS3WCR.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.17 Basic Timing for Burst Write (Auto Precharge)



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.18 Basic Timing for Single Write (Auto-Precharge)

Bank Active: The synchronous DRAM bank function is used to support high-speed acc the same row address. When the BACTV bit in SDCR is 1, accesses are performed usin commands without auto-precharge (READ or WRIT). This function is called bank-activ This function is valid only for either the upper or lower bits of area 3. When area 3 is se active mode, area 2 should be set to normal space or byte-selection SRAM. When areas are both set to SDRAM, auto precharge mode must be set.

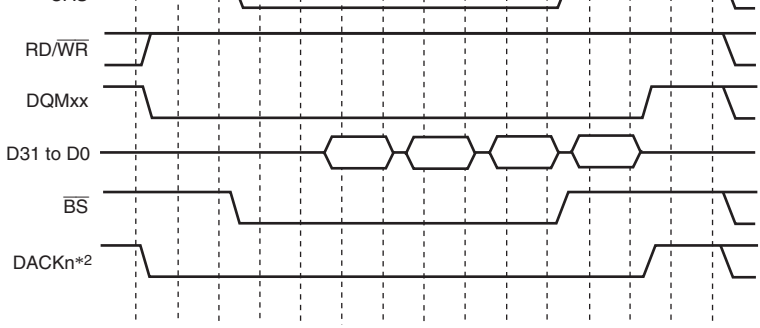
When a bank-active function is used, precharging is not performed when the access end accessing the same row address in the same bank, it is possible to issue the READ or W command immediately, without issuing an ACTV command. As synchronous DRAM is divided into several banks, it is possible to activate one row address in each bank. If the access is to a different row address, a PRE command is first issued to precharge the rele

There is a limit on tRAS, the time for placing each bank in the active state. If there is no g
that there will not be a cache hit and another row address will be accessed within the peri
which this value is maintained by program execution, it is necessary to set auto-refresh ar
refresh cycle to no more than the maximum value of tRAS.

A burst read cycle without auto-precharge is shown in figure 12.19, a burst read cycle for
row address in figure 12.20, and a burst read cycle for different row addresses in figure 1
Similarly, a single write cycle without auto-precharge is shown in figure 12.22, a single w
cycle for the same row address in figure 12.23, and a single write cycle for different row
in figure 12.24.

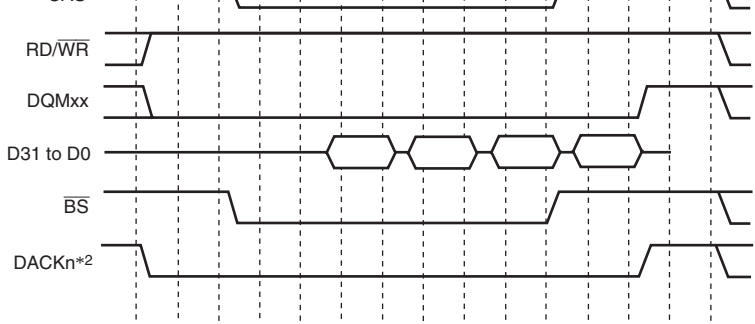
In figure 12.20, a Tnop cycle in which no operation is performed is inserted before the Tc
that issues the READ command. The Tnop cycle is inserted to acquire two cycles of CAS
for the DQMxx signal that specifies the read byte in the data read from the SDRAM. If th
latency is specified as two cycles or more, the Tnop cycle is not inserted because the two
latency can be acquired even if the DQMxx signal is asserted after the Tc cycle.

When bank active mode is set, if only accesses to the respective banks in the area 3 space
considered, as long as accesses to the same row address continue, the operation starts wit
cycle in figure 12.19 or 12.22, followed by repetition of the cycle in figure 12.20 or 12.23
access to a different area during this time has no effect. If there is an access to a different
address in the bank active state, after this is detected the bus cycle in figure 12.21 or 12.2
executed instead of that in figure 12.20 or 12.23. In bank active mode, too, all banks beco
inactive after a refresh cycle or after the bus is released as the result of bus arbitration.



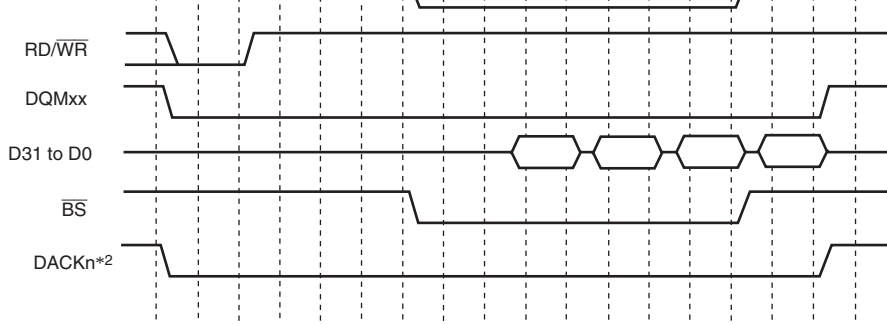
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.19 Burst Read Timing (No Auto Precharge)



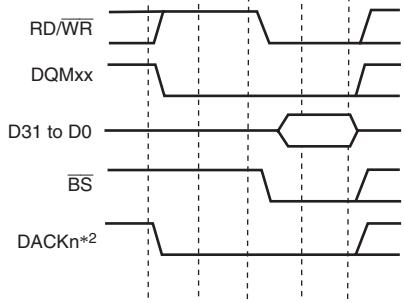
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.20 Burst Read Timing (Bank Active, Same Row Address)



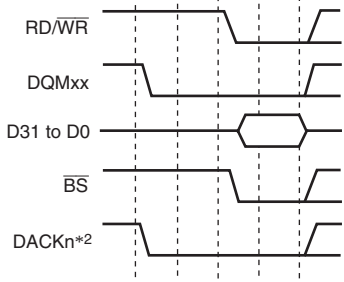
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.21 Burst Read Timing (Bank Active, Different Row Addresses



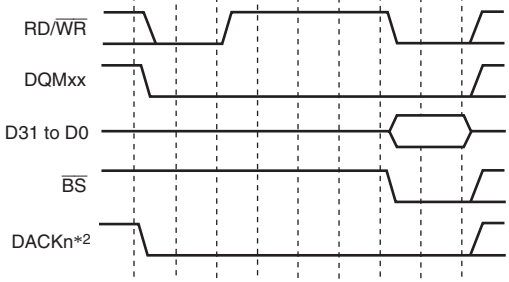
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.22 Single Write Timing (No Auto Precharge)



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.23 Single Write Timing (Bank Active, Same Row Address)

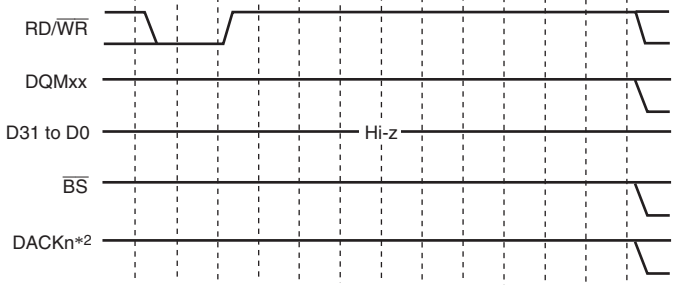


Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.24 Single Write Timing (Bank Active, Different Row Addresses)

Refreshing: This LSI has a function for controlling synchronous DRAM refreshing. Aut refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to SDCR. A continuous refreshing can be performed by setting the RRC[2:0] bits in RTCSI synchronous DRAM is not accessed for a long period, self-refresh mode, in which the po consumption for data retention is low, can be activated by setting both the RMODE bit an RFSH bit to 1.

After starting, the auto refreshing, PALL command is issued in the Tp cycle to make
banks to precharged state from active state when some bank is being precharged. Th
command is issued in the Trr cycle after inserting idle cycles of which number is spe
the TRP[1:0]bits in CSnWCR. A new command is not issued for the duration of the
cycles specified by the TRC[1:0] bits in CSnWCR after the Trr cycle. The TRC[1:0]
be set so as to satisfy the SDRAM refreshing cycle time stipulation (tRC). A NOP cy
inserted between the Tp cycle and Trr cycle when the setting value of the TRP[1:0] b
CSnWCR is longer than or equal to 1 cycle.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.25 Auto-Refresh Timing

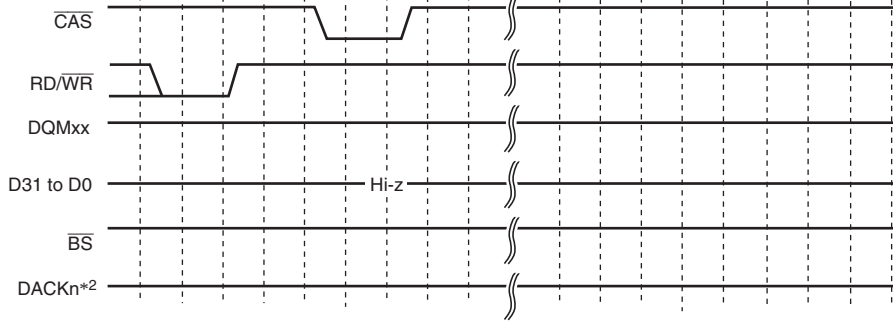
2. Self-refreshing

   Self-refresh mode in which the refresh timing and refresh addresses are generated wit
   synchronous DRAM. Self-refreshing is activated by setting both the RMODE bit and
   RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issued i
   cycle after the completion of the pre-charging bank. A SELF command is then issued
   inserting idle cycles of which number is specified by the TRP[1:0] bits in CSnWSR.
   Synchronous DRAM cannot be accessed while in the self-refresh state. Self-refresh m
   cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, co
   issuance is disabled for the number of cycles specified by the TRC[1:0] bits in CSnW

mode by an interrupt.

The self-refresh state is not cleared by a manual reset.

In case of a power-on reset, the bus state controller’s registers are initialized, and the self-refresh state is cleared.



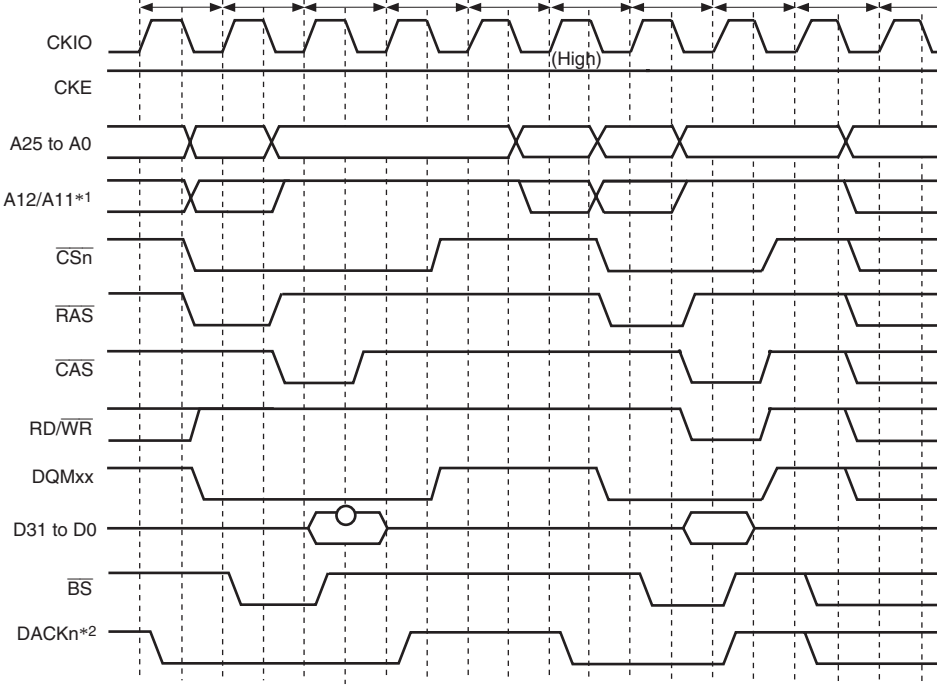
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.26 Self-Refresh Timing

Relationship between Refresh Requests and Bus Cycles: If a refresh request occurs du cycle execution, the refresh cycle must wait for the bus cycle to be completed. If a refresh occurs while the bus is released by the bus arbitration function, the refresh will not be exe until the bus mastership is acquired. This LSI supports requests by the $\overline{\text{REFOUT}}$ pin for t mastership while waiting for the refresh request. The $\overline{\text{REFOUT}}$ pin is asserted low until th mastership is acquired.

If a new refresh request occurs while waiting for the previous refresh request, the previou request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the mastership occupation must be prevented from occurring. If a bus mastership is requested self-refresh, the bus will not be released until the self-refresh is completed.

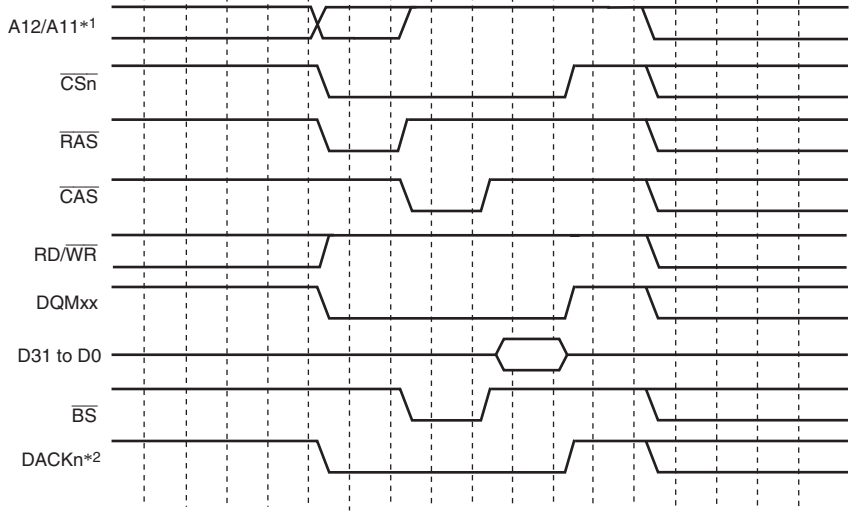
Low-Frequency Mode: When the SLOW bit in SDCR is set to 1, output of commands, a and write data, and fetch of read data are performed at a timing suitable for operating SDR low frequency.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.27 Access Timing in Low-Frequency Mode

Power-Down Mode: If the PDOWN bit in SDCR is set to 1, the SDRAM is placed in th
down mode by bringing the CKE signal to the low level in the non-access cycle. This po
mode can effectively lower the power consumption in the non-access cycle. However, p



Notes: 1.Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.28 Access Timing in Power-Down Mode

Power-On Sequence: In order to use synchronous DRAM, mode setting must first be pe
after powering on. To perform synchronous DRAM initialization correctly, the bus state c
registers must first be set, followed by a write to the synchronous DRAM mode register. I
synchronous DRAM mode register setting, the address signal value at that time is latched
combination of the CSn, RAS, CAS, and RD/WR signals. If the value to be set is X, the b
controller provides for value X to be written to the synchronous DRAM mode register by
performing a write to address H'A4FD4000 + X for area 2 synchronous DRAM, and to ad
H'A4FD5000 + X for area 3 synchronous DRAM. In this operation the data is ignored, bu
mode write is performed as a byte-size access. To set burst read/single write, CAS latency

	3	H′A4FD4460	H′0000460
32 bits	2	H′A4FD4880	H′0000880
	3	H′A4FD48C0	H′00008C0

Burst read/burst write (burst length 1):

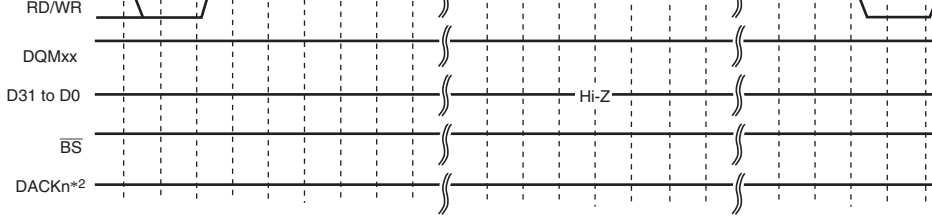
Data Bus Width	CAS Latency	Access Address	External Add
16 bits	2	H′A4FD4040	H′0000040
	3	H′A4FD4060	H′0000060
32 bits	2	H′A4FD4080	H′0000080
	3	H′A4FD40C0	H′00000C0

- Setting for Area 3 (SDMR3)
  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Add
16 bits	2	H′A4FD5440	H′0000440
	3	H′A4FD5460	H′0000460
32 bits	2	H′A4FD5880	H′0000880
	3	H′A4FD58C0	H′00008C0

MRS command (mode register write command) is finally issued. Idle cycles, of which nu
specified by the TRP[1:0] bits in CSnWCR, are inserted between the PALL and the first
cycles, of which number is specified by the TRC[1:0]bits in CSnWCR, are inserted betwe
and REF, and between the 8th REF and MRS. Idle cycles, of which number is one or mo
inserted between the MRS and a command to be issued next.

It is necessary to keep idle time of certain cycles for SDRAM before issuing PALL comm
power-on. Refer the manual of the SDRAM for the idle time to be needed. When the puls
of the reset signal is longer then the idle time, mode register setting can be started immed
after the reset, but care should be taken when the pulse width of the reset signal is shorter
idle time.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

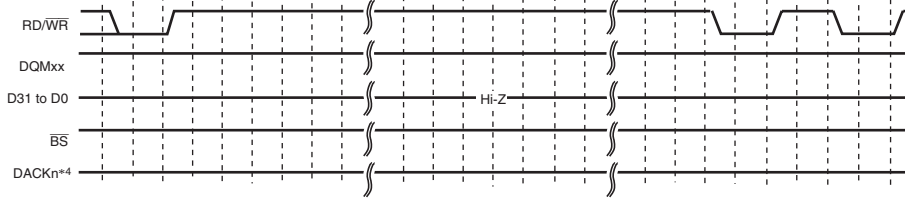
Figure 12.29 Write Timing for SDRAM Mode Register (Based on JEDEC

Low-Power SDRAM: The low-power SDRAM can be accessed using the same protoco normal SDRAM. The differences between the low-power SDRAM and normal SDRAM partial refresh takes place that puts only a part of the SDRAM in the self-refresh state du self-refresh function, and that power consumption is low during refresh under user cond as the operating temperature. The partial refresh is effective in systems in which data in area other than the specific area can be lost without severe repercussions. For details, ref data sheet for the low-power SDRAM to be used.

The low-power SDRAM supports the extension mode register (EMRS) in addition to th registers as the normal SDRAM. This LSI supports issuing of the EMRS command.

The EMRS command is issued according to the conditions specified in table 12.20. For if data H'0YYYYYYY is written to address H'A4FD5XXX in long-word, the command issued to the CS3 space in the following sequence: PALL -> REF x 8 -> MRS -> EMRS case, the MRS and EMRS issue addresses are H'0000XXX and H'YYYYYYY, respecti data H'1YYYYYYY is written to address H'A4FD5XXX in long-word, the commands a to the CS3 space in the following sequence: PALL -> MRS -> EMRS.

(with refresh)					
CS2 MRS +EMRS (without refresh)	H'A4FD4XXX	H'1YYYYYYY	32 bits	H'0000XXX	H'YYYYY
CS3 MRS +EMRS (without refresh)	H'A4FD5XXX	H'1YYYYYYY	32 bits	H'0000XXX	H'YYYYY



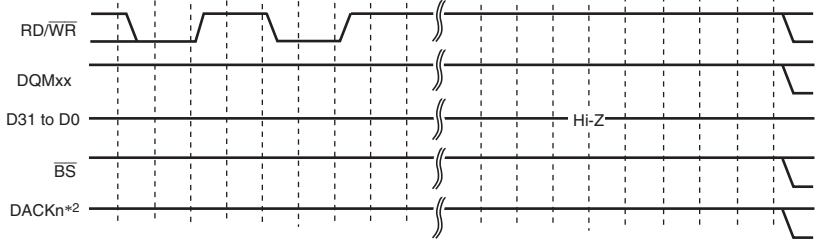
Notes: 1. Address pin to be connected to the BA1 pin of SDRAM.
2. Address pin to be connected to the BA0 pin of SDRAM.
3. Address pin to be connected to the A10 pin of SDRAM.
4. The waveform for DACKn is when active low is specified.

Figure 12.30 EMRS Command Issue Timing

- Deep power-down mode

  The low-power SDRAM supports the deep power-down mode as a low-power consu
  mode. In the partial self-refresh function, self-refresh is performed on a specific area
  deep power-down mode, self-refresh will not be performed on any memory area. Th
  effective in systems where all of the system memory areas are used as work areas.

If the RMODE bit of the SDCR is set to 1 while the DEEP and RFSH bits of the SDCR
1, the low-power SDRAM enters the deep power-down mode. If the RMODE bit is clea
the CKE signal is pulled high to cancel the deep power-down mode. Before executing a
after returning from the deep power-down mode, the power-up sequence must be re-exe



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.31 Transition Timing in Deep Power-Down Mode

12.5.6 Burst ROM (Clock Asynchronous) Interface

The burst ROM (clock asynchronous) interface is used to access a memory with a high-sp function using a method of address switching called the burst mode or page mode. In a bu (clock asynchronous) interface, basically the same access as the normal space is performe the 2nd and subsequent accesses are performed only by changing the address, without neg $\overline{RD}$ signal at the end of the 1st cycle. In the 2nd and subsequent accesses, addresses are cl the falling edge of the CKIO.

For the 1st access cycle, the number of wait cycles specified by the W[3:0] bits in CSnW inserted. For the 2nd and subsequent access cycles, the number of wait cycles specified b BW[1:0] bits in CSnWCR is inserted.

In the access to the burst ROM (clock asynchronous), the $\overline{BS}$ signal is asserted only to the access cycle. An external wait input is valid only to the first access cycle.

In the single access or write access that do not perform the burst operation in the burst RO (clock asynchronous) interface, access timing is same as a normal space.

	1		4	4
16 bits	Not affected	8 bits	1	1
	Not affected	16 bits	1	1
	Not affected	32 bits	2	1
	0	16 bytes	8	1
	1		2	4
32 bits	Not affected	8 bits	1	1
	Not affected	16 bits	1	1
	Not affected	32 bits	1	1
	Not affected	16 bytes	4	1

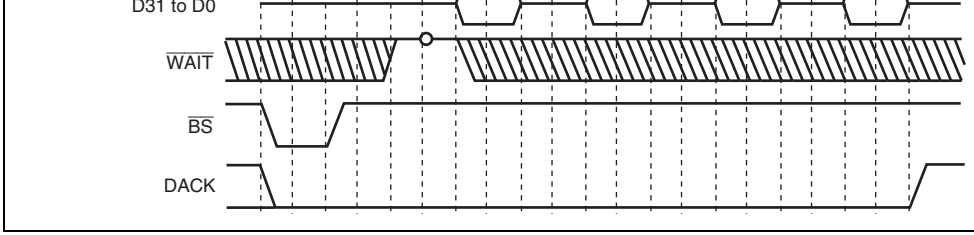


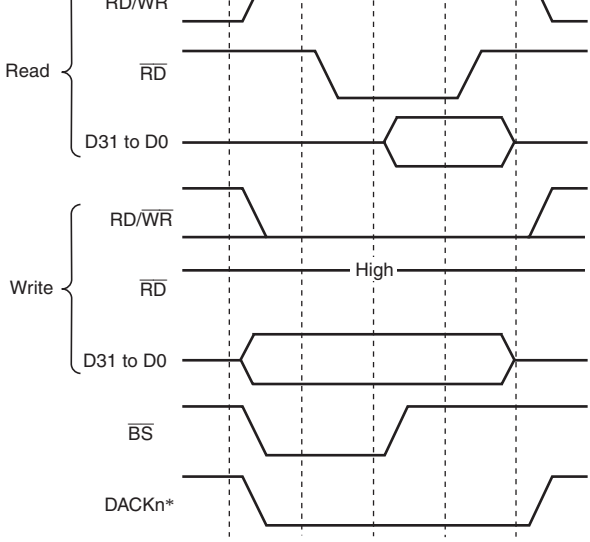
Figure 12.32 Burst ROM (Clock Asynchronous) Access (Bus Width = 32 Bit 16-byte Transfer (Number of Bursts = 4), Access Wait for First Time = 2, Access Wait for 2nd Time and after = 1)

12.5.7 Byte-Selection SRAM Interface

The byte-selection SRAM interface is for access to an SRAM which has a byte-selection ($\overline{WEn}$ ($\overline{BEn}$)). This interface has 16-bit data pins and accesses SRAMs having upper and byte selection pins, such as UB and LB.

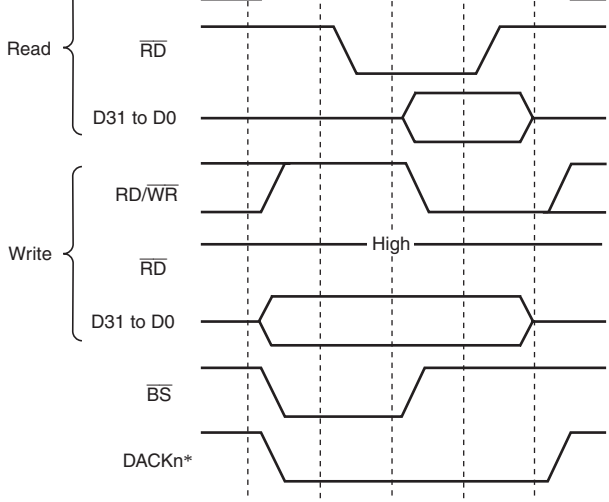
When the BAS bit in CSnWCR is cleared to 0 (initial value), the write access timing of th selection SRAM interface is the same as that for the normal space interface. While in read of a byte-selection SRAM interface, the byte-selection signal is output from the $\overline{WEn}$ ($\overline{BE}$ which is different from that for the normal space interface. The basic access timing is sho figure 12.33. In write access, data is written to the memory according to the timing of the selection pin ($\overline{WEn}$ ($\overline{BEn}$)). For details, refer to the data sheet for the corresponding mem

If the BAS bit in CSnWCR is set to 1, the $\overline{WEn}$ ($\overline{BEn}$) pin and RD/$\overline{WR}$ pin timings chang 12.34 shows the basic access timing. In write access, data is written to the memory accord the timing of the write enable pin (RD/$\overline{WR}$). The data hold timing from RD/$\overline{WR}$ negation write must be acquired by setting the HW[1:0] bits in CSnWCR. Figure 12.35 shows the timing when a software wait is specified.



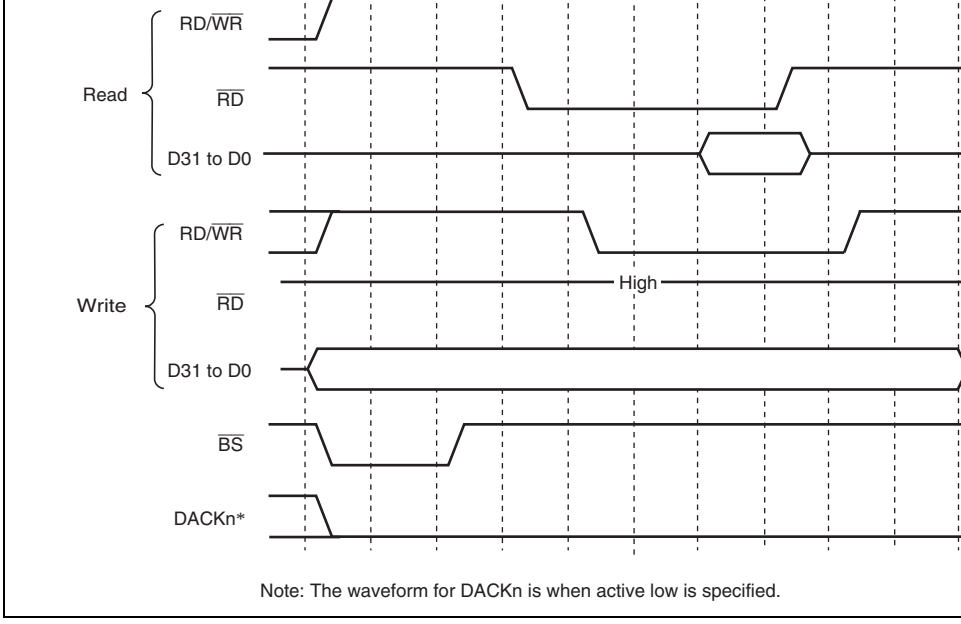
Note: The waveform for DACKn is when active low is specified.

Figure 12.33 Basic Access Timing for Byte-Selection SRAM (BAS = 0)



Note: The waveform for DACKn is when active low is specified.

Figure 12.34 Basic Access Timing for Byte-Selection SRAM (BAS = 1)



Note: The waveform for DACKn is when active low is specified.

Figure 12.35 Wait Timing for Byte-Selection SRAM (BAS = 1) (Software Wait

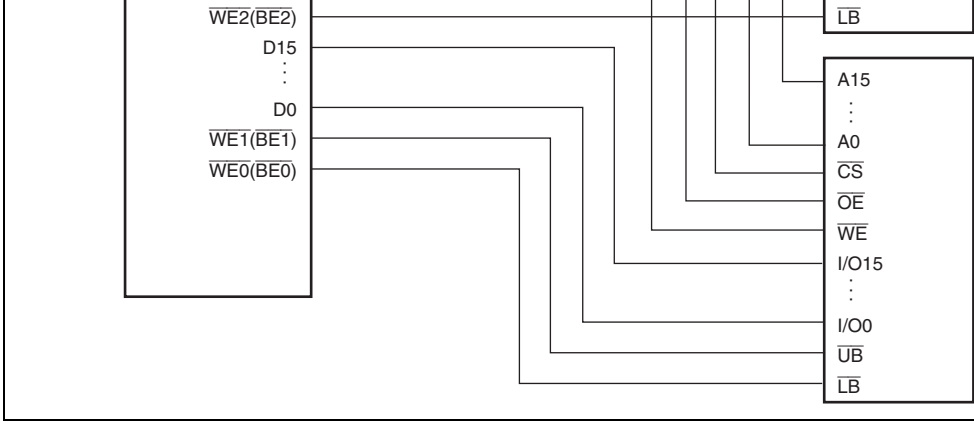


Figure 12.36 Example of Connection with 32-Bit Data-Width Byte-Selection SI

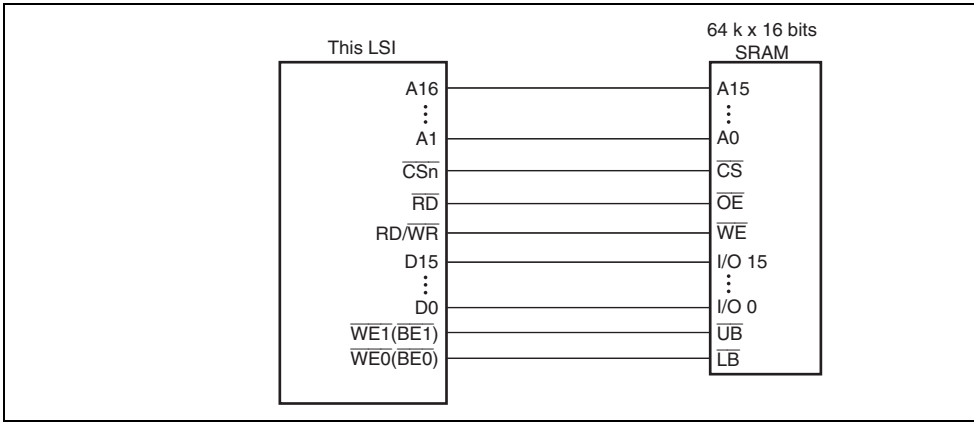


Figure 12.37 Example of Connection with 16-Bit Data-Width Byte-Selection SI

When the PCMCIA interface is used, the bus size must be specified as 8 bits or 16 bits u
BSZ[1:0] bits in CS5BBCR or CS6BBCR.

Figure 12.38 shows an example of a connection between this LSI and the PCMCIA card
enable insertion and removal of the PCMCIA card during system power-on, a three-state
must be connected between the LSI and the PCMCIA card.

In the JEIDA and PCMCIA standards, operation in the big endian mode is not clearly de
Consequently, an original definition is provided for the PCMCIA interface in big endian
this LSI.

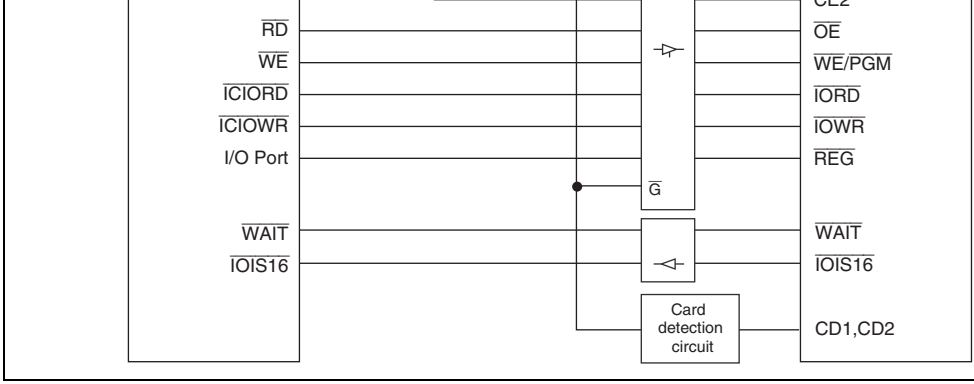


Figure 12.38 Example of PCMCIA Interface Connection

Basic Timing for Memory Card Interface: Figure 12.39 shows the basic timing of the I IC memory card interface. If areas 5 and 6 in the physical space are specified as the PCM interface, accessing the common memory areas in areas 5 and 6 automatically accesses th memory card interface. If the external bus frequency (CKIO) increases, the setup times ar times for the address pins (A25 to A0) to RD and WE, card enable signals ($\overline{\text{CE1A}}$, $\overline{\text{CE2A}}$ $\overline{\text{CE2B}}$), and write data (D15 to D0) become insufficient. To prevent this error, the LSI ca the setup times and hold times for areas 5 and 6 in the physical space independently, usin CS5BWCR and CS6BWCR. In the PCMCIA interface, as in the normal space interface, a software wait or hardware wait can be inserted using the WAIT pin. Figure 12.40 shows t PCMCIA memory bus wait timing.

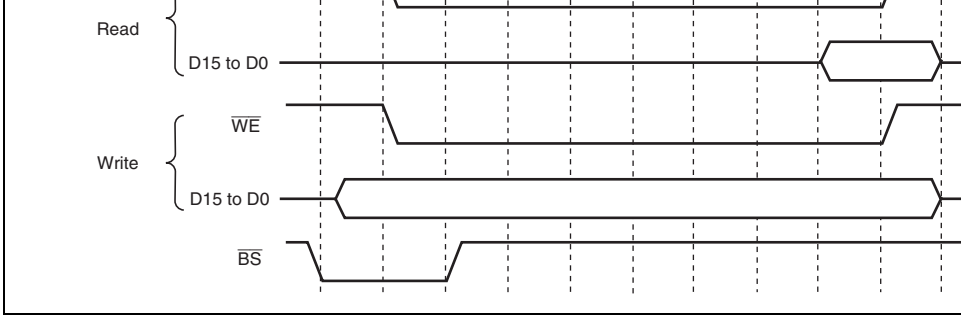


Figure 12.39 Basic Access Timing for PCMCIA Memory Card Interface

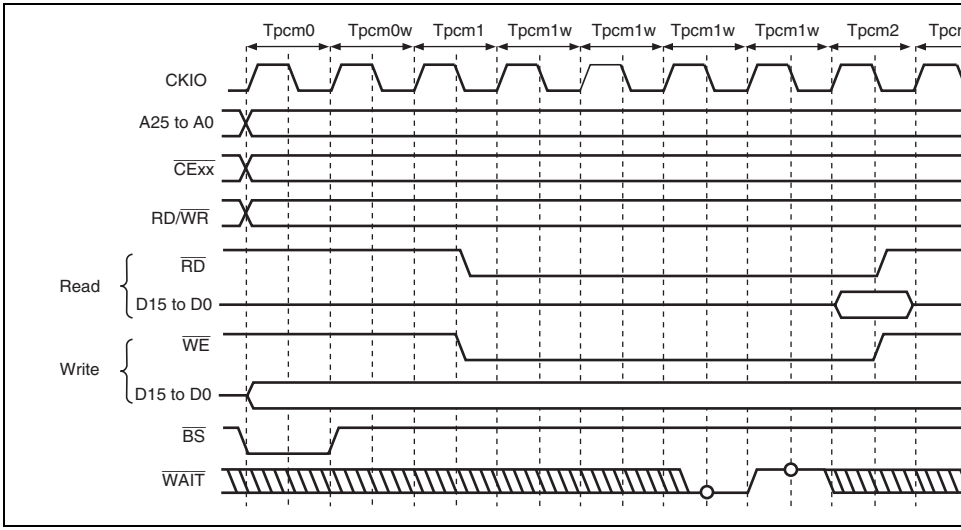


Figure 12.40 Wait Timing for PCMCIA Memory Card Interface (TED[3:0] = B′0010, TEH[3:0] = B′0001, Software Wait = 1, Hardware Wait

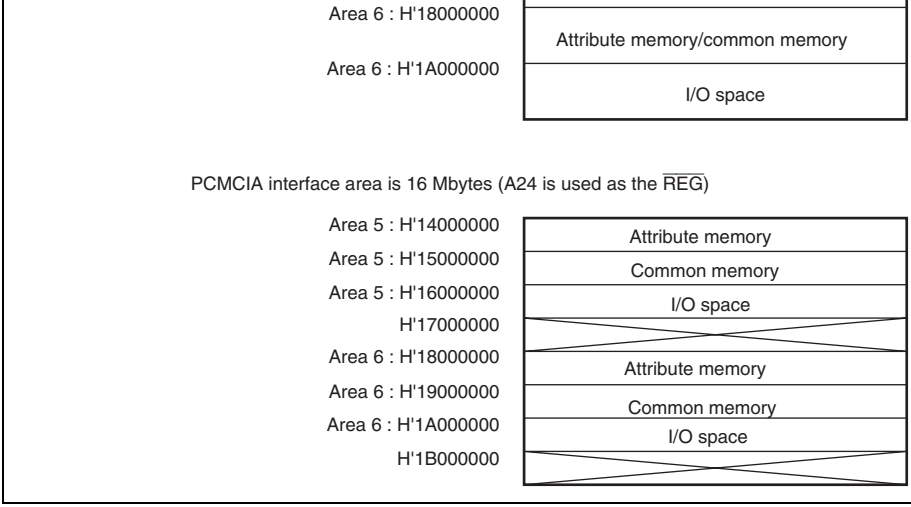


Figure 12.41 Example of PCMCIA Space Assignment (CS5BWCR.SA[1:0] = I CS6BWCR.SA[1:0] = B′10)

Basic Timing for I/O Card Interface: Figures 12.42 and 12.43 show the basic timings f PCMCIA I/O card interface.

The I/O card and IC memory card interfaces can be switched using an address to be acces area 5 of the physical space is specified as the PCMCIA, the I/O card interface can autom be accessed by accessing the physical addresses from H'16000000 to H'17FFFFFF. If are the physical space is specified as the PCMCIA, the I/O card interface can automatically b accessed by accessing the physical addresses from H′1A000000 to H′1BFFFFFF.

Note that areas to be accessed as the PCMCIA I/O card must be non-cached if they are lo space (space P2 or P3) areas, or a non-cached area specified by the MMU.

Note that the $\overline{\text{IOIS16}}$ signal is not supported in big endian mode. In the big endian mode $\overline{\text{IOIS16}}$ signal must be fixed low.

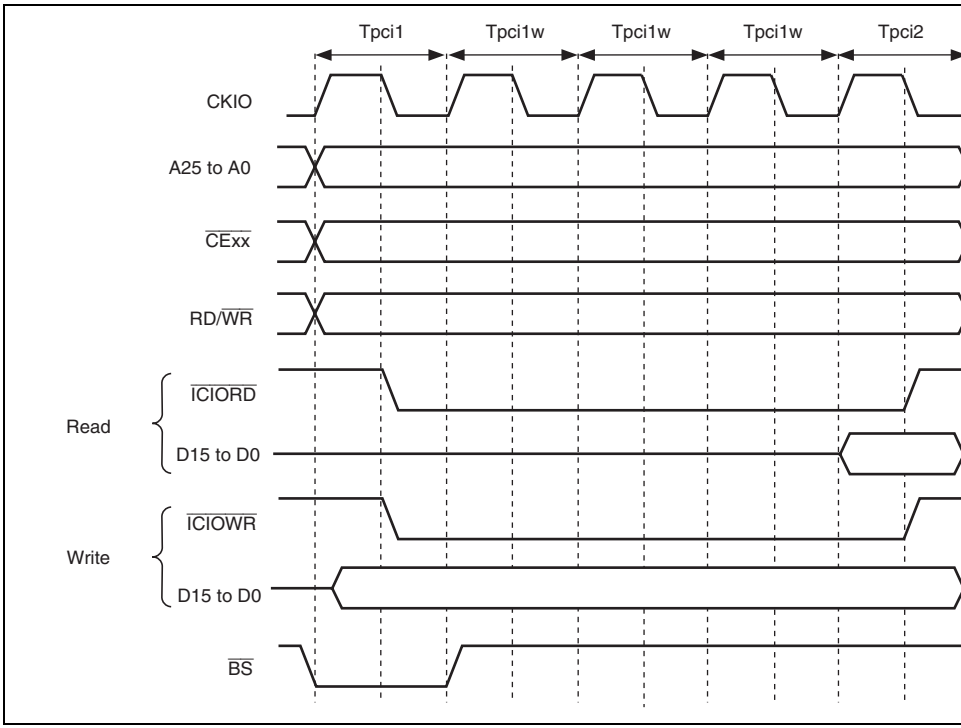


Figure 12.42 Basic Timing for PCMCIA I/O Card Interface

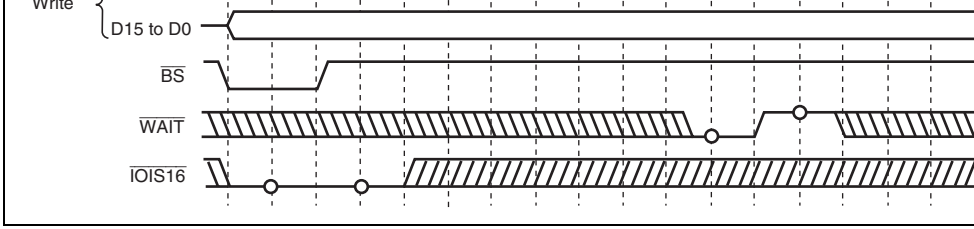


Figure 12.43 Wait Timing for PCMCIA I/O Card Interface (TED[3:0] = B′0010, TEH[3:0] = B′0001, Software Wait = 1, Hardware Wait =

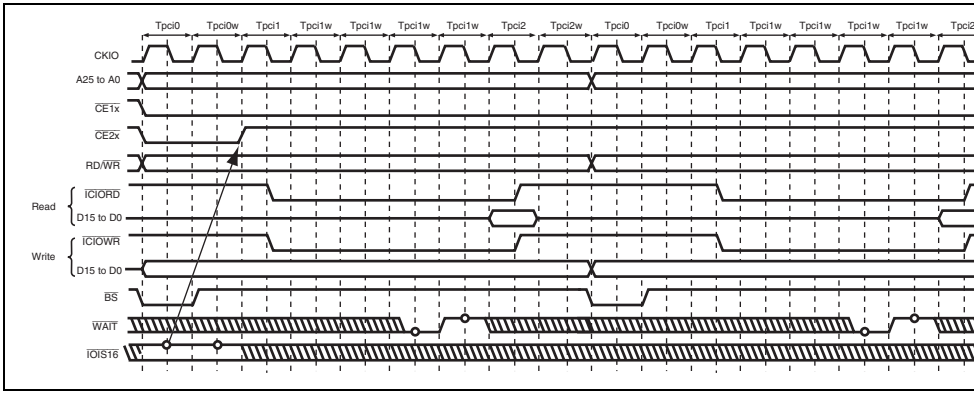


Figure 12.44 Timing for Dynamic Bus Sizing of PCMCIA I/O Card Interfac (TED[3:0] = B′0010, TEH[3:0] = B′0001, Software Waits = 3)

access cycle and an external wait input is also valid for the first access cycle.

If the bus width is 16 bits, the burst length must be specified as 8. If the bus width is 32 burst length must be specified as 4. The burst ROM interface does not support the 8-bit for the burst ROM. The burst ROM interface performs burst operations for all read acce example, in a longword access over a 16-bit bus, valid 16-bit data is read two times and 16-bit data is read six times.

These invalid data read cycles increase the memory access time and degrade the progra execution speed and DMA transfer speed. To prevent this problem, a 16-byte read by ca 16-byte read by the DMA should be used. The burst ROM interface performs write acce same way as normal space access.

Note: The burst ROM (clock synchronous) must be accessed as cacheable space.

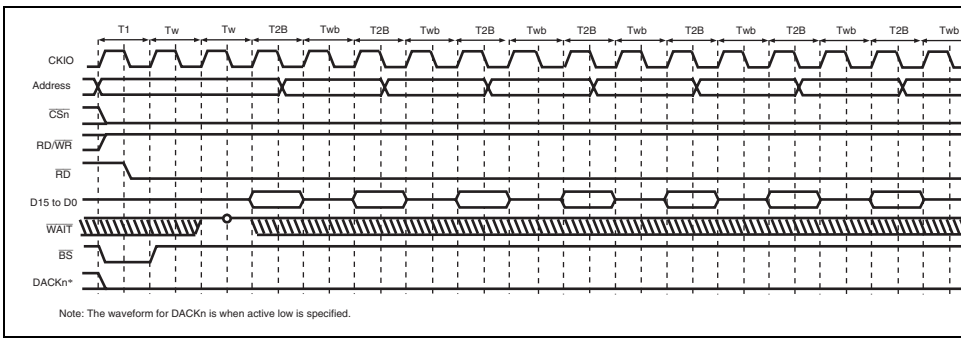


Figure 12.45 Burst ROM (Clock Synchronous) Access Timing (Burst Length = 8, Wait Cycles inserted in First Access = 2, Wait Cycles inserted in Second and Subsequent Accesses = 1)

shown below.

1. Continuous accesses are write-read or write-write
2. Continuous accesses are read-write for different spaces
3. Continuous accesses are read-write for the same space
4. Continuous accesses are read-read for different spaces
5. Continuous accesses are read-read for the same space
6. Data output from an external device caused by DMA single transfer is followed by da from another device that includes this LSI (DMAIWA = 0)
7. Data output from an external device caused by DMA single transfer is followed by an access (DMAIWA = 1)

12.5.11 Bus Arbitration

To prevent device malfunction while the bus mastership is transferred between master an the LSI negates all of the bus control signals before bus release. When the bus mastership received, all of the bus control signals are first negated and then driven appropriately. In t output buffer contention can be prevented because the master and slave drive the same sig with the same values. In addition, to prevent noise while the bus control signal is in the hi impedance state, pull-up resistors must be connected to these control signals.

Bus mastership is transferred at the boundary of bus cycles. Namely, bus mastership is re immediately after receiving a bus request when a bus cycle is not being performed. The re bus mastership is delayed until the bus cycle is complete when a bus cycle is in progress. when from outside the LSI it looks like a bus cycle is not being performed, a bus cycle m performing internally, started by inserting wait cycles between access cycles. Therefore, i be immediately determined whether or not bus mastership has been released by looking a

6. Setting the BLOCK bit in CMNCR to 1

Bits DPRTY[1:0] in CMNCR can select whether or not the bus request is received durin
burst transfer.

This LSI has the bus mastership until a bus request is received from another device. Upo
acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LS
the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the
acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the slave has r
the bus, it negates the $\overline{\text{BACK}}$ signal and resumes the bus usage.

The SDRAM issues a all bank precharge command (PALL) when active banks exist and
the bus after completion of a PALL command.

The bus sequence is as follows. The address bus and data bus are placed in a high-impec
synchronized with the rising edge of CKIO. The bus mastership enable signal is asserted
cycles after the above timing, synchronized with the falling edge of CKIO. The bus cont
($\overline{\text{BS}}$, $\overline{\text{CS}}$n, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, DQMxx, $\overline{\text{WE}}$n ($\overline{\text{BE}}$n), $\overline{\text{RD}}$, and RD/$\overline{\text{WR}}$) are placed in the high-in
state at subsequent rising edges of CKIO. Bus request signals are sampled at the falling
CKIO.

The sequence for reclaiming the bus mastership from a slave is described below. 1.5 cyc
the negation of $\overline{\text{BREQ}}$ is detected at the falling edge of CKIO, the bus control signals ar
high. The $\overline{\text{BACK}}$ is negated at the next falling edge of the clock. The fastest timing at w
bus cycles can be resumed after bus control signal assertion is at the rising edge of the C
where address and data signals are driven. Figure 12.46 shows the bus arbitration timing

In an original slave device designed by the user, multiple bus accesses are generated con
to reduce the overhead caused by bus arbitration. In this case, to execute SDRAM refres
correctly, the slave device must be designed to release the bus mastership within the refr
interval time. To achieve this, the LSI instructs the $\overline{\text{REFOUT}}$ pin to request the bus mast

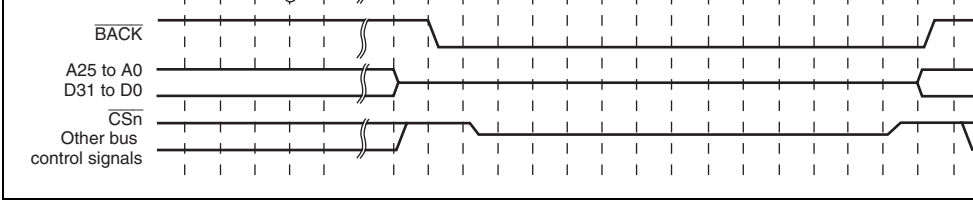


Figure 12.46 Bus Arbitration Timing

12.5.12 Others

Reset: The bus state controller (BSC) can be initialized completely only at power-on rese
power-on reset, all signals are negated and output buffers are turned off regardless of the
state. All control registers are initialized. In standby, sleep, and manual reset, control regi
the bus state controller are not initialized. At manual reset, the current bus cycle being ex
completed and then the access wait state is entered. If a 16-byte transfer is performed by
or if another LSI on-chip bus master module is executed when a manual reset occurs, the
access is cancelled in longword units because the access request is cancelled by the bus m
manual reset. If a manual reset is requested during cache fill operations, the contents of th
cannot be guaranteed. Since the RTCNT continues counting up during manual reset signa
assertion, a refresh request occurs to initiate the refresh cycle. Note, however, a bus arbitr
request by the $\overline{\text{BREQ}}$ signal can't be accepted during manual reset signal assertion.

Some flash memories may specify a minimum time from reset release to the first access.
ensure this minimum time, the bus state controller supports a 5-bit reset wait counter (RW
At power-on reset, the RWTCNT is cleared to 0. After a power-on reset, RWTCNT is co
synchronously together with CKIO and an external access will not be generated until RW
counted up to H′007F. At a manual reset, RWTCNT is not cleared. RWTCNT cannot be
or written to.

than the CPU writes data to an external memory other than the cache, the contents of the memory may differ from that of the cache memory. To prevent this problem, if the exter memory whose contents is cached is written by an on-chip bus master other than the CP corresponding cache memory should be purged by software.

If the CPU initiates read access for the cache, the cache is searched. If the cache stores d CPU latches the data and completes the read access. If the cache does not store data, the performs four contiguous longword read cycles to perform cache fill operations via the i bus. If a cache miss occurs in byte or word operand access or at a branch to an odd word (4n + 2), the CPU performs four contiguous longword accesses to perform a cache fill o on the external interface. For a non-Cacheable area, the CPU performs access according actual access addresses. For an instruction fetch to an even word boundary (4n), the CPU longword access. For an instruction fetch to an odd word boundary (4n + 2), the CPU pe word access.

For a read cycle of a cache-through area or an on-chip peripheral module, the read cycle accepted and then read cycle is initiated. The read data is sent to the CPU via the cache b

In a write cycle for the cache area, the write cycle operation differs according to the cach methods.

In write-back mode, the cache is first searched. If data is detected at the address corresp the cache, the data is then re-written to the cache. In the actual memory, data will not be until data in the corresponding address is re-written. If data is not detected at the address corresponding to the cache, the cache is modified. In this case, data to be modified is firs the internal buffer, 16-byte data including the data corresponding to the address is then r data in the corresponding access of the cache is finally modified. Following these operat write-back cycle for the saved 16-byte data is executed.

continue the process after the data write to the device has been completed, perform a dum
to the same address to check for completion of the write before the next process to be exe

The write buffer of the BSC functions in the same way for an access by a bus master othe
the CPU such as the DMAC or E-DMAC. Accordingly, to perform dual address DMA tra
the next read cycle is initiated before the previous write cycle is completed. Note, howeve
both the DMA source and destination addresses exist in external memory space, the next
cycle will not be initiated until the previous write cycle is completed.

On-Chip Peripheral Module Access: To access an on-chip module register, two or mor
peripheral module clock (Pϕ) cycles are required. Care must be taken in system design.

- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, Word (two bytes), Longword (four bytes), and (longword × 4)
- Maximum transfer count: 16,777,216 transfers (24 bits)
- Address mode: Dual address mode and single address mode are supported.
- Transfer requests:

  An external request, on-chip peripheral module request, or auto request can be selec

  The following modules can issue an on-chip peripheral module request.

  SCIF0, SCIF1, SIOF0, SIOF1
- Bus mode: Cycle steal mode or burst mode can be selected.
- Channel priority levels: The channel priority levels are selectable between fixed mod round-robin mode.
- Interrupt request: An interrupt request can be generated to the CPU at the end of the counts of data transfer.
- External request detection: There are following four types of DREQ input detection.

  Low-level detection

  High-level detection

  Rising-edge detection

  Falling-edge detection
- Transfer request acknowledge and transfer end signals: Active levels for DACK and can be set independently.

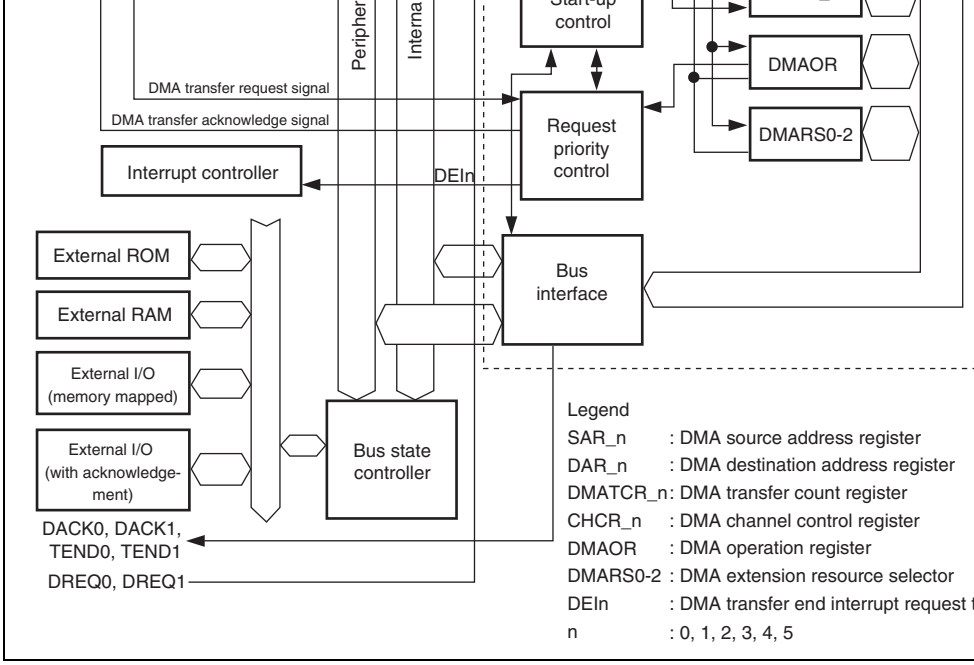


Figure 13.1 Block Diagram of DMAC

				external device to channel 0
	DMA transfer request acknowledge	DACK0	O	DMA transfer request acknow output from channel 0 to exte device
	DMA transfer end	TEND0	O	DMA transfer end output for c
1	DMA transfer request	DREQ1	I	DMA transfer request input fr external device to channel 1
	DMA transfer request acknowledge	DACK1	O	DMA transfer request acknow output from channel 1 to exte device
	DMA transfer end	TEND1	O	DMA transfer end output for c

- DMA channel control register_0 (CHCR_0)

Channel 1:

- DMA source address register_1 (SAR_1)
- DMA destination address register_1 (DAR_1)
- DMA transfer count register_1 (DMATCR_1)
- DMA channel control register _1 (CHCR_1)

Channel 2:

- DMA source address register_2 (SAR_2)
- DMA destination address register_2 (DAR_2)
- DMA transfer count register_2 (DMATCR_2)
- DMA channel control register_2 (CHCR_2)

Channel 3:

- DMA source address register_3 (SAR_3)
- DMA destination address register_3 (DAR_3)
- DMA transfer count register_3 (DMATCR_3)
- DMA channel control register_3 (CHCR_3)

Channel 4:

- DMA source address register_4 (SAR_4)
- DMA destination address register_4 (DAR_4)
- DMA transfer count register_4 (DMATCR_4)
- DMA channel control register_4 (CHCR_4)

- DMA extension resource selector 1 (DMARS1)
- DMA extension resource selector 2 (DMARS2)

13.3.1 DMA Source Address Register (SAR)

SAR is a 32-bit readable/writable register that specifies the source address of a DMA tra
During a DMA transfer, SAR indicates the next source address. When the data is transfe
an external device with the DACK in single address mode, SAR is ignored.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address
When transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set fo
source address value.

SAR is undefined at reset and retains the current value in standby or module standby mo

13.3.2 DMA Destination Address Register (DAR)

DAR is a 32-bit readable/writable register that specifies the destination address of a DM
During a DMA transfer, DAR indicates the next destination address. When the data is tr
to an external device with the DACK in single address mode, DAR is ignored.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address
When transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set fo
source address value.

DAR is undefined at reset and retains the current value in standby or module standby m

DMATCR is undefined at reset and retains the current value in standby or module standb

13.3.4 DMA Channel Control Register (CHCR)

CHCR is a 32-bit readable/writable register that controls the DMA transfer mode.

CHCR is initialized to H'00000000 at reset and retains the current value in the standby or standby mode.

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 sl always be 0.
23	DO	0	R/W	DMA Overrun Selects whether the DREQ is detected by overrun overrun 1. This bit is valid only in CHCR0 and CHCR1.This bi reserved and always read as 0 in CHCR2 to CHCF write value should always be 0. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1

21 to 18		All 0	R	Reserved These bits are always read as 0. The write value always be 0.
17	AM	0	R/W	Acknowledge Mode Specifies whether the DACK is output in data rea in data write cycle in dual address mode. In single address mode, the DACK is always outp regardless of the specification by this bit. This bit is valid only in CHCR0 and CHCR1.This reserved and always read as 0 in CHCR2 to CHC write value should always be 0. 0: DACK output in read cycle (Dual address mod 1: DACK output in write cycle (Dual address mod
16	AL	0	R/W	Acknowledge Level Specifies the DACK signal output is high active o active. This bit is valid only in CHCR0 and CHCR1.This reserved and always read as 0 in CHCR2 to CHC write value should always be 0. 0: Low-active output of DACK 1: High-active output of DACK

				byte transfer) 10: Destination address is decremented (–1 in byte –2 in word transfer, –4 in longword transfer; se prohibited in 16-byte transfer) 11: Reserved (setting prohibited)
13	SM1	0	R/W	Source Address Mode
12	SM0	0	R/W	Specify whether the DMA source address is increm decremented, or left fixed. (In single address mode SM1 and SM0 bits are ignored when data is transfe from an external device with the DACK.) 00: Fixed source address (setting prohibited in 16-b transfer) 01: Source address is incremented (+1 in byte tran in word transfer, +4 in longword transfer, +16 in transfer) 10: Source address is decremented (–1 in byte tran in word transfer, –4 in longword transfer; setting prohibited in 16-byte transfer) 11: Reserved (setting prohibited)

				0100: Auto request 1000: DMA extension resource selector Other than above: Reserved (setting prohibited) Note: An external request specification is valid only CHCR0 and CHCR1. None of the external re specification can be selected in CHCR2 to C
7	DL	0	R/W	DREQ Level and DREQ Edge Select
6	DS	0	R/W	Specify the sampling method of the DREQ pin inpu sampling level. These bits are valid only in CHCR0 and CHCR1. T are reserved and always read as 0 in CHCR2 to C The write value should always be 0. In channels 0 and 1, also, if the transfer request so specified as an on-chip peripheral module or if an a request is specified, the specification by this bit is i 00: DREQ detected in low level 01: DREQ detected at falling edge 10: DREQ detected in high level 11: DREQ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode Specifies the bus mode when the DMA transfers d 0: Cycle steal mode 1: Burst mode

				11: 16-byte (four longwords) size
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is g to the CPU at the end of the DMA transfer. Settin to 1 generates an interrupt request (DEI) to the C when the TE bit is set to 1. 0: Interrupt request is disabled 1: Interrupt request is enabled
1	TE	0	R/(W)*	Transfer End Flag The TE bit is set to 1 when data transfer ends wh DMATCR becomes to 0. The TE bit is not set to 1 in the following cases. • DMA transfer ends due to an NMI interrupt or address error before DMATCR becomes 0. • DMA transfer is ended by clearing the DE bit DME bit in the DMA operation register (DMAC Even if the DE bit is set to 1 while this bit is set to transfer is not enabled. 0: During the DMA transfer or DMA transfer has b interrupted 1: DMA transfer ends by the specified count (DM 0) [Clearing condition] Writing 0 after reading 1 from this bit

				bit to 0 can terminate the DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

Note: * Only 0 can be written to clear the flag.

13.3.5 DMA Operation Register (DMAOR)

DMAOR is a 16-bit readable/writable register that specifies the priority level of channel DMA transfer. This register indicates the DMA transfer status.

DMAOR is initialized to H′0000 at a reset and retains the current value in standby or mo standby mode.

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 s always be 0.
9	PR1	0	R/W	Priority Mode
8	PR0	0	R/W	Select the priority level between channels when th transfer requests for multiple channels simultaneo 00: Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH 01: Fixed mode 2: CH0 > CH2 > CH3 > CH1 > CH 10: Reserved (setting prohibited) 11: All channel round-robin mode

				1: DMAC address error [Clear conditions] Writing 0 after reading 1 from this bit
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. When th set, DMA transfer is disabled even if the DE bit in and the DME bit in DMAOR are set to 1. This bit c be cleared by writing 0 after reading 1. When the NMI is input, the DMA transfer in progre be done in one transfer unit. Even if the DMAC is operational, this bit is set to 1 when the NMI interr input. 0: No NMI interrupt 1: NMI interrupt occurs [Clearing conditions] Writing 0 after reading 1 from this bit
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfers on all channel DME bit and the DE bit in CHCR are set to 1, DM transfer is enabled. Note that transfer is enabled i bit in CHCR and the NMIF and AE bits in DMAOR 0. If this bit is cleared, DMA transfers in all the cha can be terminated. 0: Disables DMA transfers on all channels 1: Enables DMA transfers on all channels

Note: * Only 0 can be written to clear the flag.

DMARS is initialized to H'0000 at reset and retains the current value in standby or modu
standby mode.

- DMARS0

Bit	Bit Name	Initial Value	R/W	Description
15	C1MID5	0	R/W	Transfer request module ID for DMA channel 1 (I
14	C1MID4	0	R/W	See table 13.2.
13	C1MID3	0	R/W	
12	C1MID2	0	R/W	
11	C1MID1	0	R/W	
10	C1MID0	0	R/W	
9	C1RID1	0	R/W	Transfer request register ID for DMA channel 1 (I
8	C1RID0	0	R/W	See table 13.2.
7	C0MID5	0	R/W	Transfer request module ID for DMA channel 0 (I
6	C0MID4	0	R/W	See table 13.2.
5	C0MID3	0	R/W	
4	C0MID2	0	R/W	
3	C0MID1	0	R/W	
2	C0MID0	0	R/W	
1	C0RID1	0	R/W	Transfer request register ID for DMA channel 0 (I
0	C0RID0	0	R/W	See table 13.2.

9	C3RID1	0	R/W	Transfer request module ID for DMA channel 3 (RI
8	C3RID0	0	R/W	See table 13.2.
7	C2MID5	0	R/W	Transfer request module ID for DMA channel 2 (MI
6	C2MID4	0	R/W	See table 13.2.
5	C2MID3	0	R/W	
4	C2MID2	0	R/W	
3	C2MID1	0	R/W	
2	C2MID0	0	R/W	
1	C2RID1	0	R/W	Transfer request module ID for DMA channel 2 (RI
0	C2RID0	0	R/W	See table 13.2.

9	C5RID1	0	R/W	Transfer request module ID for DMA channel 5 (
8	C5RID0	0	R/W	See table 13.2.
7	C4MID5	0	R/W	Transfer request module ID for DMA channel 4 (
6	C4MID4	0	R/W	See table 13.2.
5	C4MID3	0	R/W	
4	C4MID2	0	R/W	
3	C4MID1	0	R/W	
2	C4MID0	0	R/W	
1	C4RID1	0	R/W	Transfer request module ID for DMA channel 4 (
0	C4RID0	0	R/W	See table 13.2.

SIOF0	H'51	B'010100	B'01	Transmit
	H'52		B'10	Receive
SIOF1	H'55	B'010101	B'01	Transmit
	H'56		B'10	Receive

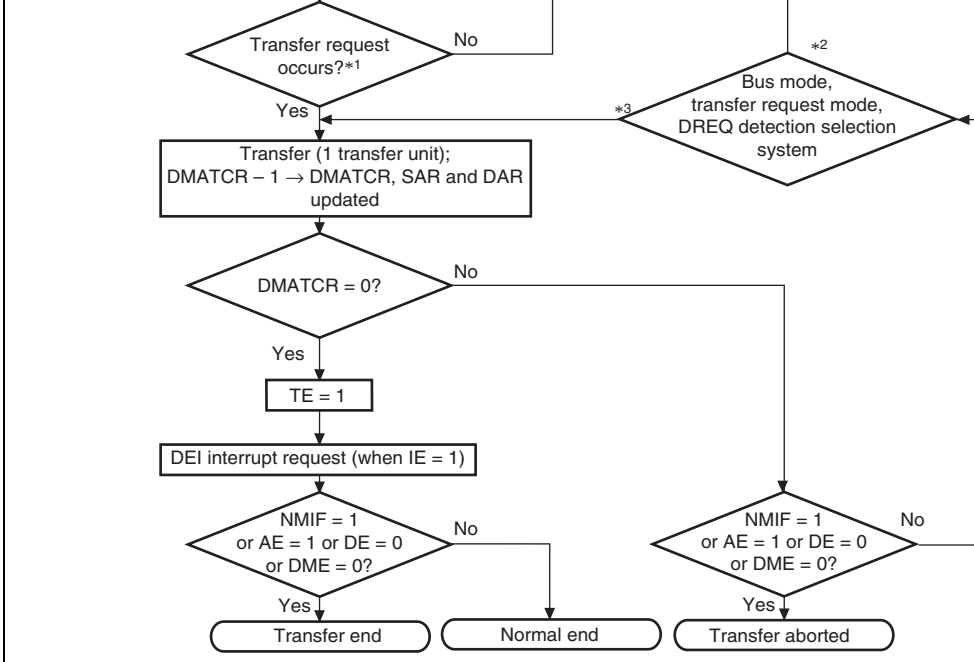
13.4 Operation

When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it en transfer. Transfers can be requested in three modes: auto request, external request, and or peripheral module request. In the bus mode, the burst mode or the cycle steal mode can b selected.

13.4.1 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA oper register (DMAOR), and DMA extension resource selector (DMARS) are set, the DMAC 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 comes and transfer is enabled, the DMAC transfers 1 transfer data (depending on the TS0 and TS1 settings). For an auto request, 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 bu



Notes: *1 In auto-request mode, transfer begins when NMIF, AE and TE are all 0 and the DE and DME bits are set to 1.
*2 DREQ = level detection in burst mode (external request) or cycle-steal mode.
*3 DREQ = edge detection in burst mode (external request), or auto-request mode in burst mod

Figure 13.2 DMA Transfer Flowchart

Auto-Request Mode: When there is no transfer request signal from an external source, memory-to-memory transfer or a transfer between memory and an on-chip peripheral m unable to request a transfer, the auto-request mode allows the DMAC to automatically g transfer request signal internally. When the DE bits of CHCR_0 to CHCR_5 and the DM the DMAOR are set to 1, the transfer begins so long as the TE bits of CHCR_0 to CHCI of DMAOR, and the NMIF bit of DMAOR are all 0.

External Request Mode: In this mode a transfer is performed at the request signals (DI DREQ1) of an external device. Choose one of the modes shown in table 13.3 according application system. When this mode is selected, if the DMA transfer is enabled (DE = 1 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon a request at the DREQ inpu

Table 13.3 Selecting External Request Modes with RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destinatio
0	0	0	0	Dual address mode	Any	Any
0	0	1	0	Single address mode	External memory, memory-mapped external device	External de DACK
			1		External device with DACK	External m memory-ma external de

Whether the DREQ is detected by either the edge or level of the signal input is selected DREQ level (DL) bit and DREQ select (DS) bit in CHCR_0 and CHCR_1 as shown in t The source of the transfer request does not have to be the data transfer source or destinat

period). After issuing acknowledge signal DACK for the accepted DREQ, the DREQ pin becomes request accept enabled state.

When DREQ is used by level detection, there are following two cases by the timing to de next DREQ after outputting DACK.

Overrun 0: Transfer is aborted after the same number of transfer has been performed as re

Overrun 1: Transfer is aborted after transfers have been performed for (the number of req plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

Table 13.5 Selecting External Request Detection with DO Bit

CHCR	
DO	**External Request**
0	Overrun 0
1	Overrun 1

On-Chip Peripheral Module Request Mode: In this mode, the transfer is performed in to the transfer request signal of an on-chip peripheral module.

The DMA transfer request signals comprise the transmit data empty transfer request and r data full transfer request from the SCIF0, SCIF1, SIOF0, and SIOF1 set by DMARS0 to DMARS2.

When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, A NMIF = 0), a transfer is performed upon the input of a transfer request signal. When a tra request is set to TXI of the SCIF0, the transfer destination must be the SCIF0's transmit d

1000	001000	01	SCIF0 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR_0
		10	SCIF0 receiver	RXI (receive FIFO data full interrupt)	SCFRDR_0	Any
	001010	01	SCIF1 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR_1
		10	SCIF1 receiver	RXI (receive FIFO data full interrupt)	SCFRDR_1	Any
	010100	01	SIOF0 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SITDR_0
		10	SIOF0 receiver	RXI (receive FIFO data full interrupt)	SIOF0/ SIRDR_0	Any
	010101	01	SIOF1 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SITDR_1
		10	SIOF1 receiver	RXI (receive FIFO data full interrupt)	SIOF1/ SIRDR_1	Any

13.4.3 Channel Priority

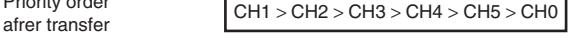
When the DMAC receives simultaneous transfer requests on two or more channels, it se
channel according to a predetermined priority order. The two modes (fixed mode and ro
mode) can be selected using bits PR0 and PR1 in DMAOR.

Fixed Mode: In this mode, the priority levels among the channels remain fixed. There a
kinds of fixed modes as follows:

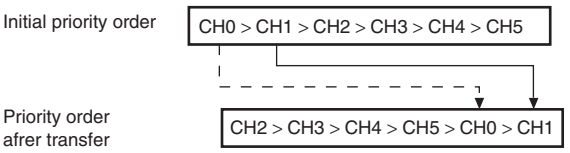
Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5

Fixed mode 2: CH0 > CH2 > CH3 > CH1 > CH4 > CH5

These are selected by the PR1 and the PR0 bits in DMAOR.

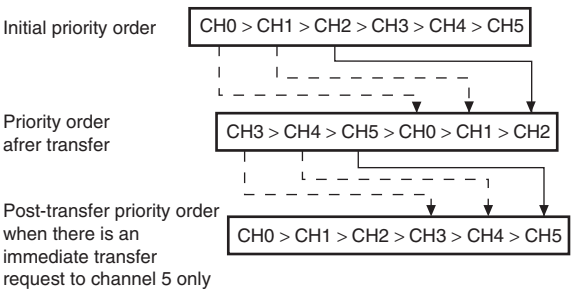


(2) When channel 1 transfers



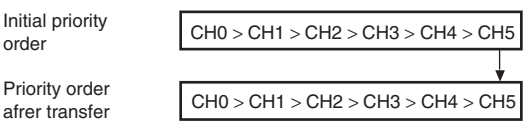
Channel 1 becomes bottom priority.
The priority of channel 0, which was higher than channel 1, is also shifted.

(3) When channel 2 transfers



Channel 2 becomes bottom priority.
The priority of channels 0 and 1, which were higher than channel are also shifted. If immediately after there is a request to transfe channel 5 only, channel 5 becom bottom priority and the priority of channels 3 and 4, which were higher than channel 5, are also shifted.

(4) When channel 5 transfers



Priority order does not change

Figure 13.3 Round-Robin Mode

5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfe (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 tha 3 becomes the lowest priority.

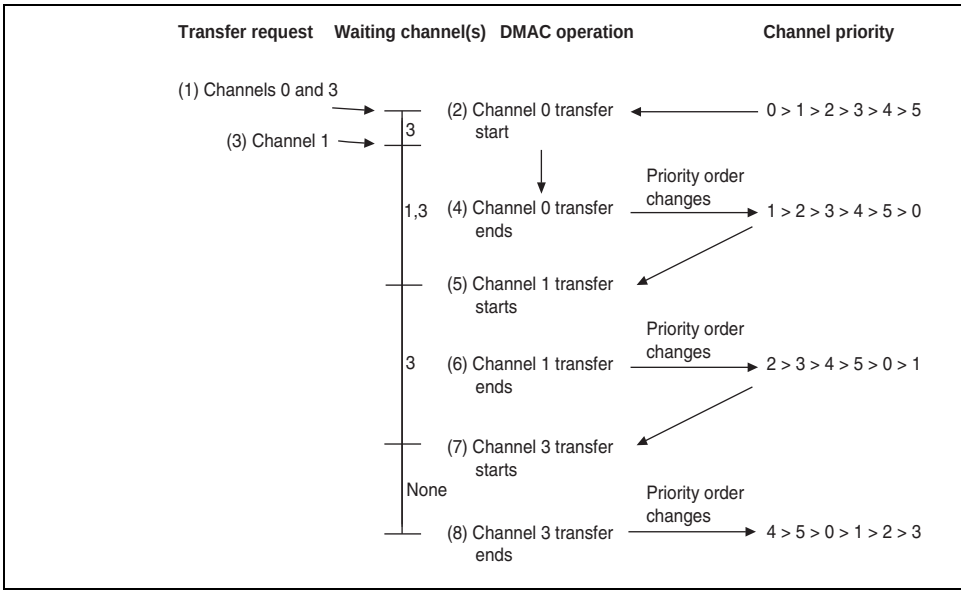


Figure 13.4 Changes in Channel Priority in Round-Robin Mode

Source	External Device with DACK	External Memory	Memory-Mapped External Device	On-Chip Peripheral Module	X/Y I
External device with DACK	Not available	Single	Single	Not available	Not a
External memory	Single	Dual	Dual	Dual	Dual
Memory-mapped external device	Single	Dual	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual	Dual
X/Y memory	Not available	Dual	Dual	Dual	Dual

Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. 16-byte transfer is not available for on-chip peripheral modules.

Address Modes:

1. Dual Address Mode

   In the dual address mode, both the transfer source and destination are accessed (select 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 read cycle and written to the transfer destination in a data write cycle. At this time, tra data is temporarily stored in the DMAC. In the transfer between external memories as in figure 13.5, data is read to the DMAC from one external memory in a data read cyc then that data is written to the other external memory in a write cycle.

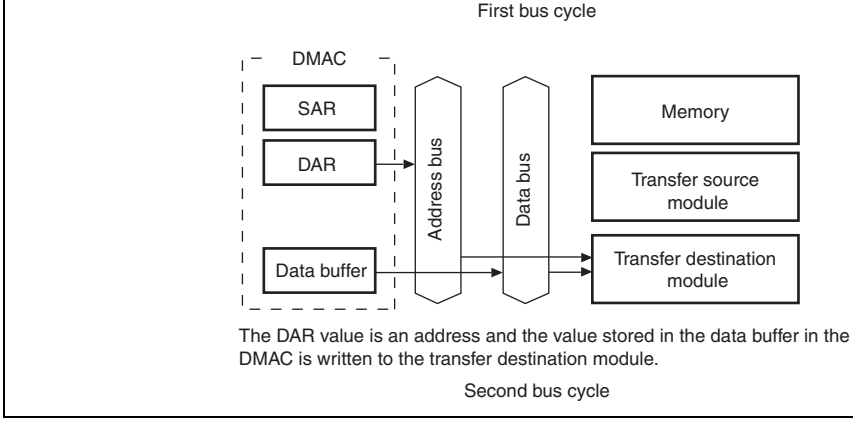
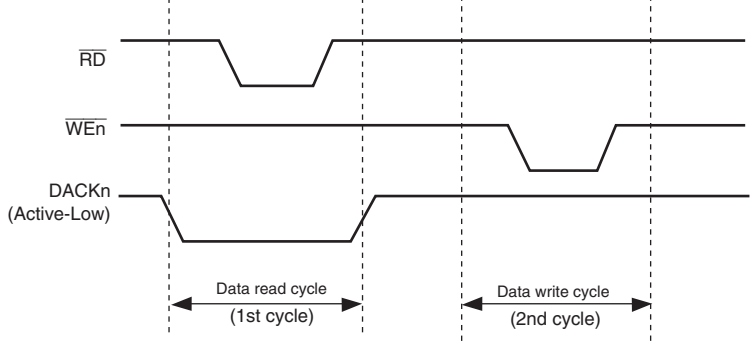


Figure 13.5 Data Flow in Dual Address Mode

Auto request, external request, and on-chip peripheral module request are available f transfer request. DACK can be output in read cycle or write cycle in dual address mo AM bit of the channel control register (CHCR) can specify whether the DACK is ou read cycle or write cycle.

Figure 13.6 shows an example of DMA transfer timing in dual address mode.



Note: In transfer between external memories, with DACK output in the read cycle, DACK output timing is the same as that of CSn.

Figure 13.6 Example of DMA Transfer Timing in Dual Address Mode (Source: O memory, Destination: Ordinary memory)

2. Single Address Mode

   In single address mode, either the transfer source or transfer destination external devic accessed (selected) by means of the DACK signal, and the other device is accessed by In this mode, the DMAC performs one DMA transfer in one bus cycle, accessing one external devices by outputting the DACK transfer request acknowledge signal to it, ar same time outputting an address to the other device involved in the transfer. For exam the case of transfer between external memory and an external device with DACK sho figure 13.7, when the external device outputs data to the data bus, that data is written external memory in the same bus cycle.

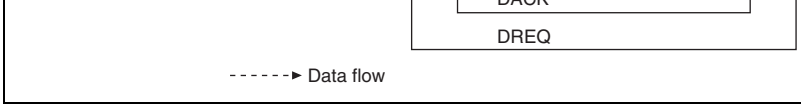
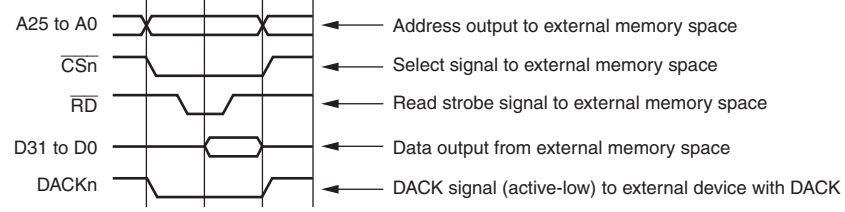


Figure 13.7 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an ex device with DACK and a memory-mapped external device, and (2) transfer between external device with DACK and external memory. In both cases, only the external re signal (DREQ) is used for transfer requests.

Figures 13.8 shows example of DMA transfer timing in single address mode.



(b) External memory space (ordinary memory) → external device with DACK

Figure 13.8 Example of DMA Transfer Timing in Single Address Mode

Bus Modes: There are two bus modes: cycle steal and burst. Select the mode in the TB b channel control register (CHCR).

- Cycle-Steal Mode

  In the cycle-steal mode, the bus mastership is given to another bus master after a one- unit (byte, word, long-word, or 16 bytes unit) DMA transfer. When another transfer re occurs, the bus masterships are obtained from the other bus master and a transfer is pe for one transfer unit. When that transfer ends, the bus mastership is passed to the othe master. This is repeated until the transfer end conditions are satisfied.

  In the cycle-steal mode, transfer areas are not affected regardless of settings of the tra request source, transfer source, and transfer destination.

Figure 13.9 shows an example of DMA transfer timing in the cycle steal mode. Transfer conditions shown in the figure are:

1. Dual address mode
2. DREQ low level detection

until the transfer end condition is satisfied. In the external request mode with low le
detection of the DREQ pin, however, when the DREQ pin is driven high, the bus pas
other bus master after the DMAC transfer request that has already been accepted end
the transfer end conditions have not been satisfied.

The burst mode cannot be used when the on-chip peripheral module is the transfer re
source. Figure 13.10 shows DMA transfer timing in burst mode.

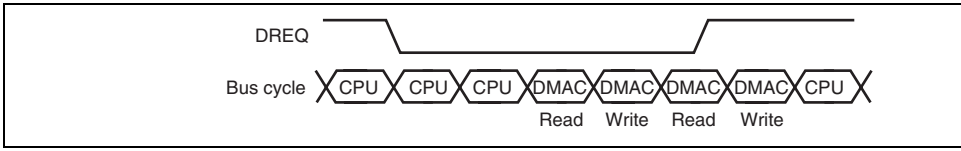


Figure 13.10 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

Relationship between Request Modes and Bus Modes by DMA Transfer Category:
13.8 shows the relationship between request modes and bus modes by DMA transfer cat

	Memory-mapped external device and memory-mapped external device	External, auto	B/C	8/16/32/128	0
	External memory and on-chip peripheral module	All*[1]	C	8/16/32*[3]	0
	Memory-mapped external device and on-chip peripheral module	All*[1]	C	8/16/32*[3]	0
	On-chip peripheral module and on-chip peripheral module	All*[1]	C	8/16/32*[3]	0
	X/Y memory and X/Y memory	External, auto	B/C	8/16/32/128	0
	X/Y memory and memory-mapped external device	External, auto	B/C	8/16/32/128	0
	X/Y memory and on-chip peripheral module	All*[1]	C	8/16/32*[3]	0
	X/Y memory and external memory	External, auto	B/C	8/16/32/128	0
Single	External device with DACK and external memory	External	B/C	8/16/32	0
	External device with DACK and memory-mapped external device	External	B/C	8/16/32	0

[Legend] B: Burst, C: Cycle steal

Notes: 1. External requests, auto requests, and on-chip peripheral module requests are available. However, the request-source register must be designated as the tra source or the transfer destination.

2. If the transfer request is an external request, channels 0 and 1 are only availab

3. Access size permitted for each module must be used when accessing the on-c peripheral module.

the burst-mode transfer. (This operation is hereinafter referred to as burst-mode priority execution.) Figure 13.11 shows an example.

If multiple channels are conflicting in burst mode, the channel with the highest priority i for execution.

If multiple channels perform DMA transfers, bus mastership is not released to the bus m all conflicting burst transfers are completed.

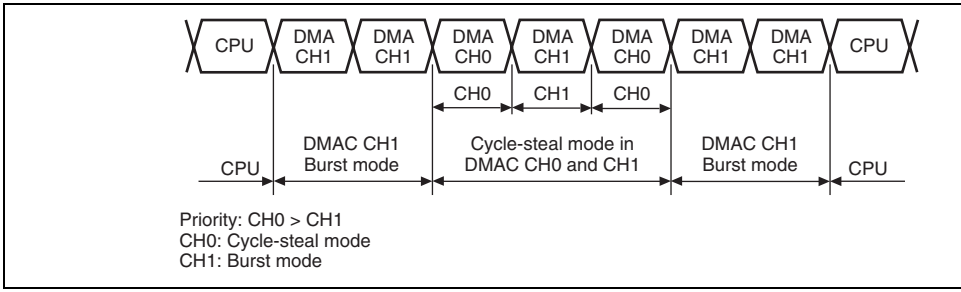


Figure 13.11 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes according to the specification shown in figure However, no mixture of channels in cycle-steal mode and channels in burst mode is allo

13.4.5 Number of Bus Cycle States and DREQ Pin Sampling Timing

Number of Bus Cycle States: When the DMAC is the bus master, the number of bus cy is controlled by the bus state controller (BSC) in the same way as when the CPU is the b For details, see section 12, Bus State Controller (BSC).

Figure 13.12 Example of DREQ Input Detection in Cycle Steal Mode Edge Dete

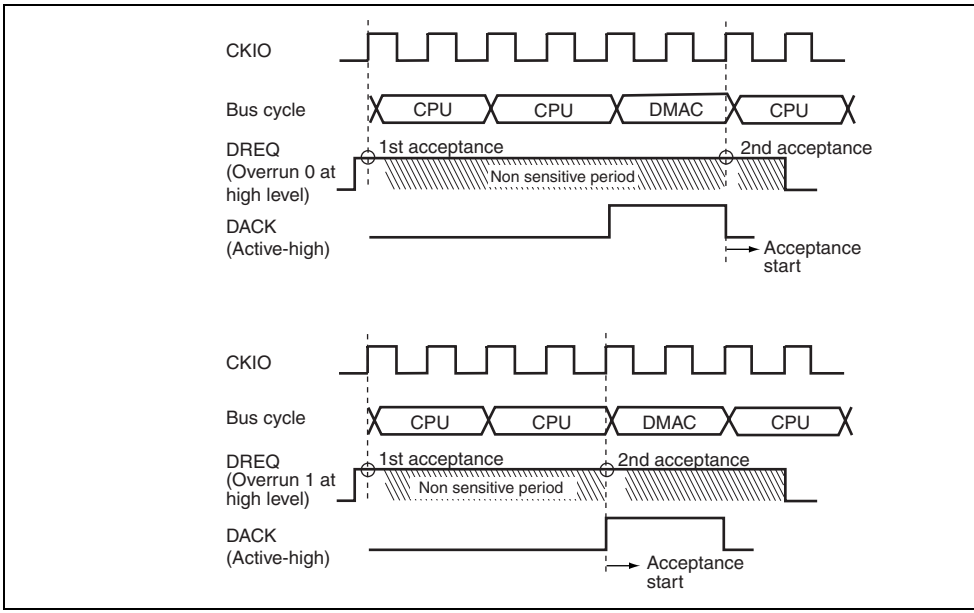


Figure 13.13 Example of DREQ Input Detection in Cycle Steal Mode Level Dete

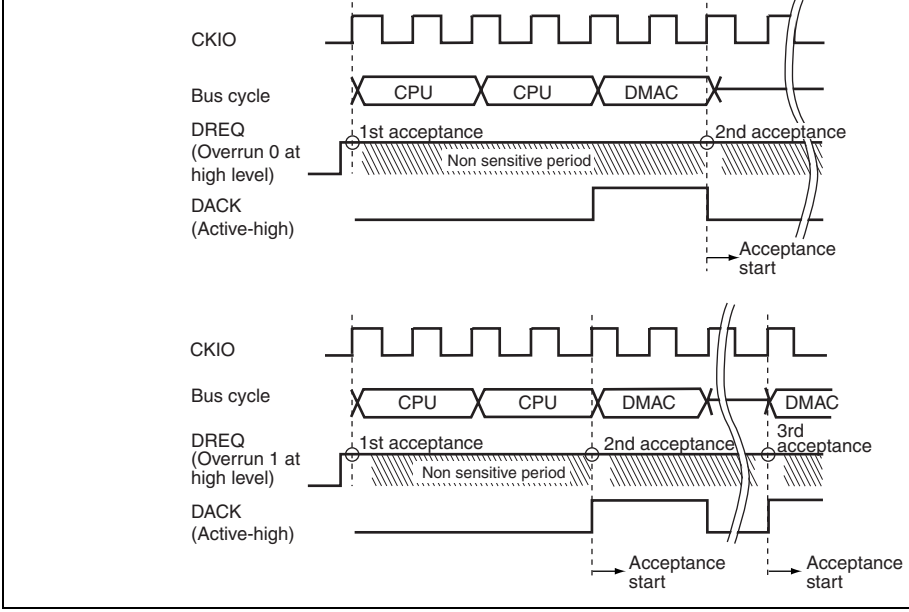


Figure 13.15 Example of DREQ Input Detection in Burst Mode Level Detec

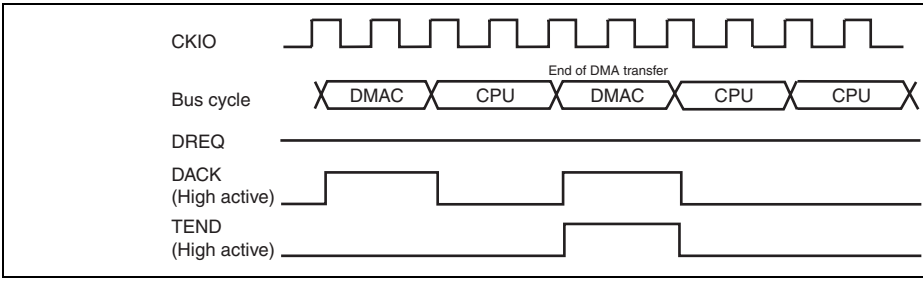
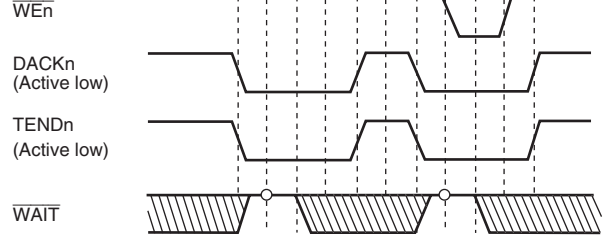


Figure 13.16 Example of DMA Transfer End Timing (Cycle Steal Level Detec



Note: TEND is asserted for the last transfer unit of DMA transfers.
If a transfer unit is divided into multiple bus cycles and
if CS is negated during the bus cycle, TEND is also divided.

Figure 13.17 Example of BSC Ordinary Memory Access (No Wait, Idle Cycle = 1, Longword Access to 16-bit Device)

Conditions:

- DACK is output in a dual address mode read cycle (with the AM bit in CHCR cleare and the DMA transfer source address (SAR) is in external memory space.
- DACK is output in a dual address mode write cycle (with the AM bit in CHCR set to the DMA transfer destination address (DAR) is in external memory space.
- Single address mode

Method of Avoidance:

Perform a dummy DMA transfer under one of the settings below. After start of a dummy transfer, clear all bits in the DMA channel control register (CHCR) of the corresponding to suspend the dummy DMA transfer forcibly.

- DACK is output in a dual address mode read cycle (with the AM bit in CHCR cleare and the DMA transfer source address (SAR) is in external memory space.
- DACK is output in a dual address mode write cycle (with the AM bit in CHCR set to the DMA transfer destination address (DAR) is in external memory space.

reload function that can be read or written to at any time

- All channels generate interrupt requests when the 32-bit down counter underflows (H'00000000 → H'FFFFFFFF)
- Allows selection among 4 counter input clocks: Pϕ/4, Pϕ/16, Pϕ/64, and Pϕ/256

Note: Pϕ is the internal clock for peripheral modules. See section 11, On-Chip Oscillat Circuits, for more information on the clock pulse generator.

14.1.1 Block Diagram

Figure 14.1 shows a block diagram of the TMU.

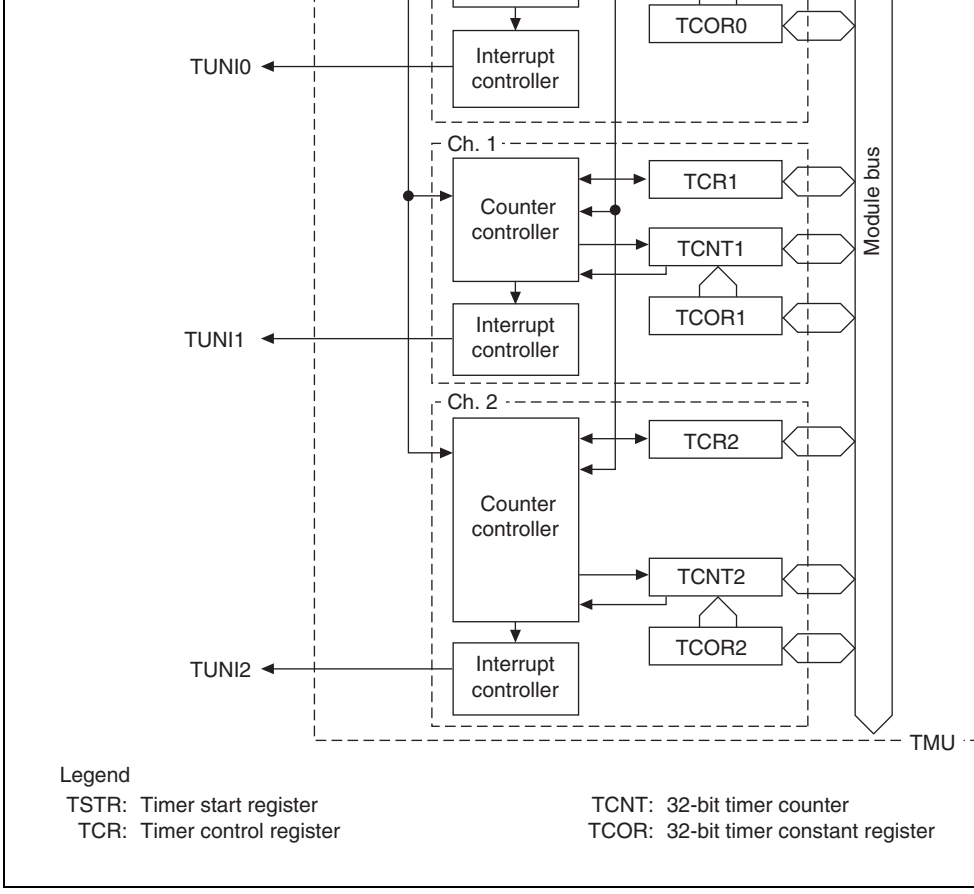


Figure 14.1 TMU Block Diagram

- Timer counter 1 (TCNT1)
- Timer control register 1 (TCR1)
- Timer constant register 2 (TCOR2)
- Timer counter 2 (TCNT2)
- Timer control register 2 (TCR2)

14.2.1 Timer Start Register (TSTR)

The timer start register (TSTR) selects whether to run or halt the timer counters (TCNT) channels 0 to 2.

TSTR is an 8-bit readable/writable register. It is initialized to H'00 by a power-on reset o reset. It is initialized in standby mode when the multiplication ratio of PLL circuit 1 (PL changed or when the MSTP2 bit in STBCR is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
2	STR2	0	R/W	Counter Start 2 Selects whether to run or halt TCNT2. 0: TCNT2 count halted 1: TCNT2 counts
1	STR1	0	R/W	Counter Start 1 Selects whether to run or halt TCNT1. 0: TCNT1 count halted 1: TCNT1 counts

three TCR registers, one for each channel.

The TCR registers control the issuance of interrupts when the flag indicating timer counte (TCNT) underflow has been set to 1, and also carry out counter clock selection.

The TCR registers are 16-bit readable/writable registers. They are initialized to H'0000 by power-on reset and manual reset, but are not initialized, and retain their contents, in stand

Bit	Bit Name	Initial Value	R/W	Description
15 or 9	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
8	UNF	0	R/W	Underflow Flag Status flag that indicates occurrence of a TCN underflow. 0: TCNT has not underflowed [Clearing condition] 0 is written to UNF 1: TCNT has underflowed [Setting condition] TCNT underflows* Note: * Contents do not change when 1 is writt UNF.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.

2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	Select the TCNT count clock.
0	TPSC0	0	R/W	000: Count on Pϕ/4
				001: Count on Pϕ/16
				010: Count on Pϕ/64
				011: Count on Pϕ/256
				100: Reserved (Setting prohibited)
				101: Reserved (Setting prohibited)
				110: Reserved (Setting prohibited)
				111: Reserved (Setting prohibited)

14.2.3 Timer Constant Registers (TCOR)

The TMU has three timer constant registers (TCOR), one for each channel. The TCOR r
set the value to be set in TCNT when TCNT underflows.

The TCOR registers are 32-bit readable/writable registers. They are initialized to H'FFF
a power-on reset or manual reset, but are not initialized, and retain their contents, in stan
mode.

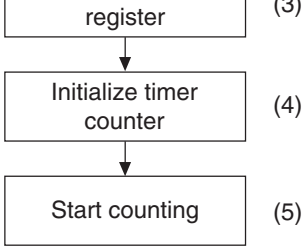
14.2.4 Timer Counters (TCNT)

The TMU has three timer counters (TCNT), one for each channel. The timer counters (T
counts down upon input of a clock. The input clock is selected using the TPSC2 to TPS
the timer control registers (TCR).

14.3.1 Counter Operation

When the STR0 to STR2 bits in TSTR are set to 1, the corresponding TCNT starts count
When TCNT underflows, the underflow flag (UNF) of the corresponding TCR is set. At
if the UNIE bit in TCR is 1, an interrupt request is sent to the CPU. Also at this time, the
copied from TCOR to TCNT and the down-count operation is continued.

Count Operation Setting Procedure: An example of the procedure for setting the count
operation is shown in figure 14.2.



1).
(4) Set the initial value in TCNT.
(5) Set the STR bit in TST to 1 to start operation.

Note: When an interrupt has been generated, clear the flag in the interrupt handler that
If interrupts are enabled without clearing the flag, another interrupt will be generat

Figure 14.2 Setting Count Operation

Auto-Reload Count Operation: Figure 14.3 shows the TCNT auto-reload operation.

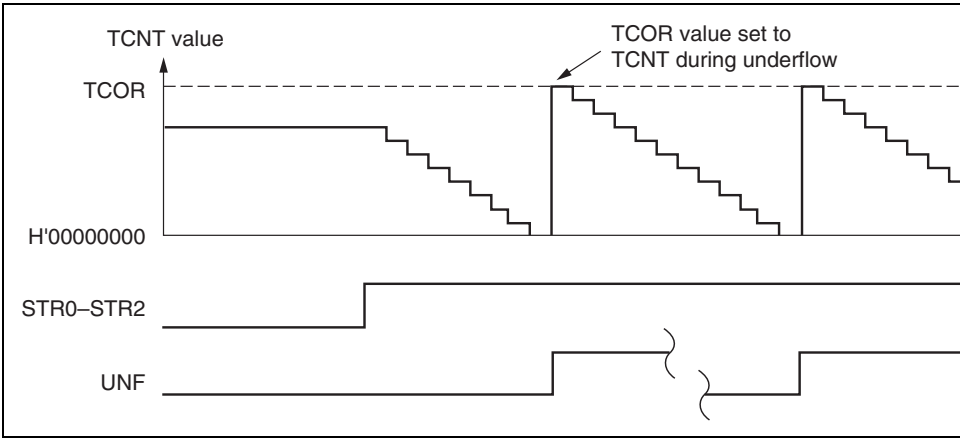


Figure 14.3 Auto-Reload Count Operation

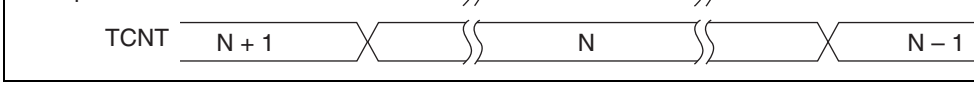


Figure 14.4 Count Timing when Internal Clock Is Operating

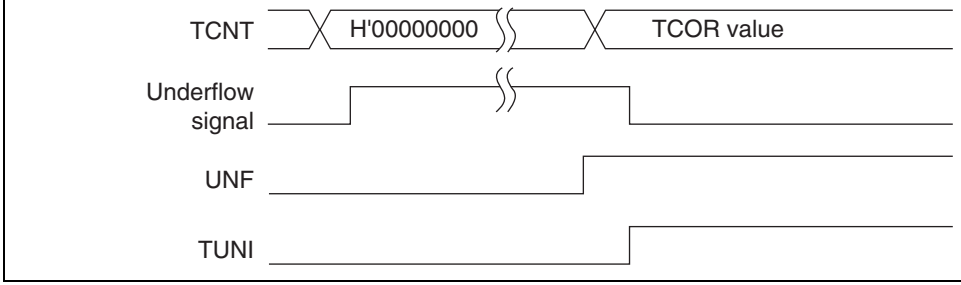


Figure 14.5 UNF Set Timing

14.4.2 Status Flag Clear Timing

The status flag can be cleared by writing 0 from the CPU. Figure 14.6 shows the timing.

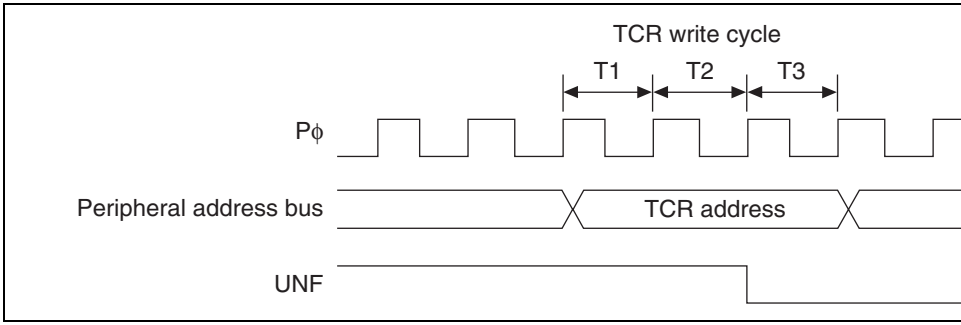


Figure 14.6 Status Flag Clear Timing

Channel	Interrupt Source	Description	Priority
0	TUNI0	Underflow interrupt 0	High
1	TUNI1	Underflow interrupt 1	↕
2	TUNI2	Underflow interrupt 2	Low

14.5 Usage Notes

14.5.1 Writing to Registers

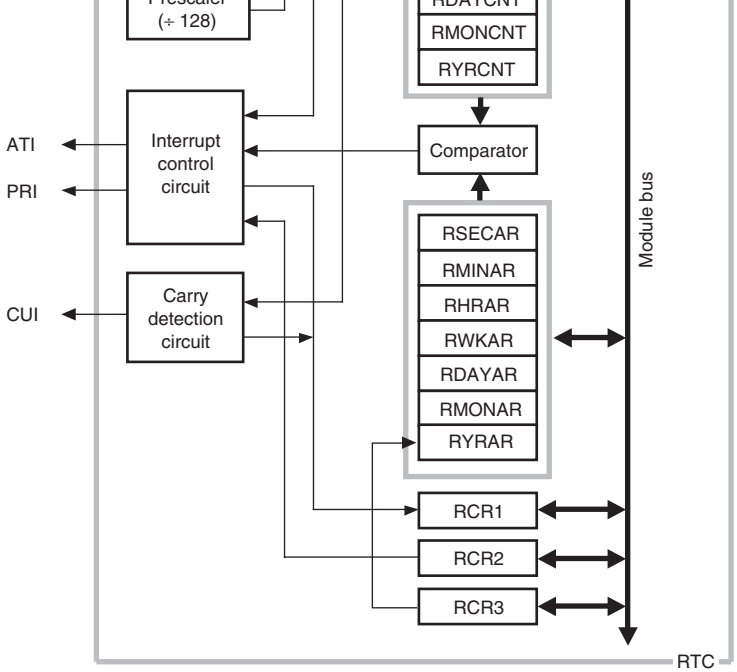
Synchronization processing is not performed for timer counting during register writes. W
writing to registers, always clear the appropriate start bits for the channel (STR2 to STR0
TSTR to halt timer counting.

14.5.2 Reading Registers

Synchronization processing is performed for timer counting during register reads. When t
counting and register read processing are performed simultaneously, the register value be
TCNT counting down (with synchronization processing) is read.

month, and year

- 1-Hz to 64-Hz timer (binary format)
- Start/stop function
- 30-second adjust function
- Alarm interrupt: Frame comparison of seconds, minutes, hours, date, day of the week and year can be used as conditions for the alarm interrupt
- Periodic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year adjustment



[Legend]

R64CNT:	64-Hz counter	RMINAR:	Minute alarm register
RSECCNT:	Second counter	RHRAR:	Hour alarm register
RMINCNT:	Minute counter	RWKAR:	Day of the week alarm register
RHRCNT:	Hour counter	RDAYAR:	Date alarm register
RWKCNT:	Day of the week counter	RMONAR:	Month alarm register
RDAYCNT:	Date counter	RYRAR:	Year alarm register
RMONCNT:	Month counter	RCR1:	RTC control register 1
RYRCNT:	Year counter	RCR2:	RTC control register 2
RSECAR:	Second alarm register	RCR3:	RTC control register 3

Figure 15.1 RTC Block Diagram

Note: 1. Pull up (VccQ-RTC) the EXTAL2 pin, and open (NC) the XTAL2 pin when the clock (RTC) is not used.
2. RTC in this LSI does not operate even if VccQ-RTC is turned on. The crystal circuit for RTC operates with VccQ-RTC. The control circuit and the RTC cou operate with Vcc (common to the internal circuit). Therefore, all power supplie than VccQ-RTC should always be turned on even if only RTC operates.

15.3 Register Descriptions

The RTC has the following registers. Refer to section 23, List of Registers, for more det address and access size.

- 64-Hz counter (R64CNT)
- Second counter (RSECCNT)
- Minute counter (RMINCNT)
- Hour counter (RHRCNT)
- Day of week counter (RWKCNT)
- Date counter (RDAYCNT)
- Month counter (RMONCNT)
- Year counter (RYRCNT)
- Second alarm register (RSECAR)
- Minute alarm register (RMINAR)
- Hour alarm register (RHRAR)
- Day of week alarm register (RWKAR)
- Date alarm register (RDAYAR)
- Month alarm register (RMONAR)
- Year alarm register (RYRAR)

R64CNT is an 8-bit read-only register and not initialized by a power-on reset or manual r
in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.
6 to 0	—	—	R	64-Hz Counter Each bit (bits 6 to 0) indicates the state o RTC divider circuit between 64 and 1Hz. Bit Frequency 6: 1 Hz 5: 2 Hz 4: 4 Hz 3: 8 Hz 2: 16 Hz 1: 32 Hz 0: 64 Hz

15.3.2 Second Counter (RSECCNT)

RSECCNT is used for setting/counting in the BCD-coded second section. The count oper
performed by a carry for each second of the 64-Hz counter.

The range of second can be set is 0 to 59 (decimal). Errant operation will result if any oth
is set. Carry out write processing after stopping the count operation with the START bit i

15.3.3 Minute Counter (RMINCNT)

RMINCNT is used for setting/counting in the BCD-coded minute section. The count op performed by a carry for each minute of the second counter.

The range of minute can be set is 0 to 59 (decimal). Errant operation will result if any ot is set. Carry out write processing after stopping the count operation with the START bit

RMINCNT is an 8-bit readable/writable register and not initialized by a power-on reset reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.The write va should always be 0.
6 to 4	—	—	R/W	Counter for 10-unit of minute in the BCI The range can be set from 0 to 5 (decim
3 to 0	—	—	R/W	Counter for 1-unit of minute in the BCD The range can be set from 0 to 9 (decim

15.3.4 Hour Counter (RHRCNT)

RHRCNT is used for setting/counting in the BCD-coded hour section. The count operat performed by a carry for each 1 hour of the minute counter.

				The range can be set from 0 to 2 (decima
3 to 0	—	—	R/W	Counter for 1-unit of hour in the BCD-cod
				The range can be set from 0 to 9 (decima

15.3.5 Day of Week Counter (RWKCNT)

RWKCNT is used for setting/counting day of week section. The count operation is perfor carry for each day of the date counter.

The range for day of the week can be set is 0 to 6 (decimal). Errant operation will result i other value is set. Carry out write processing after stopping the count operation with the S bit in RCR2.

RWKCNT is an 8-bit readable/writable register and not initialized by a power-on reset or reset, or in standby mode.

3:	Wednesday
4:	Thursday
5:	Friday
6:	Saturday

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The wri should always be 0.
5, 4	—	—	R/W	Counter for 10-unit of date in the BCD-co The range can be set from 0 to 3 (decim
3 to 0	—	—	R/W	Counter for 1-unit of date in the BCD-coc The range can be set from 0 to 9 (decim

15.3.7 Month Counter (RMONCNT)

RMONCNT is used for setting/counting in the BCD-coded month section. The count ope performed by a carry for each month of the date counter.

The range of month can be set is 1 to 12 (decimal). Errant operation will result if any othe is set. Carry out write processing after stopping the count operation with the START bit i

RMONCNT is an 8-bit readable/writable register and not initialized by a power-on reset manual reset, or in standby mode.

15.3.8 Year Counter (RYRCNT)

RYRCNT is used for setting/counting in the BCD-coded year section. The 4 digits of th
displayed. The count operation is performed by a carry for each year of the month count

The range for year which can be set is 0000 to 9999 (decimal). Errant operation will resu
other value is set. Carry out write processing after stopping the count operation with the
bit in RCR2 or using a carry flag.

RYRCNT is a 16-bit readable/writable register and not initialized by a power-on reset o
reset, or in standby mode.

Leap years are recognized by dividing the year counter value by 4 and obtaining a fracti
of 0. The year counter value of 0000 is included in the leap year.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	—	R/W	Counter for 1000-unit of year in the BCI
				The range can be set from 0 to 9 (decin
11 to 8	—	—	R/W	Counter for 100-unit of year in the BCD
				The range can be set from 0 to 9 (decin
7 to 4	—	—	R/W	Counter for 10-unit of year in the BCD-c
				The range can be set from 0 to 9 (decin
3 to 0	—	—	R/W	Counter for 1-unit of year in the BCD-co
				The range can be set from 0 to 9 (decin

RSECAR is an 8-bit readable/writable register. The ENB bit in RSECAR is initialized to
power-on reset. The remaining RSECAR fields are not initialized by a power-on reset or
reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Second Alarm Enable Specifies whether comparison of RSEC RSECAR is performed as an alarm cond 0: Not compared 1: Compared
6 to 4	—	—	R/W	Setting value for 10-unit of second alarm BCD-code. The range can be set from 0 to 5 (decim
3 to 0	—	—	R/W	Setting value for 1-unit of second alarm i BCD-code. The range can be set from 0 to 9 (decim

15.3.10 Minute Alarm Register (RMINAR)

RMINAR is an alarm register corresponding to the minute counter RMINCNT. When the
is set to 1, a comparison with the RMINCNT value is performed. From among
RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR, the counter and alarm reg
comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to 1
each of those coincide, an RTC alarm interrupt is generated.

The range of minute alarm which can be set is 0 to 59 (decimal). Errant operation will res
other value is set.

6 to 4	—	—	R/W	Setting value for 10-unit of minute alarm BCD-code. The range can be set from 0 to 5 (decim
3 to 0	—	—	R/W	Setting value for 1-unit of minute alarm BCD-code. The range can be set from 0 to 9 (decim

15.3.11 Hour Alarm Register (RHRAR)

RHRAR is an alarm register corresponding to the hour counter RHRCNT of the RTC. W
ENB bit is set to 1, a comparison with the RHRCNT value is performed. From among
RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR, the counter and alarm re
comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to
each of those coincide, an RTC alarm interrupt is generated.

The range of hour alarm which can be set is 0 to 23 (decimal). Errant operation will resu
other value is set.

RHRAR is an 8-bit readable/writable register. The ENB bit in RHRAR is initialized by
on reset. The remaining RHRAR fields are not initialized by a power-on reset or manual
in standby mode.

				BCD-code. The range can be set from 0 to 2 (decim
3 to 0	—	—	R/W	Setting value for 1-unit of hour alarm in t code. The range can be set from 0 to 9 (decim

15.3.12 Day of Week Alarm Register (RWKAR)

RWKAR is an alarm register corresponding to the day of week counter RWKCNT. Wher ENB bit is set to 1, a comparison with the RWKCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR, the counter and alarm reg comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to each of those coincide, an RTC alarm interrupt is generated.

The range of day of the week alarm which can be set is 0 to 6 (decimal). Errant operation result if any other value is set.

RWKAR is an 8-bit readable/writable register. The ENB bit in RWKAR is initialized by on reset. The remaining RWKAR fields are not initialized by a power-on reset or manual in standby mode.

The range can be set from 0 to 6 (decim

Code	Day of the Week
0:	Sunday
1:	Monday
2:	Tuesday
3:	Wednesday
4:	Thursday
5:	Friday
6:	Saturday

15.3.13 Date Alarm Register (RDAYAR)

RDAYAR is an alarm register corresponding to the date counter RDAYCNT. When the is set to 1, a comparison with the RDAYCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/ and RMONAR, the counter and alar comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to each of those coincide, an RTC alarm interrupt is generated.

The range of date alarm which can be set is 1 to 31 (decimal). Errant operation will resu other value is set. The RDAYCNT range that can be set changes with some months and years. Please confirm the correct setting.

RDAYAR is an 8-bit readable/writable register. The ENB bit in RDAYAR is initialized power-on reset. The remaining RDAYAR fields are not initialized by a power-on reset o reset, or in standby mode.

				BCD-code. The range can be set from 0 to 3 (decim
3 to 0	—	—	R/W	Setting value for 1-unit of date alarm in t code. The range can be set from 0 to 9 (decim

15.3.14 Month Alarm Register (RMONAR)

RMONAR is an alarm register corresponding to the month counter RMONCNT. When t
bit is set to 1, a comparison with the RMONCNT value is performed. From among
RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/ and RMONAR, the counter and alar
comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to
each of those coincide, an RTC alarm interrupt is generated.

The range of month alarm which can be set is 1 to 12 (decimal). Errant operation will res
other value is set.

RMONAR is an 8-bit readable/writable register. The ENB bit in RMONAR is initialized
power-on reset. The remaining RMONAR fields are not initialized by a power-on reset o
reset, or in standby mode.

				BCD-code. The range can be set from 0 to 1 (decim
3 to 0	—	—	R/W	Setting value for 1-unit of month alarm i BCD-code. The range can be set from 0 to 9 (decim

15.3.15 Year Alarm Register (RYRAR)

RYRAR is an alarm register corresponding to the year counter RYRCNT. When the YA
RCR3 is set to 1, a comparison with the RYRCNT value is performed. From among
RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/ and RMONAR, the counter and alar
comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to
each of those coincide, an RTC alarm interrupt is generated.

The range of year alarm which can be set is 0000 to 9999 (decimal). Errant operation wi
any other value is set.

RYRAR is a 16-bit readable/writable register. The contents are not initialized by a powe
or manual reset, or in standby mode.

3 to 0	—	—	R/W	Setting value for 1-unit of year alarm in tl code. The range can be set from 0 to 9 (decim

15.3.16 RTC Control Register 1 (RCR1)

RCR1 is a register that affects carry flags and alarm flags. It also selects whether to gener
interrupts for each flag. Because flags are sometimes set after an operand read, do not use
register in read-modify-write processing.

RCR1 is an 8-bit readable/writable register. RCR1 is initialized to H'00 by a power-on re
manual reset, all bits are initialized to 0 except for the CF flag, which is undefined. When
the CF flag, it must be initialized beforehand. This register is not initialized in standby m

Bit	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag Status flag that indicates that a carry has occurred. CF is set to 1 when a count-up R64CNT or RSECCNT occurs. A count r value read at this time cannot be guaran another read is required. 0: No count up of R64CNT or RSECCNT Clearing condition: When 0 is written t 1: Count up of R64CNT or RSECCNT. Setting condition: When 1 is written to the carry of R64CNT or RSECCNT oc when R64CNT or RSECCNT is read.

				is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag When the alarm flag (AF) is set to 1, the allows interrupts. 0: An alarm interrupt is not generated w AF flag is set to 1 1: An alarm interrupt is generated when flag is set to 1
2	—	0	R	Reserved
1	—	0	R	These bits are always read as 0. The w should always be 0.
0	AF	0	R/W	Alarm Flag The AF flag is set to 1 when the alarm t an alarm register (only registers with the of the corresponding alarm registers an bit in RCR3 set to 1) matches the clock calendar time. This flag is cleared to 0 w written, but holds the previous value wh be written. 0: Clock/calendar and alarm register ha matched. Clearing condition: When 0 is written 1: Clock/calendar and alarm register ha matched. Setting condition: Clock/calendar and register have matched (only registers ENB bit and YAEN bit in RCR3 set to

				Indicates interrupt generation with the pe designated by the PES2 to PES0 bits. W to 1, PEF generates periodic interrupts. 0: Interrupts not generated with the perio designated by the PES bits. Clearing condition: When 0 is written t 1: Interrupts generated with the period designated by the PES bits. Setting condition: When an interrupt is generated with the period designated PES0 to PES2 bits or when 1 is writte PEF flag
6	PES2	0	R/W	Periodic Interrupt Flags
5	PES1	0	R/W	These bits specify the periodic interrupt.
4	PES0	0	R/W	000: No periodic interrupts generated 001: Periodic interrupt generated every 1 second 010: Periodic interrupt generated every 1 second 011: Periodic interrupt generated every 1 second 100: Periodic interrupt generated every 1 second 101: Periodic interrupt generated every 1 second 110: Periodic interrupt generated every 1 111: Periodic interrupt generated every 2 seconds

				0: Runs normally. 1: 30-second adjustment.
1	RESET	0	R/W	Reset When 1 is written, initializes the divider (RTC prescaler and R64CNT). This bit reads 0. 0: Runs normally. 1: Divider circuit is reset.
0	START	1	R/W	Start Bit Halts and restarts the counter (clock). 0: Second/minute/hour/day/week/month counter halts. 1: Second/minute/hour/day/week/month counter runs normally. Note: The 64-Hz counter always runs u stopped with the RTCEN bit.

				(RYRCNT) is performed. From among R RMINAR, RHRAR, RWKAR, RDAYAR, a RMONAR, the counter and alarm registe comparison is performed only on those w bits set to 1, and if each of those coincid RTC alarm interrupt is generated.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The wri should always be 0.

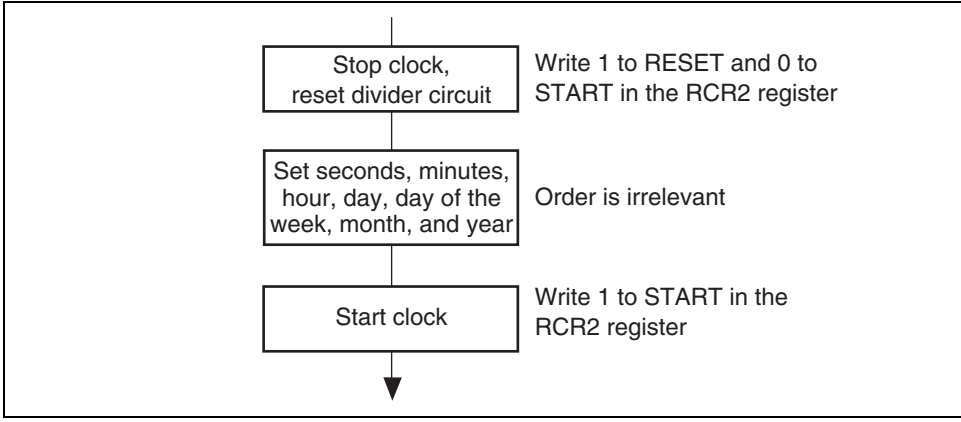


Figure 15.2 Setting Time

15.4.3 Reading Time

Figure 15.3 shows how to read the time.

If a carry occurs while reading the time, the correct time will not be obtained, so it must
again. Part (a) in figure 15.3 shows the method of reading the time without using interru
(b) in figure 15.3 shows the method using carry interrupts. To keep programming simple
(a) should normally be used.

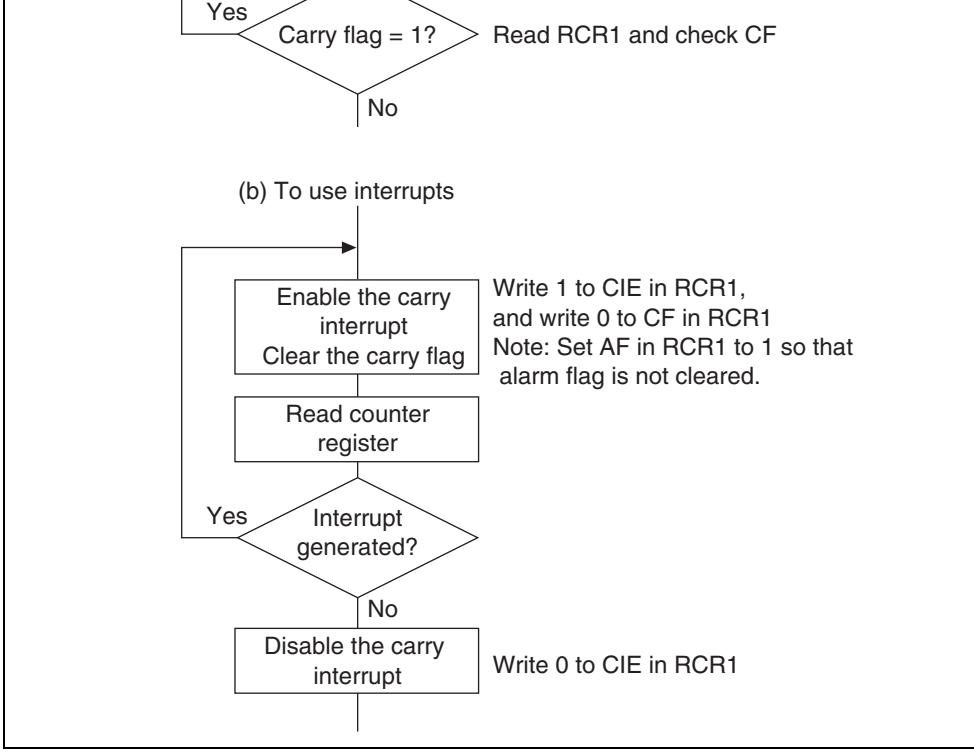


Figure 15.3 Reading Time

an interrupt is generated when an alarm occurs.

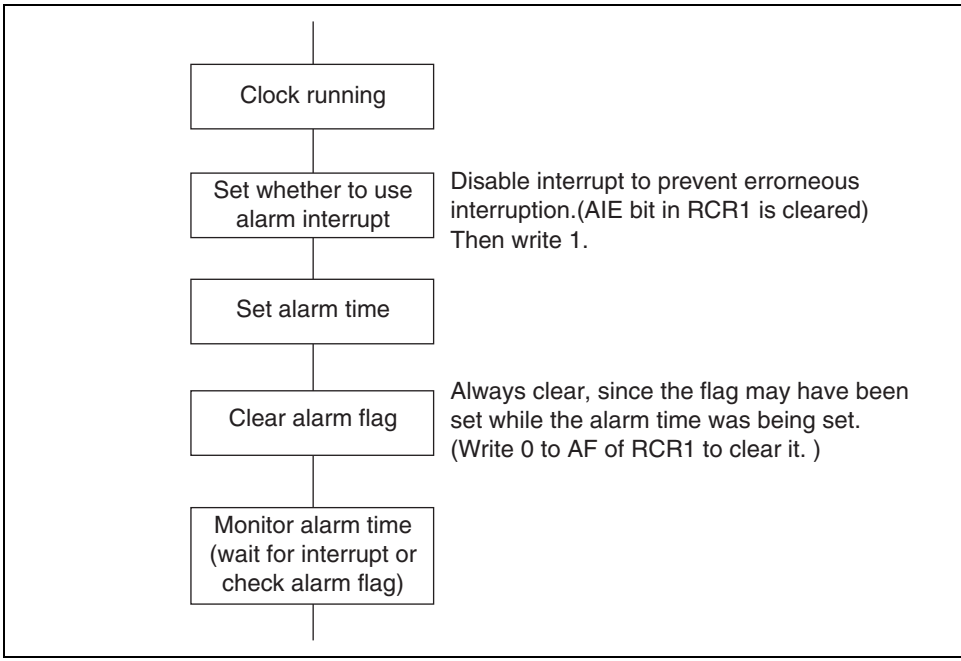
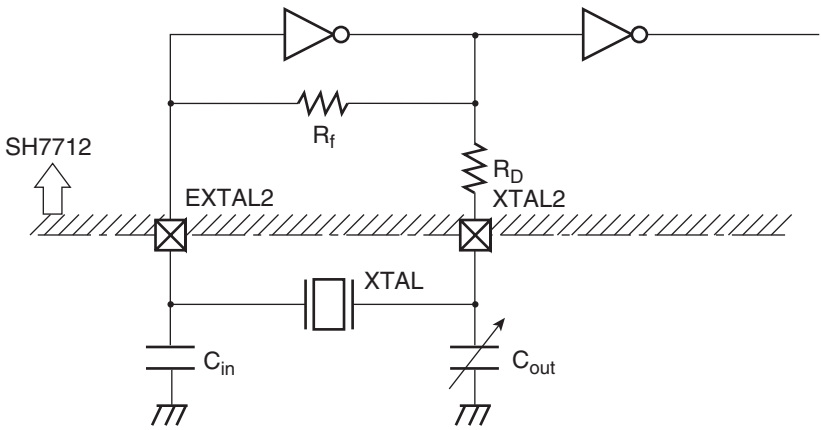


Figure 15.4 Using Alarm Function



Notes: 1. Select either the C_{in} or C_{out} side for frequency adjustment variable capacitor according to requirements such as frequency range, degree of stability, etc.
2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 kΩ
3. C_{in} and C_{out} values include stray capacitance due to the wiring. Take care w using a ground plane.
4. The crystal oscillation settling time depends on the mounted circuit constant stray capacitance, etc., and should be decided after consultation with the cr resonator manufacturer.
5. Place the crystal resonator and load capacitors C_{in} and C_{out} as close as pos to the chip.
(Correct oscillation may not be possible if there is externally induced noise i EXTAL2 and XTAL2 pins.)
6. Ensure that the crystal resonator connection pin (EXTAL2, XTAL2) wiring is routed as far away as possible from other power lines (except GND) and sig lines.

Figure 15.5 Example of Crystal Oscillator Circuit Connection

15.5.2 Use of Realtime Clock (RTC) Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 15.6.

A periodic interrupt can be generated periodically at the interval set by the periodic inter
(PES0–PES2) in RCR2. When the time set by the PES0–PES2 has elapsed, the PEF is s

The PEF is cleared to 0 upon periodic interrupt generation or when the periodic interrup
(PES0 to PES2) is set. Periodic interrupt generation can be confirmed by reading this bit
normally the interrupt function is used.

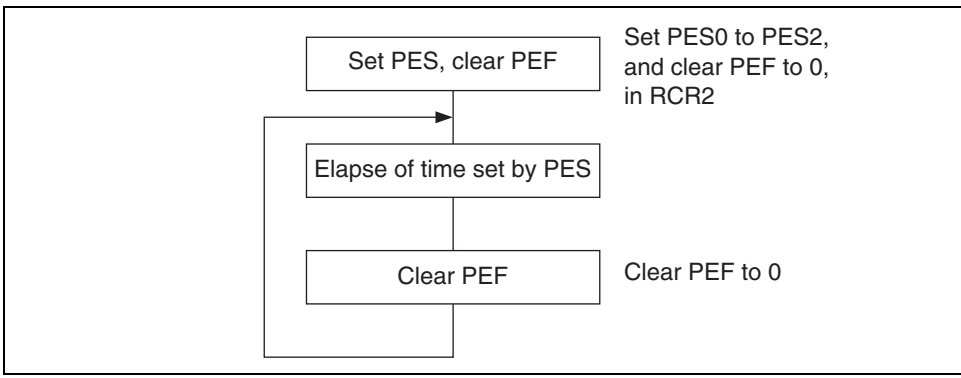


Figure 15.6 Using Periodic Interrupt Function

15.5.3 Transition to Standby Mode after Setting Register

When a transition to standby mode is made after registers in the RTC are set, sometimes
is not performed correctly. In case the registers are set, be sure to make a transition to st
mode after waiting for two RTC clocks or more.

16.1 Features

The SCIF features are listed below.

- Asynchronous mode

  Serial data communication is executed using an asynchronous system in which synchronization is achieved character by character. Serial data communication can b out with standard asynchronous communication chips such as a Universal Asynchro Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (

  There is a choice of 8 serial data communication formats.

  Data length: 7 or 8 bits

  Stop bit length: 1 or 2 bits

  Parity: Even/odd/none

  Receive error detection: Parity, framing, and overrun errors

  Break detection: If a framing error is following by at least one frame at the space “0” level, a break is detected.
- Clock synchronous mode

  Serial data communication is synchronized with a clock. Serial data communication carried out with other chips that have a synchronous communication function.

  Data length: 8 bits

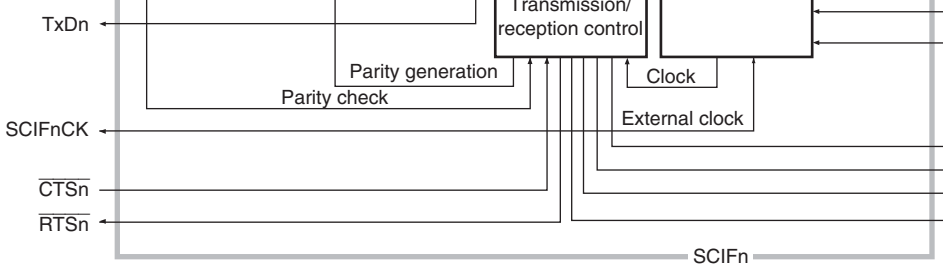
  Receive error detection: Overrun error
- Full-duplex communication capability

  The transmitter and receiver are independent units, enabling transmission and recept performed simultaneously.

  The transmitter and receiver both have a 16-stage FIFO buffer structure, enabling fas continuous serial data transmission and reception.
- On-chip baud rate generator allows any bit rate to be selected.

the receive data in the receive FIFO register, can be ascertained.

- On reception a time out error (DR) can be detected
- The contents of the transmit FIFO data register (SCFTDR) and receive FIFO data reg (SCFRDR) are undefined after a power-on or manual reset. Other registers are initiali power-on or manual reset, and retain their values in standby mode and in the module s state.



[Legend]

SCRSR_n:	Receive shift register	SCFSR_n:	Serial status register
SCFRDR_n:	Receive FIFO data register	SCBRR_n:	Bit rate register
SCTSR_n:	Transmit shift register	SCFCR_n:	FIFO control register
SCFTDR_n:	Transmit FIFO data register	SCFDR_n:	FIFO data count register
SCSMR_n:	Serial mode register	SCLSR_n:	Line status register
SCSCR_n:	Serial control register	n:	0, 1

Figure 16.1 Block Diagram of SCIF

	Modem control pin	CTS0	Input	Transmission clear
	Modem control pin	$\overline{\text{RTS0}}$	Output	Transmit request
1	Serial clock pin	SCIF1CK	Input/output	Clock input/output
	Receive data pin	RxD1	Input	Receive data input
	Transmit data pin	TxD1	Output	Transmit data outp
	Modem control pin	$\overline{\text{CTS1}}$	Input	Transmission clear
	Modem control pin	$\overline{\text{RTS1}}$	Output	Transmit request

Note: These pins function as serial pins by making the SCIF operation settings with the C
SCSMR, TE, RE, CKE1, and CKE0 bits in SCSCR, and MCE bit in SCFCR.

- Serial status register_0 (SCFSR_0)
- Receive FIFO data register_0 (SCFRDR_0)
- FIFO control register_0 (SCFCR_0)
- FIFO data count register_0 (SCFDR_0)
- Line status register_0 (SCLSR_0)
- Receive shift register_0 (SCRSR_0)
- Transmit shift register_0 (SCTSR_0)

Channel 1:

- Serial mode register_1 (SCSMR_1)
- Bit rate register_1 (SCBRR_1)
- Serial control register_1 (SCSCR_1)
- Transmit FIFO data register_1 (SCFTDR_1)
- Serial status register_1 (SCFSR_1)
- Receive FIFO data register_1 (SCFRDR_1)
- FIFO control register_1 (SCFCR_1)
- FIFO data count register_1 (SCFDR_1)
- Line status register_1 (SCLSR_1)
- Receive shift register_1 (SCRSR_1)
- Transmit shift register_1 (SCTSR_1)

SCFRDR is a 16-stage FIFO register that stores received serial data.

When the SCIF has received one byte of serial data, it transfers the received data from SC
SCFRDR where it is stored, and completes the receive operation. SCRSR is then enabled
reception, and consecutive receive operations can be performed until the receive FIFO da
register is full (16 data bytes).

SCFRDR is a read-only register, and cannot be written to by the CPU.

If a read is performed when there is no receive data in the receive FIFO data register, an u
value will be returned. When the receive FIFO data register is full of receive data, subseq
serial data is lost.

The contents of SCFRDR are undefined after a power-on reset or manual reset.

16.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is an 8-bit 16-stage FIFO data register that stores data for serial transmission.

If SCTSR is empty when transmit data has been written to SCFTDR, the SCIF transfers transmit data written in SCFTDR to SCTSR and starts serial transmission.

SCFTDR is a write-only register, and cannot be read by the CPU.

The next data cannot be written when SCFTDR is filled with 16 bytes of transmit data. I written in this case is ignored.

The contents of SCFTDR are undefined after a power-on reset or manual reset.

16.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit register used to set the SCIF's serial communication format and sele clock source of the baud rate generator.

SCSMR can be read or written to by the CPU at all times.

SCSMR is initialized to H'0000 by a power-on reset or manual reset. It is not initialized mode or in the module standby state, and retains its contents.

6	CHR	0	R/W	Character Length Selects 7 or 8 bits as the asynchronous mod length. In clock synchronous mode, a fixed d length of 8 bits is used regardless of the CHF 0: 8-bit data 1: 7-bit data* Note: * When 7-bit data is selected, the MS of the transmit FIFO data register (S is not transmitted.
5	PE	0	R/W	Parity Enable In asynchronous mode, selects whether or n bit addition is performed in transmission, and bit checking in reception. In clock synchronou parity bit addition and checking is not perform regardless of the PE bit setting. 0: Parity bit addition and checking disabled 1: Parity bit addition and checking enabled* Note: * When the PE bit is set to 1, the pari or odd) specified by the O/$\overline{E}$ bit is ad transmit data before transmission. In reception, the parity bit is checked fo parity (even or odd) specified by the

1: Odd parity*

Notes: 1. When even parity is set, parity b
is performed in transmission so
total number of 1-bits in the tran
character plus the parity bit is ev
reception, a check is performed
the total number of 1-bits in the
character plus the parity bit is ev

2. When odd parity is set, parity bit
is performed in transmission so
total number of 1-bits in the tran
character plus the parity bit is od
reception, a check is performed
the total number of 1-bits in the
character plus the parity bit is od

				before it is sent. 2. In transmission, two 1-bits (stop b added to the end of a transmit ch before it is sent. In reception, only the first stop bit is checked regardless of the STOP bit setting. If the sec bit is 1, it is treated as a stop bit; if it is 0, it is as the start bit of the next transmit character.
2	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	Select the clock source for the on-chip baud generator. 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64

Bit	Bit Name	Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables generation of a transmi data-empty interrupt (TXI) request when the flag in SCFSR is set to 1 after the serial tran is transferred from SCFTDR to SCTSR and number of data bytes in the transmit FIFO r equal to or below the trigger set number. 0: Transmit-FIFO-data-empty interrupt (TX disabled* 1: Transmit-FIFO-data-empty interrupt (TX enabled Note: * TXI interrupt requests can be clea writing transmit data exceeding the trigger set number to SCFTDR, re from the TDFE flag, then clearing by clearing the TIE bit to 0.

				(BRI) request disabled* 1: Receive-data-full interrupt (RXI) request, error interrupt (ERI) request, and break-in (BRI) request enabled Note: * An RXI request can be cleared by r from the RDF flag or DR flag, then c the flag to 0, or by clearing the RIE The ERI and BRI requests can be c by reading 1 from the ER, BRK, or flag, then clearing the flag to 0, or c the RIE and REIE bits to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the start of serial transm the SCIF. 0: Transmission disabled 1: Transmission enabled* Note: * SCSMR and SCFCR settings must made, the transmit format decided, transmit FIFO reset, before the TE to 1.

				made, the receive format decide receive FIFO reset, before the F set to 1.
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables generation of receive-e interrupt (ERI) request and break interrupt (request. The REIE bit setting is available wh RIE bit is cleared to 0. 0: Receive-error interrupt (ERI) request an interrupt (BRI) request disabled* 1: Receive-error interrupt (ERI) request an interrupt (BRI) request enabled Note: * A receive-error interrupt (ERI) requ break interrupt (BRI) request can b by reading 1 from the ER, BRK, an flags, then clearing the flags to 0, clearing the RIE and REIE bits to Even if the RIE bit is cleared to 0, the REIE bit to 1 enables generati ERI and BRI requests. This setting achieved to notify the ERI and BR to the interrupt controller at the DM transfer.
2	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

mode is selected by SCSMR.

- Asynchronous mode

00: Internal clock/SCIFnCK pin functions as i
(input signal ignored)

01: Internal clock/SCIFnCK pin functions as
output*[2]

1-*[1]: External clock/SCIFnCK pin functions a
input*[3]

- Clock synchronous mode

00: Internal clock/SCIFnCK pin functions as
synchronous clock output

01: Internal clock/SCIFnCK pin functions as
synchronous clock output

1-*[1]: External clock/SCIFnCK pin functions a
synchronous clock input

Notes: 1. When CKE1 = 1, the value of CKE
don't care.
2. The output clock frequency is 16 t
bit rate.
3. The input clock frequency is 16 ti
bit rate.

SCFSR is initialized to H'0060 by a power-on reset or manual reset. It is not initialized i
mode or in the module standby state, and retains its contents.

Bit	Bit Name	Initial Value	R/W	Description
15	PER3	0	R	Parity Error Number 3 to 0
14	PER2	0	R	Indicate the number of data bytes, in whic
13	PER1	0	R	errors are generated, in receive data store
12	PER0	0	R	SCFRDR.
				After setting the ER bit in SCFSR, the valu 15 to 12 indicate the number of parity erro generated data. When all 16 bytes of rece in SCFRDR has parity errors, the PER3 to bits indicate 0.
11	FER3	0	R	Framing Error Number 3 to 0
10	FER2	0	R	Indicate the number of data bytes, in whic
9	FER1	0	R	errors are generated, in receive data store
8	FER0	0	R	SCFRDR.
				After setting the ER bit in SCFSR, the valu 11 to 8 indicate the number of framing err generated data.
				When all 16 bytes of receive data in SCFR framing errors, the FER3 to FER0 bits ind

1: A framing error or parity error occurred d
reception

[Setting conditions]

- When the SCIF checks whether the sto
the end of the receive data is 1 when re
ends, and the stop bit is 0*[2]
- When, in reception, the number of 1-bit
receive data plus the parity bit does not
the parity setting (even or odd) specifie
O/$\overline{E}$ bit in SCSMR

Notes: 1. The ER flag is not affected and
its previous state when the RE
SCSCR is cleared to 0. When a
error occurs, the receive data is
transferred to SCFRDR, and re
continues.
The FER and PER bits in SCFS
be used to determine whether t
receive error in the data read fr
SCFRDR.

2. When the stop length is 2 bits, o
first stop bit is checked for a val
the second stop bit is not check

- When data is written to SCFTDR by th

1: Transmission has been ended

[Setting conditions]

- Power-on reset or manual reset
- The TE bit in SCSCR is cleared to 0
- When there is no transmit data in SCF transmission of the last bit of 1-byte se transmit character

SCFTDR

[Clearing conditions]

- When transmit data exceeding the trans
  trigger set number is written to SCFTDR
  is written to the TDFE bit after reading T
- When transmit data exceeding the trans
  trigger set number is written to SCFTDR
  DMAC

1: The number of transmit data bytes in SC
   does not exceed the transmit trigger set

[Setting conditions]

- Power-on reset or manual reset
- When the number of SCFTDR transmit
  bytes is equal to or below the transmit t
  set number as the result of a transmit o

Note: * As SCFTDR is a 16-byte FIFO regis
  maximum number of bytes that can
  written when TDFE = 1 is 16 – (tran
  trigger set number). Data written in
  of this will be ignored. The number
  bytes in SCFTDR is indicated by the
  bits in SCFDR.

				[Setting condition] When data with a framing error is received by the space “0” level (low level) for at lea frame length Note: * When a break is detected, the rec (H'00) following detection is not tra to SCFRDR. When the break ends receive signal returns to mark “1”, data transfer is resumed.
3	FER	0	R	Framing Error In asynchronous mode, indicates that ther framing error or not in the data read from 0: There is no framing error in the receive from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no framing error in SCF data 1: There is a framing error in the receive d from SCFRDR [Setting condition] When there is a framing error in SCFRDR

data

1: There is a parity error in the receive data from SCFRDR

[Setting condition]

When there is a parity error in SCFRDR rea

- Power-on reset or manual reset
- When SCFRDR is read until the numb
  receive data bytes in SCFRDR is less
  receive trigger set number, and 0 is w
  RDF after reading RDF = 1
- When SCFRDR is read by the DMAC
  number of receive data bytes in SCFR
  than the receive trigger set number

1: The number of receive data bytes in SC
equal to or greater than the receive trig
number

[Setting condition]

When SCFRDR contains at least the rece
set number of receive data bytes*

Note: * SCFRDR is a 16-byte FIFO registe
RDF = 1, at least the receive trigg
number of data bytes can be read.
read when SCFRDR is empty, an
value will be returned. The numbe
receive data bytes in SCFRDR is i
by the lower bits in SCFDR.

[Clearing conditions]

- Power-on reset or manual reset
- When all the receive data in SCFRDR h
  read, and 0 is written to DR after readin
  1
- When all the receive data in SCFRDR i
  the DMAC

1: No further receive data has arrived

[Setting condition]

When SCFRDR contains fewer than the re
trigger set number of receive data bytes, ar
further data has arrived for at least 15 etu a
stop bit of the last data received*

Note: * Corresponds to 1.5 frame time whe
format of 8-bit length and 1 stop bit

etu: Elementary time unit (time for t
of 1 bit)

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

Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clock synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
Pϕ: Peripheral module operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See table 16.2 for the relation between n and the clock.)

Table 16.2 Relationship between n and Clock

		SCSMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

The bit rate error in asynchronous mode is found from the following equation:

SCFCR is initialized to H'0000 by a power-on reset or manual reset. It is not initialized in mode or in the module standby state, and retains its contents.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
10	RSTRG2	0	R/W	$\overline{\text{RTS}}$ Output Active Trigger 2 to 0
9	RSTRG1	0	R/W	The $\overline{\text{RTS}}$ signal goes high when the number
8	RSTRG0	0	R/W	receive data bytes in SCFRDR is equal to or than the trigger set number shown in below. $\overline{\text{RTS}}$ active trigger: 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14

				10: 8 11: 14
5	TTRG1	0	R/W	Transmit FIFO Data Number Trigger 1 and
4	TTRG0	0	R/W	Set the number of remaining transmit data b sets the TDFE flag in SCFSR. The TDFE flag is set when, as the result of operation, the number of transmit data byte SCFTDR is equal to or below the trigger set shown in below. 00: 8 (8) 01: 4 (12) 10: 2 (14) 11: 0 (16) Note: The values in parentheses are the n empty bytes in SCFTDR when the fl
3	MCE	0	R/W	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{R}}$ bit is valid only in asynchronous mode. 0: Modem signal disabled* 1: Modem signal enabled Note: * $\overline{\text{CTS}}$ is fixed at active 0 regardless o value, and $\overline{\text{RTS}}$ is also fixed at 0.

				invalidates the receive data in the receive FI register and resets it to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * A reset operation is performed in the a power-on reset or manual reset.
0	LOOP	0	R/W	Loopback Test Internally connects the transmit output pin (T receive input pin (RxD), and $\overline{\text{RTS}}$ pin and $\overline{\text{CT}}$ enabling loopback testing. 0: Loopback test disabled 1: Loopback test enabled

16.3.10 FIFO Data Count Register (SCFDR)

SCFDR is a 16-bit register that indicates the number of data bytes stored in SCFTDR and SCFRDR.

Bits 12 to 8 show the number of transmit data bytes in SCFTDR, and bits 4 to 0 show the of receive data bytes in SCFRDR.

SCFDR can be read by the CPU at all times.

SCFDR is initialized to H'0000 by a power-on reset or manual reset. It is not initialized in mode or in the module standby state, and retains its contents.

7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
4	R4	0	R	Bits 4 to 0 in SCFDR show the number of re data bytes in SCFRDR. A value of H'00 means that there is no rece and a value of H'10 means that SCFRDR is receive data.
3	R3	0	R	
2	R2	0	R	
1	R1	0	R	
0	R0	0	R	

Indicates that an overrun error occurred du
reception and reception is ended abnormal

0: Reception is in progress, or reception h
ended successfully*[1]

[Clearing conditions]

- Power-on reset or manual reset
- When 0 is written to ORER after readin
= 1

1: An overrun error occurred during recep

[Setting condition]

When serial reception is completed while th
receive FIFO is full

Notes: 1. The ORER flag is not affected a
retains its previous state when
bit in SCSCR is cleared to 0.

2. The receive data prior to the ov
error is retained in SCFRDR, a
data received subsequently is l
Serial reception cannot be cont
while the ORER flag is set to 1.

Note: * Only 0 can be written to clear the flag.

signals are included as modem control signals. Transfer format is selected by SCSMR. T
shown in table 16.3. The SCIF clock source is determined by the combination of the C/$\overline{A}$
SCSMR and the CKE1 and CKE0 bits in SCSCR. This is shown in table 16.4.

Asynchronous Mode

- Data length: Choice of 7 or 8 bits
- Choice of parity addition and addition of 1 or 2 stop bits (the combination of these p
  determines the transfer format and character length)
- Detection of framing errors, parity errors, overrun errors, receive-FIFO-data-full stat
  data-ready state, and breaks, during reception
- Indication of the number of data bytes stored in the transmit and receive FIFO regist
- The SCIF clock source: Choice of internal or external clock

  When internal clock is selected: The SCIF operates on the baud rate generator clock.

  When external clock is selected: Clock with frequency16 times the bit rate must be i
  on-chip baud rate generator is not used.)

Clock synchronous Mode

- Transfer format: Fixed to 8-bit data
- Detection of overrun errors during reception
- The SCIF clock source: Choice of internal or external clock

  When internal clock is selected: The SCIF operates on the baud rate generator clock
  outputs the synchronous clock

  When external clock is selected: The SCIF operates on the input synchronous clock.
  chip baud rate generator is not used.

			1				2 bi
		1	0			Yes	1 bi
			1				2 bi
1	*	*	*	Clock synchronous mode	8-bit data	No	No

Table 16.4 SCSMR and SCSCR Settings for the SCIF Clock Source Selection

SCSMR	SCSCR				
Bit 7: C/$\overline{A}$	**Bit 1: CKE1**	**Bit 0: CKE0**	**Mode**	**Clock Source**	**SCIFnCK Pin Function**
0	0	0	Asynchronous mode	Internal	SCIF does not use the SCIFn
		1			Outputs a clock with frequenc times the bit rate
	1	0		External	Inputs a clock with frequency the bit rate
		1			
1	0	0	Clock synchronous mode	Internal	Outputs the synchronous cloc
		1			
	1	0		External	Inputs the synchronous clock
		1			

samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rat
Therefore, communication data is latched at the center of each bit.

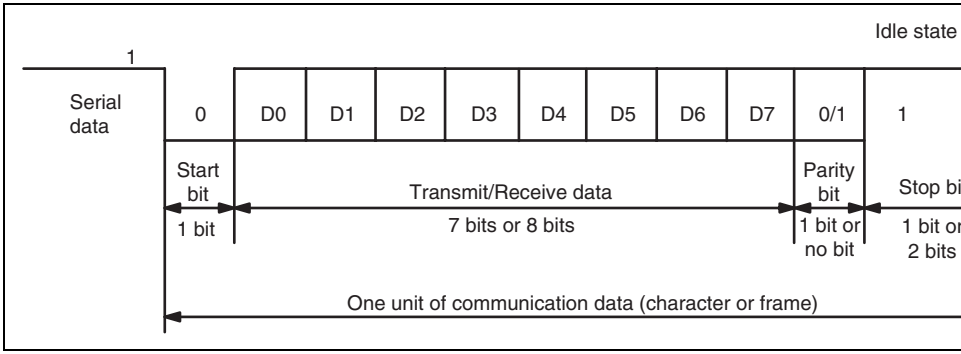


Figure 16.2 Data Format in Asynchronous Communication (Example of 8-Bit Data with Parity and 2 Stop Bits)

	1	0	S	8-bit data		P	S
		1	S	8-bit data		P	S
1	0	0	S	7-bit data	STOP		
		1	S	7-bit data	STOP	STOP	
	1	0	S	7-bit data	P	STOP	
		1	S	7-bit data	P	STOP	S

S: Start bit
STOP: Stop bit
P: Parity bit

Clock: The SCIF transfer clock is set by the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 SCSCR. For details, see table 16.4.

When an external clock is input to the SCIFnCK pin, the clock with frequency 16 times t must be input.

When the SCIF operates on an internal clock, it can output a clock from the SCIFnCK pi time output clock frequency is 16 times the bit rate.

again to start transmission, the TFRST bit in SCFCR should first be set to 1 to reset SCF

When an external clock is used, the clock should not be stopped during operation includ initialization because its operation becomes unreliable.

Figure 16.3 shows a sample the SCIF initialization flowchart.

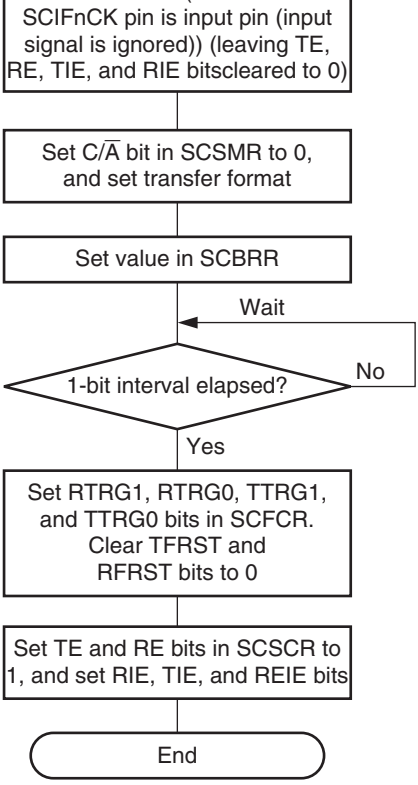
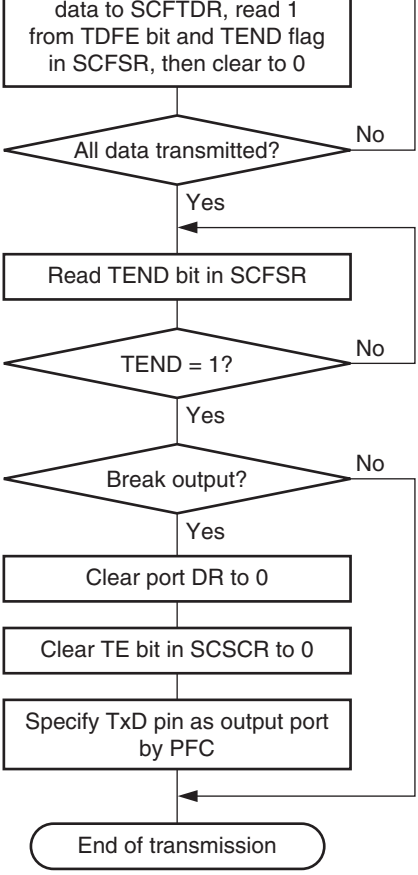


Figure 16.3 Sample the SCIF Initialization Flowchart

Serial Data Transmission: Figure 16.4 shows a sample flowchart for serial transmission

Use the following procedure for serial data transmission after enabling the SCIF for trans



then clear the TDFE bit to 0.

3. Break output at the end of serial transmission:
   To output a break in serial transmissio
   clear the port data register (DR) to 0 b
   clearing the TE bit in SCSCR to 0, and
   specify the TxD pin as an output port b
   PFC.
   In steps 1 and 2, it is possible to ascer
   the number of data bytes that can be
   written from the number of transmit da
   bytes in SCFTDR indicated by the upp
   bits of SCFDR.

Figure 16.4 Sample Serial Transmission Flowchart

The serial transmit data is sent from the TxD pin in the following order.

A. Start bit: One 0-bit is output.
B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity
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 transmiss
sent.

3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If d
present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then
transmission of the next frame is started.

   If there is no transmit data, the TEND flag in SCFSR is set to 1, the stop bit is sent, ar
the line goes to the mark state in which 1 is output.

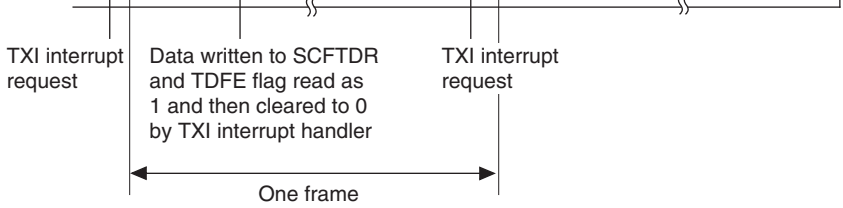
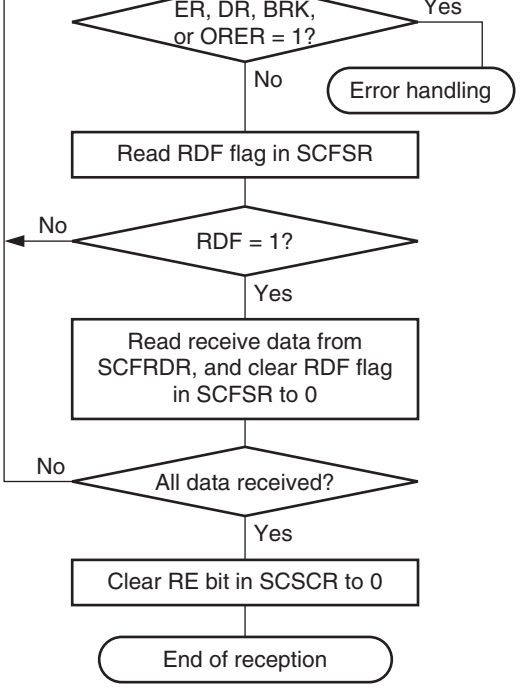


Figure 16.5 Example of Transmit Operation (Example of 8-Bit Data with Parity and 1 Stop Bit)



reading the value of the RxD pin.

2. SCIF status check and receive data
   Read SCFSR and check that RDF =
   read the receive data in SCFRDR, re
   from the RDF flag, and then clear th
   flag to 0. The transition of the RDF fl
   from 0 to 1 can also be identified by
   interrupt.
3. Serial reception continuation proced
   To continue serial reception, read at
   the receive trigger set number of dat
   from SCFRDR, read 1 from the RDF
   and then clear the RDF flag to 0. Th
   number of receive data bytes in SCF
   can be ascertained by reading the lo
   bits of SCFDR.

Figure 16.6 Sample Serial Reception Flowchart (1)

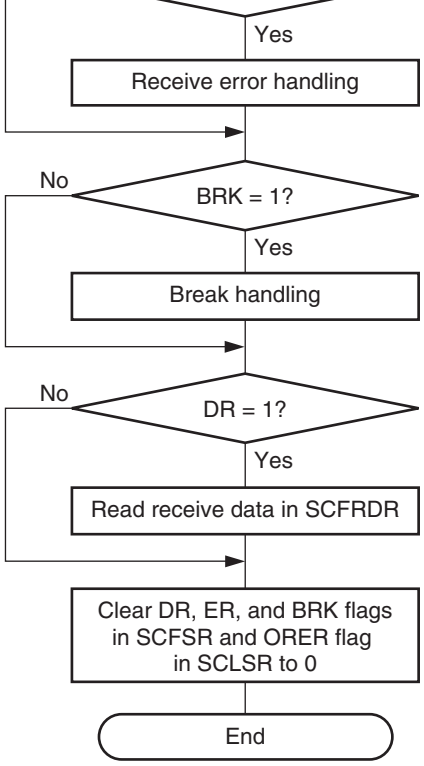


Figure 16.7 Sample Serial Reception Flowchart (2)

C. Overrun error check: The SCIF checks that the ORER flag is cleared to 0 and an o error does not occur.

D. Break check: The SCIF checks that the BRK flag is 0, indicating that the break sta set.

If all the above checks are passed, the receive data is stored in SCFRDR.

Note: Reception continues when a receive error (a framing error or parity error) occurs.

4. If the RIE bit in SCSCR is set to 1 when the RDF flag or DR flag changes to 1, a rece FIFO-data-full interrupt (RXI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a recei interrupt (ERI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the BRK flag or ORER flag char a break reception interrupt (BRI) request is generated.

Figure 16.8 shows an example of the operation for reception in asynchronous mode.

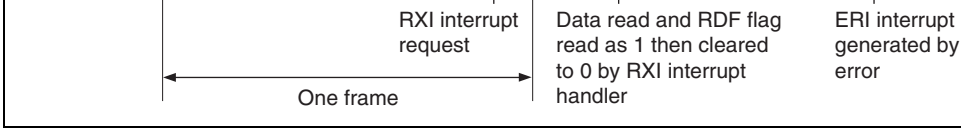


Figure 16.8 Example of SCIF Receive Operation (Example of 8-Bit Data with Parity and 1 Stop Bit)

Modem Function: When using a modem function, transmission can be stopped and star according to the $\overline{\text{CTS}}$ input value. When the $\overline{\text{CTS}}$ is set to 1 during transmission, the data mark state after transmitting one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is c starting with a start bit.

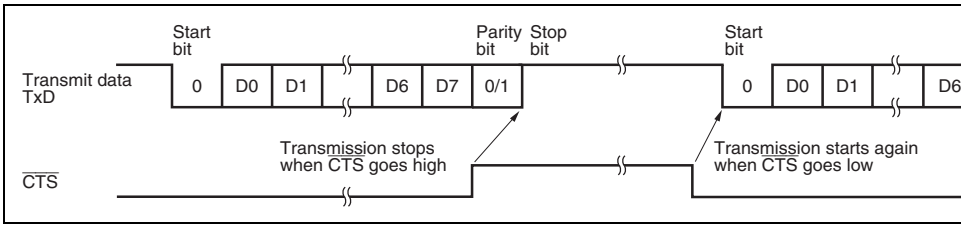


Figure 16.9 $\overline{\text{CTS}}$ Control Operation

at least number of $\overline{\text{RTS}}$ output trigger less than number of $\overline{\text{RTS}}$ output trigger

Figure 16.10 $\overline{\text{RTS}}$ Control Operation

16.4.3 Serial Operation in Clock Synchronous Mode

In clock synchronous mode, the SCIF transmits and receives data synchronizing with clo
This mode is suitable for high-speed serial communication. In the SCIF, the transmitter a
receiver are independent. Therefore, by sharing the same clock, full-duplex communicati
performed. Figure 16.11 shows the general format of clock synchronous serial communic

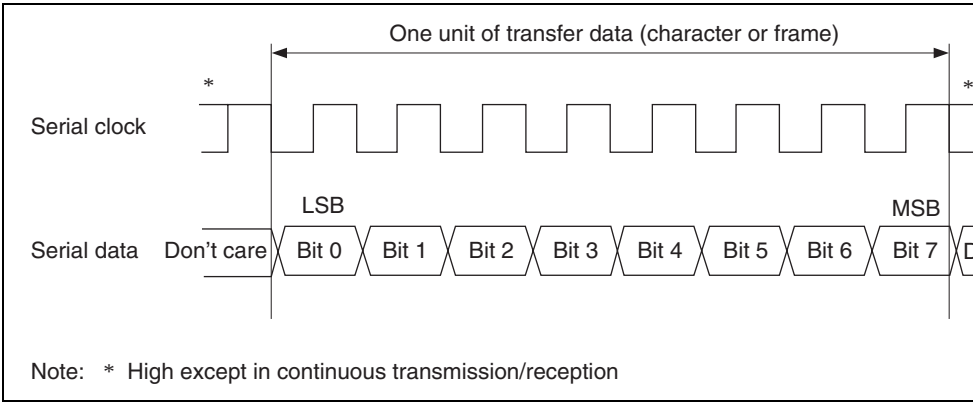


Figure 16.11 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, data on the communication line is output fro
fall of the serial clock to the next. Data is guaranteed valid at the rise of the serial clock.

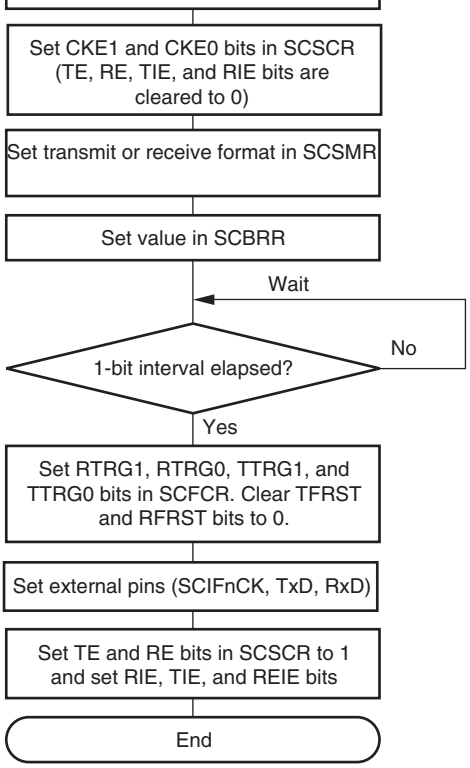
In serial communication, each character is output starting with the LSB and ending with t
After the MSB is output, the communication line remains in the state of the MSB.

only reception is performed, clock pulse is output continuously until the number of data
the receive FIFO reaches the receive trigger set number while the RE bit in SCSCR is 1.

Data Transfer Operations:

The SCIF Initialization: Before transmitting and receiving data, it is necessary to clear
and RE bits in SCSCR to 0, then initialize the SCIF as described below.

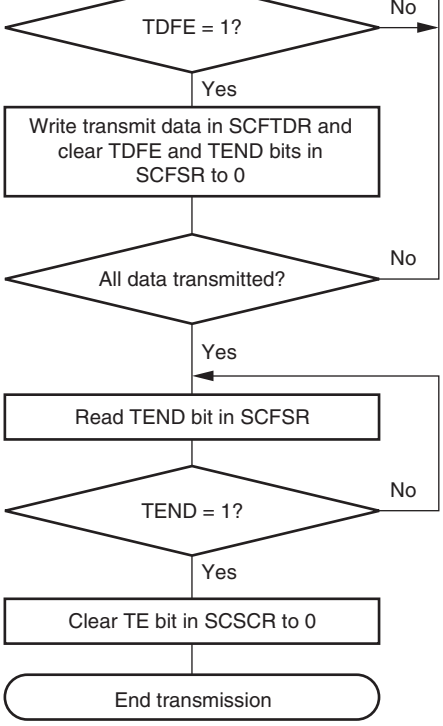
When the mode, communication format etc., is changed, the TE and RE bits must be cle
before making the change using the following procedure. When the TE bit is cleared to
is initialized. Note that the RDF, PER, FER, and ORER flags and contents of SCFRDR
retained even if the RE bit is cleared to 0.



5. Set the external pins. Specifies the pins as Rx in reception and TxD output in transmission. S SCIFnCK input/output according to the CKE1 CKE0 settings.

6. Set the TE bit or RE bit in SCSCR to 1. Also, TIE, RIE, and REIE bits. At this time, the TxD and SCIFnCK pins can be used. In transmiss TxD pin is in the mark state. When reception clock synchronous mode and synchronous cl (clock master) are selected, a clock is output SCIFnCK pin.

Figure 16.12 Sample the SCIF Initialization Flowchart



the TDFE flag to confirm that writing is po
then write data in SCFTDR and clear the
flag to 0.

Figure 16.13 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SC
starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing t
data to SCFTDR. The number of data bytes that can be written is at least 16 – (transm
set number).

the state of the TxD pin is held.

4. After serial transmission has completed, the SCIFnCK pin is fixed to high.

Figure 16.14 shows an example of the SCIF transmit operation.

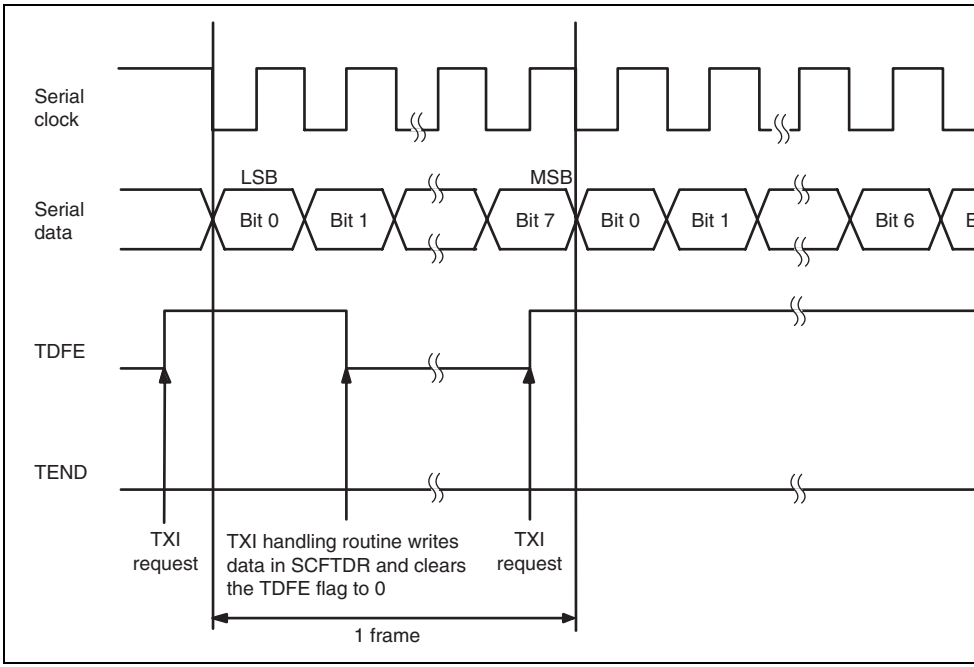
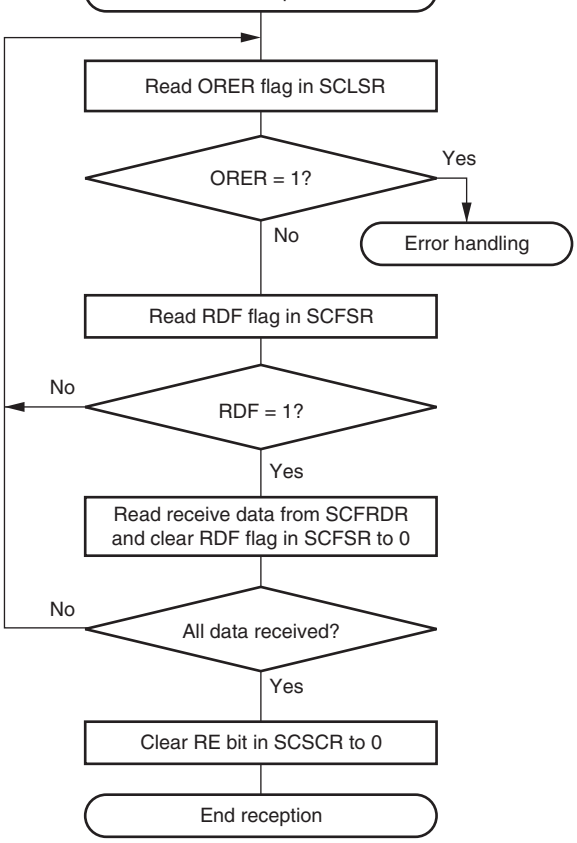


Figure 16.14 Example of the SCIF Transmit Operation



2. Receive error handling: If a receive e
occurs, read the ORER flag in SCLS
after executing the necessary error h
clear the ORER flag to 0.
Serial reception cannot be continued
ORER flag is set to 1.

3. SCIF status check and receive data
Read SCFSR, check that the RDF fl
to 1, then read receive data in SCFR
clear the RDF flag to 0. Notification t
RDF flag has changed from 0 to 1 ca
be given by the RXI.

4. Serial reception continuation proced
To continue serial reception, read at
receive trigger set number of data by
SCFRDR, read 1 from the RDF flag,
clear the RDF flag to 0. The number
data in SCFRDR can be ascertained
the lower bits of SCFDR. However, i
is activated by the RXI and the value
SCFRDR, the RDF flag is cleared au

Figure 16.15 Sample Serial Reception Flowchart

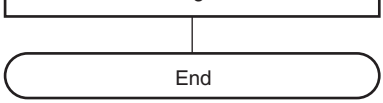


Figure 16.16 Sample Serial Reception Flowchart

In serial reception, the SCIF operates as described below.

1. The SCIF internally initializes in synchronization with the synchronous clock input or
2. The SCIF stores receive data in SCRSR in order from LSB to MSB. After reception, t checks whether receive data can be transmitted from SCRSR to SCFRDR. If this chec passed, the SCIF stores the receive data in SCFRDR. If an overrun error is detected by check, following reception can not be performed.
3. If the RDF flag is set to 1 and the RIE bit in SCSCR is set to 1, the SCIF requests a re FIFO-data-full interrupt (RXI). If the ORER flag is set to 1 and the RIE bit or REIE b SCSCR is set to 1, the SCIF requests a break interrupt (BRI).

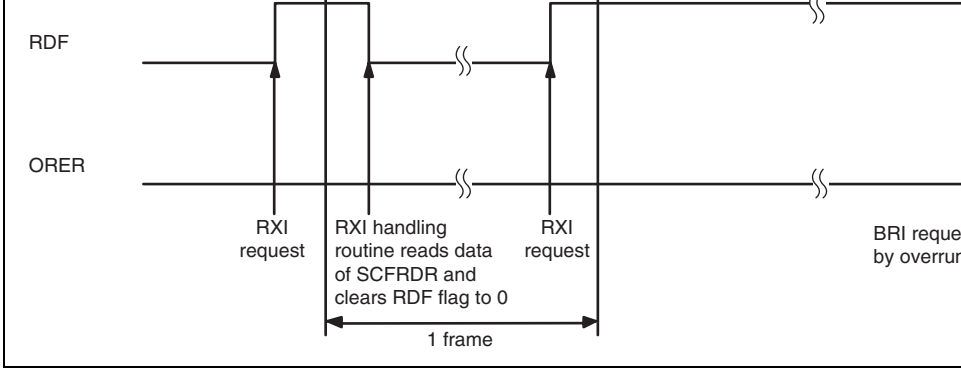
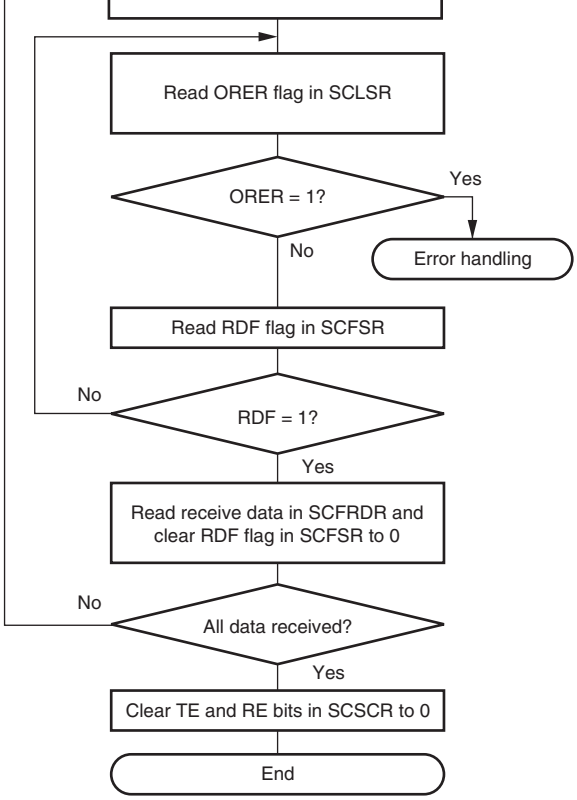


Figure 16.17 Example of the SCIF Receive Operation



4. SCIF status check and receive data r
   Read SCFSR, check that the RDF fla
   to 1, then read receive data from SCF
   and clear the RDF flag to 0. Notificati
   the RDF flag has changed from 0 to 1
   be given by the RXI.

5. Serial transmission/reception continua
   procedure: To continue serial recepti
   the MSB of the current frame is receiv
   the RDF flag and SCFRDR and clear
   flag to 0. Also, check that the TDFE fl
   to 1 and data can be written before tra
   the MSB of the current frame. Futher
   data in SCFTDR and clear the TDFE

Note: When switching from transmission
to simultaneous transmission and
clear the TE and RE bits to 0, the
bits to 1 simultaneously.

Figure 16.18 Sample Serial Data Transmission/Reception Flowchart

TIE bit, if the TDFE flag is set to 1, only the transmit-FIFO-data-empty DMA transfer re
generated. The DMAC can be activated and data transfer performed on generation of the
FIFO-data-empty DMA transfer request.

When the RXI is enabled by the RIE bit, if the RDF flag or DR flag in SCFSR is set to 1,
request and a receive-FIFO-data-full DMA transfer request are generated. When the RXI
disabled by the RIE bit, if the RDF flag or DR flag is set to 1, only the receive-FIFO-data
DMA transfer request is generated. The DMAC can be activated and data transfer perfor
generation of the receive-FIFO-data-full DMA transfer request. The generation of the RX
receive-FIFO-data-full DMA transfer requests by setting the DR flag to 1 occurs only in
asynchronous mode.

When the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1, a BRI request is g
When using the DMAC for transmission/reception, set and enable the DMAC before mak
SCIF settings. See section 13, Direct Memory Access Controller (DMAC), for details on
DMAC setting procedure. Also set the RXI and TXI requests not to be output to the inter
controller. If the interrupt requests are set to be generated, the interrupt requests to the int
controller are cleared by the DMAC regardless of the interrupt handling program.

When the RIE bit is cleared to 0 and the REIE bit is set to 1 in SCSCR, the ERI or BRI re
can be generated without generating the RXI request. Note that the TXI indicates that wri
transmit data is enabled, while the RXI indicates that the receive data is in SCFRDR.

Note: * The RXI by the DR is enabled only in asynchronous mode.
See section 4, Exception Handling, for priorities and the relationship with non
interrupts.

16.6 Usage Notes

Note the following when using the SCIF.

SCFTDR Writing and TDFE Flag: The TDFE flag in SCFSR is set when the number
transmit data bytes written in SCFTDR has fallen to or below the transmit trigger numb
bits TTRG1 and TTRG0 in SCFCR. After the TDFE flag is set, transmit data up to the n
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 tra
trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to
clearing should therefore be carried out when SCFTDR contains more than the transmit
number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from bits 12 to 8 in SCFDR

SCFRDR Reading and RDF Flag: The RDF flag in SCFSR is set when the number of
data bytes in SCFRDR has become equal to or greater than the receive trigger number s
RTRG1 and RTRG0 in SCFCR. After the RDF flag is set, receive data equivalent to the
number can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes in SCFRDR is still equal to or greater than the trig
number after a read, the RDF flag will be set to 1 again if it is cleared to 0. The RDF fla
therefore be cleared to 0 after being read as 1 after all receive data has been read.

The number of receive data bytes in SCFRDR can be found from bits 4 to 0 in SCFDR.

shown in figure 16.19.

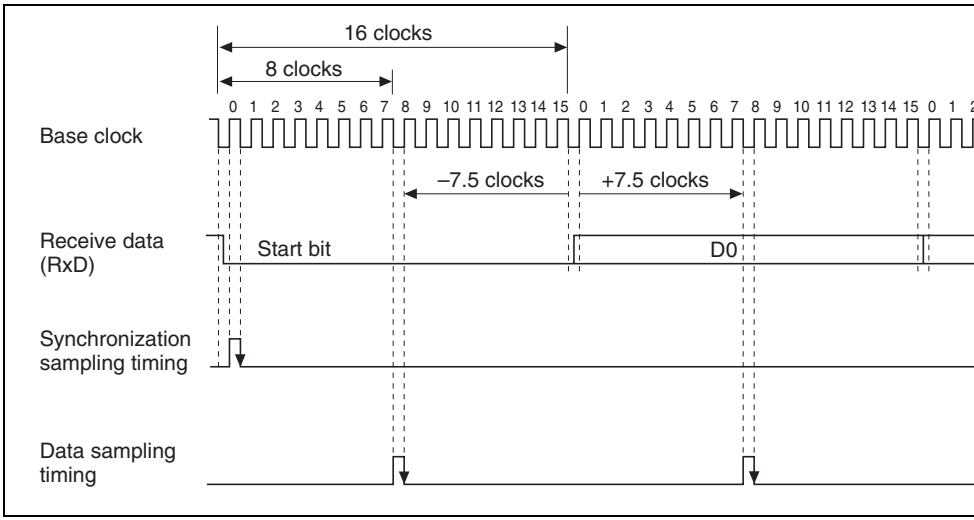


Figure 16.19 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equati

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\% \quad \text{..... (1)}$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

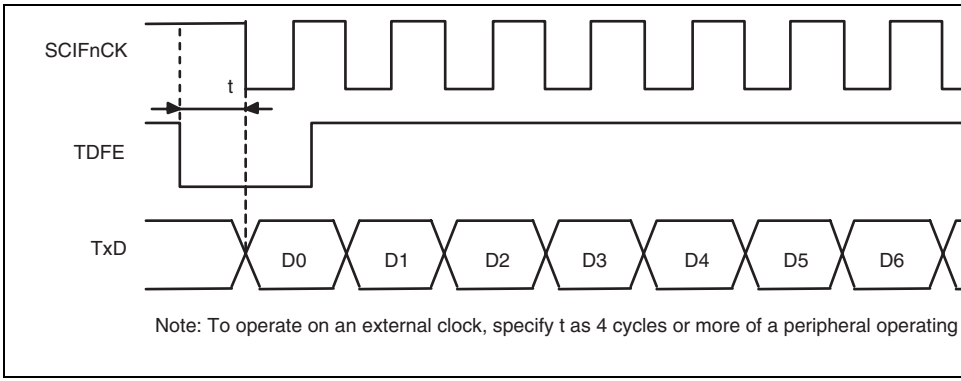


Figure 16.20 Sample Transfer of Synchronous Clock by DMAC

- Serial transfer
  Sixteen-stage 32-bit FIFOs (transmission/reception independently)
  Supports 8-bit data/16-bit data/16-bit stereo audio input/output
  MSB or LSB first for data transmission/reception
  Supports a maximum of 48-kHz sampling rate
  Synchronization by either frame synchronization pulse or left/right channel switch
  Supports CODEC control data interface
  Connectable to every A-Law or µ-Law CODEC linear audio chip manufactured by a company
  Supports both master and slave modes
- Serial clock
  An external pin input or internal clock (P_CLK) can be selected as the clock source.
- Interrupts
  Following four interrupts can be requested independently.
  Transmission interrupt
  Reception interrupt
  Error interrupt
  Control interrupt
- DMA transfer
  Supports DMA transfer by a transfer request for transmission/reception

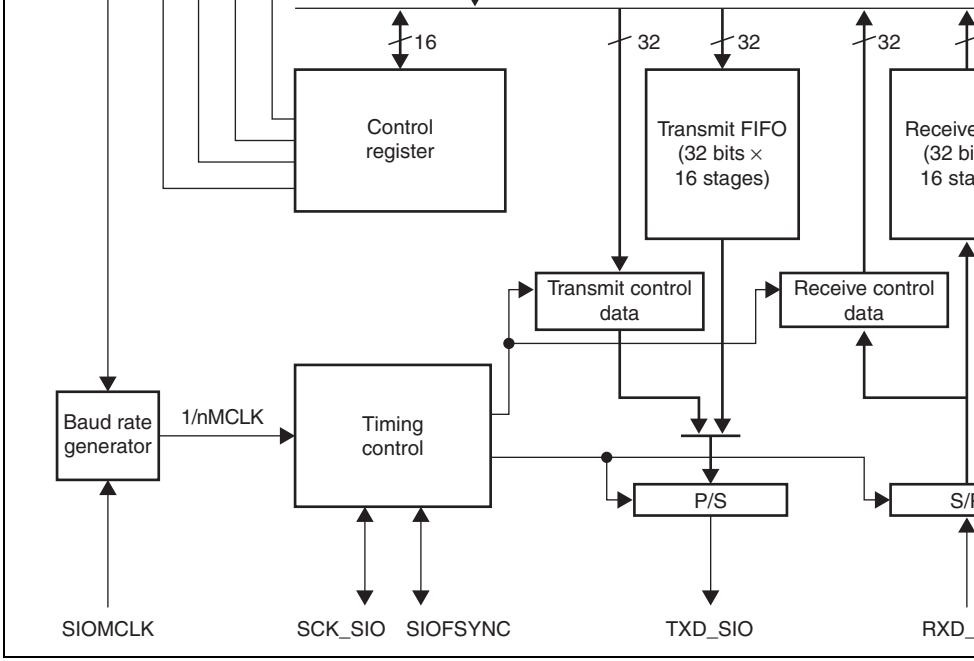


Figure 17.1 Block Diagram of SIOF

	synchronous pin		output	(common to transmission/rec
	Transmit data pin	TXD_SIO0	Output	Transmit data
	Receive data pin	RXD_ SIO0	Input	Receive data
1	Clock input pin	SIOMCLK1	Input	Master clock input
	Communication clock pin	SCK_SIO1	Input/ output	Serial clock (common to transmission/reception)
	Frame synchronous pin	SIOFSYNC1	Input/ output	Frame synchronous signal (common to transmission/rec
	Transmit data pin	TXD_SIO1	Output	Transmit data
	Receive data pin	RXD_SIO1	Input	Receive data

- Serial control data assign register_0 (SICDAR_0)
- SIOF control register_0 (SICTR_0)
- SIOF FIFO control register_0 (SIFCTR_0)
- SIOF status register_0 (SISTR_0)
- SIOF interrupt enable register_0 (SIIER_0)
- Serial transmit data register_0 (SITDR_0)
- Serial receive data register_0 (SIRDR_0)
- Serial transmit control data register_0 (SITCR_0)
- Serial receive control data register_0 (SIRCR_0)

2. Channel 1

- SIOF mode register_1 (SIMDR_1)
- Serial clock select register_1 (SISCR_1)
- Serial transmit data assign register_1 (SITDAR_1)
- Serial receive data assign register_1 (SIRDAR_1)
- Serial control data assign register_1 (SICDAR_1)
- SIOF control register_1 (SICTR_1)
- SIOF FIFO control register_1 (SIFCTR_1)
- SIOF status register_1 (SISTR_1)
- SIOF interrupt enable register_1 (SIIER_1)
- Serial transmit data register_1 (SITDR_1)
- Serial receive data register_1 (SIRDR_1)
- Serial transmit control data register_1 (SITCR_1)
- Serial receive control data register_1 (SIRCR_1)

				01: Slave mode 2 10: Master mode 1 11: Master mode 2 Note: For the operation in each mode, see section Transfer Data Format.
13	—	0	R	Reserved This bit is always read as 0. The write value should be 0.
12	REDG	0	R/W	Receive Data Sampling Edge The TXD_SIO signal is transmitted at the opposite where the RXD_SIO signal is sampled (see figure 0: RXD_SIO is sampled at the falling edge of SCK 1: RXD_SIO is sampled at the rising edge of SCK Note: This bit is valid in master mode 1 and mast 2.

				1101: Slot length is 16 bits and frame length is 64 1110: Slot length is 16 bits and frame length is 12 1111: Slot length is 16 bits and frame length is 25 Notes: 1. When slot length is specified as 8 bits, data cannot be transmitted or received 2. When LSB is first transmitted or receiv control data cannot be transmitted or r x: Don't care
7	TXDIZ	0	R/W	High-Impedance Output when Transmission is In Specifies high-impedance output when transmiss invalid. 0: High output (1 output) when invalid 1: High-impedance output when invalid Note: Invalid means when disabled, and when a is not assigned as transmit data or control being transmitted.
6	LSBF	0	R/W	LSB-First Transmission/Reception Selects the bit order of a transmit/receive frame. 0: MSB-first 1: LSB-first
5	RCIM	0	R/W	Receive Control Data Interrupt Mode Selects the set timing of the RCRDY bit in SISTR 0: Sets the RCRDY bit in SISTR when the conten SIRCR change. 1: Sets the RCRDY bit in SISTR each time when receives control data.

reset or software reset.

Bit	Bit Name	Initial Value	R/W	Description
15	MSSEL	1	R/W	Master Clock Source Selection The master clock is the clock input to the baud rat generator. 0: Uses the SIOMCLK pin input signal as the mas 1: Uses PCLK as the master clock
14	MSIMM	1	R/W	Master Clock Direct Selection 0: Uses the baud rate generator output clock as th source 1: Uses the master clock itself as the clock source
13	—	0	R	Reserved This bit is always read as 0. The write value shou be 0.
12	BRPS4	0	R/W	Baud Rate Generator's Prescalar Setting (BRPS)
11	BRPS3	0	R/W	Set the master clock division ratio BRPS.
10	BRPS2	0	R/W	00000: (× 1/32)
9	BRPS1	0	R/W	00001: (× 1/1)
8	BRPS0	0	R/W	00010: (× 1/2) 11111: (× 1/31)
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.

011: Prescalar output × 1/16

100: Prescalar output × 1/32

Note: Other than above is reserved (setting prohi

17.3.3 Serial Transmit Data Assign Register (SITDAR)

SITDAR is used to specify the position of the transmit data in a frame (slot number). SIT initialized by a power-on reset and software reset.

Bit	Bit Name	Initial Value	R/W	Description
15	TDLE	0	R/W	Transmit Left Channel Data Enable 0: Disables left channel data transmission 1: Enables left channel data transmission
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
11	TDLA3	0	R/W	Transmit Left Channel Data Assigns
10	TDLA2	0	R/W	Specify the position of left-channel data in transmi
9	TDLA1	0	R/W	as B′0000 to B′1110. Transmit data for the left cha specified in bits SITDL15 to SITDL0 in SITDR.
8	TDLA0	0	R/W	Note: If the TDLA3 to TDLA0 bits are set to B′111 operation is not guaranteed.
7	TDRE	0	R/W	Transmit Right Channel Data Enable 0: Disables right channel data transmission 1: Enables right channel data transmission

4	—	0	R	These bits are always read as 0. The write value always be 0.
3	TDRA3	0	R/W	Transmit Right Channel Data Assigns
2	TDRA2	0	R/W	Specify the position of right-channel data in trans as B′0000 to B′1110. Transmit data for the right c specified in bits SITDR15 to SITDR0 in SITDR.
1	TDRA1	0	R/W	
0	TDRA0	0	R/W	Note: If the TDRA3 to TDRA0 bits are set to B′1 operation is not guaranteed.

17.3.4 Serial Receive Data Assign Register (SIRDAR)

SIRDAR is used to specify the position of the receive data in a frame. SIRDAR is initial power-on reset or software reset.

Bit	Bit Name	Initial Value	R/W	Description
15	RDLE	0	R/W	Receive Left Channel Data Enable 0: Disables left channel data reception 1: Enables left channel data reception
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
3	RDRA3	0	R/W	Receive Right Channel Data Assigns 3 to 0
2	RDRA2	0	R/W	Specify the position of right-channel data in a rec
1	RDRA1	0	R/W	frame as B′0000 to B′1110. Receive data for the channel is stored in bits SIRDR15 to SIRDR0 in
0	RDRA0	0	R/W	Note: If the RDRA3 to RDRA0 bits are set to B′1 operation is not guaranteed.

17.3.5 Serial Control Data Assign Register (SICDAR)

SICDAR is used to specify the position of the control data in a frame. SICDAR can be sp only when the FL3 to FL0 bits in SIMDR are specified as 1xxx. SICDAR is initialized by power-on reset or software reset.

Bit	Bit Name	Initial Value	R/W	Description
15	CD0E	0	R/W	Control Channel 0 Data Enable 0: Disables transmission and reception of control 0 data 1: Enables transmission and reception of control 0 data
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.

				0: Disables transmission and reception of control data 1: Enables transmission and reception of control data
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
3	CD1A3	0	R/W	Control Channel 1 Data Assigns 3 to 0 Specify the position of control channel 1 data in a or transmit frame as B′0000 to B′1110. Transmit the control channel 1 data is specified in bits SITC SITC10 in SITCR. Receive data for the control ch data is stored in bits SIRC115 to SIRC10 in SIRC Note: If the CD1A3 to CD1A0 bits are set to B′11 operation is not guaranteed.
2	CD1A2	0	R/W	
1	CD1A1	0	R/W	
0	CD1A0	0	R/W	

				generated in the baud rate generator to the SCK_ 0: Disables the SCK_SIO output (outputs 0) 1: Enables the SCK_SIO output
14	FSE	0	R/W	Frame Synchronous Signal Output Enable This bit is valid in master mode. If this bit is set to SIOF initializes the frame counter and initiates the operation. 0: Disables the SIOFSYNC output (outputs 0) 1: Enables the SIOFSYNC output
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
9	TXE	0	R/W	Transmission Enable This bit setting becomes valid at the start of the n (at the rising edge of the SIOFSYNC signal) and v valid data is stored in the transmit FIFO. When th setting for this bit becomes valid, the SIOF issues transmission transfer request according to the set the TFWM bit in SIFCTR. When transmit data is s the transmit FIFO, transmission of data from the T pin begins. This bit is initialized by a transmit rese 0: Disables data transmission from TXD_SIO (out 1: Enables data transmission from TXD_SIO

				1: Enables data reception from RXD_SIO
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
1	TXRST	0	R/W	Transmission Reset This bit setting becomes valid immediately. Whe setting for this bit becomes valid, the SIOF imme sets transmit data from the TXD_SIO pin to 1, a initializes the transmission data register and tran related status register. The following are initializ • SITDR • Transmit FIFO write/read pointer • TCRDY, TFEMP, and TDREQ bits in SISTR • TXE bit As the SIOF is cleared automatically at the com reset operation, this bit is always read as 0. 0: Transmission operation is not reset 1: Resets transmission operation

- RXE bit

As the SIOF is cleared automatically at the comp
reset operation, this bit is always read as 0.

0: Reception operation is not reset

1: Resets reception operation

				000: Issue a transfer request when 16 stages of trans are empty. 001: Reserved (setting prohibited) 010: Reserved (setting prohibited) 011: Reserved (setting prohibited) 100: Issue a transfer request when 12 or more stages transmit FIFO are empty. 101: Issue a transfer request when 8 or more stages transmit FIFO are empty. 110: Issue a transfer request when 4 or more stages transmit FIFO are empty. 111: Issue a transfer request when 1 or more stages transmit FIFO are empty.
12	TFUA4	1	R	Transmit FIFO Usable Area Indicate the number of words that can be transferred CPU or DMAC as B′00000 to B′10000.
11	TFUA3	0	R	
10	TFUA2	0	R	
9	TFUA1	0	R	
8	TFUA0	0	R	

				011: Reserved (setting prohibited) 100: Issue a transfer request when 4 or more stages o FIFO are valid. 101: Issue a transfer request when 8 or more stages o FIFO are valid. 110: Issue a transfer request when 12 or more stages receive FIFO are valid. 111: Issue a transfer request when 16 stages of receiv are valid.
4	RFUA4	0	R	Receive FIFO Usable Area Indicate the number of words that can be transferred b CPU or DMAC as B′00000 to B′10000.
3	RFUA3	0	R	
2	RFUA2	0	R	
1	RFUA1	0	R	
0	RFUA0	0	R	

14	TCRDY	0	R	Transmit Control Data Ready This bit indicates a state of the SIOF. If SITCR is wri SIOF clears this bit. This bit is valid when the TXE bi is set to 1. If the issue of interrupts by this bit is enab SIOF issues a control interrupt. If SITCR is written w bit is cleared to 0, SITCR is over-written and the prev contents of SITCR are not output from the TXD_SIO 0: Indicates that a write to SITCR is disabled 1: Indicates that a write to SITCR is enabled Note: When using this bit, see 2 in section 17.5, Usa
13	TFEMP	0	R	Transmit FIFO Empty This bit indicates a state; if SITDR is written, the SIO this bit. This bit is valid when the TXE bit in SICTR is issue of interrupts by this bit is enabled, the SIOF iss control interrupt. 0: Indicates that transmit FIFO is not empty 1: Indicates that transmit FIFO is empty
12	TDREQ	0	R	Transmit Data Transfer Request A transmit data transfer request is issued when the e space in the transmit FIFO exceeds the size specifie TFWM bit in SIFCTR. This bit is valid when the TXE SICTR is 1. This bit indicates a state of the SIOF. If t empty space in the transmit FIFO is less than the siz specified by the TFWM bit in SIFCTR, the SIOF clea If the issue of interrupts by this bit is enabled, the SI a transmit interrupt. 0: No transfer request 1: Transfer request

				latest data. 0: Indicates that SIRCR stores no valid data 1: Indicates that SIRCR stores valid data
9	RFFUL	0	R	Receive FIFO Full This bit indicates a state. If SIRDR is read, the SIO this bit. This bit is valid when the RXE bit in SICTR the issue of interrupts by this bit is enabled, the SI issues a control interrupt. 0: Receive FIFO not full 1: Receive FIFO full
8	RDREQ	0	R	Receive Data Transfer Request A receive data transfer request is issued when the space in the receive FIFO exceeds the size specifi RFWM bit in SIFCTR. This bit is valid when the RXE bit in SICTR is 1. Th indicates a state the SIOF. If the size of valid space receive FIFO is less than the size specified by the bit in SIFCTR, the SIOF clears this bit. If the issue of interrupts by this bit is enabled, the S issues a receive interrupt. 0: Indicates that the size of valid space in the recei does not exceed the size specified by the RFWI SIFCTR. 1: Indicates that the size of valid space in the recei exceeds the size specified by the RFWM bit in S
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value sl always be 0.

				0: Indicates that no frame synchronization error occ 1: Indicates that a frame synchronization error occu
3	TFOVR	0	R/W	Transmit FIFO Overrun Transmit FIFO overrun means that there has been to write to SITDR when the transmit FIFO is full. Wh transmit overrun occurs, written data is ignored. This bit is valid when the TXE bit in SICTR is 1. Wh written to this bit, the contents are cleared. If the iss interrupts by this bit is enabled, the SIOF issues an interrupt. 0: No transmit FIFO overrun 1: Transmit FIFO overrun
2	TFUDR	0	R/W	Transmit FIFO Underrun Transmit FIFO underrun means that loading for tran has occurred when the transmit FIFO is empty. Wh transmit underrun occurs, the SIOF repeatedly send previous transmit data. This bit is valid when the TXE bit in SICTR is 1. Wh written to this bit, the contents are cleared. If the iss interrupts by this bit is enabled, the SIOF issues an interrupt. 0: No transmit FIFO underrun 1: Transmit FIFO underrun

				0: No receive FIFO underrun 1: Receive FIFO underrun
0	RFOVR	0	R/W	Receive FIFO Overrun Receive FIFO overrun means that writing has occu when the receive FIFO is full. When a receive overr occurs, the SIOF indicates the overrun, and receive lost. This bit is valid when the RXE bit in SICTR is 1. Wh written to this bit, the contents are cleared. If the iss interrupts by this bit is enabled, the SIOF issues an interrupt. 0: No receive FIFO overrun 1: Receive FIFO overrun

14	TCRDYE	0	R/W	Transmit Control Data Ready Enable 0: Disables interrupts due to transmit control data 1: Enables interrupts due to transmit control data (control interrupt)
13	TFEMPE	0	R/W	Transmit FIFO Empty Enable 0: Disables interrupts due to transmit FIFO empty 1: Enables interrupts due to transmit FIFO empty interrupt)
12	TDREQE	0	R/W	Transmit Data Transfer Request Enable 0: Disables interrupts due to transmit data transfe 1: Enables interrupts due to transmit data transfer (transmit interrupt)
11	—	0	R	Reserved This bit is always read as 0. The write value shoul be 0.
10	RCRDYE	0	R/W	Receive Control Data Ready Enable 0: Disables interrupts due to receive control data 1: Enables interrupts due to receive control data r (control interrupt)
9	RFFULE	0	R/W	Receive FIFO Full Enable 0: Disables interrupts due to receive FIFO full 1: Enables interrupts due to receive FIFO full (cor interrupt)

				0: Disables interrupts due to frame synchronization 1: Enables interrupts due to frame synchronization (error interrupt)
3	TFOVRE	0	R/W	Transmit FIFO Overrun Enable 0: Disables interrupts due to transmit FIFO overrun 1: Enables interrupts due to transmit FIFO overrun interrupt)
2	TFUDRE	0	R/W	Transmit FIFO Underrun Enable 0: Disables interrupts due to transmit FIFO underru 1: Enables interrupts due to transmit FIFO underru interrupt)
1	RFUDRE	0	R/W	Receive FIFO Underrun Enable 0: Disables interrupts due to receive FIFO underru 1: Enables interrupts due to receive FIFO underrun interrupt)
0	RFOVRE	0	R/W	Receive FIFO Overrun Enable 0: Disables interrupts due to receive FIFO overrun 1: Enables interrupts due to receive FIFO overrun interrupt)

				SITDAR. These bits are valid only when the TDLE bit in S set to 1.
15 to 0	SITDR15 to SITDR0	All 0	W	Right Channel Transmit Data Specify data to be output from the TXD_SIO pi channel data. The position of the right channel transmission frame is specified by the TDRA bi SITDAR. These bits are valid only when the TDLE bit an bit in SITDAR are set to 1 and cleared to 0, res

				These bits are valid only when the RDLE bit in SI set to 1.
15 to 0	SIRDR15 to SIRDR0	All 0	R	Right Channel Receive Data Store data received from the RXD_SIO pin as rig channel data. The position of the right channel da reception frame is specified by the RDRA bit in S These bits are valid only when the RDRE bit in S set to 1.

				control channel 0 data in the transmission or re frame is specified by the CD0A bit in SICDAR. These bits are valid only when the CD0E bit in is set to 1.
15 to 0	SITC115 to SITC10	All 0	W	Control Channel 1 Transmit Data Specify data to be output from the TXD_SIO pi control channel 1 transmit data. The position o control channel 1 data in the transmission or re frame is specified by the CD1A bit in SICDAR. These bits are valid only when the CD1E bit in is set to 1.

				channel 0 data in the transmission or reception specified by the CD0A bit in SICDAR. These bits are valid only when the CD0E bit in S is set to 1.
15 to 0	SIRC115 to SIRC10	All 0	R	Control Channel 1 Receive Data Store data received from the RXD_SIO pin as c channel 1 receive data. The position of the cont channel 1 data in the transmission or reception specified by the CD1A bit in SICDAR. These bits are valid only when the CD1E bit in S is set to 1.

Baud Rate Generator (BRG): In SIOF master mode, the baud rate generator (BRG) is
generate the serial clock. The division ratio is from 1/2 to 1/1024.

Figure 17.2 shows connections for supply of the serial clock.

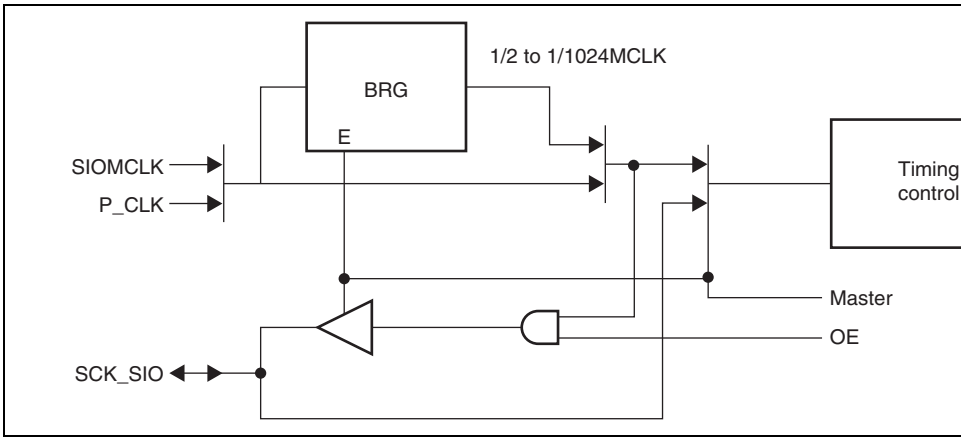


Figure 17.2 Serial Clock Supply

17.4.2 Serial Timing

SIOFSYNC: The SIOFSYNC is a frame synchronous signal. Depending on the transfer r
has the following two functions.

- Synchronous pulse: 1-bit-width pulse indicating the start of the frame
- L/R: 1/2-frame-width pulse indicating the left channel stereo data (L) in high level an
  right channel stereo data (R) in low level

Figure 17.3 shows the SIOFSYNC synchronization timing. The timing of master mode 1,
mode 1, and slave mode 2 is shown in (a) in figure 17.3. The timing of master mode 2 is s
(b) in figure 17.3.

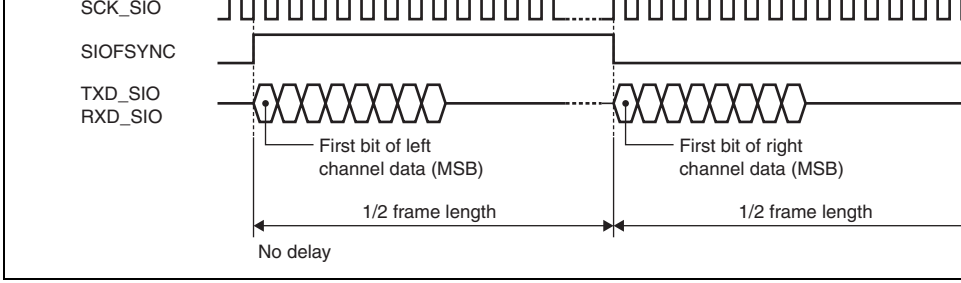


Figure 17.3 Serial Data Synchronization Timing

Transmit/Receive Timing: The TXD_SIO transmission timing and RXD_SIO receptio relative to the SCK_SIO signal can be set as the sampling timing in the following two w transmit/receive timing is set using the REDG bit in SIMDR. In slave mode 1 and slave only falling-edge sampling is available.

- Falling-edge sampling
- Rising-edge sampling

Figure 17.4 shows the transmit/receive timing.

17.4.3 Transfer Data Format

The SIOF performs the following transfer.

- Transmit/receive data: Transfer of 8-bit data/16-bit data/16-bit stereo data
- Control data: Transfer of 16-bit data (uses the specific register as interface)

Transfer Mode: The SIOF supports the following four transfer modes as listed in table 1 transfer mode can be specified by the TRMD1 and TRMD0 bits in SIMDR.

Table 17.3 Serial Transfer Modes

Transfer Mode	SIOFSYNC	Bit Delay	Control Data
Slave mode 1	Synchronous pulse	One bit	Slot position
Slave mode 2	Synchronous pulse	One bit	Secondary FS
Master mode 1	Synchronous pulse	One bit	Slot position
Master mode 2	L/R	No	Not supported

Frame Length: The length of the frame to be transferred by the SIOF is specified by the FL0 bits in SIMDR. Table 17.4 shows the relationship between the settings of the FL3 to and frame length.

1101	16	64	16-bit monaural/stereo
1110	16	128	16-bit monaural/stereo
1111	16	256	16-bit monaural/stereo

[Legend]

x: Don't care.

Slot Position: The SIOF can specify the position of transmit data, receive data, and cont (common to transmission and reception) by slot numbers. The slot number of each data specified by the following registers.

- Transmit data: SITDAR
- Receive data: SIRDAR
- Control data: SICDAR

Only 16-bit slot leugth is valid for control register. In addition, control data is always ass the same slot number both in transmission and reception.

17.4.4 Register Allocation of Transfer Data

Transmit/Receive Data: Writing and reading of transmit or receive data are performed following registers.

- Transmit data writing: SITDR (32-bit access)
- Receive data reading: SIRDR (32-bit access)

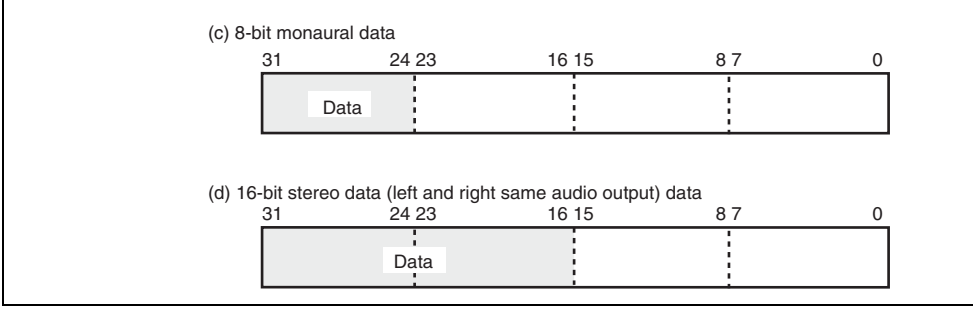


Figure 17.5 Transmit/Receive Data Bit Alignment

Note: In the figure, only the shaded areas are transmitted or received as valid data. Data unshaded areas is not transmitted or received.

Monaural or stereo can be specified for transmit data by the TDLE bit and TDRE bit in S Monaural or stereo can be specified for receive data by the RDLE bit and RDRE bit in SI To achieve left and right same audio output while stereo is specified for the transmit data the TLREP bit in SITDAR. Tables 17.5 and 17.6 show the audio mode specification for t data and that for receive data, respectively. To execute 8-bit monaural transmission or rec use the left channel.

Mode	Bit RDLE	RDRE
Monaural	1	0
Stereo	1	1

Note: Left and right same audio mode is not supported in receive data.

Control Data: Control data is written to or read from by the following registers.

- Transmit control data write: SITCR (32-bit access)
- Receive control data read: SIRCR (32-bit access)

Figure 17.6 shows the control data and bit alignment in SITCR and SIRCR.

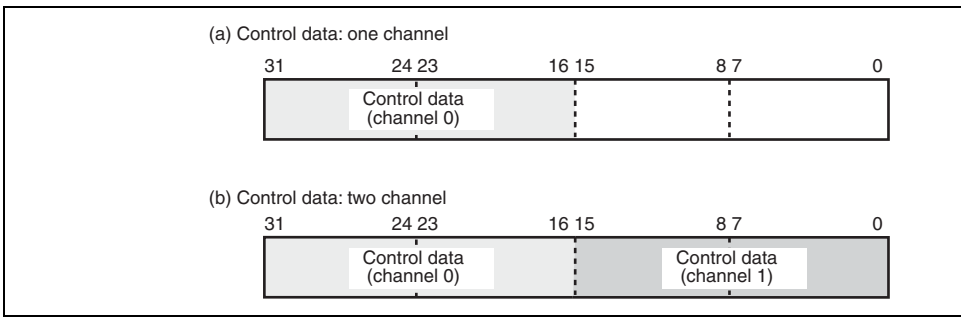


Figure 17.6 Control Data Bit Alignment

The number of channels in control data is specified by CD0E and CD1E bits in SICDAR 17.7 shows the relationship between the number of channels in control data and bit setti use only one channel in control data, use channel 0.

The SIOF supports the following two control data interface methods.

- Control by slot position
- Control by secondary FS

Control data is valid only when slot length is specified as 16 bits and MSB-first transmission/reception is selected.

Control by Slot Position (Master Mode 1): Control data is transferred for all frames tra or received by the SIOF by specifying the slot position of control data. This method can b both SIOF master and slave modes. Figure 17.7 shows an example of control data interfa by slot position control.

Note: When using this control method, use PCLK as the master clock (master clock sel (MSSEL) = 1).

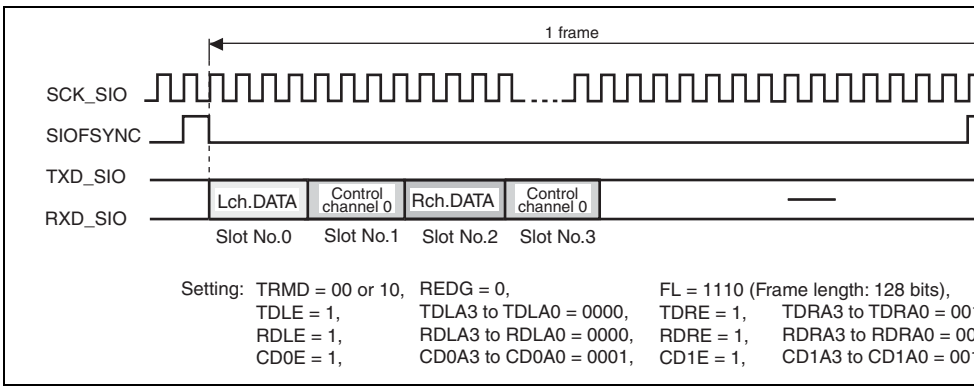


Figure 17.7 Control Data Interface (Slot Position)

SIRCR) synchronously with the secondary FT.

Figure 17.8 shows an example of control data interface timing by secondary FS.

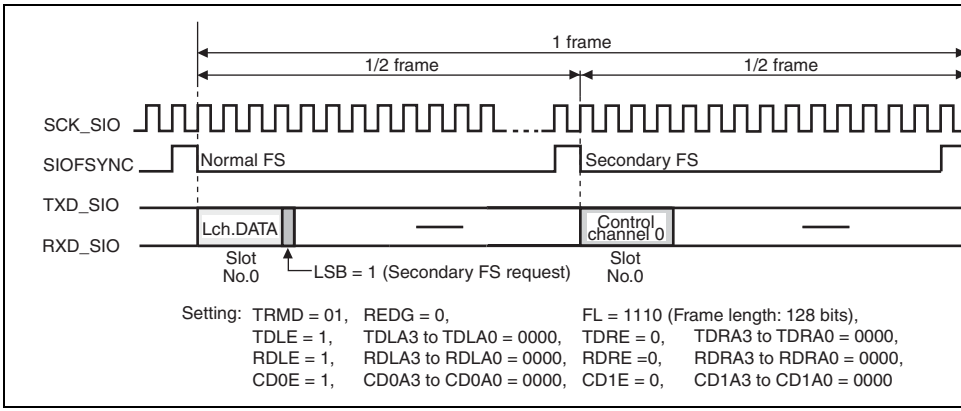


Figure 17.8 Control Data Interface (Secondary FS)

17.4.6 FIFO

Overview: The transmit and receive FIFOs of the SIOF have the following features.

- Sixteen-stage 32-bit FIFOs for transmission and reception
- The FIFO pointer can be modified in one read or write cycle regardless of access siz
  CPU and DMAC. (One-stage 32-bit FIFO access cannot be divided into multiple acc
- Regardless of access size, the number of access cycles is always two cycles of the P-

Transfer Request: The SIOF indicates a transfer request of the FIFO in the following t
SISTR.

000	1	Empty area is 16 stages	Smal
100	4	Empty area is 12 stages or more	
101	8	Empty area is 8 stages or more	
110	12	Empty area is 4 stages or more	
111	16	Empty area is 1 stage or more	Large

Table 17.9 Conditions to Issue Receive Request

RFWM2 to RFWM0	Number of Requested Stages	Receive Request	Used
000	1	Valid data is 1 stage or more	Smal
100	4	Valid data is 4 stages or more	
101	8	Valid data is 8 stages or more	
110	12	Valid data is 12 stages or more	
111	16	Valid data is 16 stages	Large

The number of stages of the FIFO is always sixteen even if the data area or empty area ex above stage number. Accordingly, an overrun error or underrun error occurs if data area o area exceeds sixteen FIFO stages. FIFO transmission or reception request is cancelled wh above condition is not satisfied even if the FIFO is not empty or full.

Number of FIFOs: The number of FIFO stages used in transmission and reception is ind the following register.

- Transmit FIFO: The number of empty FIFO stages are indicated by the TFUA4 to TF in SIFCTR.

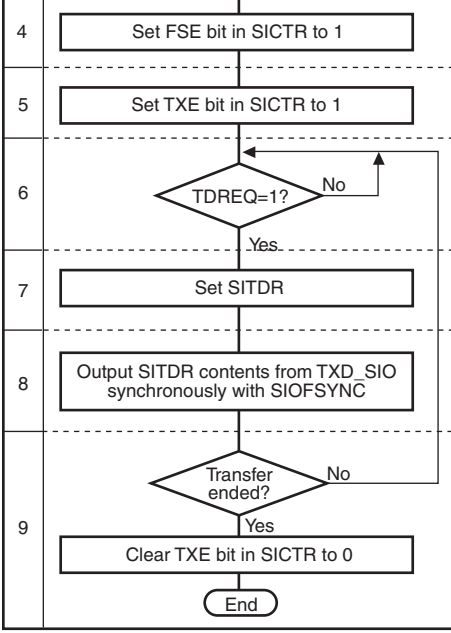


Figure 17.9 Example of Transmission Operation in Master Mode

Reception in Master Mode: Figure 17.10 shows an example of settings and operation fo mode reception.

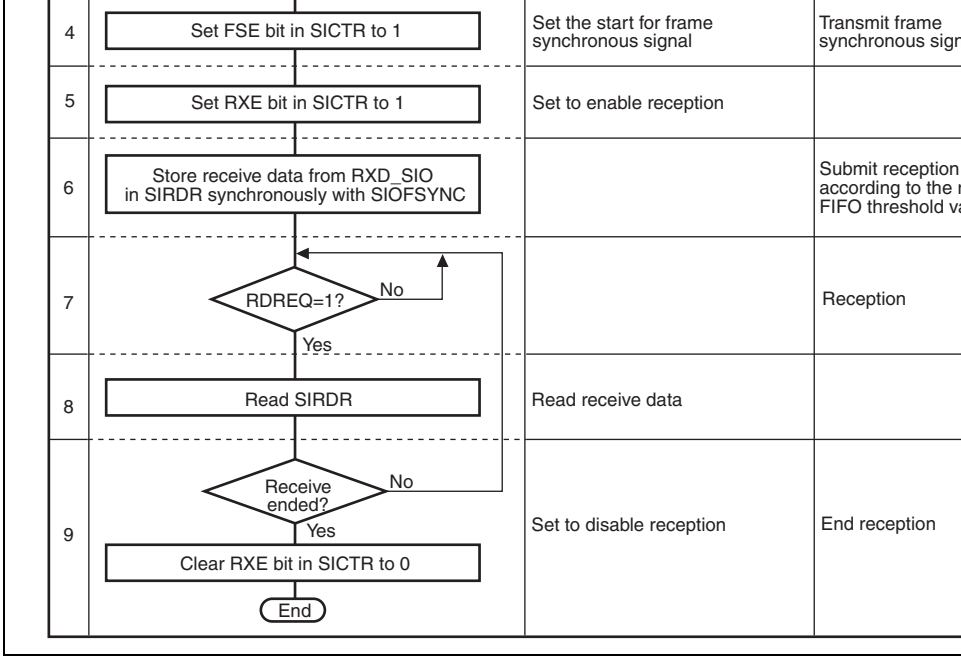


Figure 17.10 Example of Reception Operation in Master Mode

Transmission in Slave Mode: Figure 17.11 shows an example of settings and operation mode transmission.

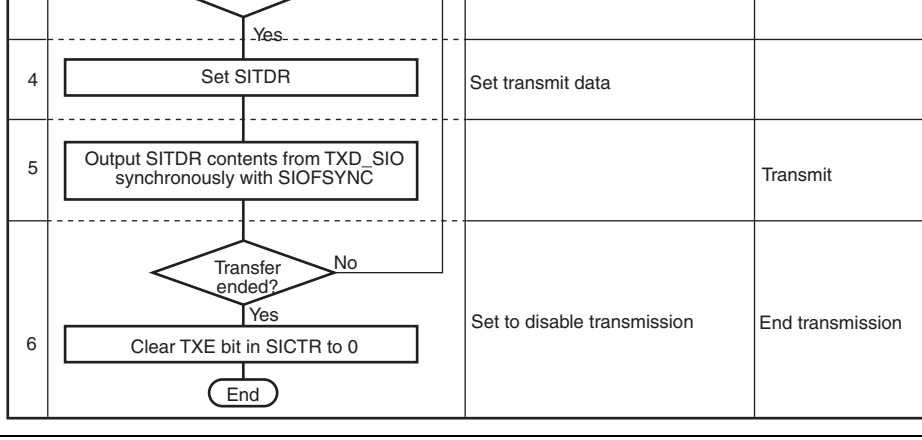


Figure 17.11 Example of Transmission Operation in Slave Mode

Reception in Slave Mode: Figure 17.12 shows an example of settings and operation for
mode reception.

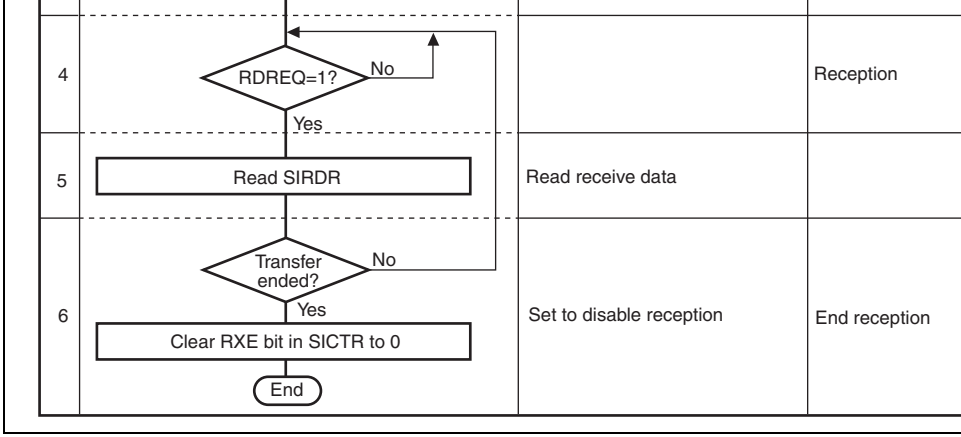


Figure 17.12 Example of Reception Operation in Slave Mode

Transmission/Reception Reset: The SIOF can separately reset the transmission and rec units by setting the following bits to 1.

- Transmission reset: TXRST bit in SICTR
- Reception reset: RXRST bit in SICTR

Module Stop: In the module stop state, the SIOF stops transmit/receive operation with co
all registers retained. If transmit/receive operation is not performed immediately after the
stop state is cleared, issue a transmit/receive reset.

17.4.8 Interrupts

The SIOF has four types of interrupts listed below. This classification is reflected to the I
(SIOF0) and IRR8 (SIOF1) of the interrupt controller (INTC).

- Transmit interrupt (TXI)
- Receive interrupt (RXI)
- Control interrupt (CCI)
- Error interrupt (ERI)

Interrupt Sources: Interrupts can each be issued by several sources. Each source is shov
SIOF status in SISTR. Table 17.11 lists the SIOF interrupt sources.

			ready	stores valid data.
5		TFEMP	Transmit FIFO empty	The transmit FIFO is emp
6		RFFUL	Receive FIFO full	The receive FIFO is full.
7	Error (ERI)	TFUDR	Transmit FIFO underrun	Serial data transmission ti arrived while the transmit empty.
8		TFOVR	Transmit FIFO overrun	Write to the transmit FIFO performed while the trans is full.
9		RFOVR	Receive FIFO overrun	Serial data is received wh receive FIFO is full.
10		RFUDR	Receive FIFO underrun	The receive FIFO is read receive FIFO is empty.
11		FSERR	Frame synchronization error	A synchronous signal is in before the specified bit tim been passed (in slave mo

Whether an interrupt is issued or not as the result of an interrupt source is determined by SIIER settings. If an interrupt source is set to 1, and the corresponding bit in SIIER is se SIOF issues each interrupt.

Transmit/Receive Interrupt Flag: Transmit and receive interrupts are sent to the INTC DMAC by this interrupt flag based on the values of bits TDREQ and RDREQ in SISTR Table17.12 shows the setting condition of the transmit/receive interrupt flag.

- Transmit FIFO underrun (TFUDR)
  The immediately preceding transmit data is again transmitted.
- Transmit FIFO overrun (TFOVR)
  The contents of the transmit FIFO are protected, and the write operation causing the o
  ignored.
- Receive FIFO overrun (RFOVR)
  Data causing the overrun is discarded and lost.
- Receive FIFO underrun (RFUDR)
  The latest read data is output on the bus (undefined value as specification).
- Frame synchronization error (FSERR)
  The internal counter is reset according to the FSYN signal in which an error occurs.

17.4.9 Transmission and Reception Timing

Examples of the SIOF serial transmission and reception are shown in figure 17.13 throug
17.19.

8-bit Monaural Data (1): Synchronous pulse method, falling edge sampling, slot No.0 u
transmit and receive data, frame length = 8 bits

Figure 17.13 Transmission and Reception Timings (8-Bit Monaural Data (

8-bit Monaural Data (2): Synchronous pulse method, falling edge sampling, slot No.0 transmit and receive data, frame length = 16 bits

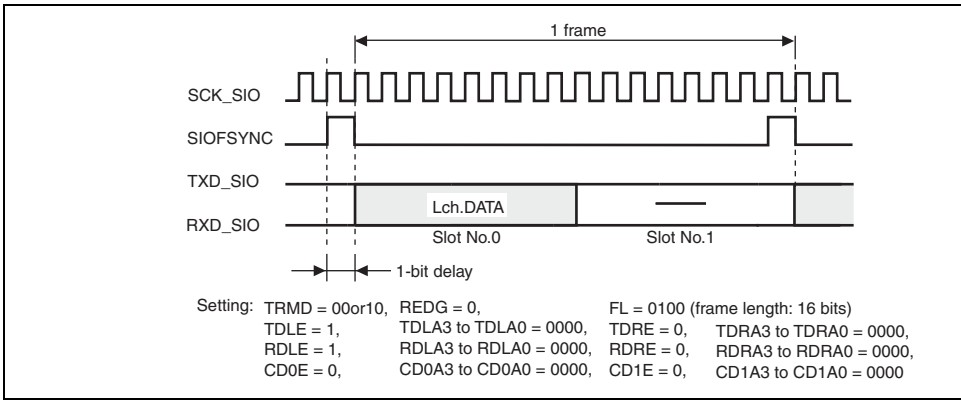


Figure 17.14 Transmission and Reception Timings (8-Bit Monaural Data (

16-bit Monaural Data (1): Synchronous pulse method, falling edge sampling, slot No.0 transmit and receive data, frame length = 64 bits

16-bit Stereo Data (1): L/R method, rising edge sampling, slot No.0 used for left channe
slot No.1 used for right channel data, frame length = 32 bits

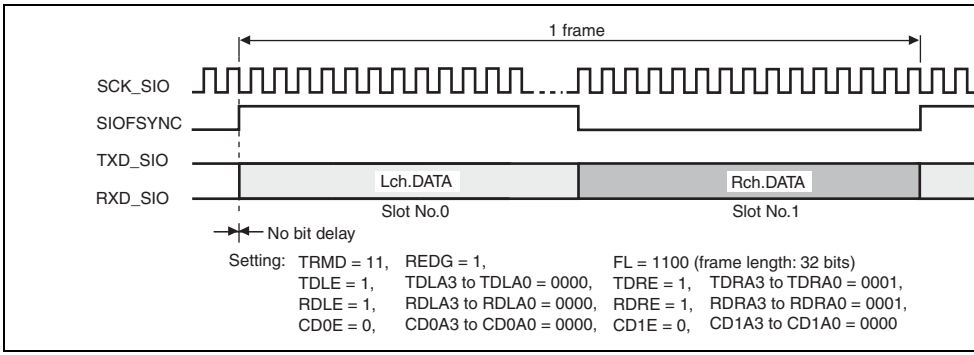


Figure 17.16 Transmission and Reception Timings (16-Bit Stereo Data (1))

16-bit Stereo Data (2): L/R method, rising edge sampling, slot No.0 used for left channe
data, slot No.1 used for left channel receive data, slot No.2 used for right channel transmi
slot No.3 used for right channel receive data, frame length = 64 bits

Figure 17.17 Transmission and Reception Timings (16-Bit Stereo Data (2

16-bit Stereo Data (3): Synchronous pulse method, falling edge sampling, slot No.0 use channel data, slot No.2 used for right channel data, slot No.1 used for control channel 0 No.3 used for control channel 1 data, frame length = 128 bits

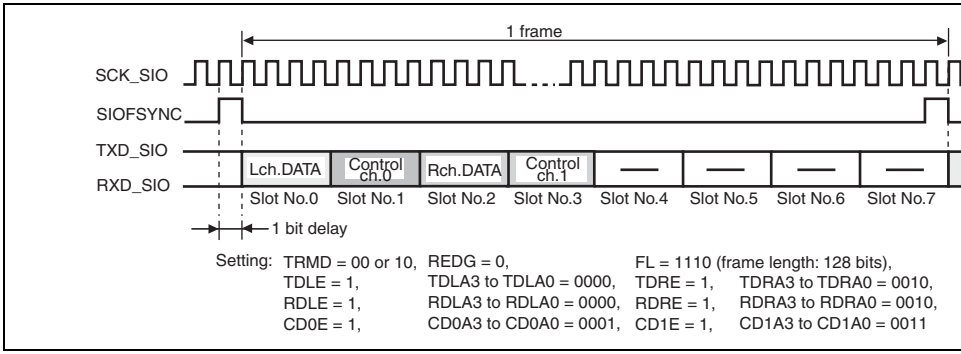


Figure 17.18 Transmission and Reception Timings (16-Bit Stereo Data (3

16-bit Monaural Data (2): Synchronous pulse method, falling edge sampling, request f secondary FS, slot No.0 used for left channel data, slot No.0 used for control channel 0 c length = 128 bits

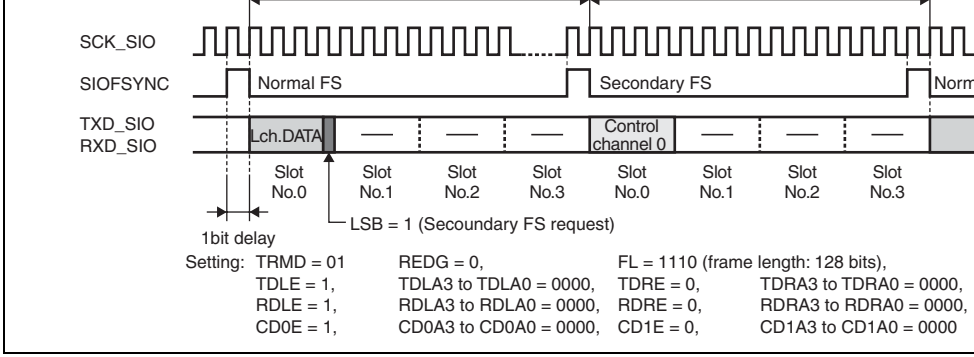


Figure 17.19 Transmission and Reception Timings (16-Bit Monaural Data (2

The TCRDY value may become 1 before transmit control data is sent, and if the nex
data is written to the control data register at this point, the control data waiting to be
be overwritten and erased.

At this time, also, the control sequence is disrupted and the SIOF switches around th
FS and secondary FS, with the result that transmission/reception of data and control
no longer be performed normally.

The control data register should therefore be written to after transmit control data has
sent.

Example:

Check RCRDY, and write to the control data register when RCRDY is 1.

After transmit control data has been written to, it is essential to read the receive cont
(SIRCR) and clear RCRDY.

3. DMA transfer

   Do not use 16-byte DMA transfer. (See section 13.4.4, DMA Transfer Types.)

4. Access from the CPU

   When performing access from the CPU, do not access the SIOF's transmit/receive FI
   consecutively, but instead insert an access to somewhere else between SIOF transmi
   FIFO accesses.

5. Transmit/receive FIFO underflow

   If the transmit/receive FIFO underflows during a transmit/receive operation, control
   SIOF's transmit/receive FIFO may fail and data may be lost.

   To prevent this, either set a watermark so that an underflow does not occur, or execu
   transmit reset (TXRST) or receive reset (RXRST) when an empty interrupt is genera

6. Transmit/receive reset execution

   When using the SIOF again after a transmit/receive operation ends, or after erroneou
   operation occurs, first execute a transmit reset (TXRST) or receive reset (RXRST).

interface input pins for determining whether to receive or transfer packets input to both
controllers. The TSU also has a total 6-kbyte transfer FIFO for retaining packets to be tr
allowing allocation of transfer FIFO capacity to be set freely for the transfer conditions
1 and port 1 to 0. The Ethernet controller is connected to the Ethernet Direct Memory A
Controller (E-DMAC) for Ethernet controller inside the LSI, and carries out high-speed
transfer to and from the memory.

Figure 18.1 shows a configuration of the EtherC.

18.1 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
- Ethernet frame relay function by the TSU
- Qtag addition and deletion functions conforming to IEEE802.1Q specifications (whe
  relay is performed by the TSU)
- MAC address filtering function by the multicast (group) address
- Ethernet frame receive and transfer control functions by the CAM (Content Address
  Memory) interface signals input externally

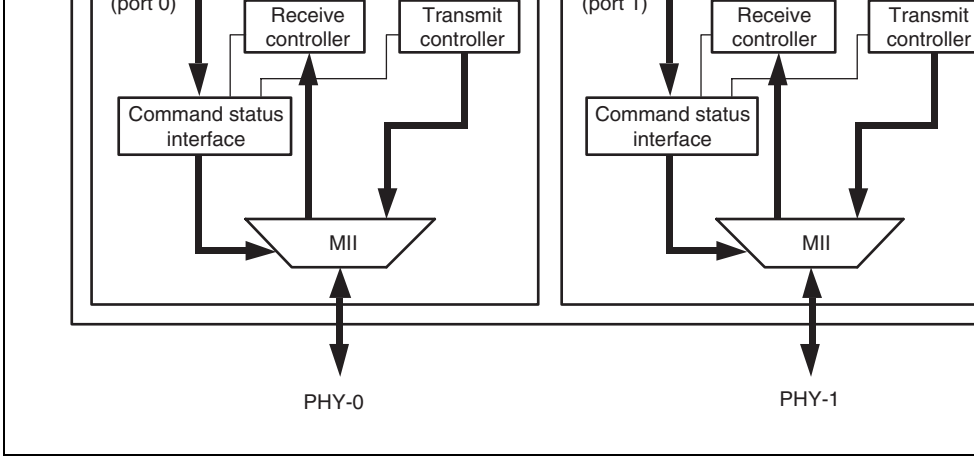


Figure 18.1 Configuration of EtherC

Transmit enable	0	TX-EN0*[1]	O	Indicates that transmit data is read ETXD3 to ETXD0
Transmit data	0	ETXD03 to ETXD00*[1]	O	4-bit transmit data
Transmit error	0	TX-ER0*[1]	O	Notifies PHY_LSI of error during transmission
Receive data valid	0	RX-DV0*[1]	I	Indicates that valid receive data is ERXD3 to ERXD0
Receive data	0	ERXD03 to ERXD00*[1]	I	4-bit receive data
Receive error	0	RX-ER0*[1]	I	Identifies error state occurred duri reception
Carrier detection	0	CRS0*[1]	I	Carrier detection signal
Collision detection	0	COL0*[1]	I	Collision detection signal
Management data clock	0	MDC0*[1]	O	Reference clock signal for informa transfer via MDIO
Management data I/O	0	MDIO0*[1]	I/O	Bidirectional signal for exchange o management information between and PHY
Link status	0	LNKSTA0	I	Inputs link status from PHY
General-purpose external output	0	EXOUT0	O	Signal indicating value of register- (ECMR0-ELB)
Wake-On-LAN	0	WOL0	O	Signal indicating reception of Mag
Transmit clock	1	TX-CLK1*[1]	I	TX-EN, ETXD3 to ETXD0, TX-ER reference signal
Receive clock	1	RX-CLK1*[1]	I	RX-DV, ERXD3 to ERXD0, RX-ER reference signal

		ERXD1*[1]		
Receive error	1	RX-ER1*[1]	I	Identifies error state occurred durin reception
Carrier detection	1	CRS1*[1]	I	Carrier detection signal
Collision detection	1	COL1*[1]	I	Collision detection signal
Management data clock	1	MDC1*[1]	O	Reference clock signal for informati transfer via MDIO
Management data I/O	1	MDIO1*[1]	I/O	Bidirectional signal for exchange of management information between t and PHY
Link status	1	LKNSTA1	I	Inputs link status from PHY
General-purpose external output	1	EXOUT1	O	Signal indicating value of register-b (ECMR1-ELB)
Wake-On-LAN	1	WOL1	O	Signal indicating reception of Magic
CAM input 0	—	CAMSEN0*[2]	I	CAM interface signal input 0
CAM input 1	—	CAMSEN1*[2]	I	CAM interface signal input 1
Bus release request	—	$\overline{\text{ARBUSY}}$*[3]	O	Signal indicating bus release reque the threshold value set for the data in the receive FIFO has been excee

Notes: 1. MII signal conforming to IEEE802.3u

2. The CAM input signal function is set by the CAMSEL03 to CAMSEL00 and CA to CAMSEL10 in the TSU_FWSLC register.

3. Refer to section 19, Ethernet Controller Direct Memory Access Controller (E-D and section 19.2.18, Overflow Alert FIFO Threshold Register (FCFTR).

Port 0

- EtherC mode register (ECMR0)
- EtherC status register (ECSR0)
- EtherC interrupt permission register (ECSIPR0)
- PHY interface register (PIR0)
- MAC address high register (MAHR0)
- MAC address low register (MALR0)
- Receive frame length register (RFLR0)
- PHY status register (PSR0)
- Transmit retry over counter register (TROCR0)
- Delayed collision detect counter register (CDCR0)
- Lost carrier counter register (LCCR0)
- Carrier not detect counter register (CNDCR0)
- CRC error frame receive counter register (CEFCR0)
- Frame receive error counter register (FRECR0)
- Too-short frame receive counter register (TSFRCR0)
- Too-long frame receive counter register (TLFRCR0)
- Residual-bit frame receive counter register (RFCR0)
- Multicast address frame receive counter register (MAFCR0)
- IPG register (IPGR0)

- Transmit retry over counter register (TROCR1)
- Delayed collision detect counter register (CDCR1)
- Lost carrier counter register (LCCR1)
- Carrier not detect counter register (CNDCR1)
- CRC error frame receive counter register (CEFCR1)
- Frame receive error counter register (FRECR1)
- Too-short frame receive counter register (TSFRCR1)
- Too-long frame receive counter register (TLFRCR1)
- Residual-bit frame receive counter register (RFCR1)
- Multicast address frame receive counter register (MAFCR1)
- IPG register (IPGR1)

TSU Control Registers:

- TSU counter reset register (TSU_CTRST)
- Relay enable register (Port 0 to 1) (TSU_FWEN0)
- Relay enable register (Port 1 to 0) (TSU_FWEN1)
- Relay FIFO size select register (TSU_FCM)
- Relay FIFO overflow alert set register (port 0) (TSU_BSYSL0)
- Relay FIFO overflow alert set register (port 1) (TSU_BSYSL1)
- Transmit/relay priority control mode register (port 0) (TSU_PRISL0)
- Transmit/relay priority control mode register (port 1) (TSU_PRISL1)
- Receive/relay function set register (port 0 to 1) (TSU_FWSL0)
- Receive/relay function set register (port 1 to 0) (TSU_FWSL1)
- Relay function set register (common) (TSU_FWSLC)
- Qtag addition/deletion set register (port 0 to 1) (TSU_QTAGM0)
- Qtag addition/deletion set register (port 1 to 0) (TSU_QTAGM1)

- Transmit frame counter register (port 0) (normal transmission only) (TXNLCR0)
- Transmit frame counter register (port 0) (normal and error transmission) (TXALCR0
- Receive frame counter register (port 0) (normal reception only) (RXNLCR0)
- Receive frame counter register (port 0) (normal and error reception) (RXALCR0)
- Relay frame counter register (port 1 to 0) (normal relay only) (FWNLCR0)
- Relay frame counter register (port 1 to 0) (normal and error relay) (FWALCR0)
- Transmit frame counter register (port 1) (normal transmission only) (TXNLCR1)
- Transmit frame counter register (port 1) (normal and error transmission) (TXALCR1
- Receive frame counter register (port 1) (normal reception only) (RXNLCR1)
- Receive frame counter register (port 1) (normal and error reception) (RXALCR1)
- Relay frame counter register (port 0 to 1) (normal relay only) (FWNLCR1)
- Relay frame counter register (port 0 to 1) (normal and error relay) (FWALCR1)

				should always be 0.
0	ARST	0	R/W	Software Reset When written with 1, a software reset is issued modules related to the Ethernet (for 64 cycles external bus clock Bϕ). Writing 0 does not affect this bit. This bit is alw as 0. While a software reset is issued, register acce modules related to the Ethernet is prohibited. following registers are not initialized by a softw reset. TSU_ADRH0 to TSU_ADRH31, TSU_ADRL0 TSU_ADRL31, TXNLCR0, TXNLCR1, TXALC TXALCR1, RXNLCR0, RXNLCR1, RXALCR0, RXALCR1, FWNLCR0, FWNLCR1, FWALCR FWALCR1

Bit	Bit Name	Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
13	MCT	0	R/W	Multicast Address Frame Receive Mode 0: Frames other than the multicast address CAM entry table 0 to 31 (H/L) registers a received. However, if the on-chip CAM e reference is disabled, all multicast addre are received. 1: Only the multicast address set by the CA table 0 to 31 (H/L) registers is received.
12	PRCEF	0	R/W	CRC Error Frame Reception Enable 0: A receive frame including a CRC error is as a frame with an error. 1: A receive frame including a CRC error is as a frame without an error. When this bit is cleared to 0, the CRC error reflected in ECSR of the E-DMAC and the s the receive descriptor. When this bit is set t frame is received as a normal frame.
11	—	0	R	Reserved
10	—	0	R	These bits are always read as 0. The write should always be 0.

6	RE	0	R/W	Reception Enable If a switch is made from receive function ena = 1) to disabled (RE = 0) while a frame is be received, the receive function will be enabled reception of the corresponding frame is com 0: Receive function is disabled 1: Receive function is enabled
5	TE	0	R/W	Transmission Enable If a switch is made from transmit function en (TE = 1) to disabled (TE = 0) while a frame is transmitted, the transmit function will be ena transmission of the corresponding frame is completed. 0: Transmit function is disabled 1: Transmit function is enabled
4	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
3	ILB	0	R/W	Internal Loop Back Mode Specifies loopback mode in the EtherC. 0: Normal data transmission/reception is per 1: Data loopback is performed inside the MA EtherC.

				1: High-level output from the EXOUT pin
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 received. All Ethernet frames means all rec frames, irrespective of differences or enabled/disabled status (destination addres broadcast address, multicast bit, etc.). 0: EtherC performs normal operation 1: EtherC performs promiscuous mode ope

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
2	LCHNG	0	R/W	Link Signal Change Indicates that the LNKSTA signal input from th LSI has changed from high to low or low to hig However, signal changes may be detected at t at which the LNKSTA function was selected us PACR of PFC. To check the current Link state, refer to the LM the PHY status register (PSR). 0: Change in the LNKSTA signal has not been 1: Change in the LNKSTA signal has been det (high to low or low to high)
1	MPD	0	R/W	Magic Packet Detection Indicates that a Magic Packet has been detect line. 0: Magic Packet has not been detected 1: Magic Packet has been detected

18.3.4 EtherC Interrupt Permission Register (ECSIPR)

ECSIPR is a 32-bit readable/writable register that enables or disables the interrupt sourc
indicated by ECSR. Each bit can disable or enable interrupts corresponding to the bits in

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
2	LCHNGIP	0	R/W	LINK Signal Changed Interrupt Enable 0: Interrupt notification by the LCHNG bit is d 1: Interrupt notification by the LCHNG bit is e
1	MPDIP	0	R/W	Magic Packet Detection Interrupt Enable 0: Interrupt notification by the MPD bit is disa 1: Interrupt notification by the MPD bit is ena
0	ICDIP	0	R/W	Illegal Carrier Detection Interrupt Enable 0: Interrupt notification by the ICD bit is disal 1: Interrupt notification by the ICD bit is enab

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 to this bit from the MD when the MMD bit is 1.
1	MMD	0	R/W	MII Management Mode Specifies the data read/write direction with r the MII. 0: Read direction is indicated 1: Write direction is indicated
0	MDC	0	R/W	MII Management Data Clock Outputs the value set to this bit from the MD and supplies the MII with the management c clock. For the method of accessing the MII r see section 18.4.6, Accessing MII Registers

	MA16			These bits are used to set the upper 32 bits o MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), the value set in this register is H'01234567.

18.3.7 MAC Address Low Register (MALR)

MALR is a 32-bit readable/writable register that specifies the lower 16 bits of the 48-bit address. The settings in this register are normally made in the initialization process after The MAC address setting must not be changed while the transmitting and receiving func enabled. To switch the MAC address setting, return the EtherC and E-DMAC to their in by means of the SWR bit in EDMR before making settings again.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
15 to 0	MA15 to MA0	All 0	R/W	MAC Address Bits 15 to 0 These bits are used to set the lower 16 bits o address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), the value set in this register is H'000089AB.

11 to 0	RFL11 to RFL0	All 0	R/W	Receive Frame Length 11 to 0 The frame length described here refers to all from the destination address up to the CRC d Frame contents from the destination address data are actually transferred to memory. CRC not included in the transfer. When data that exceeds the specified value i received, the part of the 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: 2,048 bytes

				The Link status can be read by connecting th signal output from the PHY-LSI to the LNKST the polarity, refer to the PHY-LSI specificatio connected.

18.3.10 Transmit Retry Over Counter Register (TROCR)

TROCR is a 32-bit counter that indicates the number of frames that were unable to be tr in 16 transmission attempts including the retransfer. When 16 transmission attempts hav TROCR is incremented by 1. When the value in this register reaches H'FFFFFFFF, the c 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	TROC31 to TROC0	All 0	R/W	Transmit Retry Over Count These bits indicate the number of frames tha unable to be transmitted in 16 transmission including the retransfer.

18.3.12 Lost Carrier Counter Register (LCCR)

LCCR is a 32-bit counter that indicates the number of times the carrier was lost during da transmission. When the value in this register reaches H'FFFFFFFF, the count is halted. Th counter value is cleared to 0 by writing to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	LCC31 to LCC0	All 0	R/W	Lost Carrier Count These bits indicate the number of times the ca lost during data transmission.

18.3.13 Carrier Not Detect Counter Register (CNDCR)

CNDCR is a 32-bit counter that indicates the number of times the carrier could not be det while the preamble was being sent. When the value in this register reaches H'FFFFFFFF, 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	CNDC31 to CNDC0	All 0	R/W	Carrier Not Detect Count These bits indicate the number of times the ca was not detected.

18.3.15 Frame Receive Error Counter Register (FRECR)

FRECR is a 32-bit counter that indicates the number of frames for which a receive error indicated by the RX-ER input pin from the PHY-LSI. FRECR is incremented each time ER pin becomes active. When the value in this register reaches H'FFFFFFFF, the count 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	FREC31 to FREC0	All 0	R/W	Frame Receive Error Count These bits indicate the count of errors during reception.

18.3.16 Too-Short Frame Receive Counter Register (TSFRCR)

TSFRCR is a 32-bit counter that indicates the number of frames of fewer than 64 bytes t been received. When the value in this register reaches H'FFFFFFFF, the count is halted. 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	TSFC31 to TSFC0	All 0	R/W	Too-Short Frame Receive Count These bits indicate the count of frames recei length of less than 64 bytes.

31 to 0	TLFC31 to TLFC0	All 0	R/W	Too-Long Frame Receive Count These bits indicate the count of frames receive length exceeding the value in RFLR.

18.3.18 Residual-Bit Frame Receive Counter Register (RFCR)

RFCR is a 32-bit counter that indicates the number of frames received containing residua (less than an 8-bit unit). When the value in this register reaches H'FFFFFFFF, the count i 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	RFC31 to RFC0	All 0	R/W	Residual-Bit Frame Count These bits indicate the count of frames receive containing residual bits.

18.3.20 IPG Register (IPGR)

IPGR sets the IPG (Inter Packet Gap). This register must not be changed while the trans
and receiving functions of the EtherC mode register (ECMR) are enabled. (For details, r
section 18.4.8, Operation by IPG Setting.)

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
4 to 0	IPG4 to IPG0	H′13	R/W	Inter Packet Gap Sets the IPG value every 4-bit time. H′00: 20-bit time H′01: 24-bit time : : H′13: 96-bit time (Default) : : H′1F: 144-bit time

				When 1 is written to this bit, the values of regist TXNCR0/1, TXALCR0/1, RXNLCR0/1, RXALCI FWNLCR0/1, and FWALCR0/1 are cleared to 0 0 does not affect this bit. These bits are always 0.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

18.3.22 Relay Enable Register (Port 0 to 1) (TSU_FWEN0)

TSU_FWEN0 enables or disables relay operations from the MAC-0 to MAC-1 (writing t relay FIFO).

Bit	Bit Name	Initial Value	R/W	Description
31	FWEN0	0	R/W	Port 0 to 1 Relay Operation Enable 0: Port 0 to 1 relay is disabled 1: Port 0 to 1 relay is enabled When the value of the FCM2 to FCM0 in the TS size select register TSU_FCM is set to H′4, set bit to 1 is prohibited.
30 to 0	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

				When the value of the FCM2 to FCM0 in the size select register TSU_FCM is set to H′3, s bit to 1 is prohibited.
30 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

	FCM0	H'0: Port 0 to 1: 3 kbytes Port 1 to 0: 3 kbytes H'1: Port 0 to 1: 4 kbytes Port 1 to 0: 2 kbytes H'2: Port 0 to 1 : 5 kbytes Port 1 to 0: 1 kbyte H'3: Port 0 to 1: 6 kbytes Port 1 to 0: Not used H'4: Port 0 to 1: Not used Port 1 to 0: 6 kbytes H'5: Port 0 to 1: 1 kbyte Port 1 to 0: 5 kbytes H'6: Port 0 to 1: 2 kbytes Port 1 to 0: 4 kbytes H'7: Setting prohibited Writing to this register is prohibited, after relay op have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is s

				should always be 0.
5 to 0	BSYSL05 to BSYSL00	All 1	R/W	Sets the threshold of the port 0 to 1 TSU FI capacity in 256-byte units when the TSU al MAC-0 that writing in the TSU FIFO will be during relay operations. H′00: 0 byte H′01: 256 bytes H′02: 512 bytes : : H′16: 5632 bytes H′17: 5888 bytes Settings are disabled for H′18 to H′3F. (Ale always carried out.) When H′00 is set, the TSU always alerts th that writing to the TSU FIFO will be disable the value set is above the port 0 to 1 transf capacity set by the FCM2 to FCM0 in TSU_ TSU does not alert the MAC-0 that writing t FIFO will be disabled. Writing to this register is prohibited, after re operations have been enabled once (after t FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1). When the enable bit of relay operations (th in TSU_FWEN0 or the FWEN1 in TSU_FW cleared to 0, the TSU stops alerting the MA writing to the TSU FIFO will be disabled.

				These bits are always read as 0. The write v should always be 0.
5 to 0	BSYSL15 to BSYSL10	All 1	R/W	Sets the threshold of the port 1 to 0 TSU FIF capacity in 256-byte units when the TSU ale MAC-1 that writing in the TSU FIFO will be d during relay operations. H′00: 0 byte H′01: 256 bytes H′02: 512 bytes : : H′16: 5632 bytes H′17: 5888 bytes Settings are disabled for H′18 to H′3F. (Alert always carried out.) When H′00 is set, the TSU always alerts the that writing to the transfer FIFO will be disab When the value set is above the port 1 to 0 T FIFO capacity set by the FCM2 to FCM0 in TSU_FCM, the TSU does not alert the MAC writing that the TSU FIFO will be disabled. Writing to this register is prohibited, after rela operations have been enabled once (after th FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1). When the enable bit of relay operations (the in TSU_FWEN0 or the FWEN1 in TSU_FWE cleared to 0, the TSU stops alerting the MAC writing to the TSU FIFO will be disabled.

				should always be 0.
14 to 12	PRIMD02 to PRIMD00	All 0	R/W	Sets the priority control mode of MAC-0 tra and port 1 to 0 relay operations. H′0: Round robin H′1: Transmission priority H′2: Relay priority H′4: Round robin, however switched to rela when TSU FIFO use amount exceeds value of PRISL07 to PRISL00 H′5: Transmission priority, however switche priority when TSU FIFO use amount e the set value of PRISL07 to PRISL00 Others: Setting prohibited
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.

				H′5F: 6080 bytes Settings are disabled for H′60 to H′FF. When set to H′00, relay always takes priority the value set is above the port 1 to 0 TSU FI capacity set by the FCM2 to FCM0 in TSU_F the PRIMD02 to PRIMD00 are H′4, round ro always be set. If the PRIMD02 to PRIMD00 a transmission always takes priority.

18.3.28 Transmit/Relay Priority Control Mode Register (Port 1) (TSU_PRISL1)

TSU_PRISL1 sets the priority control mode when the transmission request from the E-DI MAC-1 come into collision with port 0 to 1 relay operations. Writing to this register is pr after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the I in TSU_FWEN1 is set to 1).

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.

				priority when TSU FIFO use amount e the set value of PRISL17 to PRISL10 Others: Setting prohibited
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
7 to 0	PRISL17 to PRISL10	All 0	R/W	Sets the threshold value of the port 0 to 1 T capacity in 64-byte units in the event switch relay priority when PRIMD12 to PRIMD10 a H′4 or H′5. H′00: 0 byte H′01: 64 bytes H′02: 128 bytes : : H′5E: 6016 bytes H′5F: 6080 bytes Settings are disabled for H′60 to H′FF. When set to H′00, relay always takes priorit the value set is above the port 0 to 1 TSU F capacity set by the FCM2 to FCM0 in TSU_ the PRIMD12 to PRIMD10 are H′4, round r always be set. If the PRIMD12 to PRIMD10 transmission always takes priority.

31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
11	FW40	0	R/W	Sets the processing method when frames from are addressed to this LSI 0: Frames are not relayed 1: Frames are relayed to port 1
10	FW30	0	R/W	Sets the processing method when frames from are Broadcast. 0: Frames are not relayed 1: Frames are relayed to port 1
9	FW20	0	R/W	Sets the processing method when frames from are multicast. 0: CAM hit: Frames are relayed to port 1 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 1
8	FW10	0	R/W	Sets the processing method when frames from are addressed to other than this LSI. 0: CAM hit: Frames are relayed to port 1 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 1

CAM evaluation results when the multicast frame and the destination are other than this
details, refer to section 18.4.4, CAM Function.) Writing to this register is prohibited, aft
operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 i
TSU_FWEN1 is set to 1).

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
11	FW41	0	R/W	Sets the processing method when frames fro are addressed to this LSI 0: Frames are not relayed 1: Frames are relayed to port 0
10	FW31	0	R/W	Sets the processing method when frames fro are Broadcast. 0: Frames are not relayed 1: Frames are relayed to port 0
9	FW21	0	R/W	Sets the processing method when frames fro are multicast. 0: CAM hit: Frames are relayed to port 0 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 0

should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
13	POSTENU	0	R/W	Enables the settings of the POST field of C tables 0 to 15 (settings by the TSU_POST1 TSU_POST2 registers). 0: Disables the settings of the POST field. (entry table is referred only in port 0 recep 1: Enables the settings of the POST field. (entry table reference conditions follow th field settings.)
12	POSTENL	0	R/W	Enables the settings of the POST field of C tables 16 to 31 (settings by the TSU_POST TSU_POST4 registers). 0: Disables the settings of the POST field. (entry table is referred only in port 1 recep 1: Enables the settings of the POST field. (entry table reference conditions follow th field settings.)
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.

				CAMSEL00: Refers signals on the CAMSEN port 1 to 0 relay
3	CAMSEL13	0	R/W	These bits set the conditions for referring sig the CAMSEN1 pin. By setting multiple bits to multiple conditions can be selected.
2	CAMSEL12	0	R/W	
1	CAMSEL11	1	R/W	CAMSEL13: Refers signals on the CAMSEN port 0 reception
0	CAMSEL10	0	R/W	CAMSEL12: Refers signals on the CAMSEN port 0 to 1 relay
				CAMSEL11: Refers signals on the CAMSEN port 1 reception
				CAMSEL10: Refers signals on the CAMSEN port 1 to 0 relay

			These bits are always read as 0. The write va should always be 0.
1,0	QTAGM01, QTAGM00	All 0 R/W	These bits set Qtag adding and deleting func during port 0 to 1 relay operations. H′0: No Qtag adding and deleting functions H′1: No Qtag adding and deleting functions (H′0) H′2: Deletes Qtag from frames with Qtag H′3: Adds Qtag to frames with no Qtag Writing to this register is prohibited, after tran operations have been enabled once (after th in TSU_FWEN0 or the FWEN1 in TSU_FWE to 1).

			These bits are always read as 0. The write valu always be 0.
1, 0	QTAGM11, QTAGM10	All 0 R/W	These bits set Qtag adding and deleting functio port 1 to 0 relay operations. H′0: No Qtag adding and deleting functions H′1: No Qtag adding and deleting functions (sa H′0) H′2: Deletes Qtag from frames with Qtag H′3: Adds Qtag to frames with no Qtag Writing to this register is prohibited, after transfe operations have been enabled once (after the F TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is 1).

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write val always be 0.
27	TINT40	0	R/W	MAC-0 Carrier Not Detect Set to 1 when a carrier not detect has occured MAC-0
26	TINT30	0	R/W	MAC-0 Carrier Lost Set to 1 when a carrier is lost during data trans in the MAC-0
25	TINT20	0	R/W	MAC-0 Collision Detect Set to 1 when a collision of frames is detected MAC-0
24	TINT10	0	R/W	MAC-0 Transmission Time Out Set to 1 when frames were unable to be transr 16 transmission attempts including the retrans MAC-0
23	OVF0	0	R/W	Port 0 to 1 TSU FIFO Overflow Detect Set to 1 when a port 0 to 1 TSU FIFO overflow occured
22	RBSY0	0	R/W	MAC-0 Overflow Alert Signal Output Set to 1 when the threshold of TSU_BSYSL0 i and exceeded
21	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

				bytes are received in the MAC-0
17	RINT20	0	R/W	MAC-0 Frame Receive Error Set to 1 when a receive error is detected on th pin input from the PHY in the MAC-0
16	RINT10	0	R/W	MAC-0 CRC Error Frame Receive Set to 1 when a receive frame results in a CRC the MAC-0
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
11	TINT41	0	R/W	MAC-1 Carrier Not Detect Set to 1 when a carrier not detect has occured MAC-1
10	TINT31	0	R/W	MAC-1 Carrier Lost Set to 1 when a carrier is lost during data trans in the MAC-1
9	TINT21	0	R/W	MAC-1 Collision Detect Set to 1 when a collision of frames is detected MAC-1
8	TINT11	0	R/W	MAC-1 Transmission Time Out Set to 1 when frames were unable to be transr 16 transmission attempts including the retrans MAC-1
7	OVF1	0	R/W	Port 1 to 0 TSU FIFO Overflow Detect Set to 1 when a port 1 to 0 TSU FIFO overflow occured

				an 8-bit unit) are received in the MAC-1
3	RINT41	0	R/W	MAC-1 Exceeding Byte Frame Receive Set to 1 when frames exceeding the value se RFLR1 are received in the MAC-1
2	RINT31	0	R/W	MAC-1 Less 64-Byte Frame Receive Set to 1 when frames with a length of less tha bytes are received in the MAC-1
1	RINT21	0	R/W	MAC-1 Frame Receive Error Set to 1 when a receive error is detected on t pin input from the PHY in the MAC-1
0	RINT11	0	R/W	MAC-1 CRC Error Frame Receive Set to 1 when a receive frame results in a CR the MAC-1

18.3.35 Relay Status Interrupt Mask Register (TSU_FWINMK)

TSU_FWINMK is a 32-bit readable/writable register that sets the interrupt mask for stat
TSU_FWSR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
27	TINTM40	0	R/W	MAC-0 Carrier Not Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled

				1: Interrupts enabled
23	OVFM0	0	R/W	Port 0 to 1 TSU FIFO Overflow Detect Interrupt 0: Interrupts disabled 1: Interrupts enabled
22	RBSYM0	0	R/W	MAC-0 Overflow Alert Signal Output Interrupt M 0: Interrupts disabled 1: Interrupts enabled
21	—	0	R	Reserved This bit is always read as 0. The write value sho always be 0.
20	RINTM50	0	R/W	MAC-0 Residual Bit Frame Receive Interrupt Ma 0: Interrupts disabled 1: Interrupts enabled
19	RINTM40	0	R/W	MAC-0 Exceeding Byte Frame Receive Interrup 0: Interrupts disabled 1: Interrupts enabled
18	RINTM30	0	R/W	MAC-0 Less 64-Byte Frame Receive Interrupt M 0: Interrupts disabled 1: Interrupts enabled
17	RINTM20	0	R/W	MAC-0 Frame Receive Error Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled

				1: Interrupts enabled
10	TINTM31	0	R/W	MAC-1 Carrier Lost Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
9	TINTM21	0	R/W	MAC-1 Collision Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
8	TINTM11	0	R/W	MAC-1 Transmission Time Out Interrupt Mas 0: Interrupts disabled 1: Interrupts enabled
7	OVFM1	0	R/W	Port 1 to 0 TSU FIFO Overflow Detect Interru 0: Interrupts disabled 1: Interrupts enabled
6	RBSYM1	0	R/W	MAC-1 Overflow Alert Signal Output Interrupt 0: Interrupts disabled 1: Interrupts enabled
5	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
4	RINTM51	0	R/W	MAC-1 Residual Bit Frame Receive Interrupt 0: Interrupts disabled 1: Interrupts enabled

				1: Interrupts enabled
0	RINTM11	0	R/W	MAC-1 CRC Error Frame Receive Interrupt Mas
				0: Interrupts disabled
				1: Interrupts enabled

	QTAG016			(QTAG031 to QTAG016) as H'8100 (indica is the Qtag extension frame format). The va is H'8100.
15 to 13	QTAG015 to QTAG013	H'0	R/W	Priority Setting (PRT) These bits set the processing priority of fra Qtag. For details on the settings, refer to th specifications on Qtag control specified in IEEE802.1Q.
12	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
11 to 0	QTAG011 to QTAG000	H'000	R/W	V-LAN ID Setting (VID) These bits set the flames with Qtag to be u systems supporting V-LAN. For details on s refer to the specifications on Qtag control s IEEE802.1Q.

	QTAG116			(QTAG131 to QTAG116) as H′8100 (indicat is the Qtag extension frame format). The val is H′8100.
15 to 13	QTAG115 to QTAG113	H′0	R/W	Priority Setting (PRT) These bits set the processing priority of fram Qtag. For details on the settings, refer to the specifications on Qtag control specified in IEEE802.1Q.
12	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
11 to 0	QTAG111 to QTAG100	H′000	R/W	V-LAN ID Setting (VID) These bits set the flames with Qtag to be us systems supporting V-LAN. For details on se refer to the specifications on Qtag control sp IEEE802.1Q.

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
0	ADSBSY	0	R	CAM Entry Table Setting Busy When TSU_ADRH0 to TSU_ADRH31 and TSU to TSU_ADRL31 are set by register writing, the bit is set to 1. When the process of reflecting th contents of the CAM entry table register in the controller is completed inside the TSU, the ADS is automatically restored to 0. Accessing to TSU to TSU_ADRH31 and TSU_ADRL0 to TSU_AD prohibited, while this bit is set to 1. Writing to th is also prohibited.

30	TEN1	0	R/W	CAM Entry Table 1 (TSU_ADRH1 and TSU_AD Setting 0: Disabled 1: Enabled
29	TEN2	0	R/W	CAM Entry Table 2 (TSU_ADRH2 and TSU_AD Setting 0: Disabled 1: Enabled
28	TEN3	0	R/W	CAM Entry Table 3 (TSU_ADRH3 and TSU_AD Setting 0: Disabled 1: Enabled
27	TEN4	0	R/W	CAM Entry Table 4 (TSU_ADRH4 and TSU_AD Setting 0: Disabled 1: Enabled
26	TEN5	0	R/W	CAM Entry Table 5 (TSU_ADRH5 and TSU_AD Setting 0: Disabled 1: Enabled
25	TEN6	0	R/W	CAM Entry Table 6 (TSU_ADRH6 and TSU_AD Setting 0: Disabled 1: Enabled

22	TEN9	0	R/W	CAM Entry Table 9 (TSU_ADRH9 and TSU_A Setting 0: Disabled 1: Enabled
21	TEN10	0	R/W	CAM Entry Table 10 (TSU_ADRH10 and TSU Setting 0: Disabled 1: Enabled
20	TEN11	0	R/W	CAM Entry Table 11 (TSU_ADRH11 and TSU Setting 0: Disabled 1: Enabled
19	TEN12	0	R/W	CAM Entry Table 12 (TSU_ADRH12 and TSU Setting 0: Disabled 1: Enabled
18	TEN13	0	R/W	CAM Entry Table 13 (TSU_ADRH13 and TSU Setting 0: Disabled 1: Enabled
17	TEN14	0	R/W	CAM Entry Table 14 (TSU_ADRH14and TSU_ Setting 0: Disabled 1: Enabled

14	TEN17	0	R/W	CAM Entry Table 17 (TSU_ADRH17 and TSU_A Setting 0: Disabled 1: Enabled
13	TEN18	0	R/W	CAM Entry Table 18 (TSU_ADRH18 and TSU_A Setting 0: Disabled 1: Enabled
12	TEN19	0	R/W	CAM Entry Table 19 (TSU_ADRH19 and TSU_A Setting 0: Disabled 1: Enabled
11	TEN20	0	R/W	CAM Entry Table 20 (TSU_ADRH20 and TSU_A Setting 0: Disabled 1: Enabled
10	TEN21	0	R/W	CAM Entry Table 21 (TSU_ADRH21 and TSU_A Setting 0: Disabled 1: Enabled
9	TEN22	0	R/W	CAM Entry Table 22 (TSU_ADRH22 and TSU_A Setting 0: Disabled 1: Enabled

6	TEN25	0	R/W	CAM Entry Table 25 (TSU_ADRH20 and TSU_ Setting 0: Disabled 1: Enabled
5	TEN26	0	R/W	CAM Entry Table 26 (TSU_ADRH20 and TSU_ Setting 0: Disabled 1: Enabled
4	TEN27	0	R/W	CAM Entry Table 27 (TSU_ADRH27 and TSU_ Setting 0: Disabled 1: Enabled
3	TEN28	0	R/W	CAM Entry Table 28 (TSU_ADRH28 and TSU_ Setting 0: Disabled 1: Enabled
2	TEN29	0	R/W	CAM Entry Table 29 (TSU_ADRH29 and TSU_ Setting 0: Disabled 1: Enabled
1	TEN30	0	R/W	CAM Entry Table 30 (TSU_ADRH30 and TSU_ Setting 0: Disabled 1: Enabled

using the TSU_POST1 to TSU_POST4 registers. TSU_POST1 specifies the conditions fo
referring to TSU_ADRH0 to TSU_ADRH7 and TSU_ADRL0 to TSU_ADRL7. The sett
this register are valid when the POSENU bit in TSU_FWSLC is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	POST03 to POST00	All 0	R/W	These bits set the conditions for referring to CAM entry table 0. By setting multiple bits t multiple conditions can be selected. POST03: The CAM entry table 0 is referred reception. POST02: The CAM entry table 0 is referred to 1 relay. POST01: The CAM entry table 0 is referred reception. POST00: The CAM entry table 0 is referred to 0 relay.
27 to 24	POST13 to POST10	All 0	R/W	These bits set the conditions for referring to CAM entry table 1. By setting multiple bits t multiple conditions can be selected. POST13: The CAM entry table 1 is referred reception. POST12: The CAM entry table 1 is referred to 1 relay. POST11: The CAM entry table 1 is referred reception. POST10: The CAM entry table 1 is referred to 0 relay.

				POST20: The CAM entry table 2 is referr 1 to 0 relay.
19 to 16	POST33 to POST30	All 0	R/W	These bits set the conditions for referring CAM entry table 3. By setting multiple bit multiple conditions can be selected. POST33: The CAM entry table 3 is referr 0 reception. POST32: The CAM entry table 3 is referr 0 to 1 relay. POST31: The CAM entry table 3 is referr 1 reception. POST30: The CAM entry table 3 is referr 1 to 0 relay.
15 to 12	POST43 to POST40	All 0	R/W	These bits set the conditions for referring CAM entry table 4. By setting multiple bit multiple conditions can be selected. POST43: The CAM entry table 4 is referr 0 reception. POST42: The CAM entry table 4 is referr 0 to 1 relay. POST41: The CAM entry table 4 is referr 1 reception. POST40: The CAM entry table 4 is referr 1 to 0 relay.

				POST50: The CAM entry table 5 is referred to 0 relay.
7 to 4	POST63 to POST60	All 0	R/W	These bits set the conditions for referring to entry table 6. By setting multiple bits to 1, mu conditions can be selected. POST63: The CAM entry table 6 is referred reception. POST62: The CAM entry table 6 is referred to 1 relay. POST61: The CAM entry table 6 is referred reception. POST60: The CAM entry table 6 is referred to 0 relay.
3 to 0	POST73 to POST70	All 0	R/W	These bits set the conditions for referring to entry table 7. By setting multiple bits to 1, mu conditions can be selected. POST73: The CAM entry table 7 is referred reception. POST72: The CAM entry table 7 is referred to 1 relay. POST71: The CAM entry table 7 is referred reception. POST70: The CAM entry table 7 is referred to 0 relay.

				conditions can be selected. POST83: The CAM entry table 8 is referred reception. POST82: The CAM entry table 8 is referred to 1 relay. POST81: The CAM entry table 8 is referred reception. POST80: The CAM entry table 8 is referred to 0 relay.
27 to 24	POST93 to POST90	All 0	R/W	These bits set the conditions for referring to entry table 9. By setting multiple bits to 1, m conditions can be selected. POST93: The CAM entry table 9 is referred reception. POST92: The CAM entry table 9 is referred to 1 relay. POST91: The CAM entry table 9 is referred reception. POST90: The CAM entry table 9 is referred to 0 relay.

				POST100: The CAM entry table 10 is referre 1 to 0 relay.
19 to 16	POST113 to POST110	All 0	R/W	These bits set the conditions for referring to t entry table 11. By setting multiple bits to 1, m conditions can be selected. POST113: The CAM entry table 11 is referre 0 reception. POST112: The CAM entry table 11 is referre 0 to 1 relay. POST111: The CAM entry table 11 is referre 1 reception. POST110: The CAM entry table 11 is referre 1 to 0 relay.
15 to 12	POST123 to POST120	All 0	R/W	These bits set the conditions for referring to t entry table 12. By setting multiple bits to 1, m conditions can be selected. POST123: The CAM entry table 12 is referre 0 reception. POST122: The CAM entry table 12 is referre 0 to 1 relay. POST121: The CAM entry table 12 is referre 1 reception. POST120: The CAM entry table 12 is referre 1 to 0 relay.

				POST130: The CAM entry table 13 is referred to 0 relay.
7 to 4	POST143 to POST140	All 0	R/W	These bits set the conditions for referring to th entry table 14. By setting multiple bits to 1, mu conditions can be selected. POST143: The CAM entry table 14 is referred reception. POST142: The CAM entry table 14 is referred to 1 relay. POST141: The CAM entry table 14 is referred reception. POST140: The CAM entry table 14 is referred to 0 relay.
3 to 0	POST153 to POST150	All 0	R/W	These bits set the conditions for referring to th entry table 15. By setting multiple bits to 1, mu conditions can be selected. POST153: The CAM entry table 15 is referred reception. POST152: The CAM entry table 15 is referred to 1 relay. POST151: The CAM entry table 15 is referred reception. POST150: The CAM entry table 15 is referred to 0 relay.

				conditions can be selected. POST163: The CAM entry table 16 is referre 0 reception. POST162: The CAM entry table 16 is referre 0 to 1 relay. POST161: The CAM entry table 16 is referre 1 reception. POST160: The CAM entry table 16 is referre 1 to 0 relay.
27 to 24	POST173 to POST170	All 0	R/W	These bits set the conditions for referring to entry table 17. By setting multiple bits to 1, m conditions can be selected. POST173: The CAM entry table 17 is referre 0 reception. POST172: The CAM entry table 17 is referre 0 to 1 relay. POST171: The CAM entry table 17 is referre 1 reception. POST170: The CAM entry table 17 is referre 1 to 0 relay.

				POST180: The CAM entry table 18 is refer 1 to 0 relay.
19 to 16	POST193 to POST190	All 0	R/W	These bits set the conditions for referring to entry table 19. By setting multiple bits to 1, conditions can be selected. POST193: The CAM entry table 19 is refer 0 reception. POST192: The CAM entry table 19 is refer 0 to 1 relay. POST191: The CAM entry table 19 is refer 1 reception. POST190: The CAM entry table 19 is refer 1 to 0 relay.
15 to 12	POST203 to POST200	All 0	R/W	These bits set the conditions for referring to entry table 20. By setting multiple bits to 1, conditions can be selected. POST203: The CAM entry table 20 is refer 0 reception. POST202: The CAM entry table 20 is refer 0 to 1 relay. POST201: The CAM entry table 20 is refer 1 reception. POST200: The CAM entry table 20 is refer 1 to 0 relay.

				POST210: The CAM entry table 21 is referre 1 to 0 relay.
7 to 4	POST223 to POST220	All 0	R/W	These bits set the conditions for referring to entry table 22. By setting multiple bits to 1, m conditions can be selected. POST223: The CAM entry table 22 is referre 0 reception. POST222: The CAM entry table 22 is referre 0 to 1 relay. POST221: The CAM entry table 22 is referre 1 reception. POST220: The CAM entry table 22 is referre 1 to 0 relay.
3 to 0	POST233 to POST230	All 0	R/W	These bits set the conditions for referring to entry table 23. By setting multiple bits to 1, m conditions can be selected. POST233: The CAM entry table 23 is referre 0 reception. POST232: The CAM entry table 23 is referre 0 to 1 relay. POST231: The CAM entry table 23 is referre 1 reception. POST230: The CAM entry table 23 is referr 1 to 0 relay.

				conditions can be selected. POST243: The CAM entry table 24 is referred reception. POST242: The CAM entry table 24 is referred to 1 relay. POST241: The CAM entry table 24 is referred reception. POST240: The CAM entry table 24 is referred to 0 relay.
27 to 24	POST253 to POST250	All 0	R/W	These bits set the conditions for referring to th entry table 25. By setting multiple bits to 1, mu conditions can be selected. POST253: The CAM entry table 25 is referred reception. POST252: The CAM entry table 25 is referred to 1 relay. POST251: The CAM entry table 25 is referred reception. POST250: The CAM entry table 25 is referred to 0 relay.

				POST260: The CAM entry table 26 is referre 1 to 0 relay.
19 to 16	POST273 to POST270	All 0	R/W	These bits set the conditions for referring to entry table 27. By setting multiple bits to 1, m conditions can be selected. POST273: The CAM entry table 27 is referre 0 reception. POST272: The CAM entry table 27 is referre 0 to 1 relay. POST271: The CAM entry table 27 is referre 1 reception. POST270: The CAM entry table 27 is referre 1 to 0 relay.
15 to 12	POST283 to POST280	All 0	R/W	These bits set the conditions for referring to entry table 28. By setting multiple bits to 1, m conditions can be selected. POST283: The CAM entry table 28 is referre 0 reception. POST282: The CAM entry table 28 is referre 0 to 1 relay. POST281: The CAM entry table 28 is referre 1 reception. POST280: The CAM entry table 28 is referre 1 to 0 relay.

				POST290: The CAM entry table 29 is refer 1 to 0 relay.
7 to 4	POST303 to POST300	All 0	R/W	These bits set the conditions for referring to entry table 30. By setting multiple bits to 1, conditions can be selected. POST303: The CAM entry table 30 is refer 0 reception. POST302: The CAM entry table 30 is refer 0 to 1 relay. POST301: The CAM entry table 30 is refer 1 reception. POST300: The CAM entry table 30 is refer 1 to 0 relay.
3 to 0	POST313 to POST310	All 0	R/W	These bits set the conditions for referring to entry table 31. By setting multiple bits to 1, conditions can be selected. POST313: The CAM entry table 31 is refer 0 reception. POST312: The CAM entry table 31 is refer 0 to 1 relay. POST311: The CAM entry table 31 is refer 1 reception. POST310: The CAM entry table 31 is refer 1 to 0 relay.

	(n: 0 to 31)	These bits set the upper 32 bits of the MAC When the MAC address is 01-23-45-67-89-A (displayed in hexadecimal), H'01234567 is s register.

Notes: Set the CAM entry table as follows:

1. Check that the ADSBSY bit in TSU_ADSBSY is cleared to 0.
2. Set the upper 32 bits of the MAC address by TSU_ADRH0 to TSU_ADRH31.
3. Set the lower 16 bits of the MAC address by TSU_ADRL0 to TSU_ADRL31.

				These bits are always read as 0. The write should always be 0.
15 to 0	ADRLn15 to ADRLn0 (n: 0 to 31)	All 0	R/W	MAC Address Bit These bits set the lower 16 bits of the MAC When the MAC address is 01-23-45-67-89- (displayed in hexadecimal), H'000089AB is register.

Notes: Set the CAM entry table as follows:

1. Check that the ADSBSY bit in TSU_ADSBSY is cleared to 0.
2. Set the upper 32 bits of the MAC address by TSU_ADRH0 to TSU_ADRH31.
3. Set the lower 16 bits of the MAC address by TSU_ADRL0 to TSU_ADRL31.

18.3.46 Transmit Frame Counter Register (Port 0) (Normal Transmission Only) (TXNLCR0)

TXNLCR0 is a 32-bit counter indicating the number of frames successfully transmitted 0. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NTC031 to NTC000	All 0	R	Port 0 Transmit Frame Counter Bit These bits indicate the number of frames successfully transmitted.

	TC000			These bits indicate the number of frames succes transmitted and frames transmitted with error.

18.3.48 Receive Frame Counter Register (Port 0) (Normal Reception Only) (RXNI

RXNLCR0 is a 32-bit counter indicating the number of frames successfully received in M When the value in this register reaches H'FFFFFFFF, the count is halted. The counter val cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NRC031 to NRC000	All 0	R	Port 0 Receive Frame Counter Bit These bits indicate the number of frames succes received.

				These bits indicate the number of frames succe received and frames received with error.

18.3.50 Relay Frame Counter Register (Port 1 to 0) (Normal Relay Only) (FWNL

FWNLCR0 is a 32-bit counter indicating the number of frames successfully relayed in p relay operations. When the value in this register reaches H'FFFFFFFF, the count is halte counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NFC031 to NFC000	All 0	R	Port 1 to 0 Relay Frame Counter Bit These bits indicate the number of frames succe relayed.

	FC000			These bits indicate the number of frames succes relayed and frames relayed with error.

18.3.52 Transmit Frame Counter Register (Port 1) (Normal Transmission Only) (TXNLCR1)

TXNLCR1 is a 32-bit counter indicating the number of frames successfully transmitted i 1. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter v cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NTC131 to NTC100	All 0	R	Port 1 Transmit Frame Counter Bit These bits indicate the number of frames succes transmitted.

	TC100			These bits indicate the number of frames succe transmitted and frames transmitted with error.

18.3.54 Receive Frame Counter Register (Port 1) (Normal Reception Only) (RXN

RXNLCR1 is a 32-bit counter indicating the number of frames successfully received in When the value in this register reaches H'FFFFFFFF, the count is halted. The counter va cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NRC131 to NRC100	All 0	R	Port 1 Receive Frame Counter Bit These bits indicate the number of frames succe received.

				These bits indicate the number of frames success received and frames received with error.

18.3.56 Relay Frame Counter Register (Port 0 to 1) (Normal Relay Only) (FWNLC

FWNLCR1 is a 32-bit counter indicating the number of frames successfully relayed in po relay operations. When the value in this register reaches H'FFFFFFFF, the count is halted counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NFC131 to NFC100	All 0	R	Port 0 to 1 Relay Frame Counter Bit These bits indicate the number of frames success relayed.

	FC100			These bits indicate the number of frames succe relayed and frames relayed with error.

relay sides by means of registers TSU_TEN and TSU_POST1 to TSU_POST4. It also ha
kbyte TSU FIFO for temporarily retaining the frames relayed. This TSU FIFO can vary c
allotment with port 0 to 1 transfer and port 1 to 0 transfer using the TSU FIFO size select
(TSU_FCM).

TSU FIFO Overflow Prevention Function: By supporting relay operations, the MAC c
needs to transmit relay frames other than transmit frames requested by the E-DMAC norr
Arbitration is carried out between these two frames. The procedure of arbitration is specif
registers TSU_PRISL0 and TSU_PRISL1. It has a function which relays frames of the TS
with priority when the using rate of the TSU FIFO exceeds the value set by registers
TSU_PRISL0 and TSU_PRISL1, thus preventing frame losses by TSU FIFO overflow.

QoS (IEEE802.1Q) Frame Transmit/Receive, Relay Function: QoS frames can be tra
and received. At the using relay function, if the Ethernet device connected to one of the M
controllers cannot transmit/receive QoS frames, this LSI can convert to the normal IEEE8
frames and relay it.

Figure 18.2 shows the data path and outline of various settings.

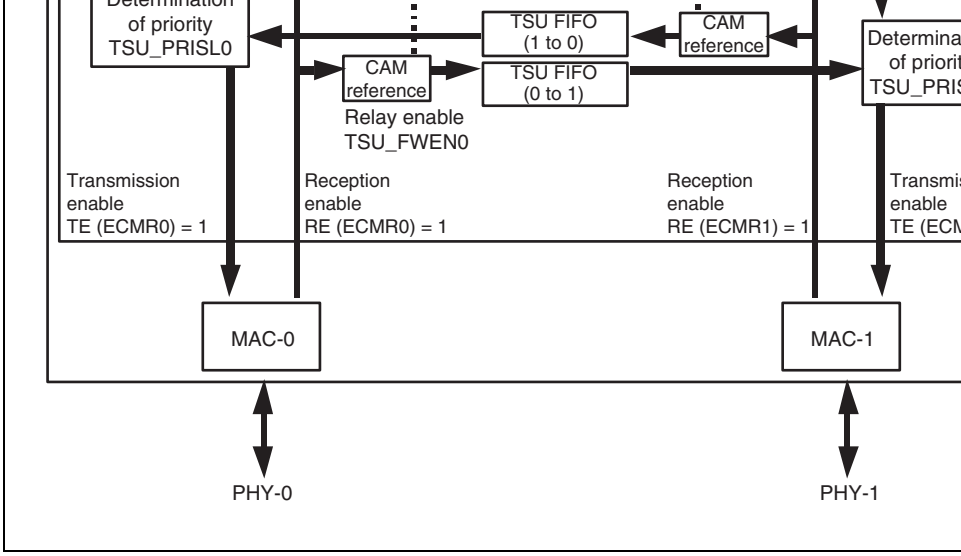
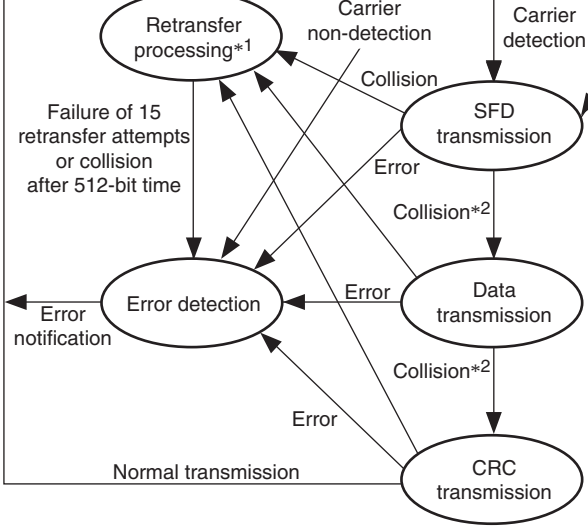


Figure 18.2 EtherC Data Path and Various Settings

18.4.1 Transmission

The EtherC transmitter assembles the transmit data on the frame and outputs to MII whe
a transmit request from the E-DMAC. The data transmitted via the MII is transmitted to
by PHY-LSI. Figure 18.3 shows the status change of the Ether-C transmitter. This opera
same between ports 0 and 1. The priority of the process when transmit frame from E-DM
relay frame transmission collide can be set by the Transmit/Relay Priority Control Mode
(TSU_PRISL0/1).



Legend
FDPX: Full-duplex
HDPX: Half-duplex

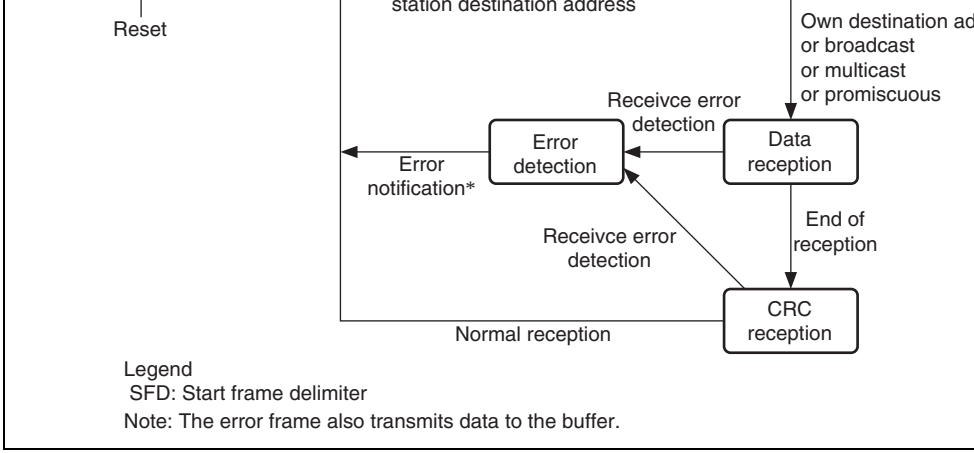
Notes:
1. Transmission retry processing includes both jam transmission that depends on collision detection and the adjustment of transmission intervals based on the back-off algorithm.
2. Transmission is retried only when data of 512 bits or less (including the preamble and SFD)is transmitted. When a collision is detected during the transmission of data greater than 512 bits, only jam is transmitted and transmission based on the back-off algorithm is not retried.

Figure 18.3 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 pre after a transmission delay equivalent to the frame interval time. If full-duplex transfer

18.4.2 Reception

The EtherC receiver separates the frame from the MII into preamble, SFD, data and CR
fields from DA (destination address) to the CRC data are transferred to the receive E-DM
Figure 18.4 shows the state transitions of the EtherC receiver. These operations are the s
ports 0 and 1. In frame processing during reception, CAM evaluation can be referenced.
using the CAM function, refer to section 18.4.4, CAM Function.)



Legend
SFD: Start frame delimiter
Note: The error frame also transmits data to the buffer.

Figure 18.4 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 r 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 b or multicast transmission or promiscuous mode is specified, the receiver starts data re
4. Following data reception from the MII, the receiver carries out a CRC check. The resu indicated as a status bit in the descriptor after the frame data has been written to mem 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 Ethe register, the receiver prepares to receive the next frame.

frames from the E-DMAC may occur. The priority of the process when collision occurs by TSU_PRISL0/1. For multicast frames and frames their destinations are other than thi CAM evaluation in frame relay processing can be referenced (for details on the CAM fu refer to section 18.4.4, CAM Function). Table 18.2 shows the settings of the relay frame processing (without CAM).

Table 18.2 Transfer Frame Processing (Without CAM)

Name	TSU-FWSL	Frame Processing
Frame for this LSI	FW40/1 = 0	Discarded
	FW40/1 = 1	Relayed
Broadcast frame	FW30/1 = 0	Discarded
	FW30/1 = 1	Relayed
Multicast frame	FW20/1 = 0	Discarded
	FW20/1 = 1	Relayed
Frames to destinations other than this LSI	FW10/1 = 0	Discarded
	FW10/1 = 1	Relayed

18.4.4 CAM Function

Frames input to the MAC are grouped into the following four types; unicast for this LSI broadcast, multicast, and unicast to other destinations. Of this, the MAC addresses of un this LSI and broadcast are fixed, and determination is carried out only by register setting Consequently, only multicast and unicast to other destinations determine whether to rece and whether to relay or not by using the CAM (unicast frames whose destination MAC a match this LSI are called unicast frames to this LSI, and those that do not are called unic to other destinations).

as when CAM is not used shown in table 18.2. The difference between the on-chip CAM table and externally connected CAM logic lies in how the POST table is set. In the intern entry table, there are 32 POST tables (same as the number of entries) and the POST table set for each entry. The internal CAM entry table has 32 entries and 32 POST tables, and t table can be specified in each entry. The external connection CAM logic configuration is pins because POST tables (total of 2) are allocated to the CAMSEN0 and CAMSEN1 pin

When On-Chip CAM Entry Table is Used: The on-chip CAM has entry tables which c register the MAC address of 32 entries, the details of which can be set by TSU_ADRH0 t TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31. The setting to enable/disable refere the on-chip CAM entry table is carried out by the CAM entry table enable setting register sets whether to perform CAM evaluation or not, and the CAM entry table POST setting r for setting whether to use the CAM determination results for determining receive or relay on-chip CAM entry table referencing during receive is enabled, the destination address in frame and MAC address registered in the CAM entry table are compared, and it is determ whether to transfer the frames input to the MAC to E-DMAC (have E-DMAC receive the or discard the frames. When relaying and CAM entry table referencing during relay are b enabled, whether to transfer or discard multicast frames and frames for destinations other LSI can be determined by comparing the destination address in the frame and MAC addr registered in the CAM entry table. Table 18.3 shows the processing method of frames (re discard) in MAC0 to E-DMAC0 and MAC1 to E-DMAC1 reception, while table 18.4 sho processing for frames in MAC0 to MAC1 and MAC1 to MAC0 relay (relay or discard).

CAM mishit (when addresses do not match)	Frames to this LSI	Received		Received	
	Broadcast frame	Received		Received	
	Multicast frame	Received	Discarded	Received	Dis
	Frames to destinations other than this LSI	Discarded		Received	

[Legend]

MCT (Bit 13 in ECMR): Multicast receive mode (0: Receive when CAM mishit/
1: Receive when CAM hit)

Table 18.4 Relay Frame Process (With CAM)

Frame	Relay Function Setting Register Bit	CAM Hit	CAM
Multicast frame	FW40/1 = 0	Relayed	Discar
	FW40/1 = 1	Discarded	Relay
Frames to destinations other than this LSI	FW40/1 = 0	Relayed	Discar
	FW40/1 = 1	Discarded	Relay

Note: CAM can be referenced only for multicast frames and frames to destinations othe
LSI. The processing of frames to this LSI and broadcast frames conforms to the v
the relay function setting register regardless of CAM reference.

When External CAM Logic is Used: In addition to the on-chip CAM entry table, use o
CAMSEN0 and CAMSEN1 pins allows referencing of evaluation results of the external
logic connected externally to this LSI for frame processing evaluation. This function ext
connects the CAM logic for comparing the destination address in receive frames, and re
results of comparing destination addresses corresponding to the signals (RXD3 to RXD0
from the MII to determine whether to receive or discard the corresponding frame. Figure

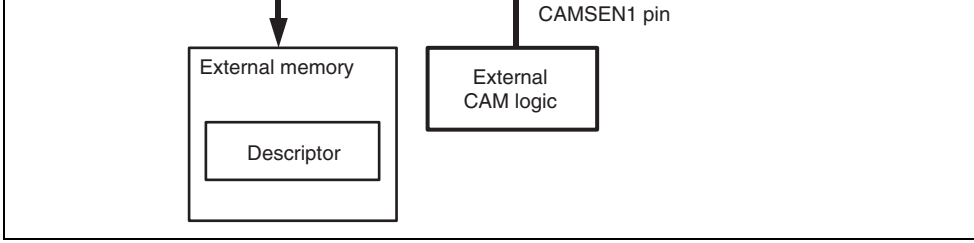


Figure 18.5 Example of External CAM Connection

The setting on whether to enable or disable the referencing of external CAM logic evalua
results by the CAMSEN0 and CAMSEN1 pins is carried out by the transfer function setti
register (common) (TSU_FWSLC). When referencing of the CAMSEN0 and CAMSEN1
enabled during receive, it is determined whether to send or discard the frames input from
0/1 to E-DMAC0/1 (have E-DMAC receive the frames) according to the value of the CA
or CAMSEN1 pin. When relaying and CAMSEN0/1 pin referencing are enabled at the sa
the transfer or discard of multicast frames and frames to destinations other than this LSI c
determined by the value of the CAMSEN0 and CAMSEN1 pins.

Table 18.5 shows the processing method (receive or discard) for frames in MAC0 to E-D
MAC1 to E-DMAC1 reception, while Table 18.6 shows the processing method (receive c
discard) for frames in MAC0 to MAC1 or MAC1 to MAC relay. The external CAM logic
memorized with MAC addresses different from the CAM entry table in this LSI. When th
address received from the PHY matches the destination address memorized in the externa
logic, the CAMSEN0 or CAMSEN1 pin is asserted*. EtherC receives or discards the fram
CAMSEN0/1 was asserted according to the settings in table 18.5.

Figure 18.6 shows the valid range of CAMSEN0/1 assertion for the corresponding receiv

CAMSEN1 Pin	Frame	MCT = 0	MCT = 1	MCT = 0	MCT
Assertion (when addresses match)	Frame to this LSI	Discarded		Discarded	
	Broadcast frame	Discarded		Discarded	
	Multicast frame	Discarded	Received	Discarded	Rec
	Frames to destinations other than this LSI	Received		Discarded	
Negation (when addresses do not match)	Frames to this LSI	Received		Received	
	Broadcast frame	Received		Received	
	Multicast frame	Received	Discarded	Received	Disc
	Frames to destinations other than this LSI	Discarded		Received	

[Legend]

MCT (Bit 13 in ECMR): Multicast receive mode (0: Received when CAM mishit/
1: Received when CAM hit)

the relay function setting register regardless of CAM reference.

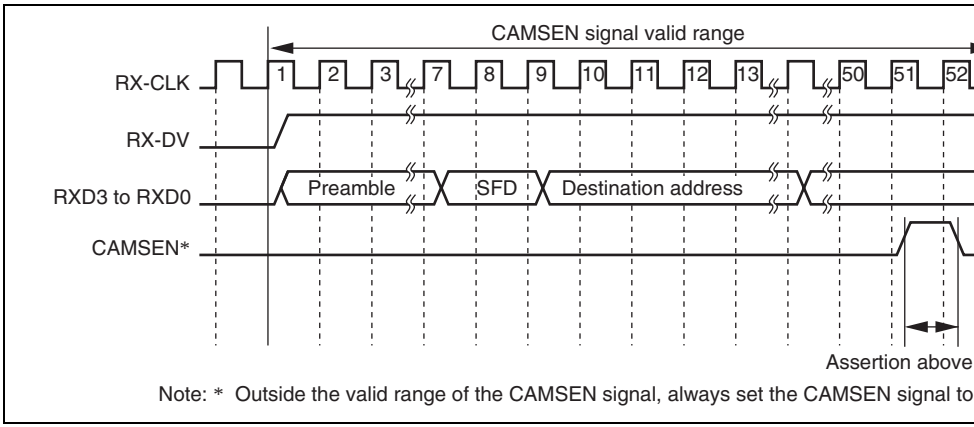


Figure 18.6 External CAM Signal Timing

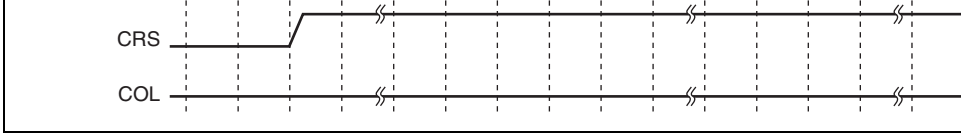


Figure 18.7 (1) MII Frame Transmit Timing (Normal Transmission)

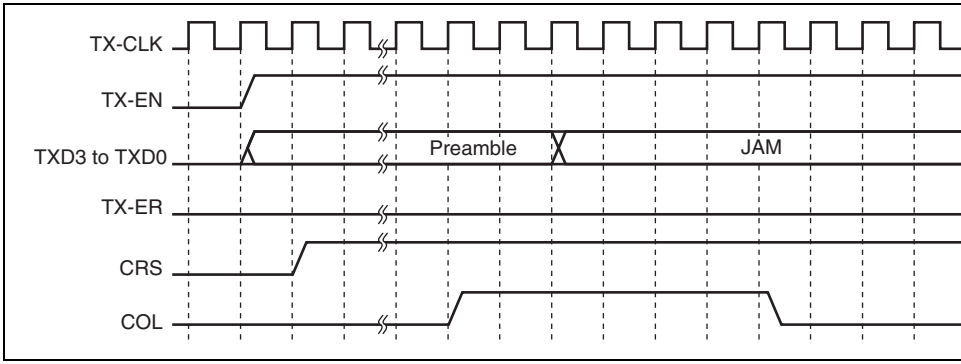


Figure 18.7 (2) MII Frame Transmit Timing (Collision)

Figure 18.7 (3) MII Frame Transmit Timing (Transmit Error)

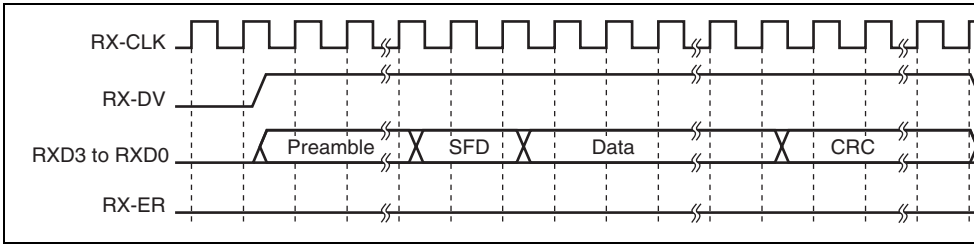


Figure 18.7 (4) MII Frame Receive Timing (Normal Reception)

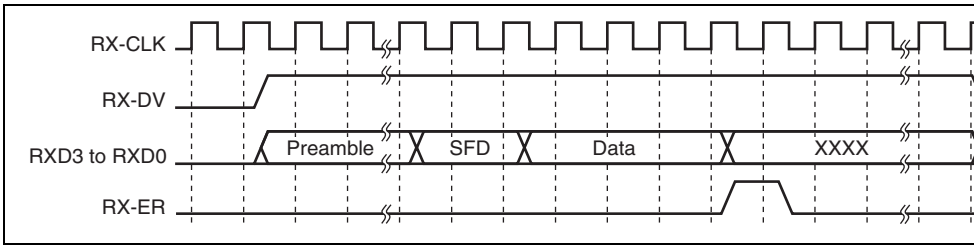


Figure 18.7 (5) MII Frame Receive Timing (Reception Error (1))

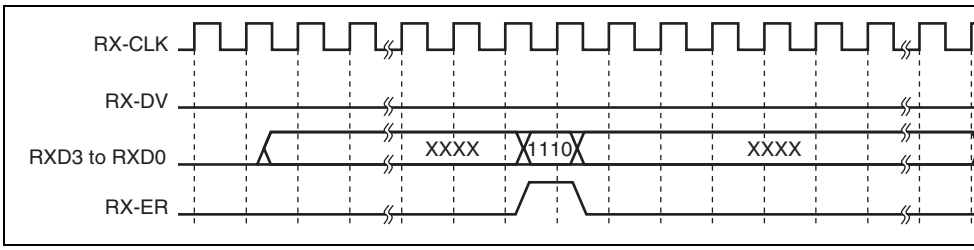


Figure 18.7 (6) MII Fame Receive Timing (Reception Error (2))

Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	
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	

[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 18.8 MII Management Frame Format

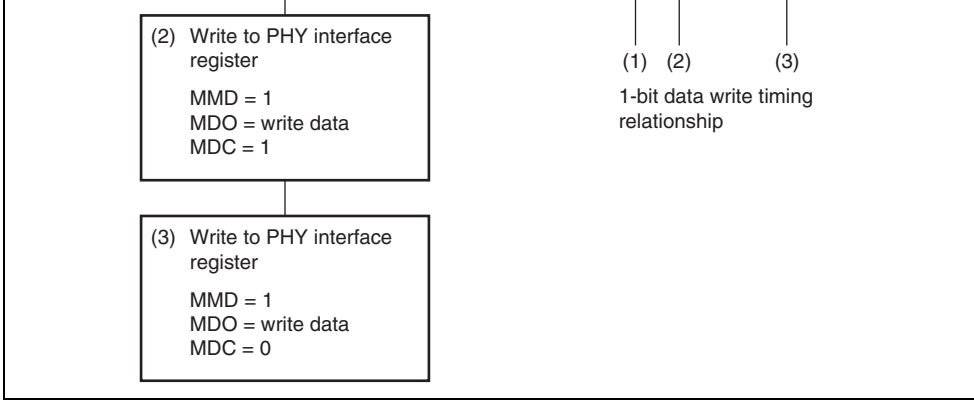
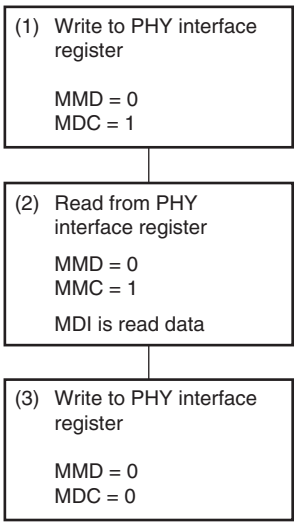


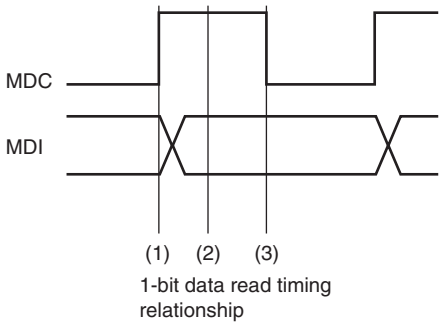
Figure 18.9 (1) 1-Bit Data Write Flowchart

(3) Write to PHY interface register

MMD = 0
MDC = 0

Figure 18.9 (2) Bus Release Flowchart (TA in Read in Figure 18.8)





1-bit data read timing relationship

Figure 18.9 (3) 1-Bit Data Read Flowchart

Figure 18.9 (4) Independent Bus Release Flowchart (IDLE in Write in Figure

18.4.7 Magic Packet Detection

The EtherC has a Magic Packet detection function. This function provides a Wake-On-L
(WOL) facility that activates various peripheral devices connected to a LAN from the ho
or other source. This makes it possible to construct a system in which a peripheral device
a Magic Packet sent from the host device or other source, and activates itself. When the M
Packet is detected, data is stored in the FIFO of the E-DMAC by the broadcast packet tha
received data previously and the EtherC is notified of the receiving status. To return to no
operation from the interrupt processing, initialize the EtherC and E-DMAC by using ARS
the software reset register (ARSTR).

With a Magic Packet, reception is performed regardless of the destination address. As a r
function is valid, and the WOL pin enabled, only in the case of a match with the destinati
address specified by the format in the Magic Packe
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 registe
2. Set the Magic Packet detection enable bit (MPDE) in the EtherC mode register (ECM
3. Set the Magic Packet detection interrupt enable bit (MPDIP) in the EtherC interrupt e
register (ECSIPR) to the enable setting.
4. If necessary, set the CPU operating mode to sleep mode or set peripheral modules to
standby mode.
5. When a Magic Packet is detected, an interrupt is sent to the CPU. The WOL pin notif
peripheral LSIs that the Magic Packet has been detected.

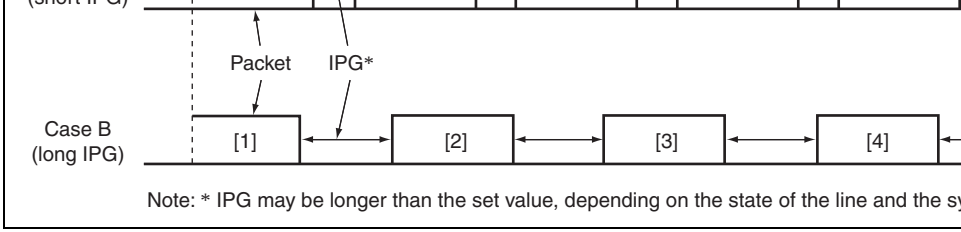


Figure 18.10 Changing IPG and Transmission Efficiency

18.4.9 Direction for IEEE802.1Q Qtag

The EtherC supports IEEE802.1Q frame processing. It can add or delete Qtags to or fro processed in relay. This function can also transmit and receive QoS frames. During relay Ethernet device connected to one MAC controller cannot transmit or receive QoS frame converted to the normal IEEE802.3 frames and relayed in this LSI. Whether to add or de is determined by the added Qtag value set register (TSU_ADQT0/1). Figure 18.11 show outline of the Qtag add function while figure 18.12 shows the comparison between the f Ethernet frames and IEEE802.1Q frames (with Qtag). For details on setting Qtag, refer t specifications on Qtag control specified in IEEE802.1Q.

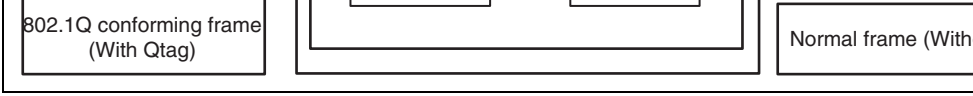


Figure 18.11 Diagram of Qtag Additional Functions

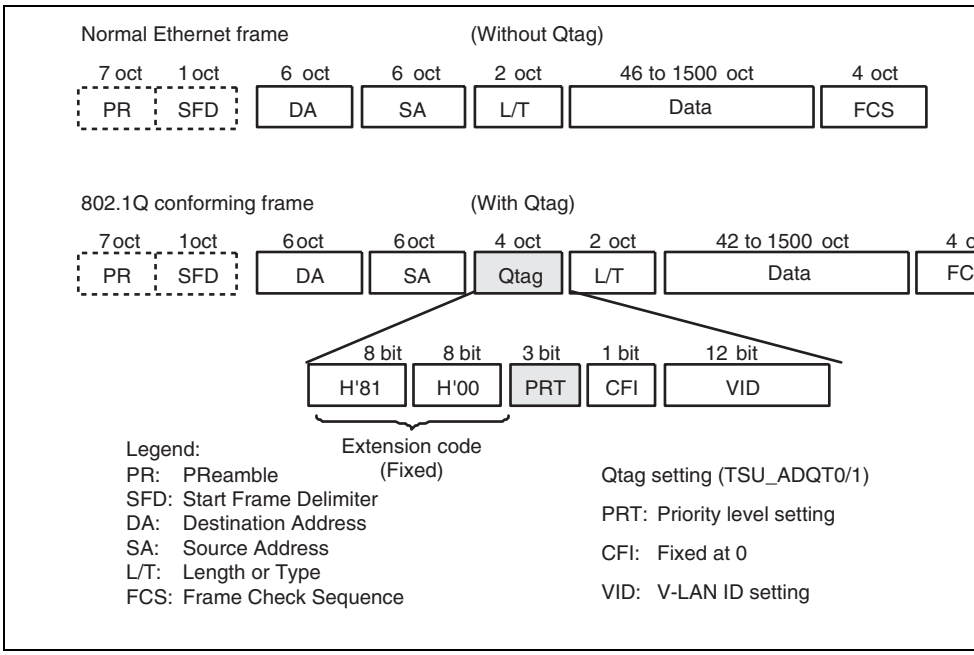


Figure 18.12 Comparison of Normal Ethernet Frame and IEEE802.1Q Frame (wi

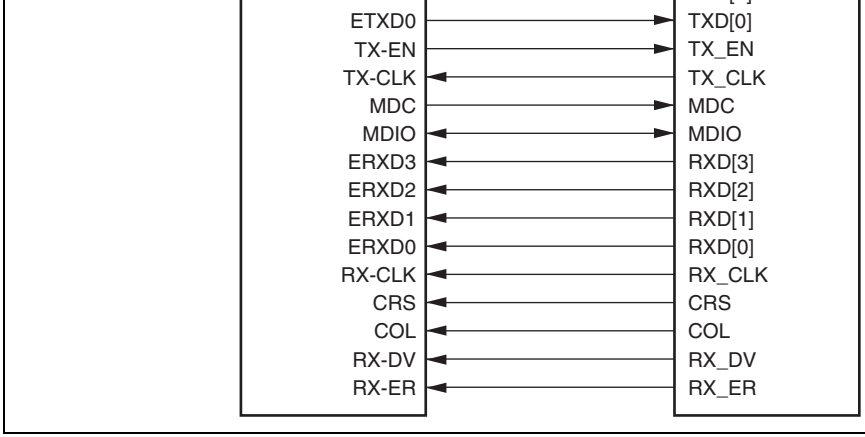


Figure 18.13 Example of Connection to DP83847

This function reduces the load on the CPU and enables efficient data transfer control to
achieved. The E-DMAC0 and E-DMAC1 control the data transmission/reception from t
and MAC-1 of EtherC respectively. (Hereafter the channel controlled by the E-DMAC0
0. The channel controlled by the E-DMAC1 is channel 1.)

Figure 19.1 shows the configuration of the E-DMAC, and the descriptors and transmit/r
buffers in memory.

19.1 Features

The E-DMAC has the following features:

- Contains two-channel independent transmit/receive DMAC
- 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 (16
  units)
- Supports single-frame/single-descriptor operation and single-frame/multi-frame (mu
  operation

Note: The E-DMAC cannot access peripheral modules.

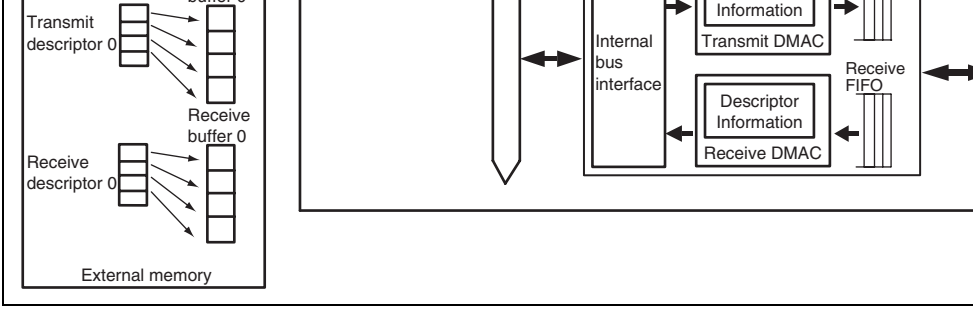


Figure 19.1 Configuration of E-DMAC, and Descriptors and Buffers

19.2 Register Descriptions

The E-DMAC has the following registers. The number at the end of the register abbreviat represents the number of corresponding E-DMAC (E-DMAC0 or E-DMAC1). In this sec some numbers are not mentioned. For addresses and access sizes of these registers, see se List of Registers.

Channel 0:

- E-DMAC mode register (EDMR0)
- E-DMAC transmit request register (EDTRR0)
- E-DMAC receive request register (EDRRR0)
- Transmit descriptor list address register (TDLAR0)
- Receive descriptor list address register (RDLAR0)
- EtherC/E-DMAC status register (EESR0)
- EtherC/E-DMAC status interrupt permission register (EESIPR0)
- Transmit/receive status copy enable register (TRSCER0)

- Overflow alert FIFO threshold register (FCFTR0)
- Transmit interrupt register (TRIMD0)

Channel 1:

- E-DMAC mode register (EDMR1)
- E-DMAC transmit request register (EDTRR1)
- E-DMAC receive request register (EDRRR1)
- Transmit descriptor list address register (TDLAR1)
- Receive descriptor list address register (RDLAR1)
- EtherC/E-DMAC status register (EESR1)
- EtherC/E-DMAC status interrupt permission register (EESIPR1)
- Transmit/receive status copy enable register (TRSCER1)
- Receive missed-frame counter register (RMFCR1)
- Transmit FIFO threshold register (TFTR1)
- FIFO depth register (FDR1)
- Receiving method control register (RMCR1)
- E-DMAC operation control register (EDOCR1)
- Receive buffer write address register (RBWAR1)
- Receive descriptor fetch address register (RDFAR1)
- Transmit buffer read address register (TBRAR1)
- Transmit descriptor fetch address register (TDFAR1)
- Overflow alert FIFO threshold register (FCFTR1)
- Transmit interrupt register (TRIMD1)

the EtherC and E-DMAC should be accessed after 64 cycles of the internal bus clock Bφ elapsed.

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
5	DL1	0	R/W	Descriptor Length
4	DL0	0	R/W	These bits specify the descriptor length. (See 19.3.1, Descriptors and Descriptor List.) 00: 16 bytes 01: 32 bytes 10: 64 bytes 11: Reserved (setting prohibited)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

register (Port 0 to 1) (TSU_FWEN0) and the
enable register (Port 1 to 0) (TSU_FWEN1)
EtherC, software reset should not be perfor
using this bit. While a software reset is issu
cycles of the internal bus clock Bϕ), accesse
all Ethernet-related registers are prohibited.

Software reset period (example):

When Bϕ = 100 MHz: 0.64 µS

When Bϕ = 66 MHz: 0.97 µS

When Bϕ = 50 MHz: 1.28 µS

When Bϕ = 33 MHz: 1.94 µS

This bit is always read as 0.

1: EtherC and E-DMAC are reset (when wri

19.2.2 E-DMAC Transmit Request Register (EDTRR)

EDTRR is a 32-bit readable/writable register that issues transmit directives to the E-DM
writing 1 to the TR bit in this register, the E-DMAC reads the transmit descriptor at the
specified by TDLAR. If the TACT bit of this descriptor is set to 1 (valid), transmit DMA
by the E-DMAC starts. When DMA transfer based on the first transmit descriptor is con
the E-DMAC reads the next transmit descriptor. If the TACT bit of that descriptor is set
(valid), the E-DMAC continues transmit DMA operation. If the TACT bit of a transmit
is cleared to 0 (invalid), the E-DMAC clears the TR bit and stops transmit DMAC opera

For details of writing to the TR bit, see section 19.4.1, Using of EDTRR and EDRRR.

1: Transmit DMA operation being performed
DMAC. After writing 1 to this bit, the E-DM
starts reading a transmit descriptor.

19.2.3 E-DMAC Receive Request Register (EDRRR)

EDRRR is a 32-bit readable/writable register that issues receive directives to the E-DMA
writing 1 to the RR bit in this register, the E-DMAC reads the receive descriptor at the ad
specified by RDLAR. If the RACT bit of this descriptor is set to 1 (valid), and the receive
holds a receive frame, the E-DMAC starts receive DMA transfer. When DMA transfer ba
the first receive descriptor is completed, the E-DMAC reads the next receive descriptor. I
RACT bit of that descriptor is set to 1 (valid), the E-DMAC continues receive DMA oper
However, if the receive FIFO holds no receive data, the E-DMAC places receive DMA o
in the standby state. If the RACT bit of the receive descriptor is cleared to 0 (invalid), the
DMAC clears the RR bit and stops receive DMAC operation.

For details of writing to the RR bit, see section 19.4.1, Using of EDTRR and EDRRR.

successfully to the receive descriptor. Following pointers to read a receive de
become abnormal and the E-DMAC can not operate successfully. In this case
the E-DMAC reception enabled again, execute a software reset by the SWR
EDMR0 (EDMR1). To make the E-DMAC reception disabled without executin
software reset, specify the RE bit in ECMR0 (ECMR1). Next, after the E_DM
completed the reception and write-back to the receive descriptor has been co
disable the receive function of this register.

19.2.4 Transmit Descriptor List Address Register (TDLAR)

TDLAR is a 32-bit readable/writable register that specifies the start address of the transm
descriptor list. Descriptors have a boundary configuration in accordance with the descrip
indicated by the DL bit in EDMR. This register must not be written to during transmissi
Modifications to this register should only be made while transmission is disabled by the
(= 0) in the E-DMAC transmit request register (EDTRR).

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDLA31 to TDLA0	All 0	R/W	Transmit Descriptor Start Address The lower bits are set as follows according specified descriptor length. 16-byte boundary: TDLA3 and TDLA0 = 00 32-byte boundary: TDLA4 and TDLA0 = 00 64-byte boundary: TDLA5 and TDLA0 = 00

	RDLA0	The lower bits are set as follows according to specified descriptor length. 16-byte boundary: RDLA3 and RDLA0 = 000 32-byte boundary: RDLA4 and RDLA0 = 000 64-byte boundary: RDLA5 and RDLA0 = 000

19.2.6 EtherC/E-DMAC Status Register (EESR)

EESR is a 32-bit readable/writable register that shows communications status information E-DMAC in combination with the EtherC. The information in this register is reported in t of interrupt sources. Individual bits are cleared by writing 1 (however, bit 22 (ECI) is a re bit and not to be cleared by writing 1) and are not affected by writing 0. Each interrupt so also be masked by means of the corresponding bit in the EtherC/E-DMAC status interrup permission register (EESIPR).

The interrupts generated by this register are EINT0 for channel 0 and EINT1 for channel interrupt priorities, see section 8, Interrupt Controller (INTC) and section 8.3.5, Interrupt Exception Handling and Priority.

The EINT2 is an interrupt generated by the TSU_FNSR in the EtherC.

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

				These bits are always read as 0. The write v should always be 0.
26	TABT	0	R/W	Transmit Abort Detection Indicates that the EtherC aborts transmittin because of failures during transmitting the f 0: Frame transmission has not been aborte transmit directive 1: Frame transmit has been aborted
25	RABT	0	R/W	Receive Abort Detection Indicates that the EtherC aborts receiving a because of failures during receiving the fra 0: Frame reception has not been aborted or receive directive 1: Frame receive has been aborted
24	RFCOF	0	R/W	Receive Frame Counter Overflow Indicates that the receive FIFO frame count overflowed. 0: Receive frame counter has not overflowe 1: Receive frame counter overflows

				reset with the SWR bit in EDMR.
22	ECI	0	R	EtherC Status Register Interrupt Source This bit is a read-only bit. When the source o ECSR interrupt in the EtherC is cleared, this also cleared. 0: EtherC status interrupt source has not bee detected 1: EtherC status interrupt source has been d
21	TC	0	R/W	Frame Transmit Complete Indicates that all the data specified by the tra descriptor has been transmitted from the Eth bit is set to 1, assuming the completion of transmission, when transmission of one fram completed in single-frame/single-descriptor c or when the last data of a frame has been tra and the transmit descriptor valid bit (TACT) c next descriptor is not set in for the processin multi-buffer frame based on single-frame/mu descriptor operation. After frame transmissio DMAC writes the transmission status back to relevant descriptor. 0: Transfer not complete, or no transfer direc 1: Transfer complete

				When transmission descriptor empty (TDE = occurs, execute a software reset and initiate transmission. In this case, the address that in the transmit descriptor list address registe (TDLAR) is transmitted first.
19	TFUF	0	R/W	Transmit FIFO Underflow Indicates that underflow has occurred in the FIFO during frame transmission. Incomplete sent onto the line. 0: Underflow has not occurred 1: Underflow has occurred
18	FR	0	R/W	Frame Reception Indicates that a frame has been received an receive descriptor has been updated. This b 1 each time a frame is received. 0: Frame not received 1: Frame received

				1: Receive descriptor active bit RACT = 0 de
16	RFOF	0	R/W	Receive FIFO Overflow Indicates that the receive FIFO has overflowe frame reception. 0: Overflow has not occurred 1: Overflow has occurred
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
11	CND	0	R/W	Carrier Not Detect Indicates the carrier detection status during p transmission. 0: A carrier is detected when transmission st 1: A carrier is not detected
10	DLC	0	R/W	Detect Loss of Carrier Indicates that loss of the carrier has been de during frame transmission. 0: Loss of carrier not detected 1: Loss of carrier detected
9	CD	0	R/W	Delayed Collision Detect Indicates that a delayed collision has been d during frame transmission. 0: Delayed collision not detected 1: Delayed collision detected

				0: Multicast address frame has not been rec 1: Multicast address frame has been receiv
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write v 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 the frame more than the numb receive frame length upper limit set by RFLI 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 byte been received. 0: Too-short frame has not been received 1: Too-short frame has been received
1	PRE	0	R/W	PHY-LSI Receive Error 0: PHY-LSI receive error not detected 1: PHY-LSI receive error detected
0	CERF	0	R/W	CRC Error on Received Frame 0: CRC error not detected 1: CRC error detected

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 val should always be 0.
26	TABTIP	0	R/W	Transmit Abort Detection Interrupt Enable 0: Transmit abort detection interrupt is disable 1: Transmit abort detection interrupt is enable
25	RABTIP	0	R/W	Receive Abort Detection Interrupt Enable 0: Receive abort detection interrupt is disabled 1: Receive abort detection interrupt is enabled
24	RFCOFIP	0	R/W	Receive Frame Counter Overflow Interrupt En 0: Receive frame counter overflow interrupt is 1: Receive frame counter overflow interrupt is
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 Interrupt Enable 0: EtherC status interrupt is disabled 1: EtherC status interrupt is enabled
21	TCIP	0	R/W	Frame Transmit Complete Interrupt Enable 0: Frame transmit complete interrupt is disable 1: Frame transmit complete interrupt is enable

				1: Frame received interrupt is enabled
17	RDEIP	0	R/W	Receive Descriptor Empty Interrupt Enable
				0: Receive descriptor empty interrupt is disab
				1: Receive descriptor empty interrupt is enab
16	RFOFIP	0	R/W	Receive FIFO Overflow Interrupt Enable
				0: Receive FIFO overflow interrupt is disabled
				1: Receive FIFO overflow interrupt is enabled
15 to 12	—	All 0	R	Reserved
				These bits are always read as 0. The write va 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 disable
				1: Delayed collision detect interrupt is enable
8	TROIP	0	R/W	Transmit Retry Over Interrupt Enable
				0: Transmit retry over interrupt is disabled
				1: Transmit retry over interrupt is enabled

				1: Receive residual-bit frame interrupt is enabl
3	RTLFIP	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
2	RTSFIP	0	R/W	Receive Too-Short Frame Interrupt Enable
				0: Receive too-short frame interrupt is disablec
				1: Receive too-short frame interrupt is enablec
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
				0: CRC error on received frame interrupt is dis
				1: CRC error on received frame interrupt is ena

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
7	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Reflects the RMAF bit status in the RFE bi receive descriptor 1: Occurrence of the corresponding source is reflected in the RFE bit of the receive desc
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				These bits are always read as 0. The write va should always be 0.
15 to 0	MFC15 to MFC0	All 0	R	Missed-Frame Counter Indicate the number of frames that are discar not transferred to the receive buffer during re

H′01 to H′0C: Setting prohibited

H′0D: 52 bytes

H′0E: 56 bytes

: :

H′1F: 124 bytes

H′20: 128 bytes

: :

H′3F: 252 bytes

H′40: 256 bytes

: :

H′7F: 508 bytes

H′80: 512 bytes

: :

H′FF: 1020 bytes

H′100: 1024 bytes

: :

H′1FF: 2044 bytes

H′200: 2048 bytes

Note: When starting transmission before one frame of data write has completed, take ca generation of the underflow.

				Specifies 256 bytes to 2 kbytes in 256-byte the size of the transmit FIFO. The setting m changed after transmission/reception has st
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
2 to 0	RFD2 to RFD0	All 1	R/W	Receive FIFO Size Specifies 256 bytes to 2 kbytes in 256-byte the size of the receive FIFO. The setting mu changed after transmission/reception has st

0	RNC	0	R/W	Receive Enable Control Sets whether to continue frame reception. 0: Upon completion of reception of one frame DMAC writes receive status to the descrip clears the RR bit in EDRRR to 0 1: Upon completion of reception of one frame DMAC writes (writes back) receive status descriptor. In addition, the E-DMAC reads descriptor and prepares for the reception o next frame

3	TEC	0	R/W	FIFO Error Control Specifies E-DMAC operation when transmit underflow or receive FIFO overflow occurs. 0: E-DMAC operation continues when unde overflow occurs 1: E-DMAC operation halts when underflow overflow occurs
2	AEC	0	R/W	Address Error Control Indicates detection of an illegal memory ad an attempted E-DMAC transfer. 0: Illegal memory address not detected (nor operation) 1: Indicates that E-DMAC operation is halte an illegal memory address is detected. W written to this bit, the E-DMAC resumes
1 to 0	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.

19.2.15 Receive Descriptor Fetch Address Register (RDFAR)

RDFAR stores the descriptor start address that is required when the E-DMAC fetches des information from the receiving descriptor. Which receiving descriptor information is used processing by the E-DMAC can be recognized by monitoring addresses displayed in this The address from which the E-DMAC is actually fetching a descriptor may be different f value read from this register.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDFA31 to RDFA0	All 0	R	Receiving-Descriptor Fetch Address These bits can only be read. Writing is prohib

19.2.16 Transmit Buffer Read Address Register (TBRAR)

TBRAR stores the address of the transmission buffer when the E-DMAC reads data from transmission buffer. Which addresses in the transmission buffer are processed by the E-D can be recognized by monitoring addresses displayed in this register. The address from w E-DMAC is actually reading in the buffer may be different from the value read from this

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TBRA31 to TBRA0	All 0	R	Transmission-Buffer Read Address These bits can only be read. Writing is prohib

TDFA0	These bits can only be read. Writing is proh

19.2.18 Overflow Alert FIFO Threshold Register (FCFTR)

FCFTR is a 32-bit readable/writable register that sets the flow control of the EtherC. Th
can be specified by the size of the receive FIFO data (RFD2 to RFD0) and the number o
frames (RFF2 to RFF0).

If the same receive FIFO size as set by the FIFO size register (FDR) is set when flow co
turned on according to the RFD setting condition, flow control is turned on with (FIFO
64) bytes. For instance, when RFD in FDR = 7 and RFD in FCFTR = 7, flow control is
when (2048 – 64) bytes of data is stored in the receive FIFO. The value set in the RFD b
register should be equal to or less than those in FDR.

Flow control is turned on when any of the setting conditions of the RFF2 to RFF0 bits o
RFD2 to RFD0 bits is satisfied. Flow control is turned off when none of the conditions i
(release).

				: :
				110: When seven receive frames have been the receive FIFO
				111: When eight receive frames have been s the receive FIFO
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
2	RFD2	1	R/W	Receive FIFO Overflow Alert Signal Output T
1	RFD1	1	R/W	000: When (256 – 32) bytes of data is stored
0	RFD0	1	R/W	receive FIFO
				001: When (512 – 32) bytes of data is stored receive FIFO
				: :
				110: When (1792 – 32) bytes of data is store receive FIFO
				111: When (2048 – 64) bytes of data is store receive FIFO

0	TIS	0	R/W	Transmit Interrupt Setting 0: Write-backed completion for each frame TWB bit in EESR is notified 1: Write-back completion for each frame is notified

19.3 Operation

Using its direct memory access (DMA) function, the E-DMAC performs DMA transfer
transmit/receive data between a Ethernet frame transmission/reception data storage desti
user- specified (accessible memory space: transmit buffer/receive buffer) and the transm
FIFO in the E-DMAC. (The user cannot read and write data in the transmit/receive FIFO
via the CPU).

To enable the E-DMAC to perform DMA transfer, information (data) including a transm
data storage address and so forth, referred to as a descriptor, is required. Before Etherne
transmission/reception, the E-DMAC reads descriptor information, then reads transmit d
the transmit buffer or writes receive data to the receive buffer according to the read desc
information. By arranging multiple descriptors as a descriptor row (list) (to be placed in
readable/writable memory space), multiple Ethernet frames can be transmitted or receiv
continuously.

are prepared as a descriptor row (descriptor list), the descriptors are placed in continuous
(memory) addresses according to the descriptor length set in the DL0 and DL1 bits in ED

The E-DMAC consists of two systems: system 0 and system 1. The DMAC for transmiss
the DMAC for reception operate independently of each other, and the DMAC for system
DMAC for system 1 operate independently of each other. For normal E-DMAC operatio
descriptors for transmission and reception and descriptors for system 0 and system 1 in th
address spaces that do not overlap.

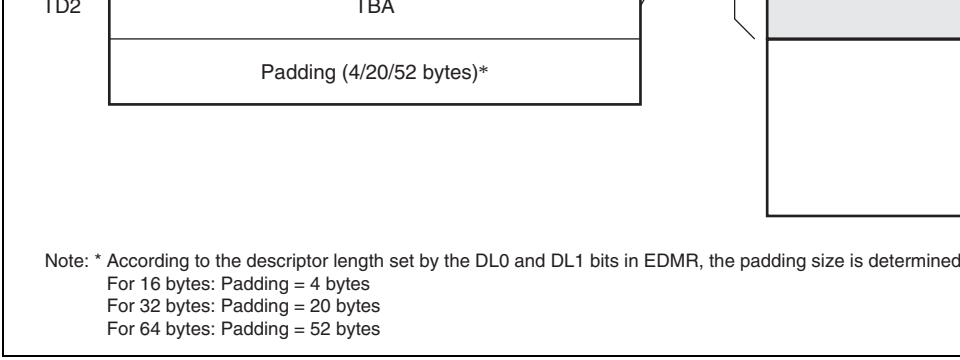
(1) **Transmit Descriptor**

Figure 19.2 shows the configuration of a transmit descriptor and the relationship with a tr
buffer.

The data of a transmit descriptor consists of TD0, TD1, TD2, and padding data in groups
bits from top to end. The length of padding data is determined according to the descriptor
specified by the DL0 and DL1 bits in EDMR. In the figure, TBA (bits 31 to 0 in TD2) inc
the start address of a transmit buffer, and TDL (bits 31 to 16 in TD1) indicates the valid d
length of the transmit buffer.

TD0 indicates whether the transmit descriptor is valid or invalid as well as information ab
descriptor configuration and status. TD1 indicates the length of data in a transmit buffer t
transferred according to the specification of the descriptor. TD2 indicates the start addres
transmit buffer that holds data to be transferred.

Depending on the descriptor specification, one transmit descriptor can specify all transmi
one frame (single-frame/single-buffer) or multiple descriptors can specify the transmit da
frame (single-frame/multi-buffer). As an example of single-frame/multi-buffer operation,
portion that is used in a fixed manner in each Ethernet frame transmission can be referenc
multiple descriptors. For example, multiple descriptors can share the destination address a



Note: * According to the descriptor length set by the DL0 and DL1 bits in EDMR, the padding size is determined
For 16 bytes: Padding = 4 bytes
For 32 bytes: Padding = 20 bytes
For 64 bytes: Padding = 52 bytes

Figure 19.2 Relationship between Transmit Descriptor and Transmit Buff

31	TACT	0	R/W	Transmit Descriptor Valid/Invalid Indicates whether the corresponding descript valid or invalid. To make this bit valid, store t data in a transmit buffer (user-specified trans storage destination) beforehand, then write 1 bit. The E-DMAC clears this bit to 0 upon cor of data transfer. 0: Indicates that the transmit descriptor is inv Indicates the initial setting state, the state written, or (in case the user writes 1 to this this bit is cleared to 0 because of completi processing of the E-DMAC data transfer. If this state is recognized when the E-DMA a descriptor, the E-DMAC clears the TR b EDTRR to 0, and halts transfer operation transmission by the E-DMAC. 1: Indicates that the transmit descriptor is val After the user writes 1 to this bit, this bit in that data is not transferred yet or data is b transferred. When there is a descriptor row (descriptor consisting of multiple continuous descripto E-DMAC can continue operation when this the next descriptor is valid.

Upon completion of transfer of the corres
descriptor, the E-DMAC reads the descri
placed at the address indicated by TDLA

10: The information of the descriptor represe
information about the start of the frame.

11: The information of the descriptor represe
information about the frame (single-frame
descriptor (single-buffer)).

Reference:

When one frame is divided for use, the meth
specifying this bit for a descriptor row accord
number of divisions is described below.

[For single-frame/single-descriptor operation]

First descriptor: TFP[1:0] = 11

[For single-frame/two-descriptor operation]

First descriptor: TFP[1:0] = 10

Second descriptor: TFP[1:0] = 01

[For single-frame/three-descriptor operation]

First descriptor: TFP[1:0] = 10

Second descriptor: TFP[1:0] = 00

Third descriptor: TFP[1:0] = 01

When the number of divisions is large, a des
row is configured by adding intermediate des
with TFP[1:0] = 00.

TFS26 to TFS9: Reserved (The write value
always be 0.)

TFS8: Transmit abort detected

Note: This bit is set when any bit of TFS3 to
set.

TFS7 to TFS4: Reserved (The write value s
always be 0)

TFS3: Failure to detect the carrier at the sta
transmission (corresponding to the C
EESR)

TFS2: Loss of the carrier during transmissio
(corresponding to the DLC bit in EES

TFS1: Late (delayed) collision (correspondi
CD bit in EESR)

TFS0: Transmit retry over (corresponding t
bit in EESR)

15 to 0	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

(c) Transmit Descriptor 2 (TD2)

TD2 indicates the start address of the corresponding 32-bit width transmit buffer. An add value on a longword boundary should be specified.

The user should set TD2 before the start of a read by the E-DMAC.

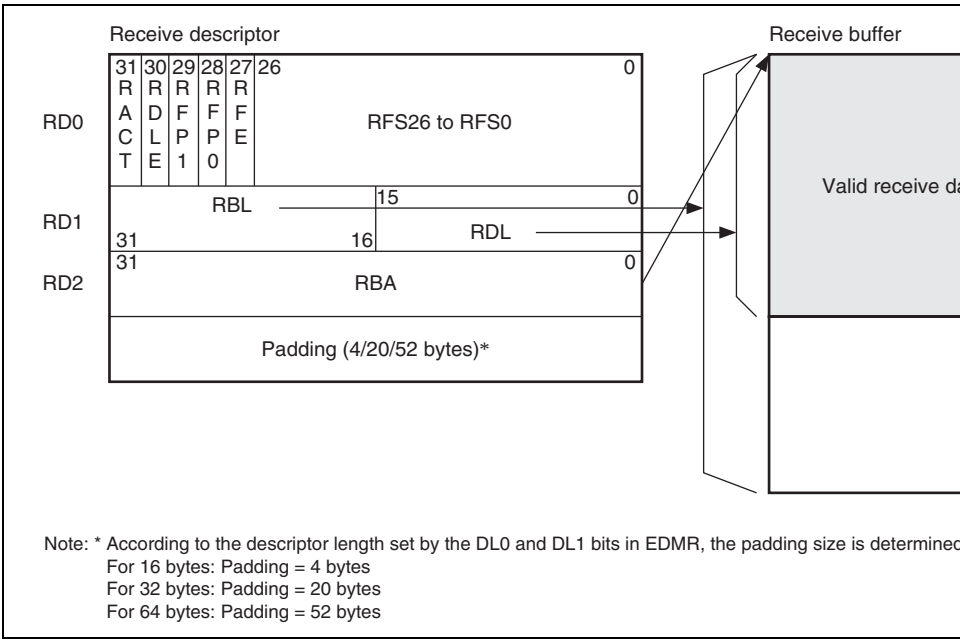
(2) Receive Descriptor

Figure 19.3 shows the relationship between a receive descriptor and receive buffer.

The data of a receive descriptor consists of RD0, RD1, RD2, and padding data in groups from top to end. The length of padding data is determined according to the descriptor leng specified by the DL0 and DL1 bits in EDMR. In the figure, RBA (bits 31 to 0 in RD2) in the start address of a receive buffer. RBL (bits 31 to 16 in RD1) indicates the usable data the receive buffer. RDL (bits 15 to 0 in RD1) indicates the data length of a received fram

RD0 indicates whether the receive descriptor is valid or invalid as well as information ab descriptor configuration and status. RD1 indicates the length (storage destination size) of the receive buffer to be received according to the specification of the descriptor. RD2 ind start address of the receive buffer for storing receive data.

Depending on the descriptor specification, one receive descriptor can specify the storing receive data of one frame in a receive buffer (single-frame/single-buffer) or multiple desc can specify the storing of the receive data of one frame in receive buffers (single-frame/m buffer). As an example of single-frame/multi-buffer operation, suppose that a row of mul descriptors (descriptor list) is prepared, RBL of each descriptor is 500 bytes, and a 1514-



Note: * According to the descriptor length set by the DL0 and DL1 bits in EDMR, the padding size is determined
For 16 bytes: Padding = 4 bytes
For 32 bytes: Padding = 20 bytes
For 64 bytes: Padding = 52 bytes

Figure 19.3 Relationship between Receive Descriptor and Receive Buffe

Bit	Bit Name	Value	R/W	Description
31	RACT	0	R/W	Receive Descriptor Valid/Invalid Indicates whether the corresponding descript valid or invalid. To make this bit valid, prepar receive buffer (user-specified receive data st destination) beforehand, then write 1 to this b E-DMAC clears this bit to 0 upon completion transfer. 0: Indicates that the receive descriptor is inva Indicates the initial setting state, the state written, or (in case the user writes 1 to this this bit is cleared to 0 because of completi processing of the E-DMAC data transfer. If this state is recognized when the E-DMA a descriptor, the E-DMAC clears the RR b EDRRR to 0, and halts transfer operation to reception by the E-DMAC. 1: Indicates that the receive descriptor is vali Indicates that data is not transferred yet af user writes 1 to this bit, or that data is beir transferred. When there is a descriptor row (descriptor consisting of multiple continuous descripto E-DMAC can continue operation when this the next descriptor is valid.

Upon completion of transfer of the corresp
descriptor, the E-DMAC reads the descrip
at the address indicated by RDLAR.

10: The information of the descriptor represe
information about the start of the frame.

11: The information of the descriptor represe
information about the frame (single-frame
descriptor (single-buffer)).

Reference:

The relationship between a frame after recep
one frame and a descriptor is described belo

[For single-frame/single-descriptor operation]

First descriptor: RFP[1:0] = 11

[For single-frame/two-descriptor operation]

First descriptor: RFP[1:0] = 10

Second descriptor: RFP[1:0] = 01

[For single-frame/three-descriptor operation]

First descriptor: RFP[1:0] = 10

Second descriptor: RFP[1:0] = 00

Third descriptor: RFP[1:0] = 01

When the number of divisions is large, a des
row is configured by adding intermediate des
with RFP[1:0] = 00.

below, which set to 1, indicates the occurrence
corresponding event. If the events of RFS8,
to RFS0 occur, frames are incompletely rec

RFS26 to RFS10: Reserved (The write valu
always be 0)

RFS9: Receive FIFO overflow (correspondi
RFOF bit in EESR)

RFS8: Receive abort detected

Note: This bit is set when any bit of RFS3 t
is set.

RFS7: Multicast address frame received
(corresponding to the RMAF bit in E

RFS6 and RFS5: Reserved (The write valu
always be 0)

RFS4: residual-bit frame receive error (corr
to the RRF bit in EESR)

RFS3: Too-long frame receive error (corres
to the RTLF bit in EESR)

RFS2: Too-short frame receive error (corre
to the RTSF bit in EESR)

RFS1: PHY-LSI receive error (correspondir
PRE bit in EESR)

RFS0: CRC error on receive frame (corresp
the CERF bit in EESR)

				Set the length of data that can be received by corresponding receive buffer in bytes. Set a receive buffer length on a 16-byte boun (with bits 19 to 16 cleared to 0). In single-frame/single-buffer (descriptor) ope the maximum receive frame length excluding data is 1514 bytes. When specifying a receiv length, set 1520 bytes (H'05F0), which is det considering the maximum receive frame leng 16-byte boundary.
15 to 0	RDL	All 0	R	Receive Data Length Indicate the data length of a receive frame st the receive buffer. Receive data transferred to the receive buffe not include CRC data (4 bytes) placed at the frame. As a receive frame length, the numbe (valid data bytes) not including CRC data are reported. In single-frame/multi-buffer (descriptor) opera only the receive data length of the last descri valid. The receive data length of an intermed descriptor has no meaning.

(c) Receive Descriptor 2 (RD2)

RD2 indicates the start address of the corresponding 32-bit width receive buffer. Set the s address of a receive buffer on a longword boundary. When an SDRAM is connected, set address of a receive buffer on a 16-byte boundary.

The user should set RD2 before the start of a read by the E-DMAC.

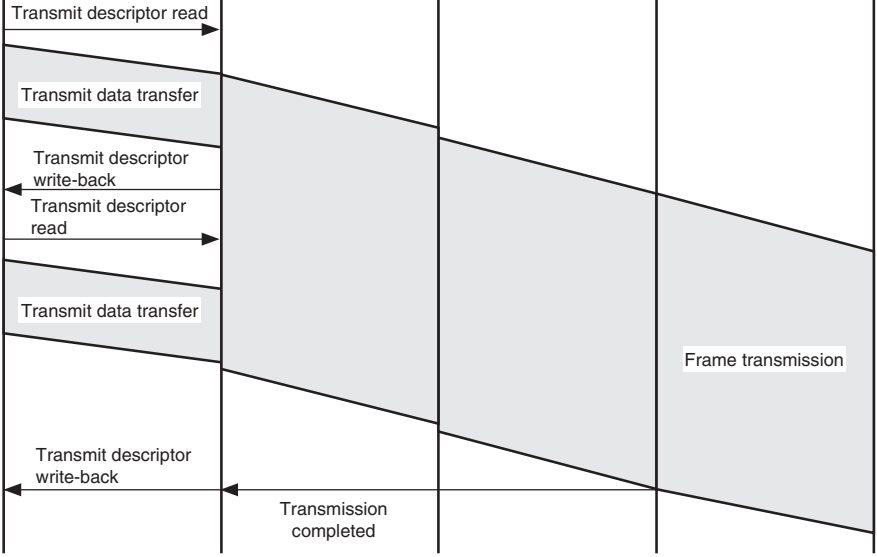
TFP value.

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 af 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 b to 0 (invalid) is read, the E-DMAC clears the TR bit in EDTRR to 0 and completes trans processing.



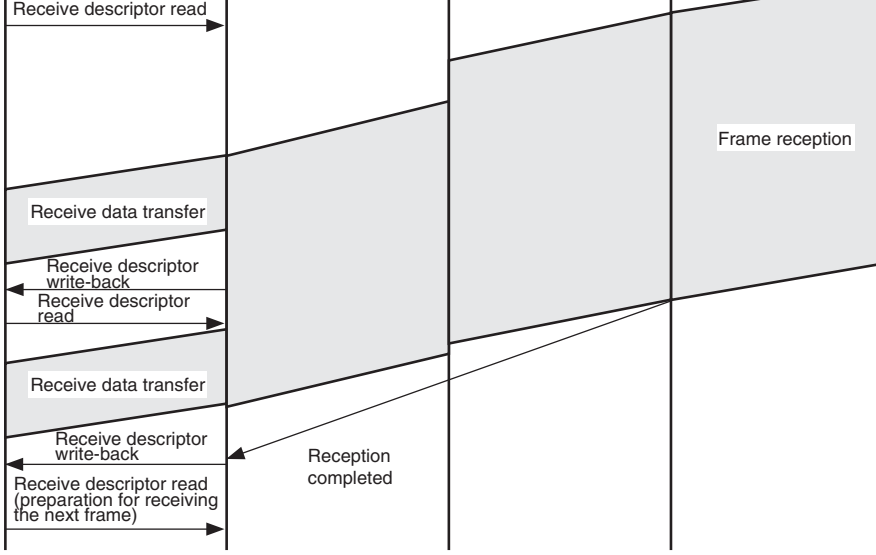
[Legend]

EtherC/E-DMAC initialization: Executes a software reset with the SWR bit in EDMR set to 1.
Transmit descriptor and transmit buffer setting: Sets transmit descriptors and transmit buffers, and sets EtherC and E-DMAC registers, then writes 1 to the TE bit in ECMR and the TR bit in EDTRR.
Start of transmission: Occurs when 1 is written to the TE bit in ECMR and the TR bit in EDTRR.
Transmit descriptor read: The E-DMAC reads a transmit descriptor.
Transmit data transfer: Writes transmit data to the transmit FIFO by using DMA transfer by the E-DMAC.
Transmit descriptor write-back: The E-DMAC writes 0 to the TACT bit and writes the transmit status to the transm

Figure 19.4 Sample Transmission Flowchart (Single-Frame/Two-Descriptor)

specified by RD1, the E-DMAC performs a write-back operation to the descriptor (with
10 or 00) when the buffer becomes full, then reads the next descriptor. The E-DMAC th
continues to transfer data to the receive buffer specified by the new RD2. When frame r
completed, or if frame reception is suspended because of a certain kind of error, the E-D
performs write-back to the relevant descriptor (with RFP set to 11 or 01), and then ends
receive processing. The E-DMAC then reads the next descriptor and enters the receive s
state again.

To receive frames continuously, the receive enable control bit (RNC) must be set to 1 in
receive method control register (RMCR). The initial value is 0.



[Legend]

EtherC/E-DMAC initialization: Executes a software reset with the SWR bit in EDMR set to 1.
Receive descriptor and receive buffer setting: Sets receive descriptors and receive buffers, and sets EtherC and registers, then writes 1 to the RE bit in ECMR and the RR bit in ED
Start of reception: Occurs when 1 is written to the RE bit in ECMR and the RR bit in EDRRR.
Receive descriptor read: The E-DMAC reads a receive descriptor.
Receive data transfer: Writes receive data from the receive FIFO to the receive buffer by using DMA transfer by t
Receive descriptor write-back: The E-DMAC writes 0 to the RACT bit and writes the receive status to the receive

Figure 19.5 Sample Reception Flowchart (Single-Frame/Two-Descriptor)

part where the transmit descriptor is active (TACT bit = 1), transmission is halted, and th
bit cleared to 0, immediately. The next descriptor is then read, and the position within th
frame is determined on the basis of bits TFP1 and TFP0 (continuing [B′00] or end [B′01
case of a continuing descriptor, the TACT bit is cleared to 0, only, and the next descripto
immediately. If the descriptor is the final descriptor, not only is the TACT bit cleared to
write-back is also performed to the TFE and TFS bits at the same time. Data in the buffe
transmitted between the occurrence of an error and write-back to the final descriptor. If
interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EESI
interrupt is generated immediately after the final descriptor write-back.

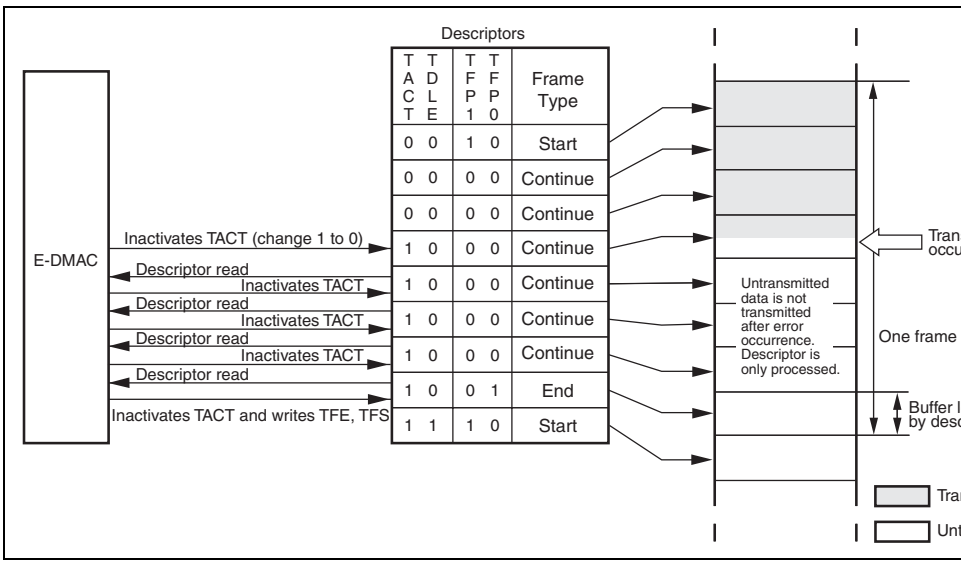


Figure 19.6 E-DMAC Operation after Transmit Error

(EESIPR), an interrupt is generated immediately after the write-back. If there is a new fra
receive request, reception is continued from the buffer after that in which the error occurr

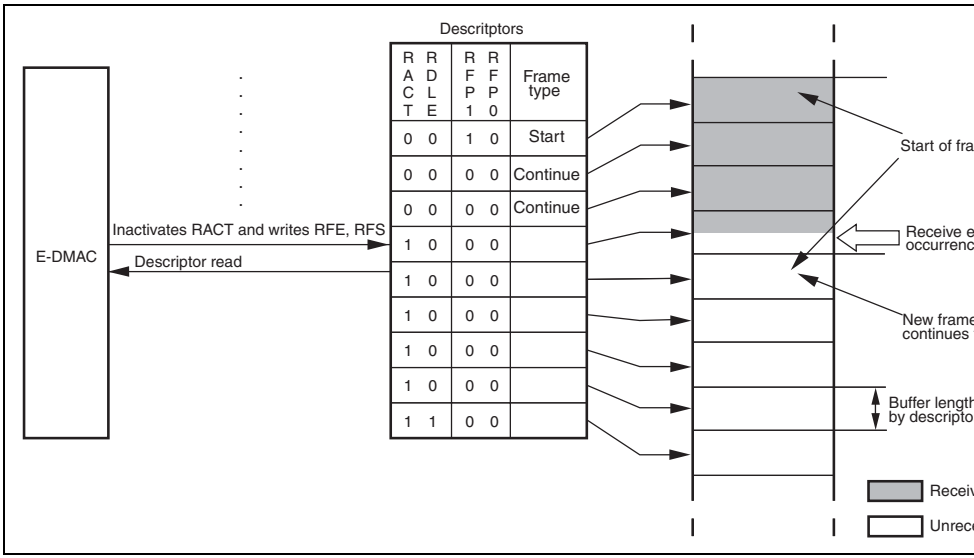


Figure 19.7 E-DMAC Operation after Receive Error

and 256 – 32 bytes.

Figure 19.9 shows the configuration of the receive FIFO overflow alert signal ($\overline{\text{ARBUSY}}$

As shown in figure 19.9, because the $\overline{\text{ARBUSY}}$ signal passes through the system clock synchronization circuit, it is behind the receive FIFO overflow alert signal received in E

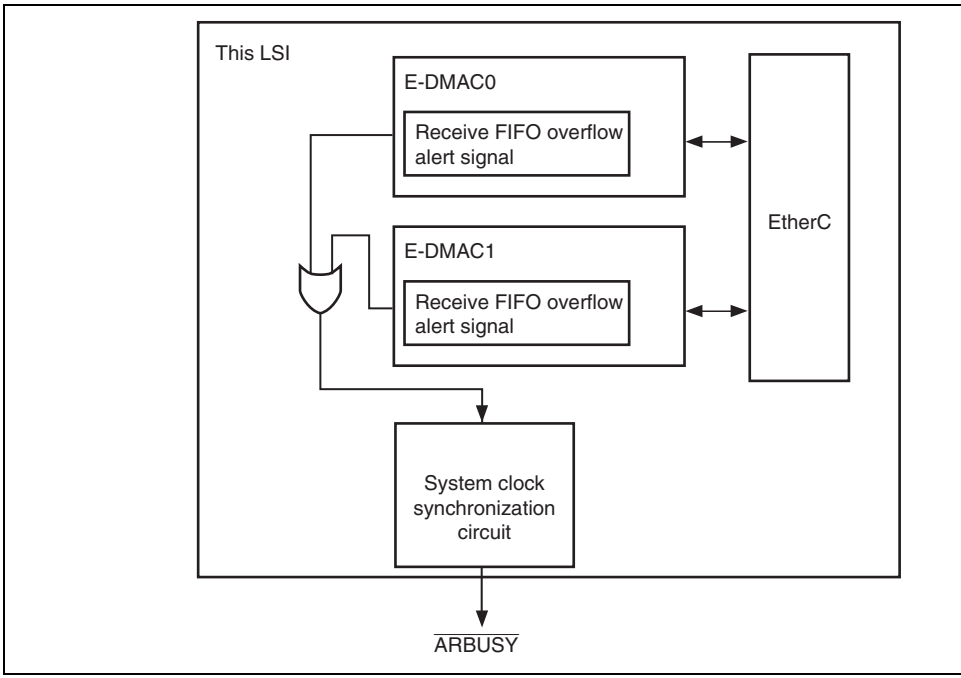


Figure 19.8 Configuration of $\overline{\text{ARBUSY}}$

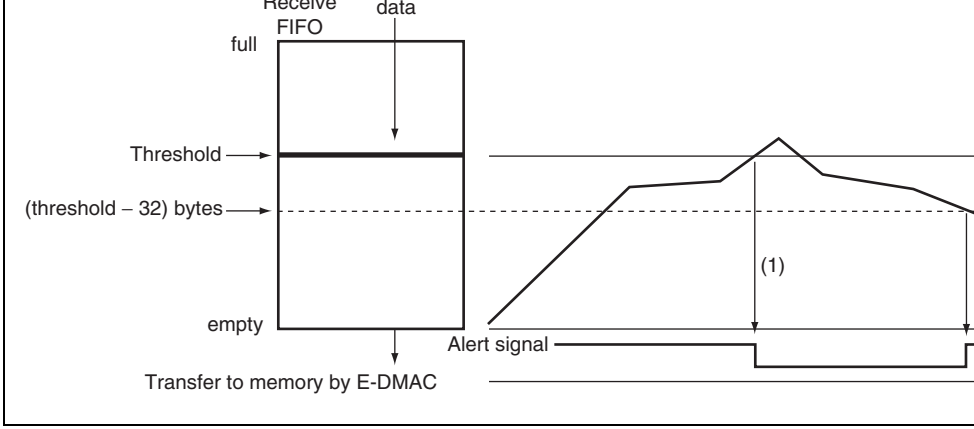


Figure 19.9 Summary of Receive FIFO Overflow Alert Signal

(a) Receive FIFO Overflow Alert Signal Changing

The receive FIFO in the E-DMAC can perform writing (reception) data from the Etherne reading data from the system simultaneously. Therefore during system operation, receive FIFO is always increased or decreased. If the change is performed near the threshold, the FIFO overflow alert signal may be seen as shown in figure 19.10. Minimum of receive da changes depending on the number of FIFO read cycles and rate of Ethernet line (10 to 10

Figure 19.10 $\overline{\text{ARBUSY}}$ Signal Change and Minimum Pulse Width Depending on Increase and Decrease of FIFO

or RR) is set to 1.

[Occurring Condition]

When the user's firmware tries to set the request bit (TR or RR), while the request bit (TR
is 1.

[Avoiding Methods]

To prevent the simultaneous occurrence of the request bit (TR or RR) being cleared by th
DMAC and the request bit (TR or RR) being set by the user's firmware, the user's firmwa
set the request bit (TR or RR) after confirming that it is cleared by the E-DMAC.

The methods to clear the RR bit with E-DMAC are as follows.

(1) Confirmation of the TR bit

As a direct method, it is possible to confirm by reading the TR bit in EDTRR as 0.

As an indirect method, it is possible to confirm by reading the TDE bit in EESR as 1.

(2) Confirmation of the RR bit

As a direct method, it is possible to confirm by reading the RR bit in EDRRR as 0.

As an indirect method, it is possible to confirm by reading the RDE bit in EESR as 0.

Port	Port Function (Related Module)	Other Function (Related Modu
A	PTA7 input/output (port)	SIOFSYNC0 input/output (SIOF
A	PTA6 input/output (port)	TXD_SIO0 output (SIOF0)
A	PTA5 input/output (port)	RXD_SIO0 input (SIOF0)
A	PTA4 input/output (port)	SIOMCLK0 input (SIOF0)
A	PTA3 input/output (port)	SCK_SIO0 input/output (SIOF0
A	PTA2 input/output (port)	SCIF0CK input/output (SCIF0)
A	PTA1 input/output (port)	TXD0 output (SCIF0)
A	PTA0 input/output (port)	RXD0 input (SCIF0)
B	PTB7 input/output (port)	$\overline{\text{RTS0}}$ output (SCIF0)
B	PTB6 input/output (port)	$\overline{\text{CTS0}}$ input (SCIF0)
B	PTB5 input/output (port)	SCIF1CK input/output (SCIF1)
B	PTB4 input/output (port)	TXD1 output (SCIF1)
B	PTB3 input/output (port)	RXD1 input (SCIF1)
B	PTB2 input/output (port)	$\overline{\text{RTS1}}$ output (SCIF1)
B	PTB1 input/output (port)	$\overline{\text{CTS1}}$ input (SCIF1)
B	PTB0 input/output (port)	Reserved (setting prohibited)*
C	PTC7 input/output (port)	$\overline{\text{IOIS16}}$ input (BSC)
C	PTC6 input/output (port)	$\overline{\text{CE2B}}$ output (BSC)
C	PTC5 input/output (port)	$\overline{\text{CE2A}}$ output (BSC)
C	PTC4 input/output (port)	SIOFSYNC1 input/output (SIOF

EXOUT1 output	TEND1 output (DMAC)
CAMSEN1 input	IRQ5 input (INTC)
EXOUT0 output	TEND0 output (DMAC)
CAMSEN0 input	IRQ4 input (INTC)

20.2 Register Configuration

The registers of the pin function controller are shown below.

- Port A control register (PACR)
- Port B control register (PBCR)
- Port C control register (PCCR)
- Ethernet controller pin control register (PETCR)

15	PA7MD1	1	R/W	Modes PA7 to PA0 Control
14	PA7MD0	0	R/W	The combination of PAnMD1 and PAnMD0 (7) selects the pin functions and control input MOS.
13	PA6MD1	1	R/W	
12	PA6MD0	0	R/W	00: Other function (see Table 20.1)
11	PA5MD1	1	R/W	01: Port output
10	PA5MD0	0	R/W	10: Port input (pull-up MOS: on)
9	PA4MD1	1	R/W	11: Port input (pull-up MOS: off)
8	PA4MD0	0	R/W	
7	PA3MD1	1	R/W	
6	PA3MD0	0	R/W	
5	PA2MD1	1	R/W	
4	PA2MD0	0	R/W	
3	PA1MD1	1	R/W	
2	PA1MD0	0	R/W	
1	PA0MD1	1	R/W	
0	PA0MD0	0	R/W	

13	PB6MD1	1	R/W	MOS.
12	PB6MD0	0	R/W	00: Other function (n = 1 to 7) or reserved (n =
11	PB5MD1	1	R/W	Table 20.1)
10	PB5MD0	0	R/W	01: Port output
9	PB4MD1	1	R/W	10: Port input (pull-up MOS: on)
8	PB4MD0	0	R/W	11: Port input (pull-up MOS: off)
7	PB3MD1	1	R/W	
6	PB3MD0	0	R/W	
5	PB2MD1	1	R/W	
4	PB2MD0	0	R/W	
3	PB1MD1	1	R/W	
2	PB1MD0	0	R/W	
1	PB0MD1	1	R/W	
0	PB0MD0	0	R/W	

13	PC6MD1	1	R/W	MOS.
12	PC6MD0	0	R/W	00: Other function (see Table 20.1.)
11	PC5MD1	1	R/W	01: Port output
10	PC5MD0	0	R/W	10: Port input (pull-up MOS: on)
9	PC4MD1	1	R/W	11: Port input (pull-up MOS: off)
8	PC4MD0	0	R/W	
7	PC3MD1	1	R/W	
6	PC3MD0	0	R/W	
5	PC2MD1	1	R/W	
4	PC2MD0	0	R/W	
3	PC1MD1	1	R/W	
2	PC1MD0	0	R/W	
1	PC0MD1	1	R/W	
0	PC0MD0	0	R/W	

				0: TEND1 (other function) is selected. 1: EXOUT1 (Ethernet controller function) is s
14	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
13	PET2MD	1	R/W	Controls input of CAMSEN1 (Ethernet control function) and IRQ5 (other function). 0: IRQ5 (other function) is selected. 1: CAMSEN1 (Ethernet controller function) is selected.
12	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
11	PET1MD	1	R/W	Controls output of EXOUT0 (Ethernet control function) and TEND0 (other function). 0: TEND0 (other function) is selected. 1: EXOUT0 (Ethernet controller function) is s
10	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
9	PET0MD	1	R/W	Controls input of CAMSEN0 (Ethernet control function) and IRQ4 (other function). 0: IRQ4 (other function) is selected. 1: CAMSEN0 (Ethernet controller function) is selected.

				always be 0.
5	—	1	R	Reserved This bit is always read as 1. The write value always be 1.
4	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
3	—	1	R	Reserved This bit is always read as 1. The write value always be 1.
2	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
1	—	1	R	Reserved This bit is always read as 1. The write value always be 1.
0	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

21.2.1 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores data for pins PTA7 to PTA0. Bits
to PA0DT correspond to pins PTA7 to PTA0. PADR is initialized to H′00 by a power-o
is not initialized by a manual reset, in standby mode, or sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DT	0	R/W	When the pin function is general output port is read, the value of the corresponding PAD returned directly. When the function is gene port, if the port is read, the corresponding pi read. Table 21.1 shows the function of PAD
6	PA6DT	0	R/W	
5	PA5DT	0	R/W	
4	PA4DT	0	R/W	
3	PA3DT	0	R/W	
2	PA2DT	0	R/W	
1	PA1DT	0	R/W	
0	PA0DT	0	R/W	

n = 0 to 7

21.2.2 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores the data for pins PTB7 to PTB0. B
PB7DT to PB0DT correspond to the pins PTB7 to PTB0. PBDR is initialized to H′00 by
on reset but is not initialized by a manual reset, in standby mode, or sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DT	0	R/W	When the pin function is general output port, is read, the value of the corresponding PBDR returned directly. When the function is genera port, if the port is read, the corresponding pin read. Tables 21.2 and 21.3 show the function PBDR.
6	PB6DT	0	R/W	
5	PB5DT	0	R/W	
4	PB4DT	0	R/W	
3	PB3DT	0	R/W	
2	PB2DT	0	R/W	
1	PB1DT	0	R/W	
0	PB0DT	0	R/W	

n = 1 to 7

Table 21.3 Port B Data Register (PBDR) Read/Write Operations (2)

PBnMD1	PBnMD0	Pin State	Read	Write
0	0	Reserved*	PBDR value	Value is written to PBDR, but does affect pin state.
	1	Output	PBDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin status	Value is written to PBDR, but does affect pin state.
	1	Input (Pull-up MOS off)	Pin status	Value is written to PBDR, but does affect pin state.

[Legend]

n = 0

Note: * When this pin is specified as a reserved pin, its operation is not guaranteed.

5	PC5DT	0	R/W
4	PC4DT	0	R/W
3	PC3DT	0	R/W
2	PC2DT	0	R/W
1	PC1DT	0	R/W
0	PC0DT	0	R/W

port, if the port is read, the corresponding pin l
read. Table 21.4 shows the function of PCDR.

Table 21.4 Port C Data Register (PCDR) Read/Write Operations

PCnMD1	PCnMD0	Pin State	Read	Write
0	0	Other function	PCDR value	Value is written to PCDR, but does affect pin state.
	1	Output	PCDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PCDR, but does affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PCDR, but does affect pin state.

[Legend]
n = 0 to 7

The H-UDI (User debugging interface) is a serial I/O interface which conforms to JTAG
Test Action Group, IEEE Standard 1149.1 and IEEE Standard Test Access Port and Bou
Scan Architecture) specifications.

The H-UDI in this LSI supports a boundary scan mode, and is also used for emulator co

When using an emulator, H-UDI functions should not be used. Refer to the emulator ma
the method of connecting the emulator.

Figure 22.1 shows a block diagram of the H-UDI.

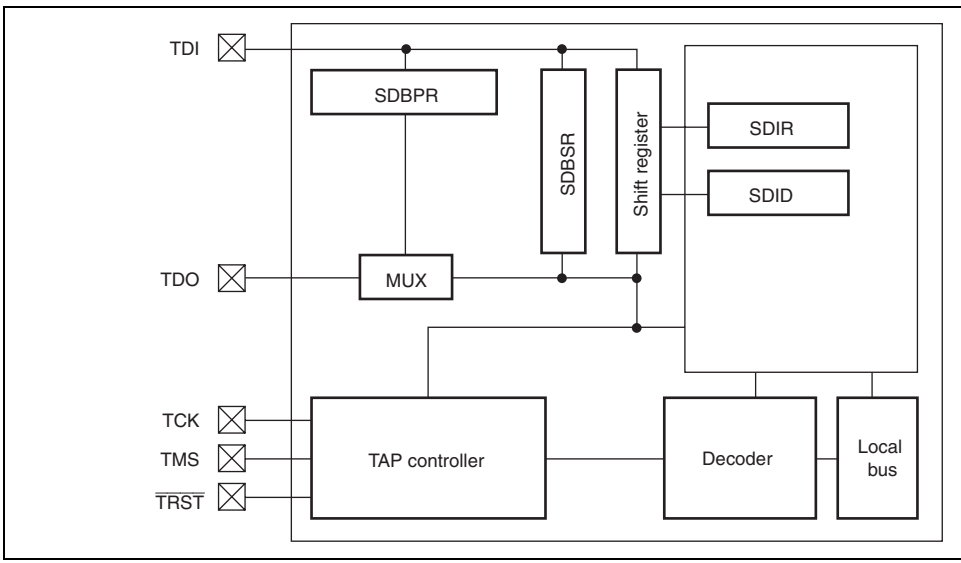


Figure 22.1 Block Diagram of H-UDI

TMS	Input	Mode Select Input Pin The state of the TAP control circuit is determined by ch this signal in synchronization with TCK. The protocol c to the JTAG standard (IEEE Std.1149.1).
$\overline{\text{TRST}}$	Input	Reset Input Pin Input is accepted asynchronously with respect to TCK, when low, the H-UDI is reset. $\overline{\text{TRST}}$ must be low for a period when power is turned on regardless of using the function. As the same as the $\overline{\text{RESETP}}$ pin, the $\overline{\text{TRST}}$ p be driven low at the power-on reset state and driven hi the power-on reset state is released. This is different fr JTAG standard. See section 22.4.2, Reset Configuration, for more infor
TDI	Input	Serial Data Input Pin Data transfer to the H-UDI is executed by changing this in synchronization with TCK.
TDO	Output	Serial Data Output Pin Data read from the H-UDI is executed by reading this p synchronization with TCK. The data output timing depe the command type set in the SDIR. See section 22.4.3 Output Timing, for more information.
$\overline{\text{ASEMD0}}$	Input	ASE Mode Select Pin If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RES}}$ is asserted, ASE mode is entered; if a high level is inpu normal mode is entered. In ASE mode, dedicated emu function can be used. The input level at the $\overline{\text{ASEMD0}}$ p should be held for at least one cycle after $\overline{\text{RESETP}}$ neg

and access size for registers.

- Bypass register (SDBPR)
- Instruction register (SDIR)
- Boundary scan register (SDBSR)
- ID register (SDID)

22.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to the
mode, SDBPR is connected between H-UDI pins TDI and TDO. The initial value is und
SDBPR is initialized to 0 if the TAP is in Capture-DR state.

22.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. The register is in JTAG IDCODE in its initial state.
initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state, and can be written to b
UDI irrespective of the CPU mode. Operation is not guaranteed if a reserved command
this register.

0	—	1	R	Reserved This bit is always read as 1.

Table 22.2 H-UDI Commands

Bits 15 to 8								
TI7	**TI6**	**TI5**	**TI4**	**TI3**	**TI2**	**TI1**	**TI0**	**Description**
0	0	0	0	—	—	—	—	JTAG EXTEST
0	0	1	0	—	—	—	—	JTAG CLAMP
0	0	1	1	—	—	—	—	JTAG HIGHZ
0	1	0	0	—	—	—	—	JTAG SAMPLE/PRE
0	1	1	0	—	—	—	—	H-UDI reset negate
0	1	1	1	—	—	—	—	H-UDI reset assert
1	0	1	—	—	—	—	—	H-UDI interrupt
1	1	1	0	—	—	—	—	JTAG IDCODE (Initi
1	1	1	1	—	—	—	—	JTAG BYPASS
Other than the above								Reserved

22.3.3 Boundary Scan Register (SDBSR)

SDBSR is a shift register, located on the PAD, for controlling the input/output pins of thi
The initial value is undefined. SDBSR cannot be accessed by the CPU.

Using the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands, a boundary
test conforming to the JTAG standard can be carried out. Table 22.3 shows the correspon
between this LSI's pins and boundary scan register bits.

356	D4	IN	327	D9	C
355	D5	IN	326	D10	C
354	D6	IN	325	D11	C
353	D7	IN	324	D12	C
352	D8	IN	323	D13	C
351	D9	IN	322	D14	C
350	D10	IN	321	D15	C
349	D11	IN	320	$\overline{WE0}$($\overline{BE0}$)/DQMLL	C
348	D12	IN	319	$\overline{WE1}$($\overline{BE1}$)/DQMLU/$\overline{WE}$	C
347	D13	IN	318	RD/$\overline{WR}$	C
346	D14	IN	317	$\overline{CAS}$	C
345	D15	IN	316	CKE	C
344	$\overline{REFOUT}$/IRQOUT/$\overline{ARBUSY}$	OUT	315	$\overline{RAS}$	C
343	$\overline{BACK}$	OUT	314	$\overline{CS2}$	C
342	$\overline{CS0}$	OUT	313	$\overline{CS3}$	C
341	$\overline{CS4}$	OUT	312	A0	C
340	$\overline{CS5A}$	OUT	311	A1	C
339	$\overline{CS6A}$	OUT	310	A2	C
338	$\overline{RD}$	OUT	309	A3	C
337	$\overline{BS}$	OUT	308	A4	C
336	D0	OUT	307	A5	C
335	D1	OUT	306	A6	C

297	CS0	Control	265	A2	C
296	$\overline{CS4}$	Control	264	A3	C
295	$\overline{CS5A}$	Control	263	A4	C
294	$\overline{CS6A}$	Control	262	A5	C
293	$\overline{RD}$	Control	261	A6	C
292	$\overline{BS}$	Control	260	A7	C
291	D0	Control	259	A8	C
290	D1	Control	258	A9	C
289	D2	Control	257	A10	C
288	D3	Control	256	A11	C
287	D4	Control	255	A12	C
286	D5	Control	254	D16	IN
285	D6	Control	253	D17	IN
284	D7	Control	252	D18	IN
283	D8	Control	251	D19	IN
282	D9	Control	250	D20	IN
281	D10	Control	249	D21	IN
280	D11	Control	248	D22	IN
279	D12	Control	247	D23	IN
278	D13	Control	246	D24	IN
277	D14	Control	245	D25	IN
276	D15	Control	244	D26	IN
275	$\overline{WE0}$($\overline{BE0}$)/DQMLL	Control	243	D27	IN
274	$\overline{WE1}$($\overline{BE1}$)/DQMLU/$\overline{WE}$	Control	242	D28	IN

233	PTB5/SCIF1CK	IN	202	D29
232	PTB6/$\overline{\text{CTS0}}$	IN	201	D30
231	PTB7/$\overline{\text{RTS0}}$	IN	200	D31
230	PTA0/RXD0	IN	199	A18
229	PTA1/TXD0	IN	198	A19
228	PTA2/SCIF0CK	IN	197	A20
227	PTA3/SCK_SIO0	IN	196	A21
226	PTA4/SIOMCLK0	IN	195	A22
225	PTA5/RXD_SIO0	IN	194	A23
224	PTA6/TXD_SIO0	IN	193	A24
223	PTA7/SIOFSYNC0	IN	192	A25
222	A13	OUT	191	PTB0
221	A14	OUT	190	PTB1/$\overline{\text{CTS1}}$
220	A15	OUT	189	PTB2/$\overline{\text{RTS1}}$
219	A16	OUT	188	PTB3/RXD1
218	A17	OUT	187	PTB4/TXD1
217	$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/$\overline{\text{ICIORD}}$	OUT	186	PTB5/SCIF1CK
216	$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/$\overline{\text{ICIOWR}}$	OUT	185	PTB6/$\overline{\text{CTS0}}$
215	D16	OUT	184	PTB7/$\overline{\text{RTS0}}$
214	D17	OUT	183	PTA0/RXD0
213	D18	OUT	182	PTA1/TXD0
212	D19	OUT	181	PTA2/SCIF0CK
211	D20	OUT	180	PTA3/SCK_SIO0

171	A17	Control	140	PTB4/TXD1	C
170	$\overline{WE2}$($\overline{BE2}$)/DQMUL/$\overline{ICIORD}$	Control	139	PTB5/SCIF1CK	C
169	$\overline{WE3}$($\overline{BE3}$)/DQMUU/$\overline{ICIOWR}$	Control	138	PTB6/$\overline{CTS0}$	C
168	D16	Control	137	PTB7/$\overline{RTS0}$	C
167	D17	Control	136	PTA0/RXD0	C
166	D18	Control	135	PTA1/TXD0	C
165	D19	Control	134	PTA2/SCIF0CK	C
164	D20	Control	133	PTA3/SCK_SIO0	C
163	D21	Control	132	PTA4/SIOMCLK0	C
162	D22	Control	131	PTA5/RXD_SIO0	C
161	D23	Control	130	PTA6/TXD_SIO0	C
160	D24	Control	129	PTA7/SIOFSYNC0	C
159	D25	Control	128	CRS1	IN
158	D26	Control	127	COL1	IN
157	D27	Control	126	TX-CLK1	IN
156	D28	Control	125	RX-ER1	IN
155	D29	Control	124	RX-CLK1	IN
154	D30	Control	123	RX-DV1	IN
153	D31	Control	122	ERXD10	IN
152	A18	Control	121	ERXD11	IN
151	A19	Control	120	ERXD12	IN
150	A20	Control	119	ERXD13	IN
149	A21	Control	118	MDIO1	IN

109	ERXD00	IN	78	ETXD11
108	ERXD01	IN	77	ETXD10
107	ERXD02	IN	76	TX-EN1
106	ERXD03	IN	75	TX-ER1
105	MDIO0	IN	74	MDC1
104	LNKSTA0	IN	73	MDIO1
103	CAMSEN0/IRQ4	IN	72	WOL1
102	MD4	IN	71	EXOUT1
101	MD5	IN	70	ETXD03
100	ETXD13	OUT	69	ETXD02
99	ETXD12	OUT	68	ETXD01
98	ETXD11	OUT	67	ETXD00
97	ETXD10	OUT	66	TX-EN0
96	TX-EN1	OUT	65	TX-ER0
95	TX-ER1	OUT	64	MDC0
94	MDC1	OUT	63	MDIO0
93	MDIO1	OUT	62	WOL0
92	WOL1	OUT	61	EXOUT0
91	EXOUT1/TEND1	OUT	60	NMI
90	ETXD03	OUT	59	IRQ0/$\overline{\text{IRL0}}$
89	ETXD02	OUT	58	IRQ1/$\overline{\text{IRL1}}$
88	ETXD01	OUT	57	IRQ2/$\overline{\text{IRL2}}$
87	ETXD00	OUT	56	IRQ3/$\overline{\text{IRL3}}$

47	PTC6/CE2B	IN	18	AUDCK	C
46	PTC7/$\overline{\text{IOIS16}}$	IN	17	AUDATA3	C
45	MD0	IN	16	AUDATA2	C
44	MD1	IN	15	AUDATA1	C
43	MD2	IN	14	AUDATA0	C
42	MD3	IN	13	STATUS0	C
41	$\overline{\text{ASEBRKAK}}$	OUT	12	STATUS1	C
40	$\overline{\text{AUDSYNC}}$	OUT	11	DACK0	C
39	AUDCK	OUT	10	DACK1	C
38	AUDATA3	OUT	9	PTC0/SCK_SIO1	C
37	AUDATA2	OUT	8	PTC1/SIOMCLK1	C
36	AUDATA1	OUT	7	PTC2/RXD_SIO1	C
35	AUDATA0	OUT	6	PTC3/TXD_SIO1	C
34	STATUS0	OUT	5	PTC4/SIOFSYNC1	C
33	STATUS1	OUT	4	PTC5/$\overline{\text{CE2A}}$	C
32	DACK0	OUT	3	PTC6/$\overline{\text{CE2B}}$	C
31	DACK1	OUT	2	PTC7/$\overline{\text{IOIS16}}$	C
30	PTC0/SCK_SIO1	OUT	1	$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$	C
29	PTC1/SIOMCLK1	OUT	0	$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$	C
28	PTC2/RXD_SIO1	OUT	To TDO		
27	PTC3/TXD_SIO1	OUT			

Note: Control is an active-low signal.
When Control is driven low, the corresponding pin is driven by the value of OUT.

Device ID in this LSI is H'081E200F. Upp
bits may be changed by the chip version

SDIDH corresponds to bits 31 to 16.

SDIDL corresponds to bits 15 to 0.

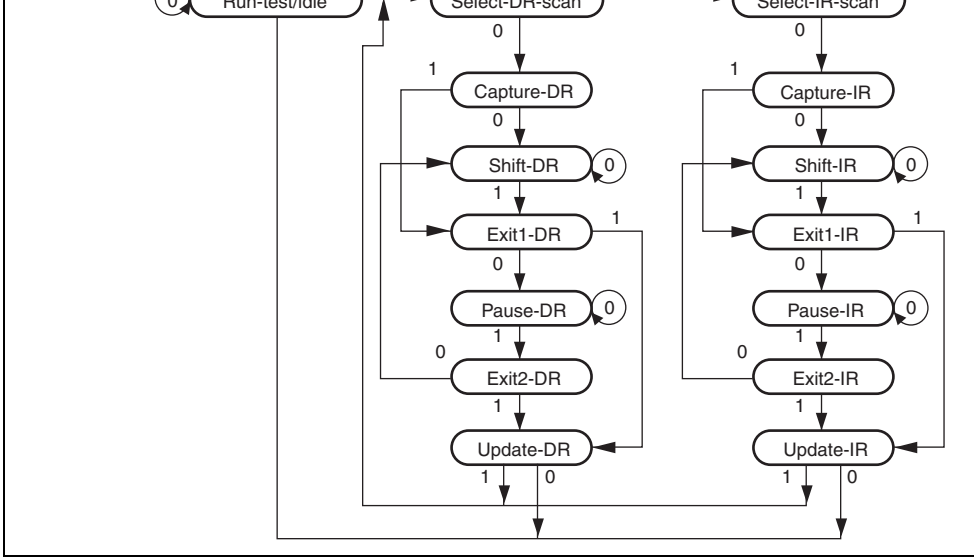


Figure 22.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI val sampled at the rising edge of TCK; shifting occurs at the falling edge of TCK. Fo on change timing of the TDO value, see section 22.4.3, TDO Output Timing. The at high impedance, except with shift-DR and shift-IR states. During the change to 0, there is a transition to test-logic-reset asynchronously with TCK.

		H	In ASE user mode*[3]: Normal reset In ASE break mode*[3]: $\overline{\text{RESETP}}$ ass masked
H		L	H-UDI reset only
		H	Normal operation

Notes: 1. Performs normal mode and ASE mode settings
$\overline{\text{ASEMD0}}$ = H, normal mode
$\overline{\text{ASEMD0}}$ = L, ASE mode
2. In ASE mode, reset hold is enabled by driving the $\overline{\text{RESETP}}$ and $\overline{\text{TRST}}$ pins lo constant cycle. In this state, the CPU does not activate, even if $\overline{\text{RESETP}}$ is dr When $\overline{\text{TRST}}$ is driven high, H-UDI operation is enabled, but the CPU does no The reset hold state is canceled by the following conditions:
- Another $\overline{\text{RESETP}}$ assertion (power-on reset)
- $\overline{\text{TRST}}$ reassertion
3. ASE mode is classified into two modes; ASE break mode to execute the firmw program of an emulator and ASE user mode to execute the user program.
4. Make sure the $\overline{\text{TRST}}$ pin is low when the power is turned on.

22.4.3 TDO Output Timing

The timing of data output from the TDO is switched by the command type set in the SD timing changes at the TCK falling edge when JTAG commands (EXTEST, CLAMP, HI SAMPLE/PRELOAD, IDCODE, and BYPASS) are set. This is a timing of the JTAG st When the H-UDI commands (H-UDI reset negate, H-UDI reset assert, and H-UDI intern set, TDO is output at the TCK rising edge earlier than the JTAG standard by a half cycle

22.4.4 H-UDI Reset

An H-UDI reset is executed by inputting an H-UDI reset assert command in SDIR. An H reset is of the same kind as a power-on reset. An H-UDI reset is released by inputting an reset negate command. The required time between the H-UDI reset assert command and reset negate command is the same as time for keeping the $\overline{\text{RESETP}}$ pin low to apply a po reset.

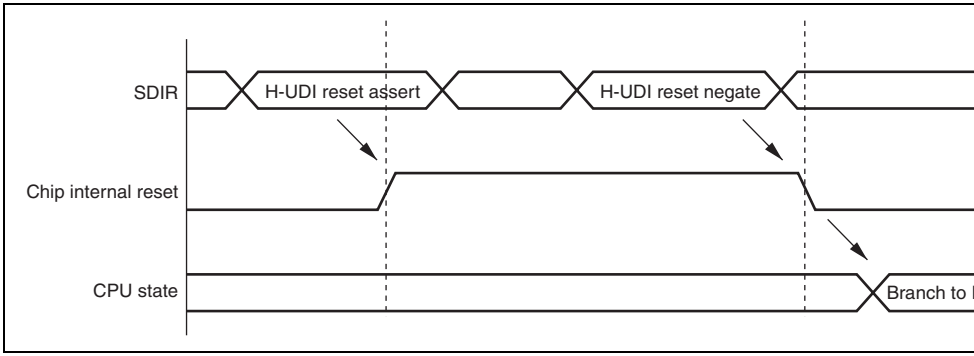


Figure 22.4 H-UDI Reset

22.4.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-U SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in a branc address based on the VBR value plus offset, and with return by the RTE instruction. This request has a fixed priority level of 15.

H-UDI interrupts are accepted in sleep mode.

BYPASS: The BYPASS instruction is an essential standard instruction that operates the register. This instruction shortens the shift path to speed up serial data transfer involving chips on the printed circuit board. While this instruction is executing, the test circuit has on the system circuits. The upper four bits of the instruction code are B'1111.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction inputs values from this L internal circuitry to the boundary scan register, outputs values from the scan path, and lo onto the scan path. When this instruction is executing, this LSI's input pin signals are tra directly to the internal circuitry, and internal circuit values are directly output externally output pins. This LSI's system circuits are not affected by execution of this instruction. four bits of the instruction code are 0100.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the circuitry, or a value to be transferred from the internal circuitry to an output pin, is latch boundary scan register and read from the scan path. Snapshot latching is performed in synchronization with the rise of TCK in the Capture-DR state. Snapshot latching does no normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the bound register from the scan path prior to the EXTEST instruction. Without a PRELOAD oper when the EXTEST instruction was executed an undefined value would be output from th pin until completion of the initial scan sequence (transfer to the output latch) (with the E instruction, the parallel output latch value is constantly output to the output pin).

EXTEST: This instruction is provided to test external circuitry when the this LSI is mo printed circuit board. When this instruction is executed, output pins are used to output te (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan registe printed circuit board, and input pins are used to latch test results into the boundary scan from the printed circuit board. If testing is carried out by using the EXTEST instruction the Nth test data is scanned-in when test data (N-1) is scanned out.

22.5.2 Points for Attention

1. Boundary scan mode does not cover clock-related signals (EXTAL, EXTAL2, XTAL
   XTAL2, CKIO, CKIO2).
2. Boundary scan mode does not cover reset-related signals ($\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$).
3. Boundary scan mode does not cover H-UDI-related signals (TCK, TDI, TDO, TMS, $\overline{\text{T}}$
4. Boundary scan mode does not cover the $\overline{\text{ASEMD0}}$ pin.
5. When the EXTEST, CLAMP, and HIGHZ commands are set, fix the $\overline{\text{RESETP}}$ pin lov
6. When a boundary scan test for other than BYPASS and IDCODE is carried out, fix th
   $\overline{\text{ASEMD0}}$ pin high.

22.6 Usage Notes

1. An H-UDI command, once set, will not be modified as long as another command is n
   issued from the H-UDI. If the same command is given continuously, the command mu
   after a command (BYPASS, etc.) that does not affect chip operations is once set.
2. In standby mode, the H-UDI function cannot be used. To retain the TAP status before
   after standby mode, keep TCK high before entering standby mode.
3. The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be us
   using an emulator.

22.7 Advanced User Debugger (AUD)

The AUD is a function only for an emulator. For details on the AUD, refer to each emula
user's manual.

presumption of a big-endian system.

Register Bits:

- Bit configurations of the registers are described in the same order as the Register Ad
(by functional module, in order of the corresponding section numbers).
- Reserved bits are indicated by — in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a co
for holding data.
- When registers consist of 16 or 32 bits, bits are described from the MSB side.
The order in which bytes are described is on the presumption of a big-endian system

Register States in Each Operating Mode:

- Register states are described in the same order as the Register Addresses (by functio
module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the correspo
section.
- The register states described are for the basic operating modes. If there is a specific
on-chip module, refer to the section on that on-chip module.

INTEVT	Exception handling	L	H'FFFF FFD8	32	32
INTEVT2		L	H'A400 0000	32	32
TRA		L	H'FFFF FFD0	32	32
EXPEVT		L	H'FFFF FFD4	32	32
TEA		L	H'FFFF FFFC	32	32
MMUCR	MMU	L	H'FFFF FFE0	32	32
PTEH		L	H'FFFF FFF0	32	32
PTEL		L	H'FFFF FFF4	32	32
TTB		L	H'FFFF FFF8	32	32
CCR1	Cache	L	H'FFFF FFEC	32	32
CCR2		L	H'A400 00B0	32	32
CCR3		L	H'A400 00B4	32	32
IPRA	INTC	P	H'A414 FEE2	16	16
IPRB		P	H'A414 FEE4	16	16
IPRC		P	H'A414 0016	16	16
IPRD		P	H'A414 0018	16	16
IPRE		P	H'A414 001A	16	16
IPRF		P	H'A408 0000	16	16
IPRG		P	H'A408 0002	16	16
IPRH		P	H'A408 0004	16	16
IPRI		P	H'A408 0006	16	16
ICR0		P	H'A414 FEE0	16	16
ICR1		P	H'A414 0010	16	16

IPRJ		P	H'A408 0020	8	8
BARA	UBC	L	H'A4FF FFB0	32	32
BAMRA		L	H'A4FF FFB4	32	32
BBRA		L	H'A4FF FFB8	16	16
BARB		L	H'A4FF FFA0	32	32
BAMRB		L	H'A4FF FFA4	32	32
BBRB		L	H'A4FF FFA8	16	16
BDRB		L	H'A4FF FF90	32	32
BDMRB		L	H'A4FF FF94	32	32
BRCR		L	H'A4FF FF98	32	32
BETR		L	H'A4FF FF9C	16	16
BRSR		L	H'A4FF FFAC	32	32
BRDR		L	H'A4FF FFBC	32	32
BASRA		L	H'FFFF FFE4	8	8
BASRB		L	H'FFFF FFE8	8	8
STBCR	Power-down mode	P	H'A415 FF82	8	8
STBCR2		P	H'A415 FF88	8	8
STBCR3		P	H'A40A 0000	8	8
FRQCR	CPG	P	H'A415 FF80	16	16
WTCNT		P	H'A415 FF84	8	8/1
WTCSR		P	H'A415 FF86	8	8/1
CMNCR	BSC	I	H'A4FD 0000	32	32
CS0BCR		I	H'A4FD 0004	32	32
CS2BCR		I	H'A4FD 0008	32	32

CS2WCR		I	H'A4FD 0028	32	32
CS3WCR		I	H'A4FD 002C	32	32
CS4WCR		I	H'A4FD 0030	32	32
CS5AWCR		I	H'A4FD 0034	32	32
CS5BWCR		I	H'A4FD 0038	32	32
CS6AWCR		I	H'A4FD 003C	32	32
CS6BWCR		I	H'A4FD 0040	32	32
SDCR		I	H'A4FD 0044	32	32
RTCSR		I	H'A4FD 0048	32	32
RTCNT		I	H'A4FD 004C	32	32
RTCOR		I	H'A4FD 0050	32	32
SDMR2		I	H'A4FD 4xxx	—	16
SDMR3		I	H'A4FD 5xxx	—	16
SAR_0	DMAC	P	H'A401 0020	32	16/3
DAR_0		P	H'A401 0024	32	16/3
DMATCR_0		P	H'A401 0028	32	16/3
CHCR_0		P	H'A401 002C	32	8/16
SAR_1		P	H'A401 0030	32	16/3
DAR_1		P	H'A401 0034	32	16/3
DMATCR_1		P	H'A401 0038	32	16/3
CHCR_1		P	H'A401 003C	32	8/16
SAR_2		P	H'A401 0040	32	16/3
DAR_2		P	H'A401 0044	32	16/3
DMATCR_2		P	H'A401 0048	32	16/3

DMATCR_4		P	H'A401 0078	32	16/
CHCR_4		P	H'A401 007C	32	8/1
SAR_5		P	H'A401 0080	32	16/
DAR_5		P	H'A401 0084	32	16/
DMATCR_5		P	H'A401 0088	32	16/
CHCR_5		P	H'A401 008C	32	8/1
DMAOR		P	H'A401 0060	16	8/1
DMARS0		P	H'A409 0000	16	16
DMARS1		P	H'A409 0004	16	16
DMARS2		P	H'A409 0008	16	16
TSTR	TMU	P	H'A412 FE92	8	8
TCOR0		P	H'A412 FE94	32	32
TCNT0		P	H'A412 FE98	32	32
TCR0		P	H'A412 FE9C	16	16
TCOR1		P	H'A412 FEA0	32	32
TCNT1		P	H'A412 FEA4	32	32
TCR1		P	H'A412 FEA8	16	16
TCOR2		P	H'A412 FEAC	32	32
TCNT2		P	H'A412 FEB0	32	32
TCR2		P	H'A412 FEB4	16	16
R64CNT	RTC	P	H'A413 FEC0	8	8
RSECCNT		P	H'A413 FEC2	8	8
RMINCNT		P	H'A413 FEC4	8	8
RHRCNT		P	H'A413 FEC6	8	8

RWKRAR		P	H'A413 FED6	8	8
RDAYAR		P	H'A413 FED8	8	8
RMONAR		P	H'A413 FEDA	8	8
RCR1		P	H'A413 FEDC	8	8
RCR2		P	H'A413 FEDE	8	8
RYRAR		P	H'A413 FEE0	16	16
RCR3		P	H'A413 FEE4	8	8
SCSMR_0	SCIF	P	H'A440 0000	16	16
SCBRR_0		P	H'A440 0004	8	8
SCSCR_0		P	H'A440 0008	16	16
SCFTDR_0		P	H'A440 000C	8	8
SCFSR_0		P	H'A440 0010	16	16
SCFRDR_0		P	H'A440 0014	8	8
SCFCR_0		P	H'A440 0018	16	16
SCFDR_0		P	H'A440 001C	16	16
SCLSR_0		P	H'A440 0024	16	16
SCSMR_1		P	H'A441 0000	16	16
SCBRR_1		P	H'A441 0004	8	8
SCSCR_1		P	H'A441 0008	16	16
SCFTDR_1		P	H'A441 000C	8	8
SCFSR_1		P	H'A441 0010	16	16
SCFRDR_1		P	H'A441 0014	8	8
SCFCR_1		P	H'A441 0018	16	16
SCFDR_1		P	H'A441 001C	16	16

SIFCTR_0		P	H'A442 0010	16	16
SISTR_0		P	H'A442 0014	16	16
SIIER_0		P	H'A442 0016	16	16
SITDR_0		P	H'A442 0020	32	32
SIRDR_0		P	H'A442 0024	32	32
SITCR_0		P	H'A442 0028	32	32
SIRCR_0		P	H'A442 002C	32	32
SIMDR_1		P	H'A443 0000	16	16
SISCR_1		P	H'A443 0002	16	16
SITDAR_1		P	H'A443 0004	16	16
SIRDAR_1		P	H'A443 0006	16	16
SICDAR_1		P	H'A443 0008	16	16
SICTR_1		P	H'A443 000C	16	16
SIFCTR_1		P	H'A443 0010	16	16
SISTR_1		P	H'A443 0014	16	16
SIIER_1		P	H'A443 0016	16	16
SITDR_1		P	H'A443 0020	32	32
SIRDR_1		P	H'A443 0024	32	32
SITCR_1		P	H'A443 0028	32	32
SIRCR_1		P	H'A443 002C	32	32
ECMR0	EtherC (MAC-0)	I	H'A700 0160	32	32
ECSR0		I	H'A700 0164	32	32
ECSIPR0		I	H'A700 0168	32	32
PIR0		I	H'A700 016C	32	32

CNDCR0		I	H'A700 018C	32	32
CEFCR0		I	H'A700 0194	32	32
FRECR0		I	H'A700 0198	32	32
TSFRCR0		I	H'A700 019C	32	32
TLFRCR0		I	H'A700 01A0	32	32
RFCR0		I	H'A700 01A4	32	32
MAFCR0		I	H'A700 01A8	32	32
IPGR0		I	H'A700 01B4	32	32
ECMR1	EtherC (MAC-1)	I	H'A700 0560	32	32
ECSR1		I	H'A700 0564	32	32
ECSIPR1		I	H'A700 0568	32	32
PIR1		I	H'A700 056C	32	32
MAHR1		I	H'A700 0570	32	32
MALR1		I	H'A700 0574	32	32
RFLR1		I	H'A700 0578	32	32
PSR1		I	H'A700 057C	32	32
TROCR1		I	H'A700 0580	32	32
CDCR1		I	H'A700 0584	32	32
LCCR1		I	H'A700 0588	32	32
CNDCR1		I	H'A700 058C	32	32
CEFCR1		I	H'A700 0594	32	32
FRECR1		I	H'A700 0598	32	32
TSFRCR1		I	H'A700 059C	32	32

TSU_FWEN1	I	H'A700 0814	32	32
TSU_FCM	I	H'A700 0818	32	32
TSU_BSYSL0	I	H'A700 0820	32	32
TSU_BSYSL1	I	H'A700 0824	32	32
TSU_PRISL0	I	H'A700 0828	32	32
TSU_PRISL1	I	H'A700 082C	32	32
TSU_FWSL0	I	H'A700 0830	32	32
TSU_FWSL1	I	H'A700 0834	32	32
TSU_FWSLC	I	H'A700 0838	32	32
TSU_QTAGM0	I	H'A700 0840	32	32
TSU_QTAGM1	I	H'A700 0844	32	32
TSU_ADQT0	I	H'A700 0848	32	32
TSU_ADQT1	I	H'A700 084C	32	32
TSU_FWSR	I	H'A700 0850	32	32
TSU_FWINMK	I	H'A700 0854	32	32
TSU_ADSBSY	I	H'A700 0860	32	32
TSU_TEN	I	H'A700 0864	32	32
TSU_POST1	I	H'A700 0870	32	32
TSU_POST2	I	H'A700 0874	32	32
TSU_POST3	I	H'A700 0878	32	32
TSU_POST4	I	H'A700 087C	32	32
TXNLCR0	I	H'A700 0880	32	32
TXALCR0	I	H'A700 0884	32	32

FWNLCR1	I	H'A700 08B0	32	32
FWALCR1	I	H'A700 08B4	32	32
TSU_ADRH0	I	H'A700 0900	32	32
TSU_ADRL0	I	H'A700 0904	32	32
TSU_ADRH1	I	H'A700 0908	32	32
TSU_ADRL1	I	H'A700 090C	32	32
TSU_ADRH2	I	H'A700 0910	32	32
TSU_ADRL2	I	H'A700 0914	32	32
TSU_ADRH3	I	H'A700 0918	32	32
TSU_ADRL3	I	H'A700 091C	32	32
TSU_ADRH4	I	H'A700 0920	32	32
TSU_ADRL4	I	H'A700 0924	32	32
TSU_ADRH5	I	H'A700 0928	32	32
TSU_ADRL5	I	H'A700 092C	32	32
TSU_ADRH6	I	H'A700 0930	32	32
TSU_ADRL6	I	H'A700 0934	32	32
TSU_ADRH7	I	H'A700 0938	32	32
TSU_ADRL7	I	H'A700 093C	32	32
TSU_ADRH8	I	H'A700 0940	32	32
TSU_ADRL8	I	H'A700 0944	32	32
TSU_ADRH9	I	H'A700 0948	32	32
TSU_ADRL9	I	H'A700 094C	32	32
TSU_ADRH10	I	H'A700 0950	32	32

TSU_ADRL14	I	H'A700 0974	32	32
TSU_ADRH15	I	H'A700 0978	32	32
TSU_ADRL15	I	H'A700 097C	32	32
TSU_ADRH16	I	H'A700 0980	32	32
TSU_ADRL16	I	H'A700 0984	32	32
TSU_ADRH17	I	H'A700 0988	32	32
TSU_ADRL17	I	H'A700 098C	32	32
TSU_ADRH18	I	H'A700 0990	32	32
TSU_ADRL18	I	H'A700 0994	32	32
TSU_ADRH19	I	H'A700 0998	32	32
TSU_ADRL19	I	H'A700 099C	32	32
TSU_ADRH20	I	H'A700 09A0	32	32
TSU_ADRL20	I	H'A700 09A4	32	32
TSU_ADRH21	I	H'A700 09A8	32	32
TSU_ADRL21	I	H'A700 09AC	32	32
TSU_ADRH22	I	H'A700 09B0	32	32
TSU_ADRL22	I	H'A700 09B4	32	32
TSU_ADRH23	I	H'A700 09B8	32	32
TSU_ADRL23	I	H'A700 09BC	32	32
TSU_ADRH24	I	H'A700 09C0	32	32
TSU_ADRL24	I	H'A700 09C4	32	32
TSU_ADRH25	I	H'A700 09C8	32	32
TSU_ADRL25	I	H'A700 09CC	32	32

TSU_ADRL29		I	H'A700 09EC	32	32
TSU_ADRH30		I	H'A700 09F0	32	32
TSU_ADRL30		I	H'A700 09F4	32	32
TSU_ADRH31		I	H'A700 09F8	32	32
TSU_ADRL31		I	H'A700 09FC	32	32
EDMR0	E-DMAC0	I	H'A700 0000	32	32
EDTRR0		I	H'A700 0004	32	32
EDRRR0		I	H'A700 0008	32	32
TDLAR0		I	H'A700 000C	32	32
RDLAR0		I	H'A700 0010	32	32
EESR0		I	H'A700 0014	32	32
EESIPR0		I	H'A700 0018	32	32
TRSCER0		I	H'A700 001C	32	32
RMFCR0		I	H'A700 0020	32	32
TFTR0		I	H'A700 0024	32	32
FDR0		I	H'A700 0028	32	32
RMCR0		I	H'A700 002C	32	32
EDOCR0		I	H'A700 0030	32	32
FCFTR0		I	H'A700 0034	32	32
TRIMD0		I	H'A700 003C	32	32
RBWAR0		I	H'A700 0040	32	32
RDFAR0		I	H'A700 0044	32	32
TBRAR0		I	H'A700 004C	32	32
TDFAR0		I	H'A700 0050	32	32

TRSCER1		I	H'A700 041C	32	32
RMFCR1		I	H'A700 0420	32	32
TFTR1		I	H'A700 0424	32	32
FDR1		I	H'A700 0428	32	32
RMCR1		I	H'A700 042C	32	32
EDOCR1		I	H'A700 0430	32	32
FCFTR1		I	H'A700 0434	32	32
TRIMD1		I	H'A700 043C	32	32
RBWAR1		I	H'A700 0440	32	32
RDFAR1		I	H'A700 0444	32	32
TBRAR1		I	H'A700 044C	32	32
TDFAR1		I	H'A700 0450	32	32
PACR	PFC	P	H'A405 0100	16	16
PBCR		P	H'A405 0102	16	16
PCCR		P	H'A405 0104	16	16
PETCR		P	H'A405 0106	16	16
PADR	I/O port	P	H'A405 0120	8	8
PBDR		P	H'A405 0122	8	8
PCDR		P	H'A405 0124	8	8

BSC: Bus State Controller
DMAC: Direct Memory Access Controller
TMU: Timer Unit
RTC: Realtime Clock
SCIF0: Serial Communication Interface with FIFO 0
SCIF1: Serial Communication Interface with FIFO 1
SIOF0: Serial I/O with FIFO 0
SIOF1: Serial I/O with FIFO 1
EtherC (MAC-0): Ethernet Controller 0
EtherC (MAC-1): Ethernet Controller 1
EtherC (TSU): Transfer Unit for Ethernet Controller
E-DMAC0: Ethernet Controller Direct Memory Access Controller 0
E-DMAC1: Ethernet Controller Direct Memory Access Controller 1
PFC: Pin Function Controller
H-UDI: User Debugging Interface

2. Bus:
   L: Connected to the CPU, DSP, CCN, Cache, MMU, and UBC.
   I: Connected to the BSC, CCN, Cache, DMAC, E-DMAC0, and E-DMAC1
   P: Connected to the BSC and peripheral modules (RTC, TMU, SCIF0, SCI SIOF0, SIOF1, DMAC, PORT, INTC, H-UDI, CPG).
3. The access size indicates the size when accessing (read/write) the control reg an access is performed in the different size as shown above, the result is not c data.
4. 16 bits when writing and 8 bits when reading.

	—	—	—	—	—	—	TRA	TRA
	TRA	TRA	TRA	TRA	TRA	TRA	—	—
EXPEVT	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	EXPEVT	EXPEVT	EXPEVT	EXPEVT
	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT
INTEVT	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	INTEVT	INTEVT	INTEVT	INTEVT
	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT
INTEVT2	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	INTEVT2	INTEVT2	INTEVT2	INTEVT2
	INTEVT2	INTEVT2	INTEVT2	INTEVT2	INTEVT2	INTEVT2	—	—
TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA
	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA
	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA
	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA
PTEH	VPN	VPN	VPN	VPN	VPN	VPN	VPN	VPN
	VPN	VPN	VPN	VPN	VPN	VPN	VPN	VPN
	VPN	VPN	VPN	VPN	VPN	VPN	—	—
	ASID	ASID	ASID	ASID	ASID	ASID	ASID	ASID

MMUCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	SV
	—	—	RC	RC	—	TF	IX	AT
CCR1	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	CF	CB	WT	CE
CCR2	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	LE
	—	—	—	—	—	—	W3LOAD	W3LOCK
	—	—	—	—	—	—	W2LOAD	W2LOCK
CCR3	—	—	—	—	—	—	—	—
	CSIZE7	CSIZE6	CSIZE5	CSIZE4	CSIZE3	CSIZE2	CSIZE1	CSIZE0
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
IPRA	TMU0				TMU1			
	TMU2				RTC			
IPRB	WDT				REF			
	—	—	—	—	—	—	—	—
IPRC	IRQ3				IRQ2			
	IRQ1				IRQ0			

		E-DMAC(3)			—	—	—	—
IPRH	—	—	—	—	—	—	—	—
	—	—	—	—		SIOF0		
IPRI	—	—	—	—	—	—	—	—
		SIOF1			—	—	—	—
ICR0	NMIL	—	—	—	—	—	—	NMIE
	—	—	—	—	—	—	—	—
ICR1	MAI	IRQLVL	BLMASK	IRLSEN	IRQ51S	IRQ50S	IRQ41S	IRQ40S
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
IRR0	—	—	IRQ5R	IRQ4R	IRQ3R	IRQ2R	IRQ1R	IRQ0R
IRR1	TXI0R	BRI0R	RXI0R	ERI0R	DEI3R	DEI2R	DEI1R	DEI0R
IRR2	—	—	—	—	TXI1R	BRI1R	RXI1R	ERI1R
IRR3	—	—	—	—	—	CUIR	PRIR	ATIR
IRR4	—	TUNI2R	TUNI1R	TUNI0R	ITIR	—	—	RCMIR
IRR5	—	—	DEI5R	DEI4R	—	EINT2R	EINT1R	EINT0R
IRR7	CCI0R	RXI0R	TXI0R	ERI0R	—	—	—	—
IRR8	CCI1R	RXI1R	TXI1R	ERI1R	—	—	—	—
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0

	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0
BBRB	—	—	—	—	—	—	XYE	XYS
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0
BRCR	—	—	—	—	—	—	—	—
	—	—	BASMA	BASMB	—	—	—	—
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—
	DBEB	PCBB	—	—	SEQ	—	—	ETBE
BETR	—	—	—	—	BET11	BET10	BET9	BET8
	BET7	BET6	BET5	BET4	BET3	BET2	BET1	BET0

	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0
BASRA	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA1	BASA0
BASRB	BASB7	BASB6	BASB5	BASB4	BASB3	BASB2	BASB1	BASB0
STBCR	STBY	—	—	—	—	MSTP2	MSTP1	—
STBCR2	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	—	MSTP3
STBCR3	—	—	—	—	MSTP33	MEST32	MSTP31	MSTP30
FRQCR	—	—	—	CKOEN	—	—	STC1	STC0
	—	—	IFC1	IFC0	—	PFC2	PFC1	PFC0
WTCNT								
WTCSR	TME	WT/IT	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0
CMNCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	WAITSEL	BSD	—	MAP	BLOCK	DPRTY1	DPRTY0	DMAIW
	DMAIW1	DMAIW0	DMAIWA	—	ENDIAN	CK2DRV	HIZMEM	HIZCNT
CSnBCR (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B)	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—

						W3	W2	W1
	WR0	WM	—	—	—	—	HW1	HW0
	W0	WM	—	—	—	—	—	—
	W0	WM	—	—	—	—	—	—
CS2WCR*[2]	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	BAS	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	WR3	WR2	WR1
	—	—	—	—	—	—	—	A2CL1
	WR0	WM	—	—	—	—	—	—
	A2CL0	—	—	—	—	—	—	—
CS3WCR*[2]	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	BAS	—	—	—	—
	—	—	—	—	—	—	—	—
CS3WCR*[2]	—	—	—	—	—	WR3	WR2	WR1
	—	TRP1	TRP0	—	TRCD1	TRCD0	—	A3CL1
	WR0	WM	—	—	—	—	—	—
	A3CL0	—	—	TRWL1	TRWL0	—	TRC1	TRC0

CS5A WCR*[4]	—	—	—	—	—	—	—	—
	—	—	—	—	—	WW2	WW1	WW0
	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0
CS5BWCR*[5]	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	BAS	—	WW2	WW1	WW0
	—	—	SA1	SA0	—	—	—	—
	—	—	—	SW1	SW0	WR3	WR2	WR1
	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1
	WR0	WM	—	—	—	—	HW1	HW0
	PCW0	WM	—	—	TEH3	TEH2	TEH1	TEH0
CS6AWCR*[4]	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0
CS6BWCR*[6]	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	BAS	—	—	—	—
	—	—	SA1	SA0	—	—	—	—
	—	—	—	SW1	SW0	WR3	WR2	WR1
	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1
	WR0	WM	—	—	—	—	HW1	HW0
	PCW0	WM	—	—	TEH3	TEH2	TEH1	TEH0

	CMF	CMIE	CKS2	CKS1	CKS0	RRC2	RRC1	RRC0
RTCNT	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
RTCOR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
SAR_n (n = 0 to 5)								
DAR_n (n = 0 to 5)								
DMATCR_n (n = 0 to 5)								

	—	—	TB	TS1	TS0	IE	TE	DE
DMAOR	—	—	—	—	—	—	PR1	PR0
	—	—	—	—	—	AE	NMIF	DME
DMARS0	C1MID5	C1MID4	C1MID3	C1MID2	C1MID1	C1MID0	C1RID1	C1RID0
	C0MID5	C0MID4	C0MID3	C0MID2	C0MID1	C0MID0	C0RID1	C0RID0
DMARS1	C3MID5	C3MID4	C3MID3	C3MID2	C3MID1	C3MID0	C3RID1	C3RID0
	C2MID5	C2MID4	C2MID3	C2MID2	C2MID1	C2MID0	C2RID1	C2RID0
DMARS2	C5MID5	C5MID4	C5MID3	C5MID2	C5MID1	C5MID0	C5RID1	C5RID0
	C4MID5	C4MID4	C4MID3	C4MID2	C4MID1	C4MID0	C4RID1	C4RID0
TSTR	—	—	—	—	—	STR2	STR1	STR0
TCRn (n = 0 to 2)	—	—	—	—	—	—	—	UNF
	—	—	UNIE	—	—	TPSC2	TPSC1	TPSC0
TCORn (n = 0 to 2)								
TCNTn (n = 0 to 2)								

				month				
RYRCNT	1000-unit of year				100-unit of year			
	10-unit of year				1-unit of year			
RSECAR	ENB	10-unit of second			1-unit of second			
RMINAR	ENB	10-unit of minute			1-unit of minute			
RHRAR	ENB	—	10-unit of hour		1-unit of hour			
RWKAR	ENB	—	—	—	—		Day of week code	
RDAYAR	ENB	—	10-unit of date		1-unit of date			
RMONAR	ENB	—	—	10-unit of month	1-unit of month			
RYRAR	1000-unit of year				100-unit of year			
	10-unit of year				1-unit of year			
RCR1	CF	—	—	CIE	AIE	—	—	AF
RCR2	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START
RCR3	YAEN	—	—	—	—	—	—	—
SCFRDR_n (n = 0, 1)								
SCFTDR_n (n = 0, 1)								
SCSMR_n (n = 0, 1)	—	—	—	—	—	—	—	—
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0
SCSCR_n (n = 0, 1)	—	—	—	—	—	—	—	—
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0

(n = 0, 1)	—	—	—	R4	R3	R2	R1	R0
SCLSR_n (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	ORER
SIMDR_n (n = 0, 1)	TRMD1	TRMD0	—	REDG	FL3	FL2	FL1	FL0
	TXDIZ	LSBF	RCIM	—	—	—	—	—
SISCR_n (n = 0, 1)	MSSEL	MSIMM	—	BRPS4	BRPS3	BRPS2	BRPS1	BRPS0
	—	—	—	—	—	BRDV2	BRDV1	BRDV0
SITDAR_n (n = 0, 1)	TDLE	—	—	—	TDLA3	TDLA2	TDLA1	TDLA0
	TDRE	TLREP	—	—	TDRA3	TDRA2	TDRA1	TDRA0
SIRDAR_n (n = 0, 1)	RDLE	—	—	—	RDLA3	RDLA2	RDLA1	RDLA0
	RDRE	—	—	—	RDRA3	RDRA2	RDRA1	RDRA0
SICDAR_n (n = 0, 1)	CD0E	—	—	—	CD0A3	CD0A2	CD0A1	CD0A0
	CD1E	—	—	—	CD1A3	CD1A2	CD1A1	CD1A0
SICTR_n (n = 0, 1)	SCKE	FSE	—	—	—	—	TXE	RXE
	—	—	—	—	—	—	TXRST	RXRST
SIFCTR_n (n = 0, 1)	TFWM2	TFWM1	TFWM0	TFUA4	TFUA3	TFUA2	TFUA1	TFUA0
	RFWM2	RFWM1	RFWM0	RFUA4	RFUA3	RFUA2	RFUA1	RFUA0
SISTR_n (n = 0, 1)	—	TCRDY	TFEMP	TDREQ	—	RCRDY	RFFUL	RDREQ
	—	—	—	FSERR	TFOVR	TFUDR	RFUDR	RFOVR
SIIER_n (n = 0, 1)	—	TCRDYE	TFEMPE	TDREQE	—	RCRDYE	RFFULE	RDREQE
	—	—	—	FSERRE	TFOVRE	TFUDRE	RFUDRE	RFOVRE

	SIRDR7	SIRDR6	SIRDR5	SIRDR4	SIRDR3	SIRDR2	SIRDR1	SIRDR0
SITCR_n (n = 0, 1)	SITC015	SITC014	SITC013	SITC012	SITC011	SITC010	SITC09	SITC08
	SITC07	SITC06	SITC05	SITC04	SITC03	SITC02	SITC01	SITC00
	SITC115	SITC114	SITC113	SITC112	SITC111	SITC110	SITC19	SITC18
	SITC17	SITC16	SITC15	SITC14	SITC13	SITC12	SITC11	SITC10
SIRCR_n (n = 0, 1)	SIRC015	SIRC014	SIRC013	SIRC012	SIRC011	SIRC010	SIRC09	SIRC08
	SIRC07	SIRC06	SIRC05	SIRC04	SIRC03	SIRC02	SIRC01	SIRC00
	SIRC115	SIRC114	SIRC113	SIRC112	SIRC111	SIRC110	SIRC19	SIRC18
	SIRC17	SIRC16	SIRC15	SIRC14	SIRC13	SIRC12	SIRC11	SIRC10
ARSTR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	ARST
ECMRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	MCT	PRCEF	—	—	MPDE	—
	—	RE	TE	—	ILB	ELB	DM	PRM
ECSRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	LCHNG	MPD	ICD

	—	—	—	—	MDI	MDO	MMD	MDC
MAHRn (n = 0, 1)	MA47	MA46	MA45	MA44	MA43	MA42	MA41	MA40
	MA39	MA38	MA37	MA36	MA35	MA34	MA33	MA32
	MA31	MA30	MA29	MA28	MA27	MA26	MA25	MA24
	MA23	MA22	MA21	MA20	MA19	MA18	MA17	MA16
MALRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8
	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
RFLRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	RFL11	RFL10	RFL9	RFL8
	RFL7	RFL6	RFL5	RFL4	RFL3	RFL2	RFL1	RFL0
PSRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	LMON
TROCRn (n = 0, 1)	TROC31	TROC30	TROC29	TROC28	TROC27	TROC26	TROC25	TROC24
	TROC23	TROC22	TROC21	TROC20	TROC19	TROC18	TROC17	TROC16
	TROC15	TROC14	TROC13	TROC12	TROC11	TROC10	TROC9	TROC8
	TROC7	TROC6	TROC5	TROC4	TROC3	TROC2	TROC1	TROC0

	LCC7	LCC6	LCC5	LCC4	LCC3	LCC2	LCC1	LCC0
CNDCRn (n = 0, 1)	CNDC31	CNDC30	CNDC29	CNDC28	CNDC27	CNDC26	CNDC25	CNDC24
	CNDC23	CNDC22	CNDC21	CNDC20	CNDC19	CNDC18	CNDC17	CNDC16
	CNDC15	CNDC14	CNDC13	CNDC12	CNDC11	CNDC10	CNDC9	CNDC8
	CNDC7	CNDC6	CNDC5	CNDC4	CNDC3	CNDC2	CNDC1	CNDC0
CEFCRn (n = 0, 1)	CEFC31	CEFC30	CEFC29	CEFC28	CEFC27	CEFC26	CEFC25	CEFC24
	CEFC23	CEFC22	CEFC21	CEFC20	CEFC19	CEFC18	CEFC17	CEFC16
	CEFC15	CEFC14	CEFC13	CEFC12	CEFC11	CEFC10	CEFC9	CEFC8
	CEFC7	CEFC6	CEFC5	CEFC4	CEFC3	CEFC2	CEFC1	CEFC0
FRECRn (n = 0, 1)	FREC31	FREC30	FREC29	FREC28	FREC27	FREC26	FREC25	FREC24
	FREC23	FREC22	FREC21	FREC20	FREC19	FREC18	FREC17	FREC16
	FREC15	FREC14	FREC13	FREC12	FREC11	FREC10	FREC9	FREC8
	FREC7	FREC6	FREC5	FREC4	FREC3	FREC2	FREC1	FREC0
TSFRCRn (n = 0, 1)	TSFC31	TSFC30	TSFC29	TSFC28	TSFC27	TSFC26	TSFC25	TSFC24
	TSFC23	TSFC22	TSFC21	TSFC20	TSFC19	TSFC18	TSFC17	TSFC16
	TSFC15	TSFC14	TSFC13	TSFC12	TSFC11	TSFC10	TSFC9	TSFC8
	TSFC7	TSFC6	TSFC5	TSFC4	TSFC3	TSFC2	TSFC1	TSFC0
TLFRCRn (n = 0, 1)	TLFC31	TLFC30	TLFC29	TLFC28	TLFC27	TLFC26	TLFC25	TLFC24
	TLFC23	TLFC22	TLFC21	TLFC20	TLFC19	TLFC18	TLFC17	TLFC16
	TLFC15	TLFC14	TLFC13	TLFC12	TLFC11	TLFC10	TLFC9	TLFC8
	TLFC7	TLFC6	TLFC5	TLFC4	TLFC3	TLFC2	TLFC1	TLFC0

	MAFC7	MAFC6	MAFC5	MAFC4	MAFC3	MAFC2	MAFC1	MAFC0
IPGRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	IPG4	IPG3	IPG2	IPG1	IPG0
TSU_ CTRST	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	CTRST
	—	—	—	—	—	—	—	—
TSU_ FWEN0	FWEN0	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
TSU_ FWEN1	FWEN1	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
TSU_FCM	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	FCM2	FCM1	FCM0

	—	—	BSYSL15	BSYSL14	BSYSL13	BSYSL12	BSYSL11	BSYSL10
TSU_ PRISL0	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	PRIMD02	PRIMD01	PRIMD00	—	—	—	—
	PRISL07	PRISL06	PRISL05	PRISL04	PRISL03	PRISL02	PRISL01	PRISL00
TSU_ PRISL1	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	PRIMD12	PRIMD11	PRIMD10	—	—	—	—
	PRISL17	PRISL16	PRISL15	PRISL14	PRISL13	PRISL12	PRISL11	PRISL10
TSU_ FWSL0	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	FW40	FW30	FW20	FW10
	—	—	—	—	—	—	—	—
TSU_ FWSL1	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	FW41	FW31	FW21	FW11
	—	—	—	—	—	—	—	—
TSU_ FWSLC	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	POSTENU	POSTENL	—	—	—	—
	CAMSEL03	CAMSEL02	CAMSEL01	CAMSEL00	CAMSEL13	CAMSEL12	CAMSEL11	CAMSEL10

	—	—	—	—	—	—	QTAGM11	QTAGM10
TSU_ FWSR	—	—	—	—	TINT40	TINT30	TINT20	TINT10
	OVF0	RBSY0	—	RINT50	RINT40	RINT30	RINT20	RINT10
	—	—	—	—	TINT41	TINT31	TINT21	TINT11
	OVF1	RBSY1	—	RINT51	RINT41	RINT31	RINT21	RINT11
TSU_ FWINMK	—	—	—	—	TINTM40	TINTM30	TINTM20	TINTM10
	OVFM0	RBSYM0	—	RINTM50	RINTM40	RINTM30	RINTM20	RINTM10
	—	—	—	—	TINTM41	TINTM31	TINTM21	TINTM11
	OVFM1	RBSYM1	—	RINTM51	RINTM41	RINTM31	RINTM21	RINTM11
TSU_ ADQT0	QTAG031	QTAG030	QTAG029	QTAG028	QTAG027	QTAG026	QTAG025	QTAG024
	QTAG023	QTAG022	QTAG021	QTAG020	QTAG019	QTAG018	QTAG017	QTAG016
	QTAG015	QTAG014	QTAG013	—	QTAG011	QTAG010	QTAG009	QTAG008
	QTAG007	QTAG006	QTAG005	QTAG004	QTAG003	QTAG002	QTAG001	QTAG000
TSU_ ADQT1	QTAG131	QTAG130	QTAG129	QTAG128	QTAG127	QTAG126	QTAG125	QTAG124
	QTAG123	QTAG122	QTAG121	QTAG120	QTAG119	QTAG118	QTAG117	QTAG116
	QTAG115	QTAG114	QTAG113	—	QTAG111	QTAG110	QTAG109	QTAG108
	QTAG107	QTAG106	QTAG105	QTAG104	QTAG103	QTAG102	QTAG101	QTAG100
TSU_ ADSBSY	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	ADSBSY

	POST63	POST62	POST61	POST60	POST73	POST72	POST71	POST70
TSU_ POST2	POST83	POST82	POST81	POST80	POST93	POST92	POST91	POST90
	POST103	POST102	POST101	POST100	POST113	POST112	POST111	POST110
	POST123	POST122	POST121	POST120	POST133	POST132	POST131	POST130
	POST143	POST142	POST141	POST140	POST153	POST152	POST151	POST150
TSU_ POST3	POST163	POST162	POST161	POST160	POST173	POST172	POST171	POST170
	POST183	POST182	POST181	POST180	POST193	POST192	POST191	POST190
	POST203	POST202	POST201	POST200	POST213	POST212	POST211	POST210
	POST223	POST222	POST221	POST220	POST233	POST232	POST231	POST230
TSU_ POST4	POST243	POST242	POST241	POST240	POST253	POST252	POST251	POST250
	POST263	POST262	POST261	POST260	POST273	POST272	POST271	POST270
	POST283	POST282	POST281	POST280	POST293	POST292	POST291	POST290
	POST303	POST302	POST301	POST300	POST313	POST312	POST311	POST310
TSU_ ADRHn (n = 0 to 31)	ADRHn31	ADRHn30	ADRHn29	ADRHn28	ADRHn27	ADRHn26	ADRHn25	ADRHn24
	ADRHn23	ADRHn22	ADRHn21	ADRHn20	ADRHn19	ADRHn18	ADRHn17	ADRHn16
	ADRHn15	ADRHn14	ADRHn13	ADRHn12	ADRHn11	ADRHn10	ADRHn9	ADRHn8
	ADRHn7	ADRHn6	ADRHn5	ADRHn4	ADRHn3	ADRHn2	ADRHn1	ADRHn0
TSU_ ADRLn (n = 0 to 31)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	ADRLn15	ADRLn14	ADRLn13	ADRLn12	ADRLn11	ADRLn10	ADRLn9	ADRLn8
	ADRLn7	ADRLn6	ADRLn5	ADRLn4	ADRLn3	ADRLn2	ADRLn1	ADRLn0

	TC007	TC006	TC005	TC004	TC003	TC002	TC001	TC000
RXNLCR0	NRC031	NRC030	NRC029	NRC028	NRC027	NRC026	NRC025	NRC024
	NRC023	NRC022	NRC021	NRC020	NRC019	NRC018	NRC017	NRC016
	NRC015	NRC014	NRC013	NRC012	NRC011	NRC010	NRC009	NRC008
	NRC007	NRC006	NRC005	NRC004	NRC003	NRC002	NRC001	NRC000
RXALCR0	RC031	RC030	RC029	RC028	RC027	RC026	RC025	RC024
	RC023	RC022	RC021	RC020	RC019	RC018	RC017	RC016
	RC015	RC014	RC013	RC012	RC011	RC010	RC009	RC008
	RC007	RC006	RC005	RC004	RC003	RC002	RC001	RC000
FWNLCR0	NFC031	NFC030	NFC029	NFC028	NFC027	NFC026	NFC025	NFC024
	NFC023	NFC022	NFC021	NFC020	NFC019	NFC018	NFC017	NFC016
	NFC015	NFC014	NFC013	NFC012	NFC011	NFC010	NFC009	NFC008
	NFC007	NFC006	NFC005	NFC004	NFC003	NFC002	NFC001	NFC000
FWALCR0	FC031	FC030	FC029	FC028	FC027	FC026	FC025	FC024
	FC023	FC022	FC021	FC020	FC019	FC018	FC017	FC016
	FC015	FC014	FC013	FC012	FC011	FC010	FC009	FC008
	FC007	FC006	FC005	FC004	FC003	FC002	FC001	FC000
TXNLCR1	NTC131	NTC130	NTC129	NTC128	NTC127	NTC126	NTC125	NTC124
	NTC123	NTC122	NTC121	NTC120	NTC119	NTC118	NTC117	NTC116
	NTC115	NTC114	NTC113	NTC112	NTC111	NTC110	NTC109	NTC108
	NTC107	NTC106	NTC105	NTC104	NTC103	NTC102	NTC101	NTC100

	NRC107	NRC106	NRC105	NRC104	NRC103	NRC102	NRC101	NRC100
RXALCR1	RC131	RC130	RC129	RC128	RC127	RC126	RC125	RC124
	RC123	RC122	RC121	RC120	RC119	RC118	RC117	RC116
	RC115	RC114	RC113	RC112	RC111	RC110	RC109	RC108
	RC107	RC106	RC105	RC104	RC103	RC102	RC101	RC100
FWNLCR1	NFC131	NFC130	NFC129	NFC128	NFC127	NFC126	NFC125	NFC124
	NFC123	NFC122	NFC121	NFC120	NFC119	NFC118	NFC117	NFC116
	NFC115	NFC114	NFC113	NFC112	NFC111	NFC110	NFC109	NFC108
	NFC107	NFC106	NFC105	NFC104	NFC103	NFC102	NFC101	NFC100
FWALCR1	FC131	FC130	FC129	FC128	FC127	FC126	FC125	FC124
	FC123	FC122	FC121	FC120	FC119	FC118	FC117	FC116
	FC115	FC114	FC113	FC112	FC111	FC110	FC109	FC108
	FC107	FC106	FC105	FC104	FC103	FC102	FC101	FC100
EDMRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	DL1	DL0	—	—	—	SWR
EDTRRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	TR

	TDLA7	TDLA6	TDLA5	TDLA4	TDLA3	TDLA2	TDLA1	TDLA0
RDLARn (n = 0, 1)	RDLA31	RDLA30	RDLA29	RDLA28	RDLA27	RDLA26	RDLA25	RDLA24
	RDLA23	RDLA22	RDLA21	RDLA20	RDLA19	RDLA18	RDLA17	RDLA16
	RDLA15	RDLA14	RDLA13	RDLA12	RDLA11	RDLA10	RDLA9	RDLA8
	RDLA7	RDLA6	RDLA5	RDLA4	RDLA3	RDLA2	RDLA1	RDLA0
EESRn (n = 0, 1)	—	TWB	—	—	—	TABT	RABT	RFCOF
	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
	—	—	—	—	CND	DLC	CD	TRO
	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
EESIPRn (n = 0, 1)	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP
	ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP
	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
TRSCERn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	RMAFCE	—	—	—	—	—	—	—
RMFCRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	MFC15	MFC14	MFC13	MFC12	MFC11	MFC10	MFC9	MFC8
	MFC7	MFC6	MFC5	MFC4	MFC3	MFC2	MFC1	MFC0

	—	—	—	—	—	RFD2	RFD1	RFD0
RMCRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	RNC
EDOCRn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	FEC	AEC	—	—
RBWARn (n = 0, 1)	RBWA31	RBWA30	RBWA29	RBWA28	RBWA27	RBWA26	RBWA25	RBWA24
	RBWA23	RBWA22	RBWA21	RBWA20	RBWA19	RBWA18	RBWA17	RBWA16
	RBWA15	RBWA14	RBWA13	RBWA12	RBWA11	RBWA10	RBWA9	RBWA8
	RBWA7	RBWA6	RBWA5	RBWA4	RBWA3	RBWA2	RBWA1	RBWA0
RDFARn (n = 0, 1)	RDFA31	RDFA30	RDFA29	RDFA28	RDFA27	RDFA26	RDFA25	RDFA24
	RDFA23	RDFA22	RDFA21	RDFA20	RDFA19	RDFA18	RDFA17	RDFA16
	RDFA15	RDFA14	RDFA13	RDFA12	RDFA11	RDFA10	RDFA9	RDFA8
	RDFA7	RDFA6	RDFA5	RDFA4	RDFA3	RDFA2	RDFA1	RDFA0
TBRARn (n = 0, 1)	TBRA31	TBRA30	TBRA29	TBRA28	TBRA27	TBRA26	TBRA25	TBRA24
	TBRA23	TBRA22	TBRA21	TBRA20	TBRA19	TBRA18	TBRA17	TBRA16
	TBRA15	TBRA14	TBRA13	TBRA12	TBRA11	TBRA10	TBRA9	TBRA8
	TBRA7	TBRA6	TBRA5	TBRA4	TBRA3	TBRA2	TBRA1	TBRA0

	—	—	—	—	—	RFD2	RFD1	RFD0
TRIMDn (n = 0, 1)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	TIS
PACR	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0
	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0
PBCR	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0
PCCR	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0
	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0
PETCR	PET3MD	—	PET2MD	—	PET1MD	—	PET0MD	—
	—	—	—	—	—	—	—	—
PADR	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT
PBDR	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT
PCDR	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT
SDIR	TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0
	—	—	—	—	—	—	—	—
SDID/ SDIDH	DID31	DID30	DID29	DID28	DID27	DID26	DID25	DID24
	DID23	DID22	DID21	DID20	DID19	DID18	DID17	DID16
SDIDL	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8
	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0

Notes: 1. Bit names in the first row of CS0WCR show the names for the normal/byte-se
SRAM interface, in the second row for the burst ROM (asynchronous) interfac
the third row for the burst ROM (synchronous) interface.

MMUCR	Initialized	Initialized	Retained	Retained	Retained	M
PTEH	Initialized	Initialized	Retained	Retained	Retained	
PTEL	Initialized	Initialized	Retained	Retained	Retained	
TTB	Initialized	Initialized	Retained	Retained	Retained	
CCR1	Initialized	Initialized	Retained	Retained	Retained	C
CCR2	Initialized	Initialized	Retained	Retained	Retained	
CCR3	Initialized	Initialized	Retained	Retained	Retained	
IPRA	Initialized	Initialized	Retained	—	Retained	I
IPRB	Initialized	Initialized	Retained	—	Retained	
IPRC	Initialized	Initialized	Retained	—	Retained	
IPRD	Initialized	Initialized	Retained	—	Retained	
IPRE	Initialized	Initialized	Retained	—	Retained	
IPRF	Initialized	Initialized	Retained	—	Retained	
IPRG	Initialized	Initialized	Retained	—	Retained	
IPRH	Initialized	Initialized	Retained	—	Retained	
IPRI	Initialized	Initialized	Retained	—	Retained	
ICR0	Initialized	Initialized	Retained	—	Retained	
ICR1	Initialized	Initialized	Retained	—	Retained	
IRR0	Initialized	Initialized	Retained	—	Retained	
IRR1	Initialized	Initialized	Retained	—	Retained	
IRR2	Initialized	Initialized	Retained	—	Retained	
IRR3	Initialized	Initialized	Retained	—	Retained	
IRR4	Initialized	Initialized	Retained	—	Retained	

BAMRB	Initialized	Initialized	Retained	Retained	Retained	
BBRB	Initialized	Initialized	Retained	Retained	Retained	
BDRB	Initialized	Initialized	Retained	Retained	Retained	
BDMRB	Initialized	Initialized	Retained	Retained	Retained	
BRCR	Initialized	Initialized	Retained	Retained	Retained	
BETR	Initialized	Initialized	Retained	Retained	Retained	
BRSR	Initialized	Initialized	Retained	Retained	Retained	
BRDR	Initialized	Initialized	Retained	Retained	Retained	
BASRA	Initialized	Initialized	Retained	Retained	Retained	
BASRB	Initialized	Initialized	Retained	Retained	Retained	
STBCR	Initialized	Retained	Retained	—	Retained	Po
STBCR2	Initialized	Retained	Retained	—	Retained	mo
STBCR3	Initialized	Retained	Retained	—	Retained	
FRQCR	Initialized	Retained	Retained	—	Retained	CK
WTCNT	Initialized	Initialized	Retained	—	Retained	
WTCSR	Initialized	Initialized	Retained	—	Retained	
CMNCR	Initialized	Retained	Retained	—	Retained	BS
CS0BCR	Initialized	Retained	Retained	—	Retained	
CS2BCR	Initialized	Retained	Retained	—	Retained	
CS3BCR	Initialized	Retained	Retained	—	Retained	
CS4BCR	Initialized	Retained	Retained	—	Retained	
CS5ABCR	Initialized	Retained	Retained	—	Retained	
CS5BBCR	Initialized	Retained	Retained	—	Retained	
CS6ABCR	Initialized	Retained	Retained	—	Retained	

CS6AWCR	Initialized	Retained	Retained	—	Retained	
CS6BWCR	Initialized	Retained	Retained	—	Retained	
SDCR	Initialized	Retained	Retained	—	Retained	
RTCSR	Initialized	Retained	Retained	—	Retained	
RTCNT	Initialized	Retained	Retained	—	Retained	
RTCOR	Initialized	Retained	Retained	—	Retained	
SAR_n (n = 0 to 5)	Initialized	Initialized	Retained	Retained	Retained	D
DAR_n (n = 0 to 5)	Initialized	Initialized	Retained	Retained	Retained	
DMATCR_n (n = 0 to 5)	Initialized	Initialized	Retained	Retained	Retained	
CHCR_n (n = 0 to 5)	Initialized	Initialized	Retained	Retained	Retained	
DMAOR	Initialized	Initialized	Retained	Retained	Retained	
DMARS0	Initialized	Initialized	Retained	Retained	Retained	
DMARS1	Initialized	Initialized	Retained	Retained	Retained	
DMARS2	Initialized	Initialized	Retained	Retained	Retained	
TSTR	Initialized	Initialized	Initialized*[3]	Initialized	Retained	T
TCOR_n (n = 0 to 2)	Initialized	Initialized	Retained	Retained	Retained	
TCNT_n (n = 0 to 2)	Initialized	Initialized	Retained	Retained	Retained	
TCR_n (n = 0 to 2)	Initialized	Initialized	Retained	Retained	Retained	

RTCNT	Retained	Retained	Retained	Retained	Retained	
RSECAR	Retained*[4]	Retained	Retained	Retained	Retained	
RMINAR	Retained*[4]	Retained	Retained	Retained	Retained	
RHRAR	Retained*[4]	Retained	Retained	Retained	Retained	
RWKAR	Retained*[4]	Retained	Retained	Retained	Retained	
RDAYAR	Retained*[4]	Retained	Retained	Retained	Retained	
RMONAR	Retained*[4]	Retained	Retained	Retained	Retained	
RYRAR	Retained	Retained	Retained	Retained	Retained	
RCR1	Initialized	Initialized	Retained	Retained	Retained	
RCR2	Initialized	Initialized*[5]	Retained	Retained	Retained	
RCR3	Initialized	Retained	Retained	Retained	Retained	
SCSMR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	SC
SCBRR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SCSCR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SCFTDR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SCFSR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SCFRDR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SCFCR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	

(n = 0, 1)						
SIRDAR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SICDAR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SICTR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SIFCTR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SISTR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SIIER_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SITDR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SIRDR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SITCR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SIRCR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
ECMRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	E
ECSRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
ECSIPRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	

(n = 0, 1)					
TROCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
CDCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
LCCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
CNDCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
CEFCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
FRECRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
TSFRCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
TLFRCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
RFCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
MAFCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
IPGRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
ARSTR	Initialized	Initialized	Retained	—	Retained
TSU_CTRST	Initialized	Initialized	Retained	—	Retained
TSU_FWEN0	Initialized	Initialized	Retained	—	Retained
TSU_FWEN1	Initialized	Initialized	Retained	—	Retained

TSU_FWSLC	Initialized	Initialized	Retained	—	Retained
TSU_QTAGM0	Initialized	Initialized	Retained	—	Retained
TSU_QTAGM1	Initialized	Initialized	Retained	—	Retained
TSU_ADQT0	Initialized	Initialized	Retained	—	Retained
TSU_ADQT1	Initialized	Initialized	Retained	—	Retained
TSU_FWSR	Initialized	Initialized	Retained	—	Retained
TSU_FWINMK	Initialized	Initialized	Retained	—	Retained
TSU_ADSBSY	Initialized	Initialized	Retained	—	Retained
TSU_TEN	Initialized	Initialized	Retained	—	Retained
TSU_POST1	Initialized	Initialized	Retained	—	Retained
TSU_POST2	Initialized	Initialized	Retained	—	Retained
TSU_POST3	Initialized	Initialized	Retained	—	Retained
TSU_POST4	Initialized	Initialized	Retained	—	Retained
TXNLCR0	Initialized	Initialized	Retained	—	Retained
TXALCR0	Initialized	Initialized	Retained	—	Retained
RXNLCR0	Initialized	Initialized	Retained	—	Retained
RXALCR0	Initialized	Initialized	Retained	—	Retained
FWNLCR0	Initialized	Initialized	Retained	—	Retained
FWALCR0	Initialized	Initialized	Retained	—	Retained
TXNLCR1	Initialized	Initialized	Retained	—	Retained
TXALCR1	Initialized	Initialized	Retained	—	Retained
RXNLCR1	Initialized	Initialized	Retained	—	Retained
RXALCR1	Initialized	Initialized	Retained	—	Retained
FWNLCR1	Initialized	Initialized	Retained	—	Retained

EDRRRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
TDLARn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
RDLARn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
EESRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
EESIPRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
TRSCERn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
RMFCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
TFTRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
FDRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
RMCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
EDOCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
RBWARn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained
RDFARn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained

PBCR	Initialized	Retained	Retained	—	Retained	
PCCR	Initialized	Retained	Retained	—	Retained	
PETCR	Initialized	Retained	Retained	—	Retained	
PADR	Initialized	Retained	Retained	—	Retained	I/
PBDR	Initialized	Retained	Retained	—	Retained	
PCDR	Initialized	Retained	Retained	—	Retained	
SDIR	Retained	Retained	Retained	Retained	Retained	H
SDID/SDIDH	Retained	Retained	Retained	Retained	Retained	
SDIDL	Retained	Retained	Retained	Retained	Retained	

Notes: 1. For the initial values of each register, see the sections for the corresponding r
the initial value is undefined, it is shown as initialized since the data is not reta
2. Some bits are initialized in standby mode. See section 8, Interrupt Controller
details.
3. If the multiplication rate of PLL1 is modified, this register is initialized.
4. Some bits are initialized by a power-on reset. See section 15, Realtime Clock
details.
5. Some bits are initialized by a manual reset. See section 15, Realtime Clock (
details.

	$V_{CC}Q$-HTU		
Power supply voltage (internal)	V_{CC} V_{CC}-PLL1 V_{CC}-PLL2	–0.3 to 2.1	V
Input voltage	V_{in}	–0.3 to $V_{CC}Q$ + 0.3	V
Operating temperature	T_{opr}	–20 to 75	°C
Storage temperature	T_{stg}	–55 to 125	°C

Caution:

- Operating the chip in excess of the absolute maximum rating may result in permaner
- Order of turning on 1.5 V power (Vcc, Vcc-PLL1, Vcc-PLL2) and 3.3 V power (Vc RTC):
  1. The 3.3 V power and the 1.5 V power should be turned on simultaneously or the power should be tuned on first. When the 3.3 V is turned on first, turn on the 1.5 within 1 ms. It is recommended that this interval will be as short as possible.
  2. Until voltage is applied to all power supplies and a low level is input at the $\overline{\text{RESE}}$ internal circuits remain unsettled, and so pin states are also undefined. The syster must ensure that these undefined states do not cause erroneous system operation.
  3. When the power is turned on, make sure that the voltage of the 1.5 V power is lo that of the 3.3 V power.

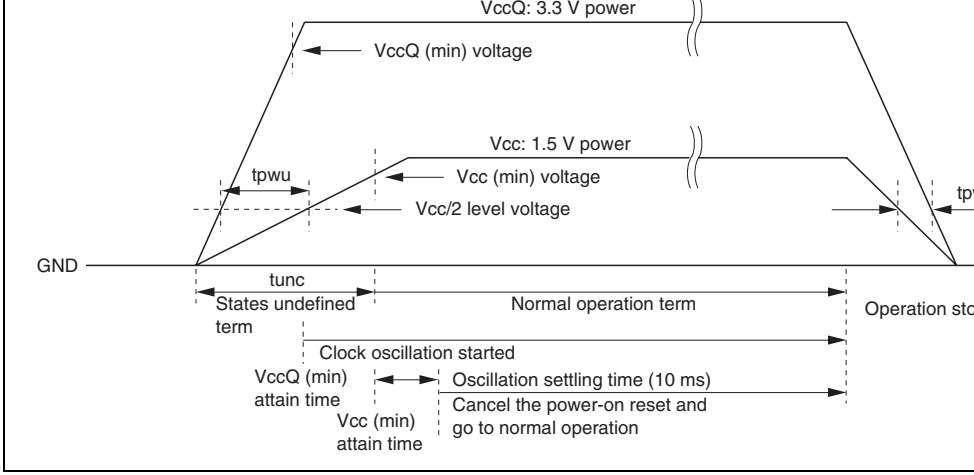


Figure 24.1 Power On/Off Sequence

Recommended Power On/Off Times

Item	Symbol	Max. Permitted Value	Uni
VccQ to Vcc power-on time interval	tpwu	1	ms
VccQ to Vcc power-off time interval	tpwd	1	ms
State undefined term	tunc	10	ms

Note: The recommended times shown above do not require strict settings.

The state undefined term indicates that pins are at the power rising stage. The pin stabilized at VccQ (min.) attain time. However, a power-on reset ($\overline{\text{RESETP}}$) is acce successfully only after VccQ (min.) attain time and clock oscillation settling time. S state undefined term less than 10 ms.

		V_{CC}, V_{CC}-PLL1 V_{CC}-PLL2	1.4	1.5	1.6	V	
Current consumption	Normal operation	I_{CC}	—	250	330	mA	V_{CC} = 1.5 Iϕ = 200
		$I_{CC}Q$	—	40	70	mA	Bϕ = 66.
	In sleep mode*	I_{CC}	—	110	160	mA	$V_{CC}Q$ = 3
		$I_{CC}Q$	—	4	7		Bϕ = 66.
	In standby mode	I_{CC}	—	1500	2600	µA	T_a = 25° (RTC on
		$I_{CC}Q$	—	75	230		$V_{CC}Q$ = 3 V_{CC} = 1.5
		I_{CC}	—	1500	2600	µA	T_a = 25° (RTC of
		$I_{CC}Q$	—	75	230		$V_{CC}Q$ = 3 V_{CC} = 1.5
Input leak current	All input pins	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 $V_{CC}Q$ – 0
Three-state leak current	I/O, all output pins (off condition)	$\lvert I_{STI} \rvert$	—	—	1.0	µA	Vin = 0.5 $V_{CC}Q$ – 0
Pull-up resistance	Port pin	R_{pull}	20	60	180	kΩ	
Pin capacitance	All pins	C	—	—	20	pF	

Note: * No external bus cycles except refresh cycles.

	EXTAL2		—	—	—		When this pi connected to crystal reson pin should be connected to $V_{CC}Q$ pin (pu
	Other input pins		2.0	—	$V_{CC}Q + 0.3$		
Input low voltage	$\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI IRQ5 to IRQ0, MD5 to MD0, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$, EXTAL, CKIO	V_{IL}	–0.3	—	$V_{CC}Q \times 0.1$	V	
	EXTAL2		—	—	—		When this pi connected to crystal reson pin should be connected to $V_{CC}Q$ pin (pu
	Other input pins		–0.3	—	$V_{CC}Q \times 0.2$		
Output high voltage	All output pins	V_{OH}	2.4	—	—	V	$V_{CC}Q$ = 3.0 V I_{OH} = –2 mA
Output low voltage	All output pins	V_{OL}	—	—	0.55	V	$V_{CC}Q$ = 3.0 V I_{OL} = 2 mA

Output low-level permissible current (total)	$\sum I_{OL}$	—	—	120
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0
Output high-level permissible current (total)	$\sum (-I_{OH})$	—	—	40

Caution: To ensure LSI reliability, do not exceed the value for output current given in Tab

24.3 AC Characteristics

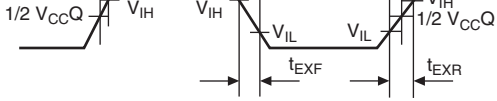
In general, inputting for this LSI should be clock synchronous. Keep the setup and hold each input signal unless otherwise specified.

Table 24.4 Maximum Operating Frequencies

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1$
$V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -2$

Item		Symbol	Min.	Typ.	Max.	Unit	Re
Operating frequency	CPU, cache (Iϕ)	f	33.34	—	200	MHz	
	External bus (Bϕ)		33.34	—	66.67		
	Peripheral module (Pϕ)		8.34	—	33.34		

EXTAL clock input high pulse width	t_{EXH}	1.5	—		
EXTAL clock input rise time	t_{EXR}	—	6		
EXTAL clock input fall time	t_{EXF}	—	6		
CKIO clock input frequency	f_{CKI}	33.34	66.67	MHz	24.3
CKIO clock input cycle time	t_{CKIcyc}	15	30	ns	
CKIO clock input low pulse width	t_{CKIL}	3	—		
CKIO clock input high pulse width	t_{CKIH}	3	—		
CKIO clock input rise time	t_{CKIR}	—	4		
CKIO clock input fall time	t_{CKIF}	—	4		
CKIO clock output frequency	f_{OP}	33.34	66.67	MHz	24.4
CKIO clock output cycle time	t_{cyc}	15	30	ns	
CKIO clock output low pulse width	t_{CKOL}	3	—		
CKIO clock output high pulse width	t_{CKOH}	3	—		
CKIO clock output rise time	t_{CKOR}	—	4		
CKIO clock output fall time	t_{CKOF}	—	4		
CKIO2 clock output delay time	t_{CK2D}	—	2.5		
CKIO2 clock output rise time	t_{CK2OR}	—	7		
CKIO2 clock output fall time	t_{CK2OF}	—	7		
Power-on oscillation settling time	t_{OSC1}	10	—	ms	24.5
$\overline{\text{RESETP}}$ setup time	t_{RESPS}	20	—	ns	24.5
$\overline{\text{RESETP}}$ assert time	t_{RESPW}	20	—	t_{cyc}	24.5, 2
$\overline{\text{RESETM}}$ assert time	t_{RESMW}	20	—	t_{cyc}	24.6
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	24.6
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	24.7
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	24.8



Note: * The clock input from the EXTAL pin.

Figure 24.2 EXTAL Clock Input Timing

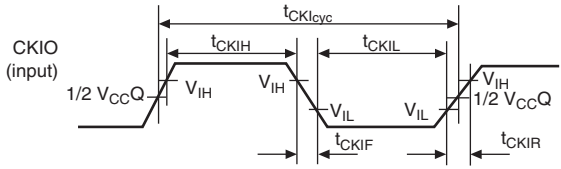


Figure 24.3 CKIO Clock Input Timing

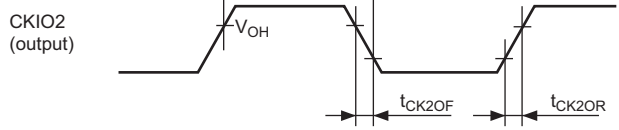
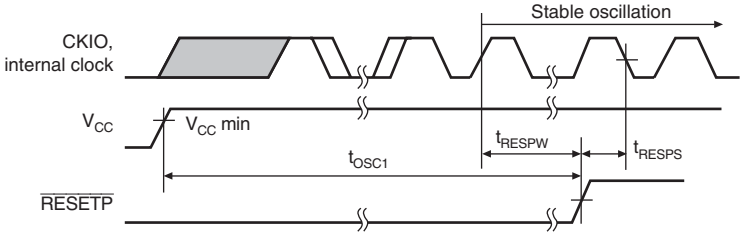
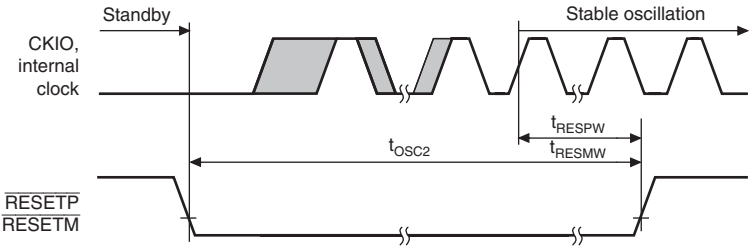


Figure 24.4 CKIO Clock Output Timing



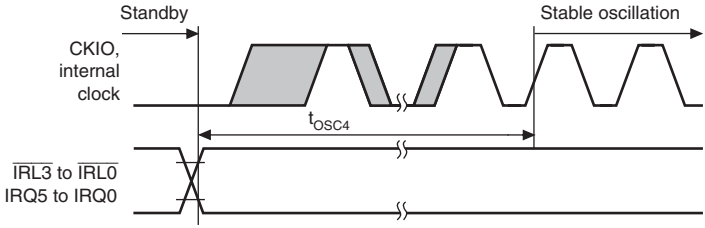
Note: Oscillation settling time when on-chip oscillator is used

Figure 24.5 Power-On Oscillation Settling Time



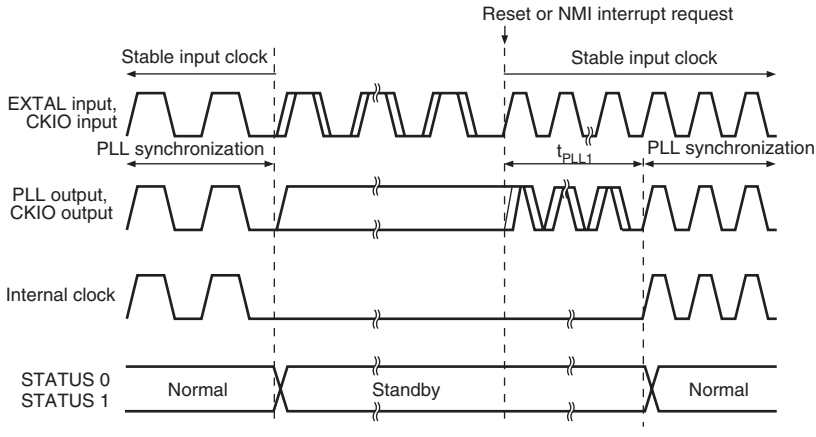
Note: Oscillation settling time when on-chip oscillator is used

Figure 24.6 Oscillation Settling Time at Standby Return (Return by Reset)



Note: Oscillation settling time when on-chip oscillator is used in oscillation stop mode

Figure 24.8 Oscillation Settling Time at Standby Return (Return by IRQ5 to IRQ0 and $\overline{IRL3}$ to $\overline{IRL0}$)

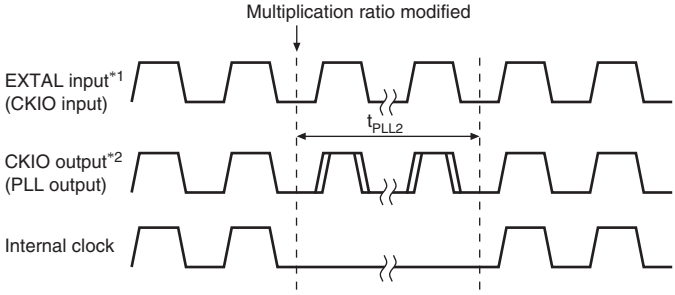


Note: PLL oscillation settling time when clock is input from EXTAL pin

Figure 24.9 PLL Synchronization Settling Time by Reset or NMI

Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin in oscillation continuous mode.

Figure 24.10 PLL Synchronization Settling Time by IRQ/IRL Interrupts



Notes: 1. CKIO input in clock mode 7
2. PLL output except in clock mode 7

Figure 24.11 PLL Synchronization Settling Time when Frequency Multiplica Ratio Modified

RESETM pulse width	t_{RESMW}	20		t_{cyc}
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	10	—	ns
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	1/2 t_{cyc}+10	—	
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	1/2 t_{cyc}+3	—	
NMI setup time*[1]	t_{NMIS}	10	—	
NMI hold time	t_{NMIH}	3	—	
IRQ5 to IRQ0 setup time*[1]	t_{IRQS}	10	—	
IRQ5 to IRQ0 hold time	t_{IRQH}	3	—	
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	1/2 t_{cyc}+13	
STATUS1, STATUS0 delay time	t_{STD}	—	18	
$\overline{\text{IRQOUT}}$ delay time	t_{IRQOTD}	—	1/2 t_{cyc}+12	
Bus tri-state delay time 1	t_{BOFF1}	0	30	
Bus tri-state delay time 2	t_{BOFF2}	0	30	
Bus buffer-on time 1	t_{BON1}	0	30	
Bus buffer-on time 2	t_{BON2}	0	30	

Notes: t_{cyc} is the external bus clock cycle (B clock cycle).

1. $\overline{\text{RESETP}}$, NMI, and IRQ5 to IRQ0 are asynchronous. Changes are detected a clock rise when the setup shown is kept. When the setup cannot be kept, det be delayed until the next clock rises.
2. The upper limit of the external bus clock is 66.67 MHz.
3. In standby mode, $t_{RESPW} = t_{OSC2}$ (10 ms). When the crystal oscillation continues clock multiplication ratio is changed in standby mode, $t_{RESPW} = t_{PLL1}$ (100 µs).
4. In standby mode, $t_{RESMW} = t_{OSC2}$ (10 ms). When the crystal oscillation continues clock multiplication ratio is changed in standby mode, $\overline{\text{RESETM}}$ must be kept STATUS (0-1) changes to reset (HH).

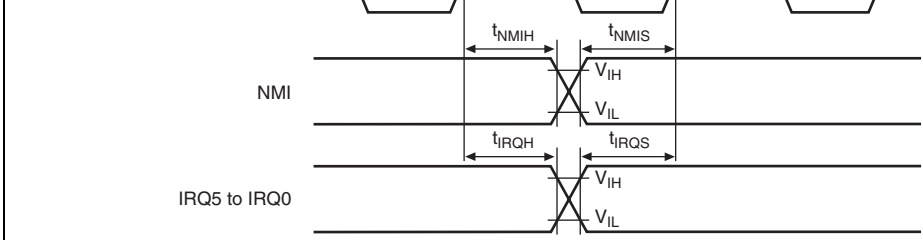


Figure 24.13 Interrupt Signal Input Timing

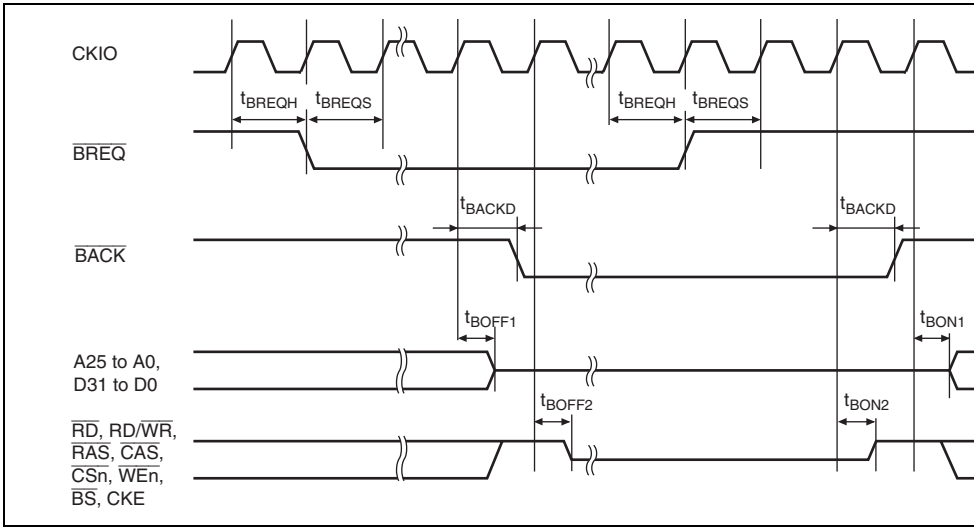


Figure 24.14 Bus Release Timing

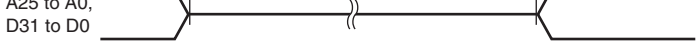


Figure 24.15 Pin Drive Timing at Standby

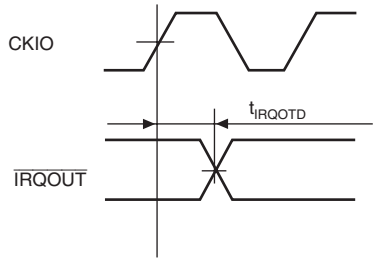
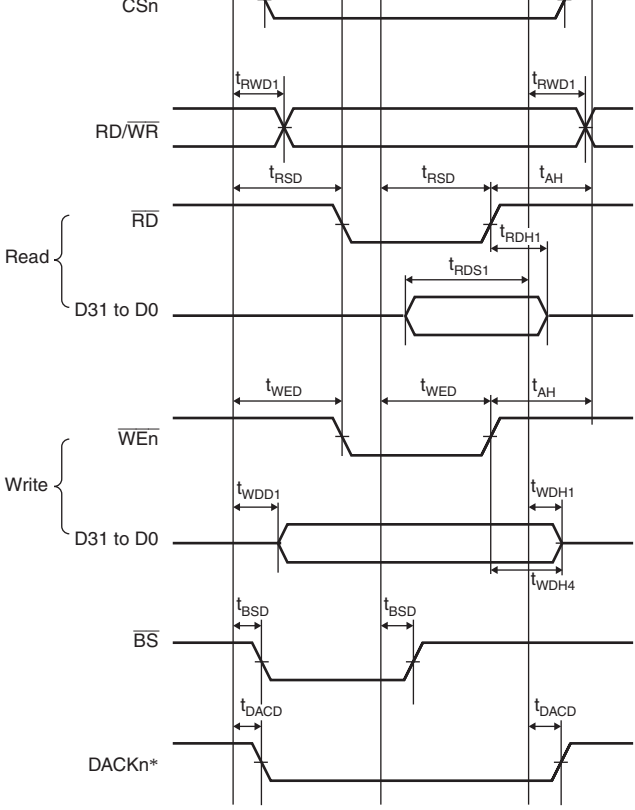


Figure 24.16 $\overline{IRQOUT}$ Output Delay Time

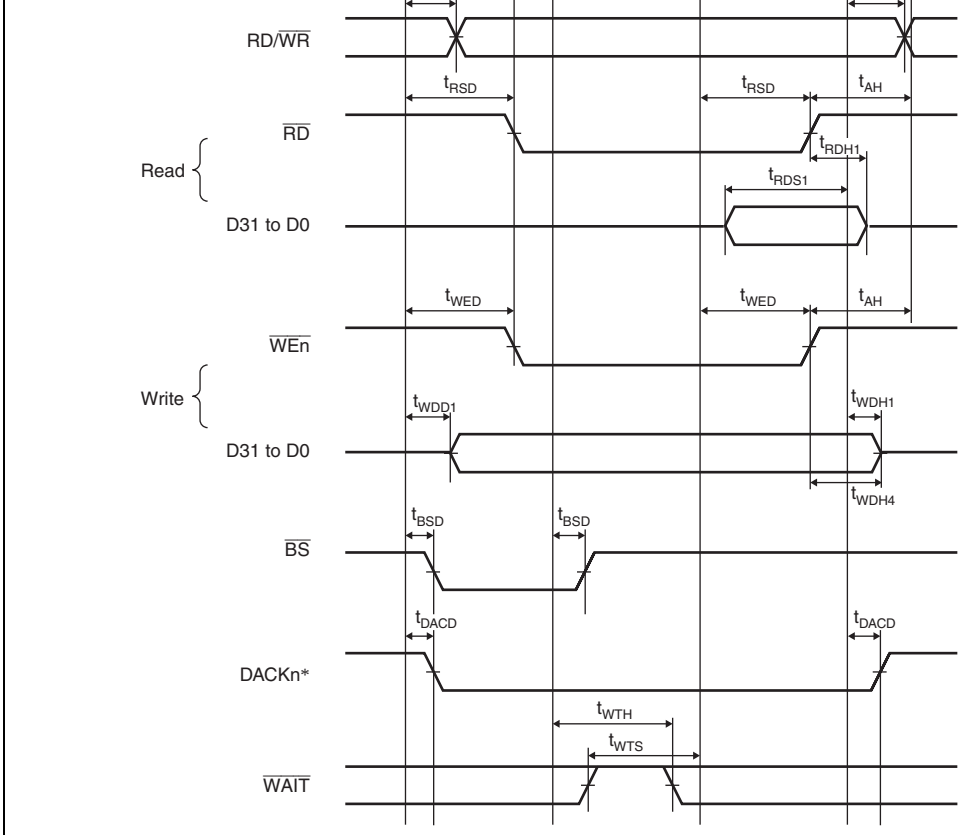
Address setup time	t_{AS}	0		24.17 to 24.20
Address hold time	t_{AH}	0	—	
$\overline{BS}$ delay time	t_{BSD}	—	10	24.17 to 24.35, 24.42
$\overline{CS}$ delay time 1	t_{CSD1}	1	10	24.17 to 24.42
Read/write delay time 1	t_{RWD1}	1	10	
Read strobe delay time	t_{RSD}	—	1/2 t_{cyc}+10	24.17 to 24.21, 24.40
Read data setup time 1	t_{RDS1}	1/2 t_{cyc}+6	—	24.17 to 24.20, 24.42
Read data setup time 2	t_{RDS2}	6	—	24.22 to 24.25, 24.32
Read data setup time 3	t_{RDS3}	1/2 t_{cyc}+6	—	24.21
Read data hold time 1	t_{RDH1}	0	—	24.17 to 24.20, 24.42
Read data hold time 2	t_{RDH2}	2	—	24.22 to 24.25, 24.32
Read data hold time 3	t_{RDH3}	0	—	24.21
Write enable delay time	t_{WED}	—	1/2 t_{cyc}+10	24.17 to 24.21, 24.40
Write data delay time 1	t_{WDD1}	—	12	24.17 to 24.20, 24.42
Write data delay time 2	t_{WDD2}	—	12	24.26 to 24.29, 24.35
Write data hold time 1	t_{WDH1}	1	—	24.17 to 24.20
Write data hold time 2	t_{WDH2}	1	—	24.26 to 24.29, 24.35

CKE delay time 1	t_{CKED1}	1	10	24.37
DACK delay time	t_{DACD}	—	10	24.17 to 24.35
$\overline{\text{ICIORD}}$ delay time	t_{ICRSD}	—	1/2 t_{cyc}+12	24.41, 24.42
$\overline{\text{ICIOWR}}$ delay time	t_{ICWSD}	—	1/2 t_{cyc}+12	
$\overline{\text{IOIS16}}$ setup time	t_{IO16S}	1/2 t_{cyc}+12	—	24.42
$\overline{\text{IOIS16}}$ hold time	t_{IO16H}	1/2 t_{cyc}+4	—	
$\overline{\text{REFOUT}}$ delay time	t_{REFOD}	—	1/2 t_{cyc}+12	24.43



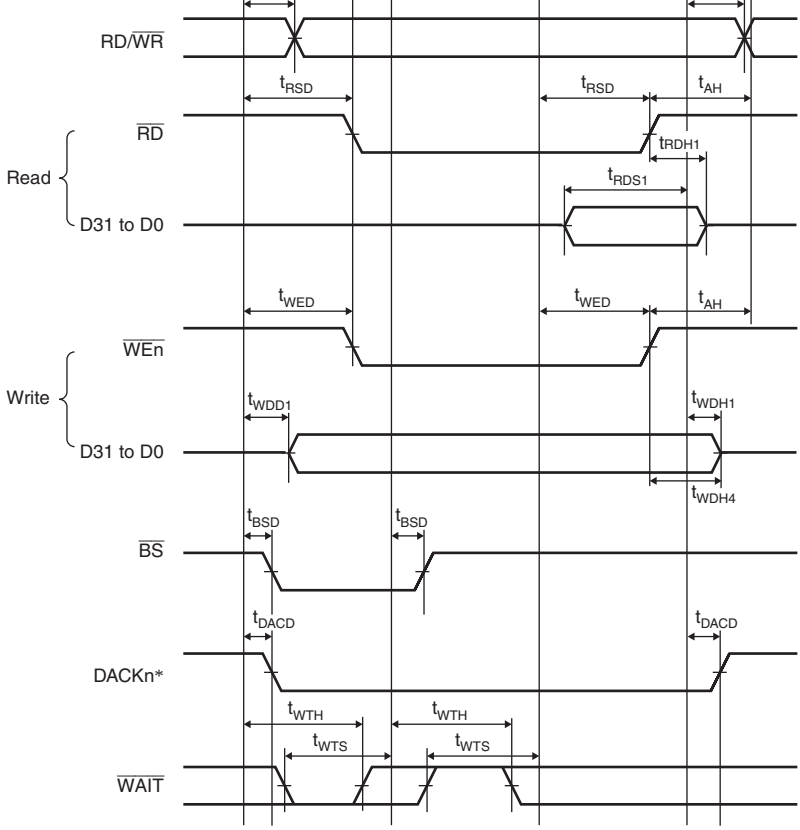
Note: * DACKn is a waveform when active-low is specified.

Figure 24.17 Basic Bus Cycle (No Wait)



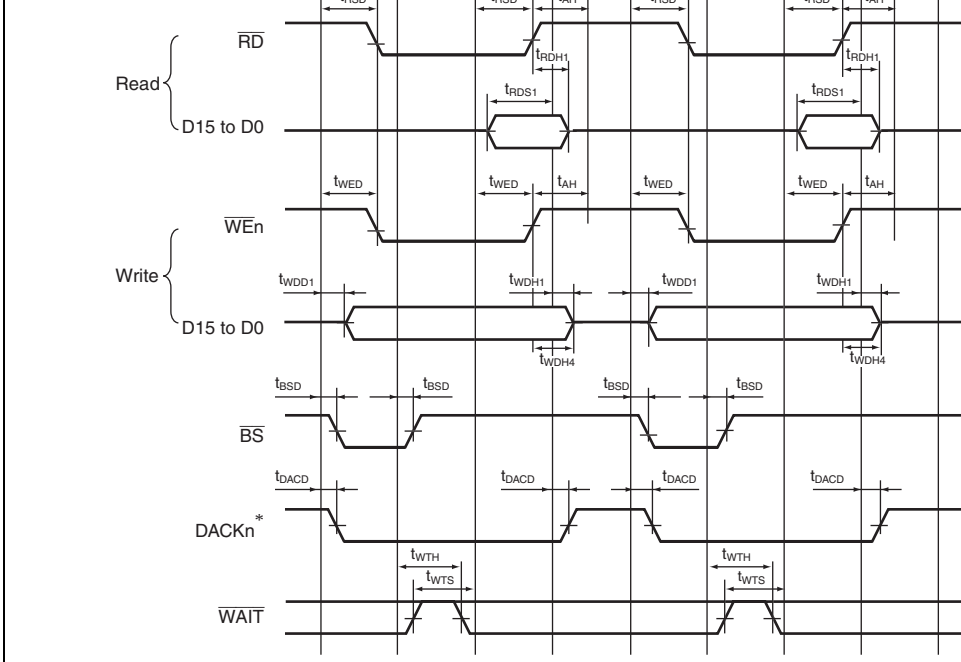
Note: * DACKn is a waveform when active-low is specified.

Figure 24.18 Basic Bus Cycle (One Software Wait)



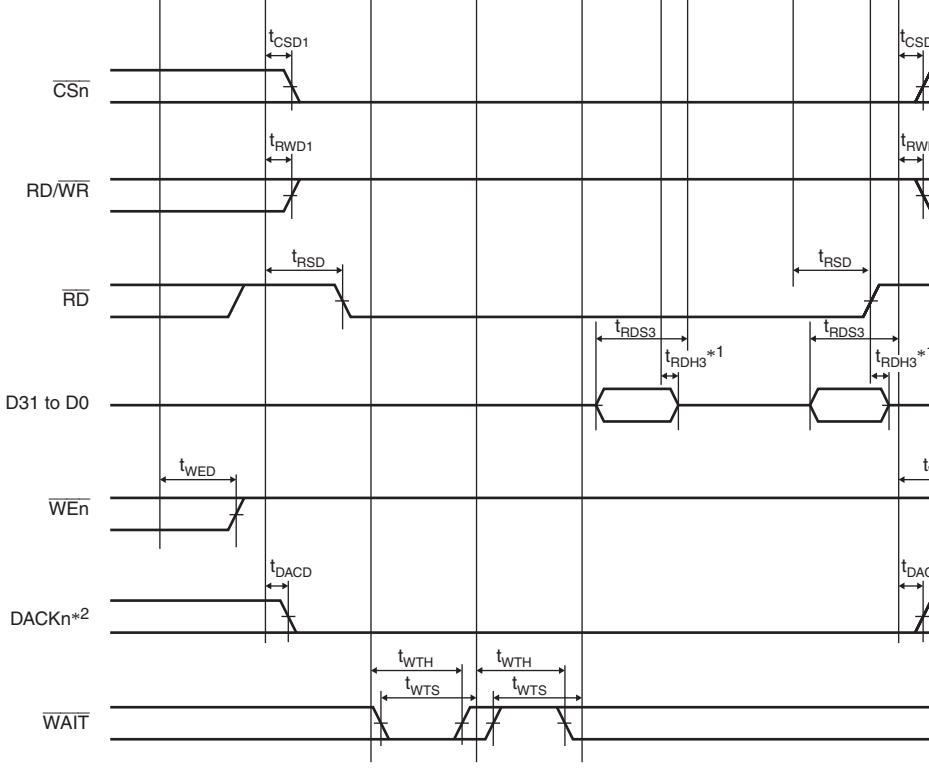
Note: * DACKn is a waveform when active-low is specified.

Figure 24.19 Basic Bus Cycle (One External Wait)



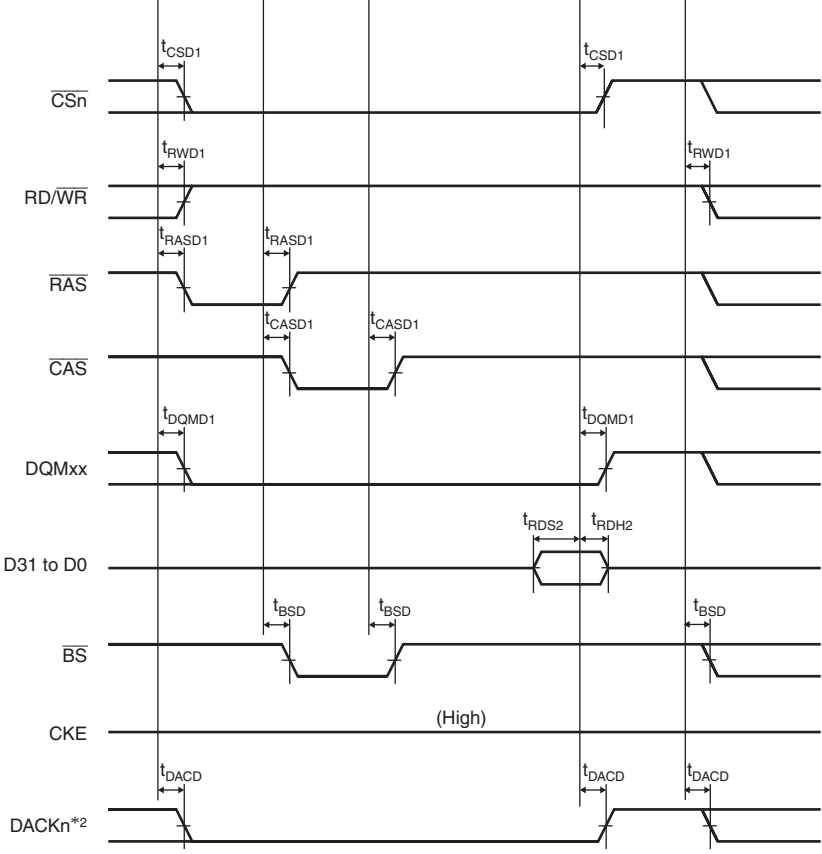
Note: * DACKn is a waveform when active-low is specified.

Figure 24.20 Basic Bus Cycle (One Software Wait, External Wait Enable (WM bit = 0), No Idle Cycle Setting)



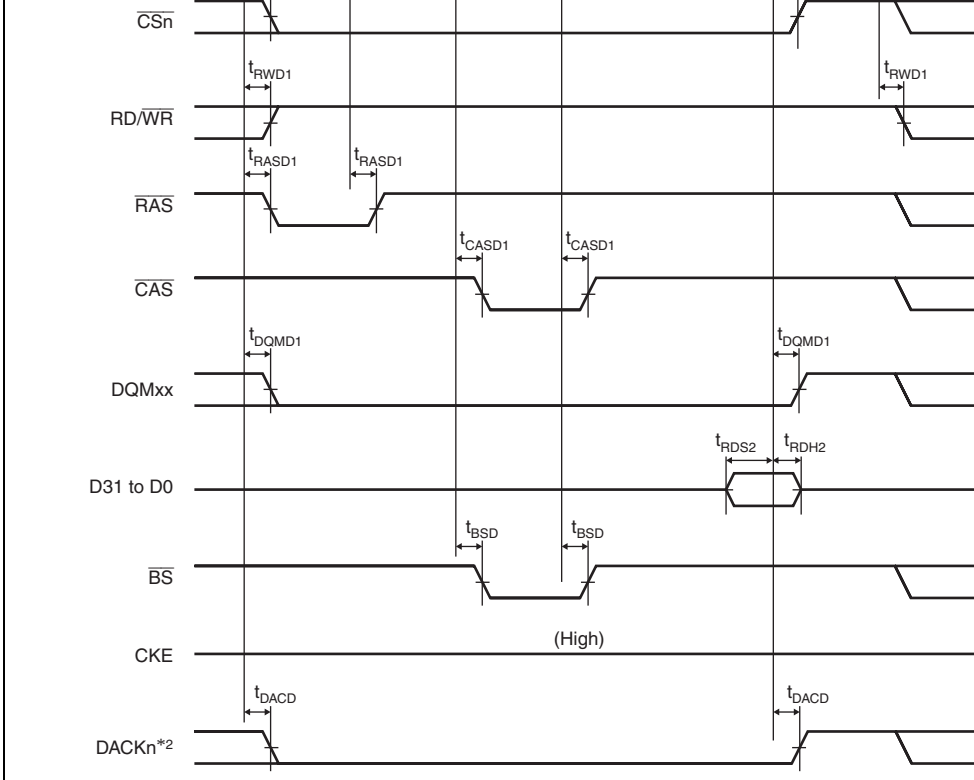
Notes: 1. t_{RDH3} is specified by earlier one of change of A25 to A0 or the $\overline{RD}$ rising edge.
2. DACKn is a waveform when active-low is specified.

Figure 24.21 Burst ROM Read Cycle (One Access Wait, One External Wai
One Burst Wait, Two Bursts)



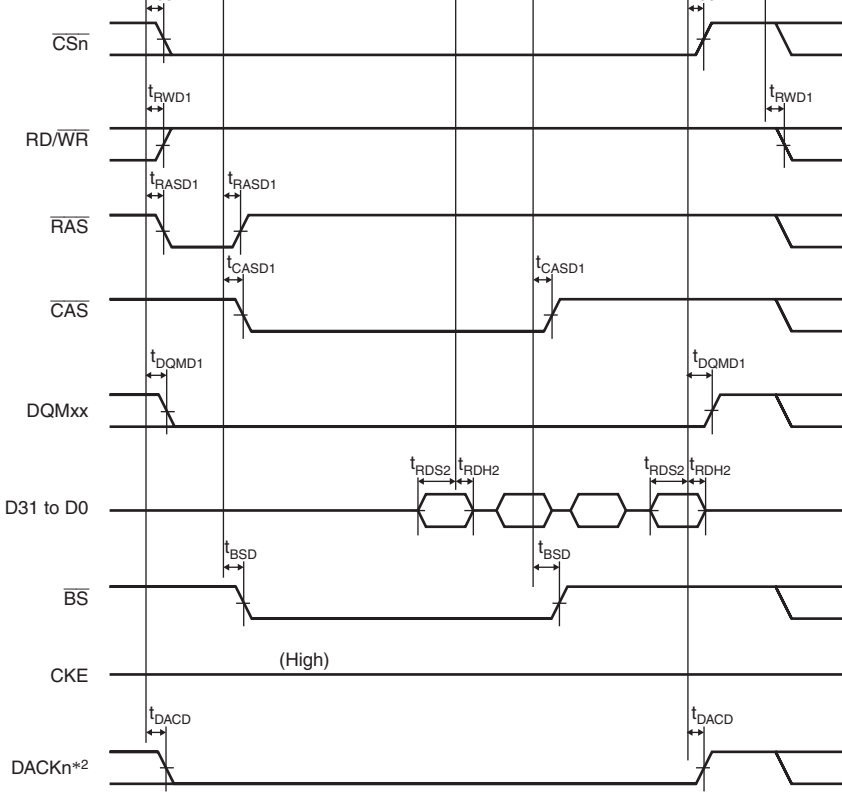
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.22 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency = 2, TRCD = 1 Cycle, TRP = 1 Cycle)



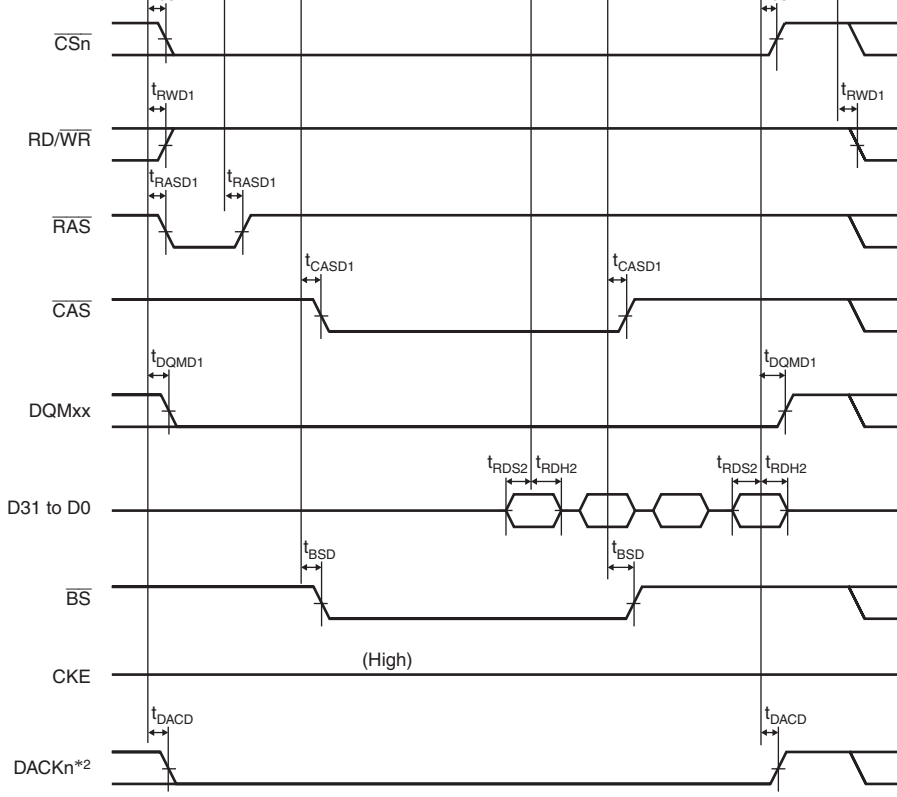
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.23 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency = 2, TRCD = 2 Cycle, TRP = 2 Cycle)



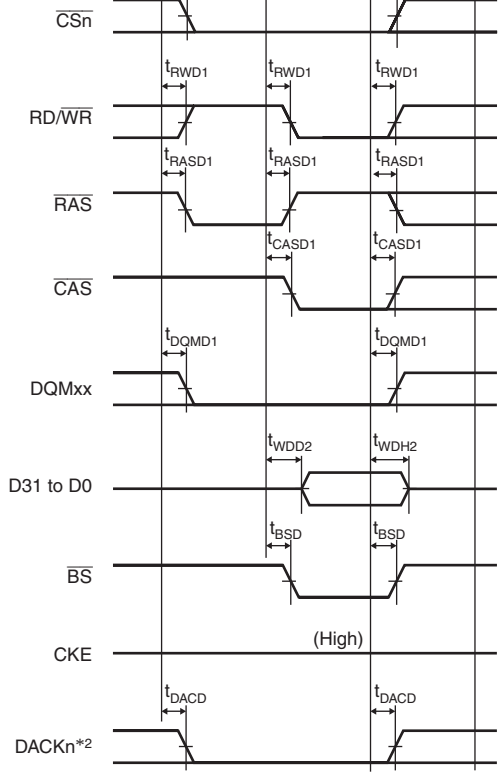
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.24 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4 (Auto Precharge, CAS Latency = 2, TRCD = 1 Cycle, TRP = 2 Cycle)



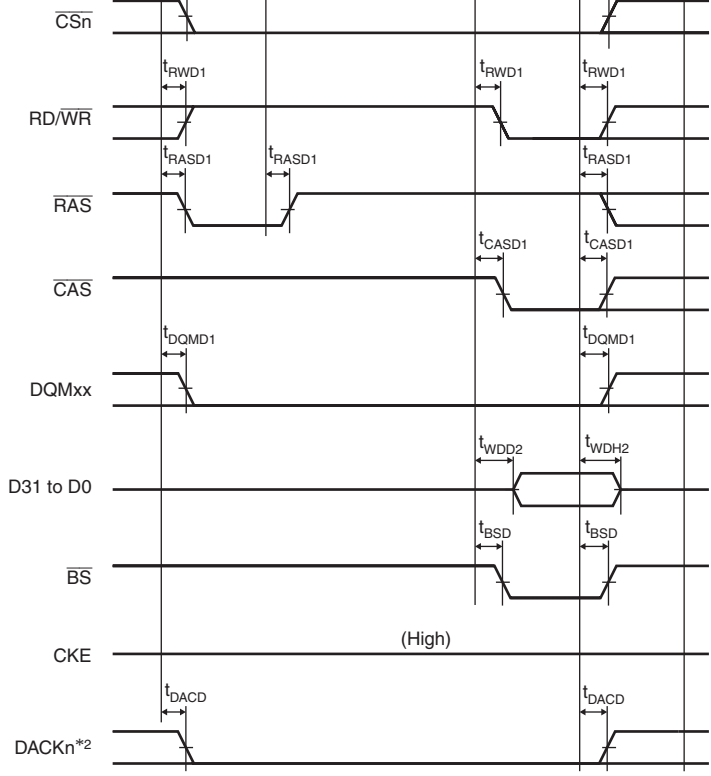
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.25 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Auto Precharge, CAS Latency = 2, TRCD = 2 Cycle, TRP = 1 Cycle)



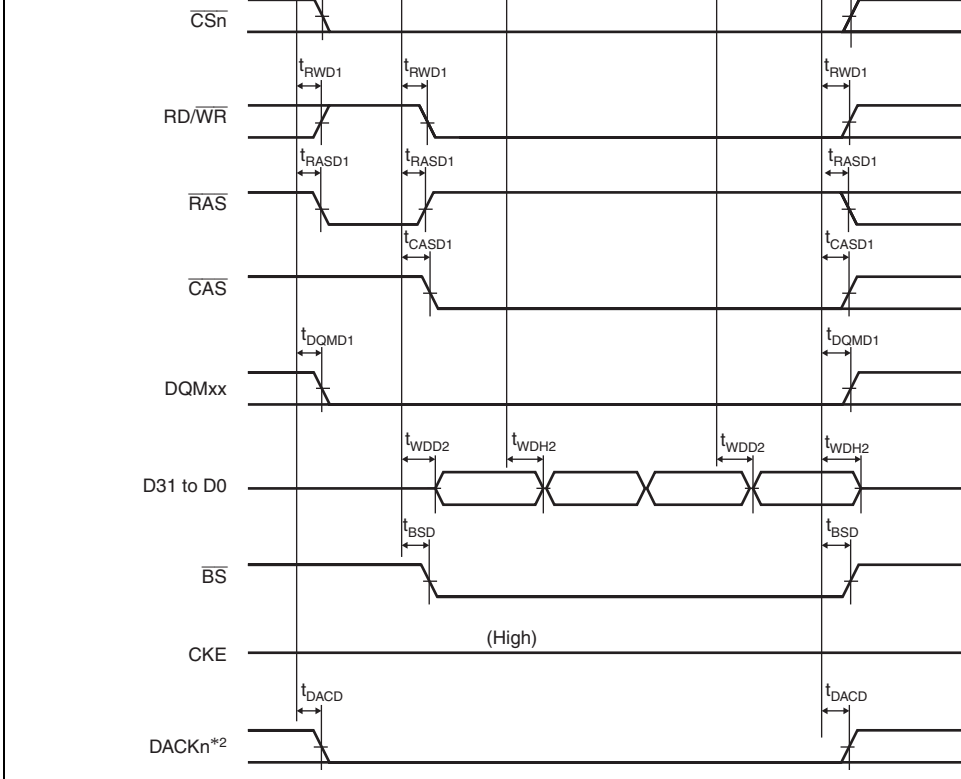
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.26 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 2 Cycle)



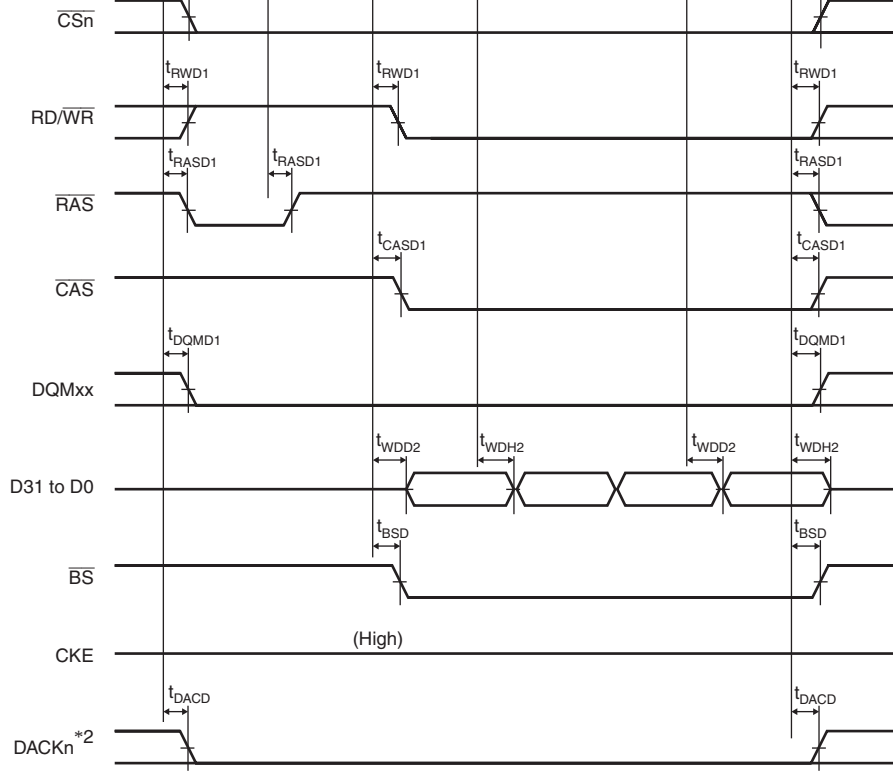
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.27 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRCD = 3 Cycle, TRWL = 2 Cycle)



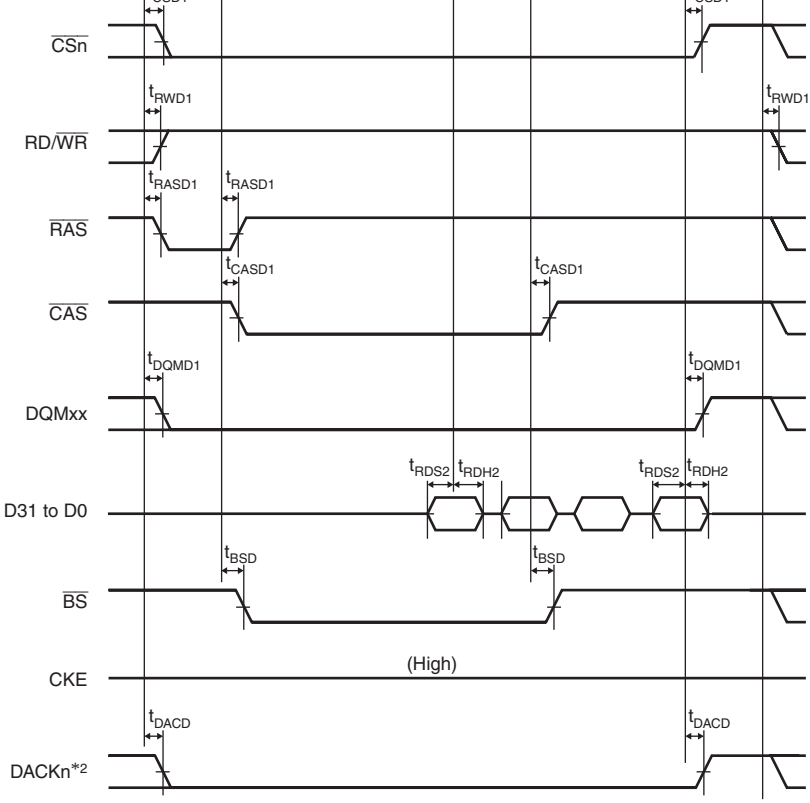
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.28 Synchronous DRAM Burst Write Bus Cycle (Single Write × (Auto Precharge, TRCD = 1 Cycle, TRWL = 2 Cycle)



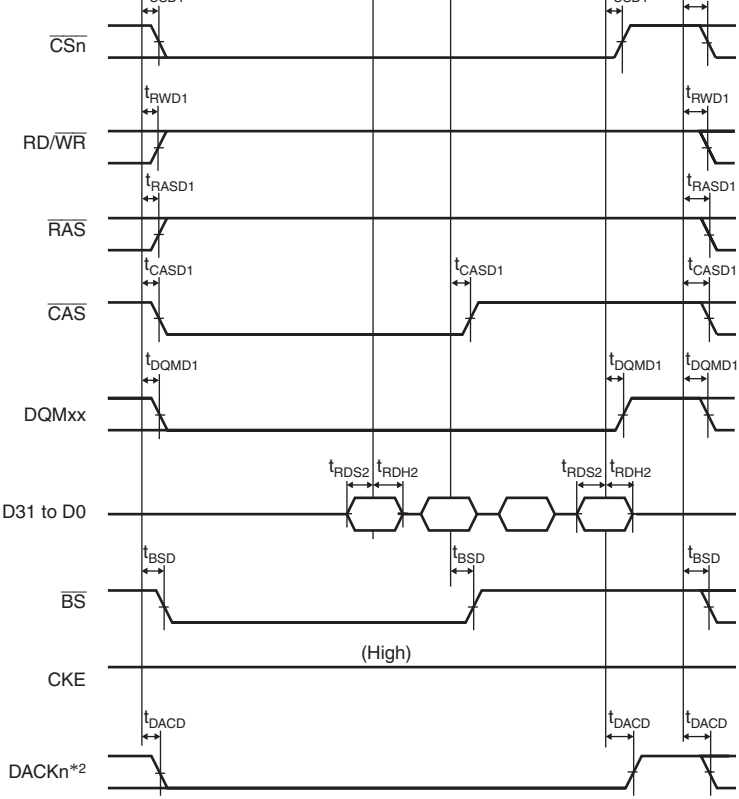
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.29 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4 (Auto Precharge, TRCD = 2 Cycle, TRWL = 2 Cycle)



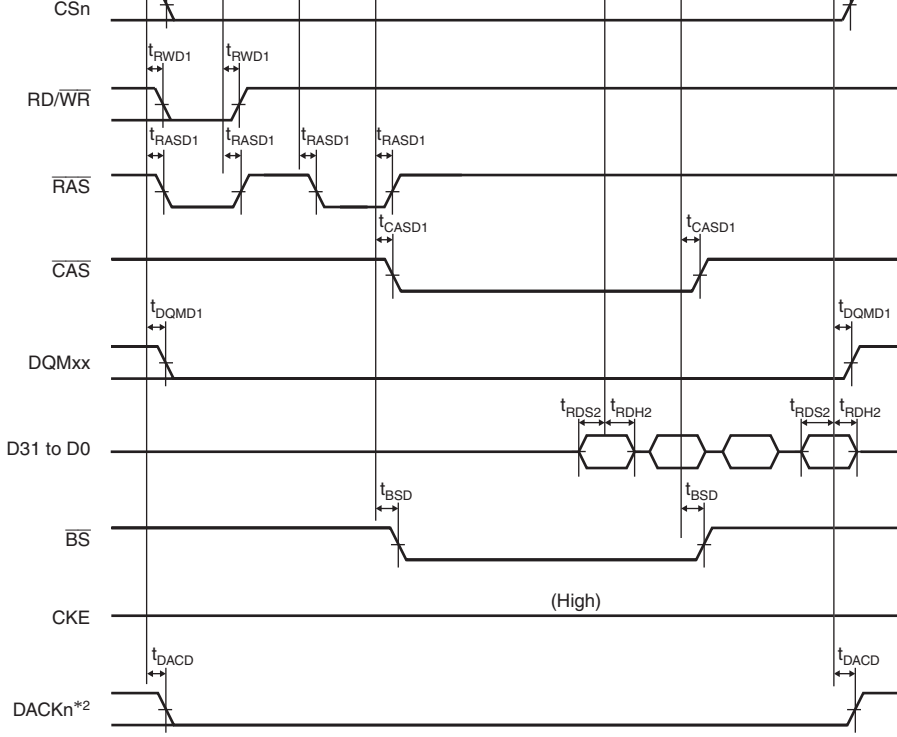
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.30 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4
(Bank Active Mode, ACTV + READ Commands, CAS Latency = 2, TRCD = 1



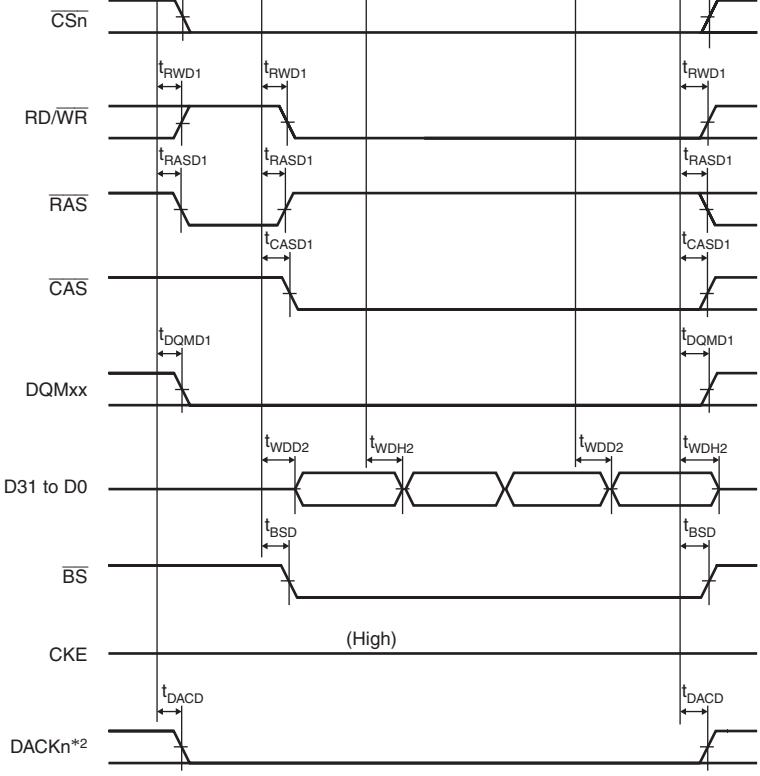
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.31 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode, READ Command, Same Row Address, CAS Latency = 2, TRCD = 1 Cycle)



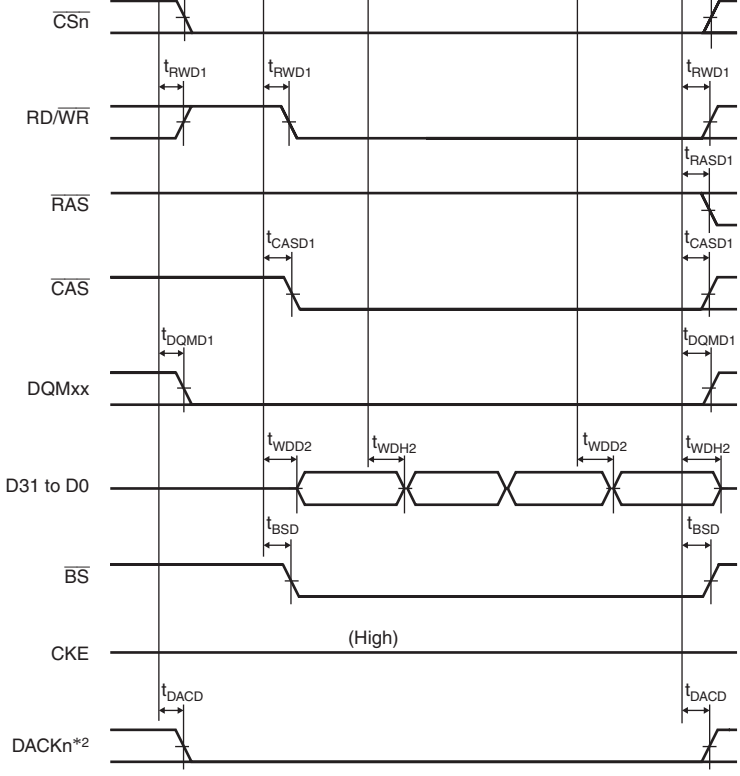
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.32 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4 (Bank Active Mode, PRE + ACTV + READ Commands, Different Row Address, CAS Latency = 2, TRCD = 1 Cycle)



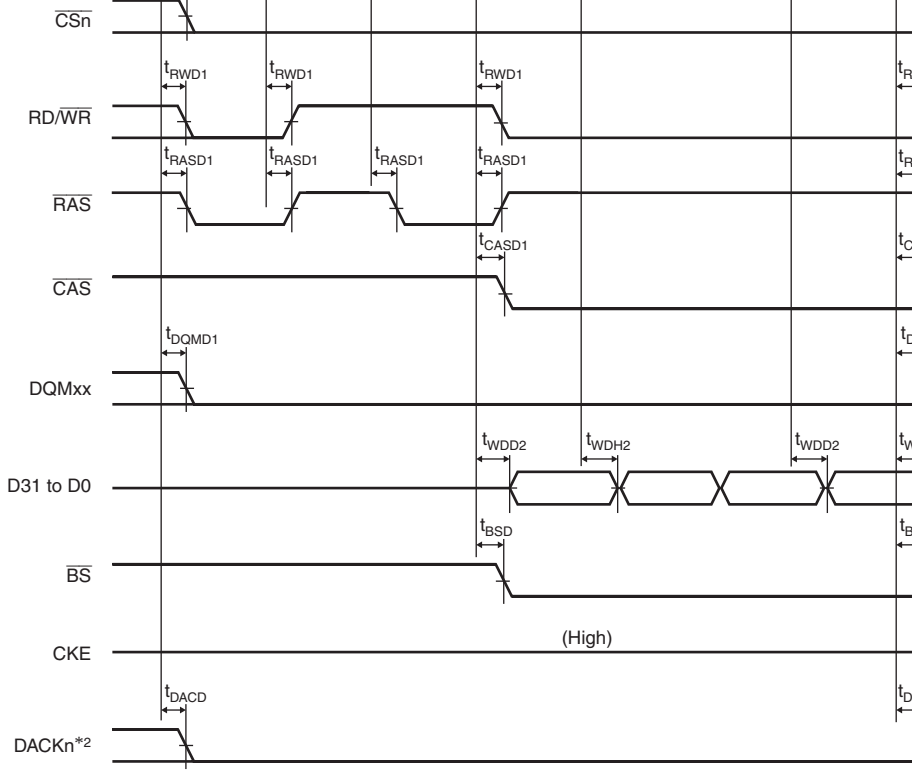
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.33 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4
(Bank Active Mode, ACTV + WRITE Commands, TRCD = 1 Cycle, TRWL = 1 C



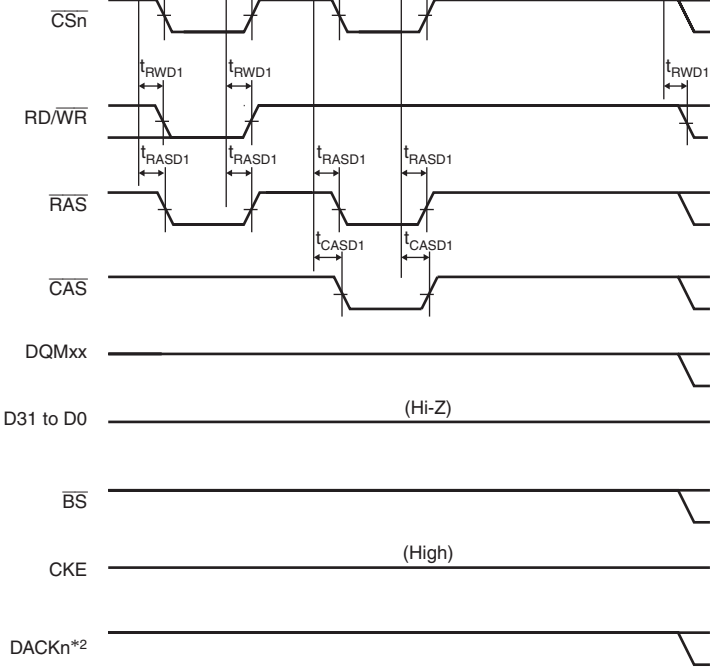
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.34 Synchronous DRAM Burst Write Bus Cycle (Single Write × (Bank Active Mode, WRITE Command, Same Row Address, TRCD = 1 Cycle, TRWL = 1 Cycle)



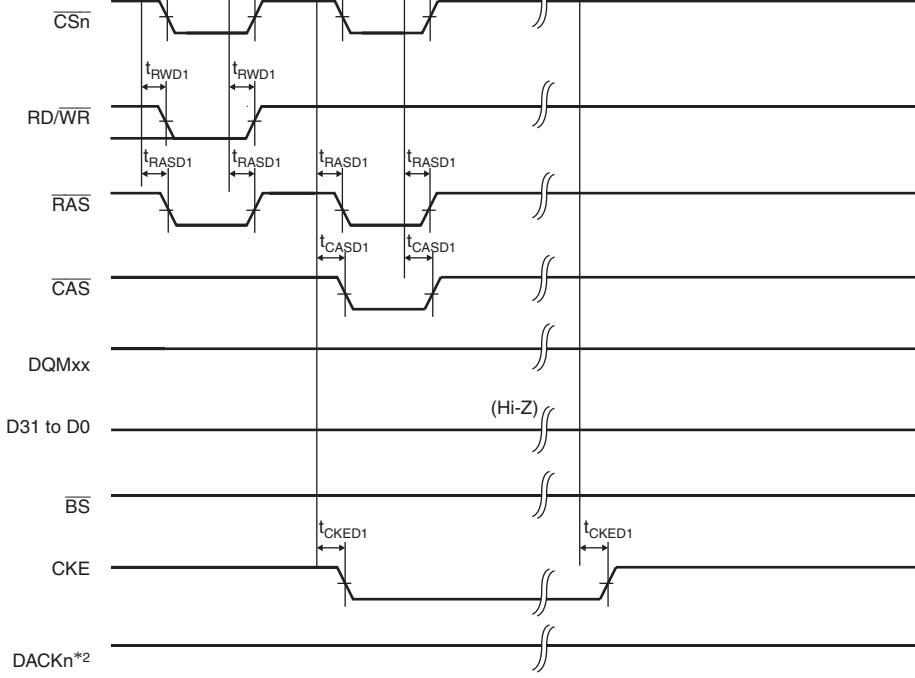
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.35 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4 (Bank Active Mode, PRE + ACTV + WRITE Commands, Different Row Address, TRCD = 1 Cycle, TRWL = 1 Cycle)



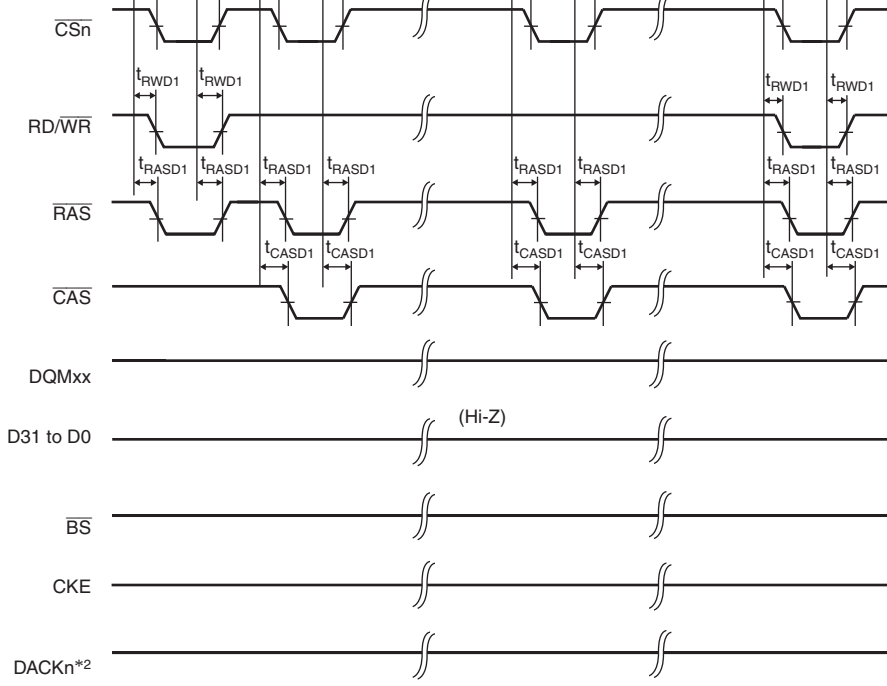
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.36 Synchronous DRAM Auto-Refresh Timing (TRP = 2 Cycle



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.37 Synchronous DRAM Self-Refresh Timing (TRP = 2 Cycle)



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.38 Synchronous DRAM Mode Register Write Timing (TRP = 2 C

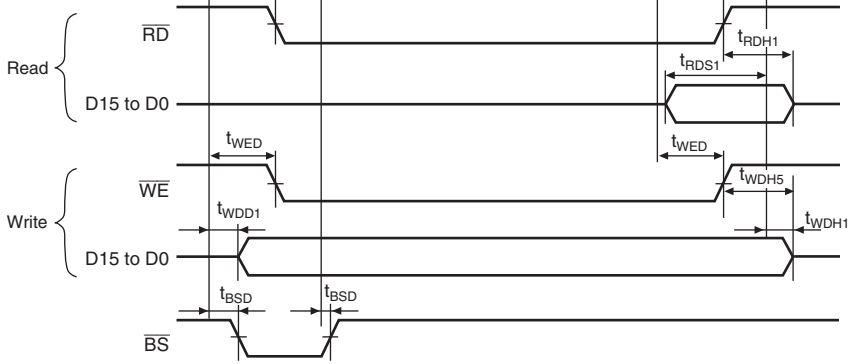


Figure 24.39 PCMCIA Memory Card Interface Bus Timing

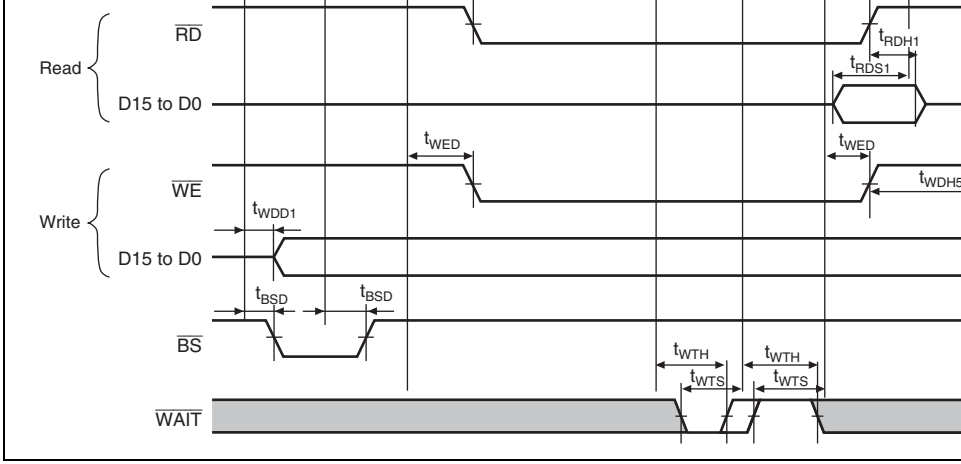


Figure 24.40 PCMCIA Memory Card Interface Bus Timing (TED[3:0] = B'0010, TEH[3:0] = B'0001, One Software Wait, One Hardware

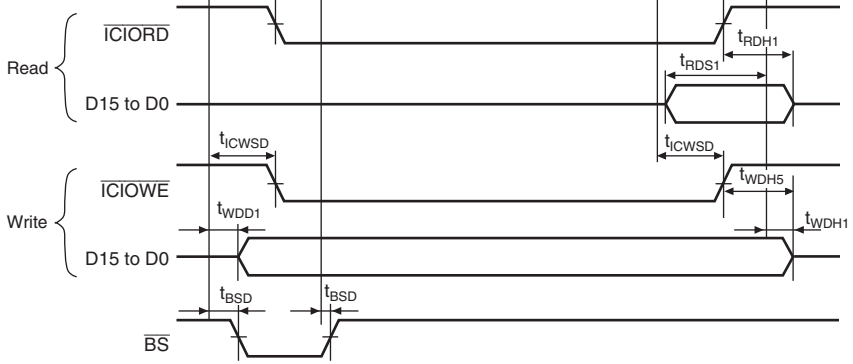


Figure 24.41 PCMCIA I/O Card Interface Bus Timing

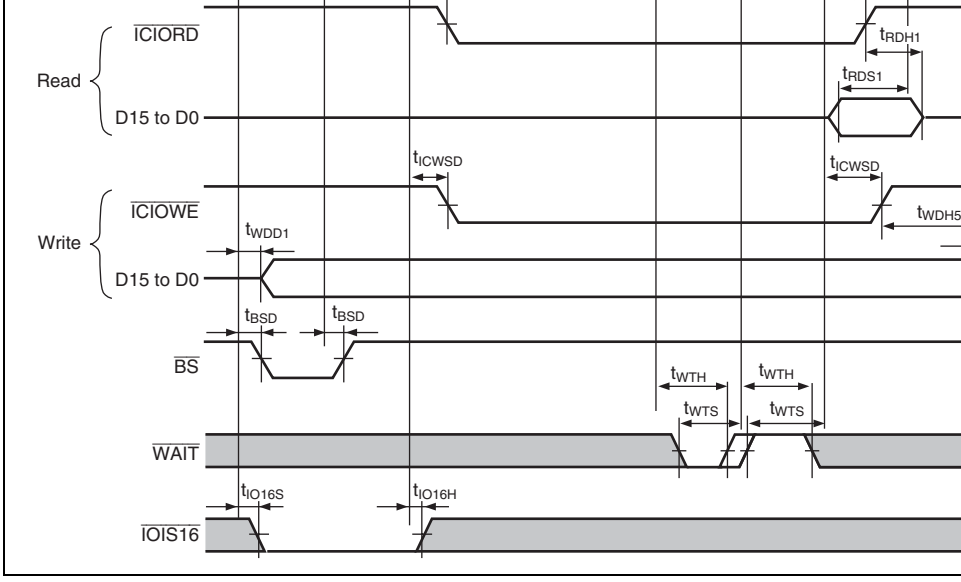


Figure 24.42 PCMCIA I/O Card Interface Bus Timing
(TED[3:0] = B'0010, TEH[3:0] = B'0001, One Software Wait, One Hardware

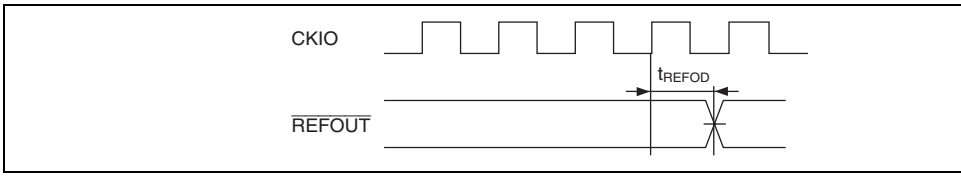
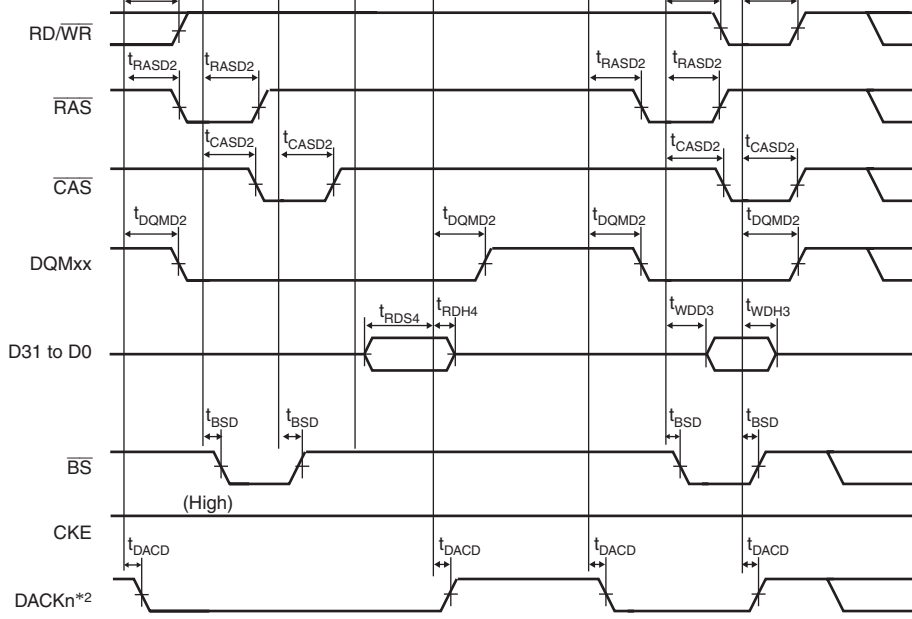


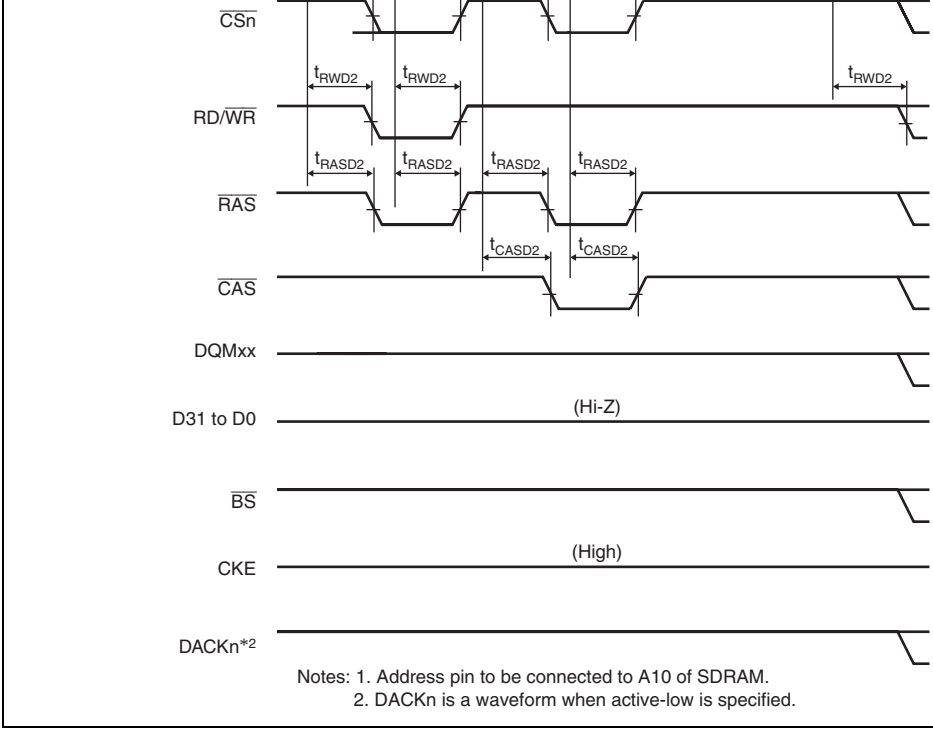
Figure 24.43 $\overline{\text{REFOUT}}$ Delay Time

Write data delay time 3	t_{WDD3}	—	1/2 t_{cyc}+12	24.44
Write data hold time 3	t_{WDH3}	1/2 t_{cyc}	—	24.44
$\overline{RAS}$ delay time 2	t_{RASD2}	1/2 t_{cyc}	1/2 t_{cyc}+10	24.44 to 24.47
$\overline{CAS}$ delay time 2	t_{CASD2}	1/2 t_{cyc}	1/2 t_{cyc}+10	24.44 to 24.47
DQM delay time 2	t_{DQMD2}	1/2 t_{cyc}	1/2 t_{cyc}+10	24.44
CKE delay time 2	t_{CKED2}	1/2 t_{cyc}	1/2 t_{cyc}+10	24.46



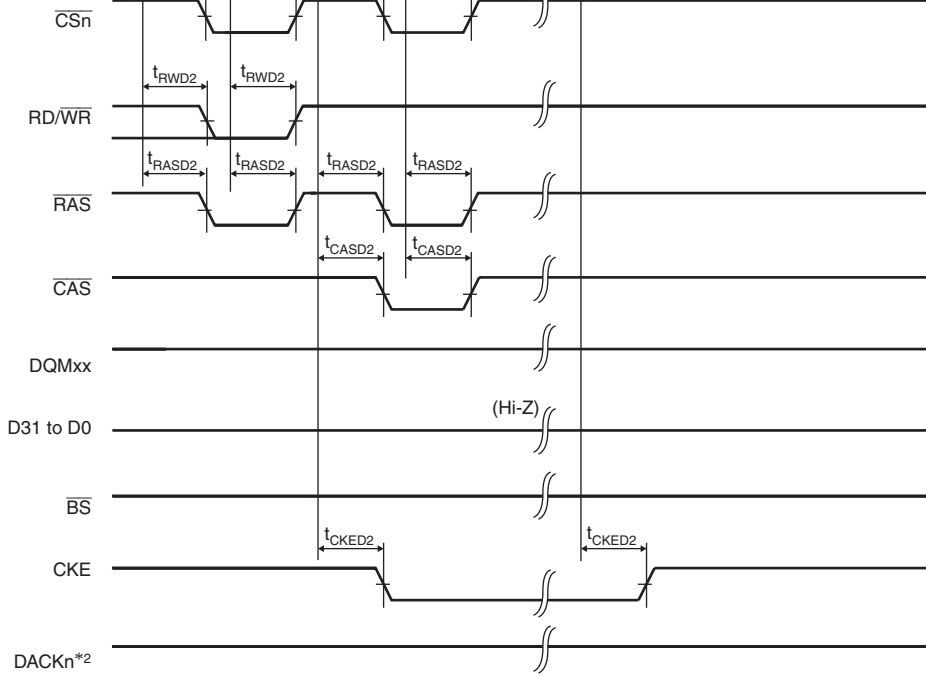
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.44 Access Timing in Low-Frequency Mode (Auto Precharge)



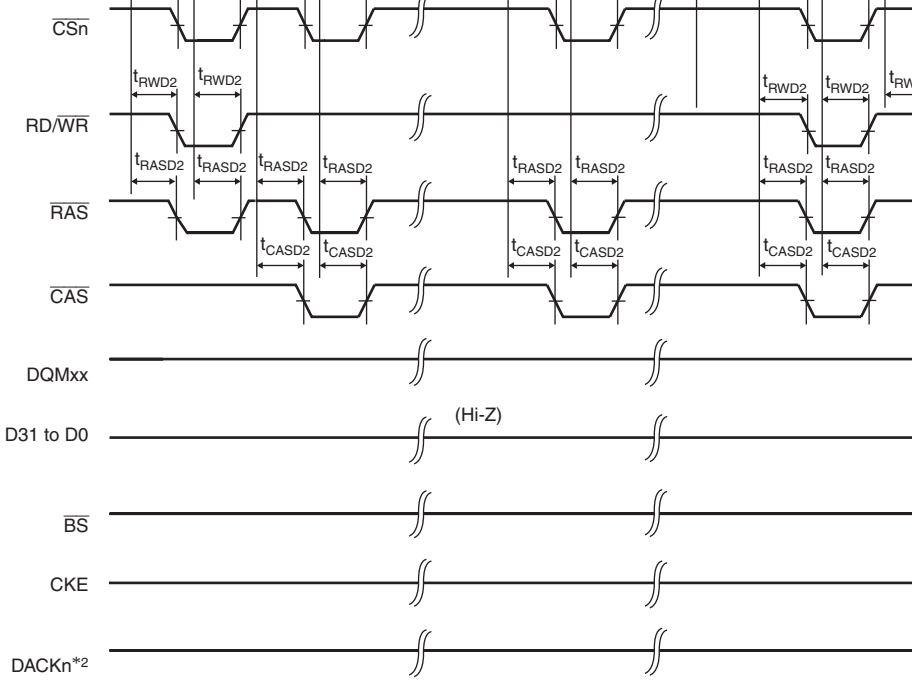
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.45 Synchronous DRAM Auto-Refresh Timing (TRP = 2 Cycle, Low-Frequency Mode)



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.46 Synchronous DRAM Self-Refresh Timing (TRP = 2 Cycle, Low-Frequency Mode)



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 24.47 Synchronous DRAM Mode Register Write Timing (TRP = 2 Cycle, Low-Frequency Mode)

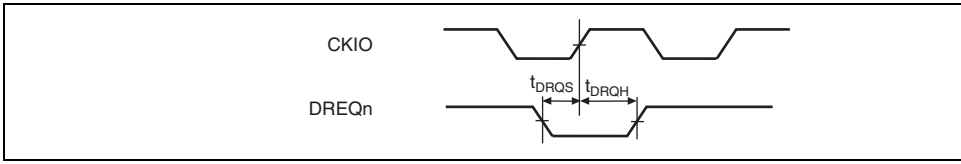


Figure 24.48 DREQn Input Timing

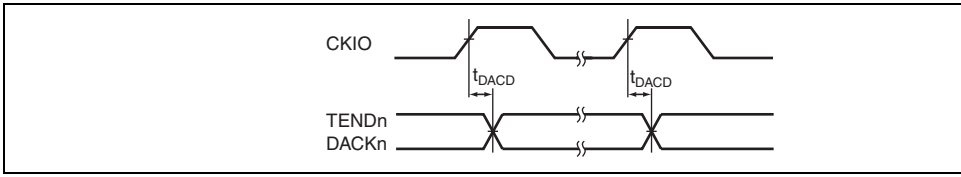


Figure 24.49 TENDn, DACKn Output Timing

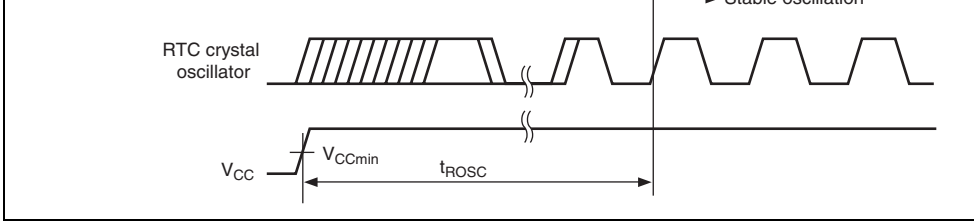


Figure 24.50 Oscillation Settling Time when RTC Crystal Oscillator is Turned

Input clock rise time	t_{SCKR}	—	1.5	
Input clock fall time	t_{SCKF}	—	1.5	
Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}
Transmission data delay time	t_{TXD}	—	3 t_{Pcyc}* + 50	ns
Receive data setup time (clock synchronization)	t_{RXS}	2 t_{Pcyc}*	—	
Receive data hold time (clock synchronization)	t_{RXH}	2 t_{Pcyc}*	—	
$\overline{RTS}$ delay time	t_{RTSD}	—	100	
$\overline{CTS}$ setup time (clock synchronization)	t_{CTSS}	100	—	
$\overline{CTS}$ hold time (clock synchronization)	t_{CTSH}	100	—	

Note: * t_{Pcyc} indicates a peripheral clock (Pϕ) cycle.

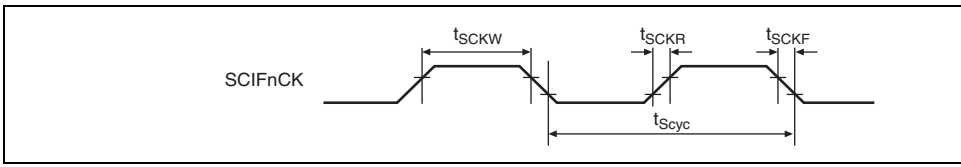


Figure 24.51 SCIFnCK Input Clock Timing

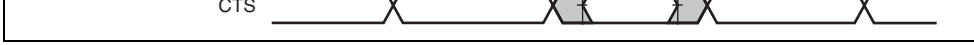


Figure 24.52 SCIF Input/Output Timing in Clock Synchronous Mode

24.3.10 SIOF Module Signal Timing

Table 24.12 SIOF Module Signal Timing

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$
$V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$

Item	Symbol	Min.	Max.	Unit	Figure
SIOMCLK clock input cycle time	t_{Mcyc}	30	—	ns	24.53
SIOMCLK input high-level width	t_{MWH}	$0.4 \times t_{Mcyc}$	—		
SIOMCLK input low-level width	t_{MWL}	$0.4 \times t_{Mcyc}$	—		
SCK_SIO clock cycle time	t_{SIcyc}	$2 \times t_{Pcyc}$	—		24.54 t
SCK_SIO output high-level width	t_{SWHO}	$0.4 \times t_{SIcyc}$	—		24.54 t
SCK_SIO output low-level width	t_{SWLO}	$0.4 \times t_{SIcyc}$	—		
SIOFSYNC output delay time	t_{FSD}	—	20		
SCK_SIO input high-level width	t_{SWHI}	$0.4 \times t_{SIcyc}$	—		24.58
SCK_SIO input low-level width	t_{SWLI}	$0.4 \times t_{SIcyc}$	—		
SIOFSYNC input setup time	t_{FSS}	20	—		
SIOFSYNC input hold time	t_{FSH}	20	—		
TXD_SIO output delay time	t_{STDD}	—	20		24.54 t
RXD_SIO input setup time	t_{SRDS}	20	—		
RXD_SIO input hold time	t_{SRDH}	20	—		

Note: t_{Pcyc} is the cycle time (ns) of the peripheral clock (Pϕ).

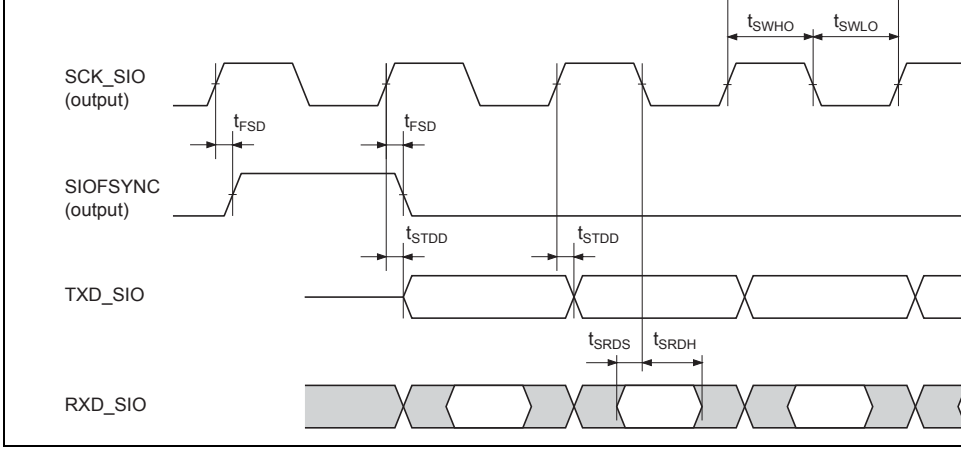


Figure 24.54 SIOF Transmit/Receive Timing (Master Mode 1: Fall Sampling

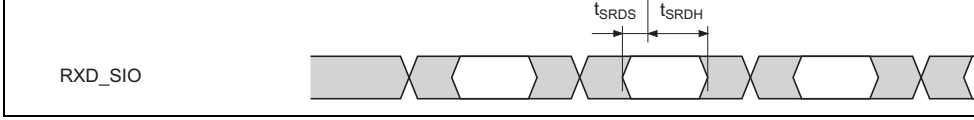


Figure 24.55 SIOF Transmit/Receive Timing (Master Mode 1: Rise Sampling T

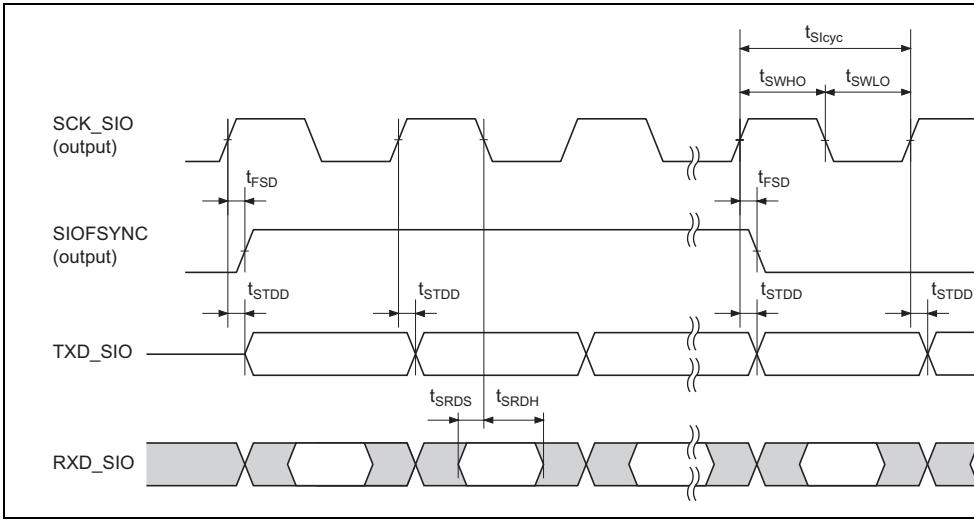


Figure 24.56 SIOF Transmit/Receive Timing (Master Mode 2: Fall Sampling T

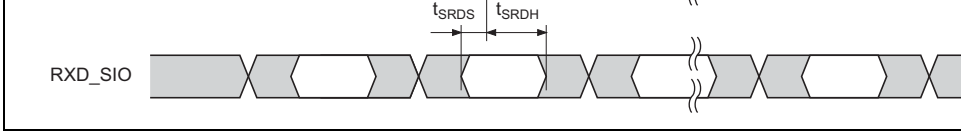


Figure 24.57 SIOF Transmit/Receive Timing (Master Mode 2: Rise Sampling

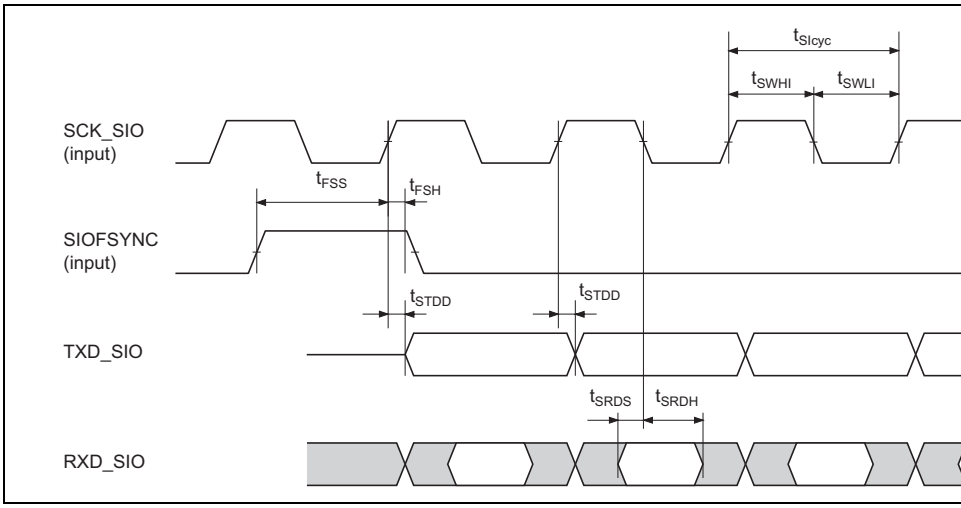


Figure 24.58 SIOF Transmit/Receive Timing (Slave Mode 1 and Slave Mod

CRS setup time	t_{CRSS}	10	—	—
CRS hold time	t_{CRSH}	10	—	—
COL setup time	t_{COLS}	10	—	—
COL hold time	t_{COLH}	10	—	—
RX-CLK cycle time	t_{Rcyc}	40	—	—
RX-DV setup time	t_{RDVS}	10	—	—
RX-DV hold time	t_{RDVH}	3	—	—
ERXD[3:0] setup time	t_{ERDS}	10	—	—
ERXD[3:0] hold time	t_{ERDH}	3	—	—
RX-ER setup time	t_{RERS}	10	—	—
RX-ER hold time	t_{RERH}	3	—	—
MDIO setup time	t_{MDIOS}	10	—	—
MDIO hold time	t_{MDIOH}	10	—	—
MDIO output data hold time*	t_{MDIODH}	5	—	18
WOL output delay time	t_{WOLD}	1	—	18
EXOUT output delay time	t_{EXOUTD}	1	—	28
CAMSEN setup time	t_{CAMS}	10	—	—
CAMSEN hold time	t_{CAMH}	3	—	—
$\overline{\text{ARBUSY}}$ output delay time	t_{ARBYD}	—	—	1/2 t_{cyc}+12

Note: * The user must ensure that the code satisfies this condition.

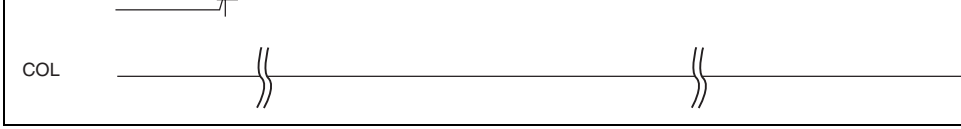


Figure 24.59 MII Transmit Timing (Normal Operation)

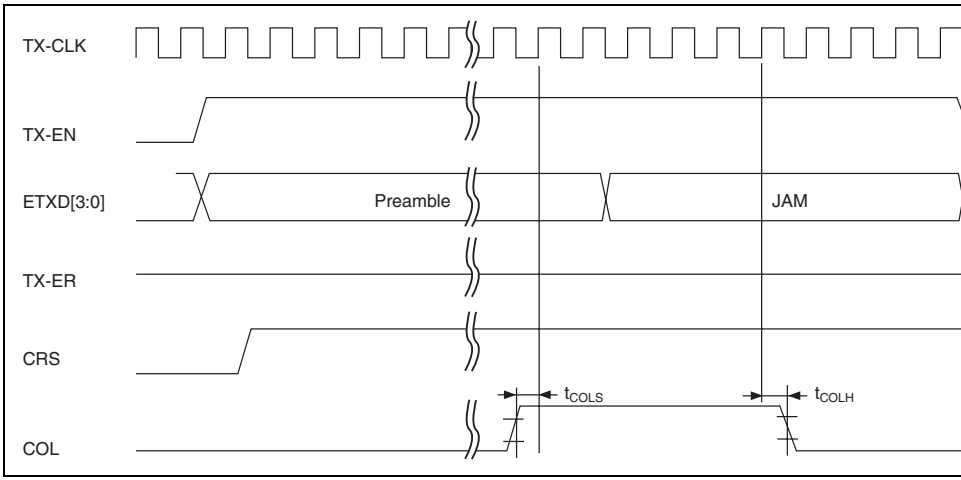


Figure 24.60 MII Transmit Timing (Case of Conflict)

Figure 24.61 MII Receive Timing (Normal Operation)

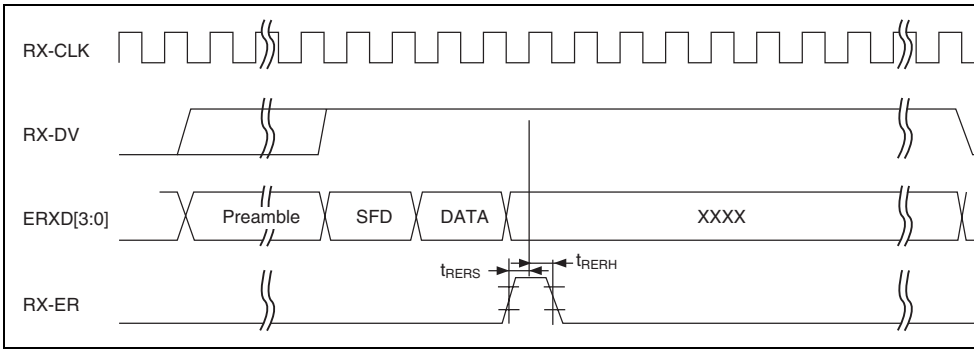


Figure 24.62 MII Receive Timing (Case of Error)

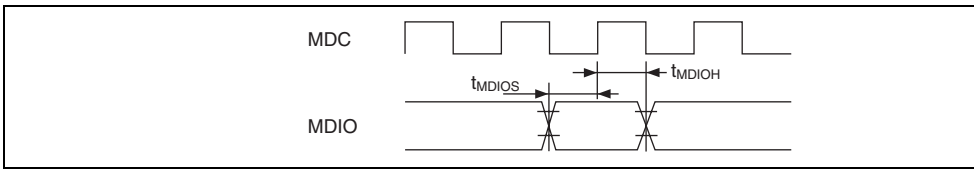


Figure 24.63 MDIO Input Timing

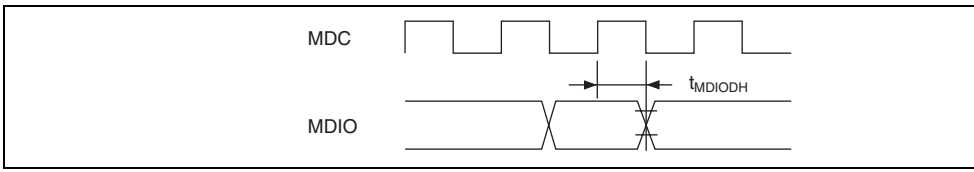


Figure 24.64 MDIO Output Timing

Figure 24.66 EXOUT Output Timing

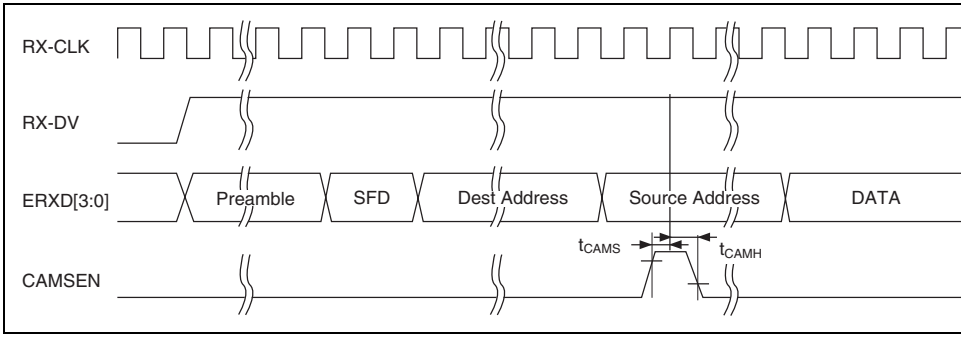


Figure 24.67 CAMSEN Input Timing

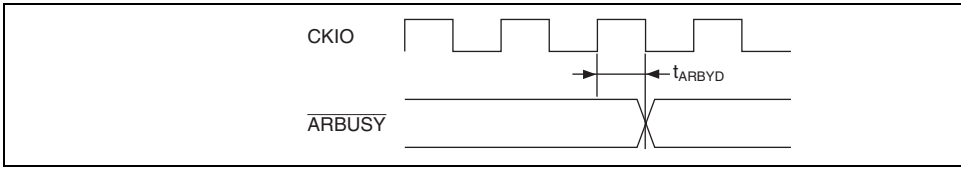


Figure 24.68 $\overline{\text{ARBUBY}}$ Output Timing

	B:P clock ratio = 4:1		$3 \times t_{cyc}+15$	—
Input data hold time		t_{PORTH}	8	—

Note: t_{cyc} is the output cycle time of the CKIO clock.

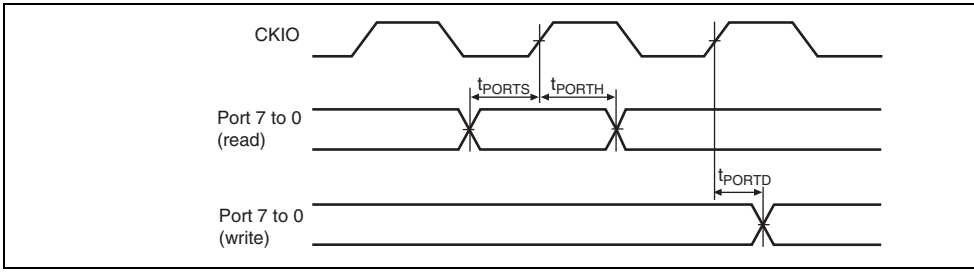


Figure 24.69 I/O Port Timing

TCK rise/fall time	t_{TCKR}, t_{TCKF}			ns
$\overline{TRST}$ setup time	t_{TRSTS}	12	—	ns
$\overline{TRST}$ hold time	t_{TRSTH}	50	—	t_{cyc}
TDI setup time	t_{TDIS}	10	—	ns
TDI hold time	t_{TDIH}	10	—	ns
TMS setup time	t_{TMSS}	10	—	ns
TMS hold time	t_{TMSH}	10	—	ns
TDO delay time	t_{TDOD}	—	15	ns
$\overline{ASEMD0}$ setup time	$t_{ASEMD0S}$	12	—	ns
$\overline{ASEMD0}$ hold time	$t_{ASEMD0H}$	12	—	ns
$\overline{ASEBRKAK}$ delay time	$t_{ASBRAKD}$	—	15	ns

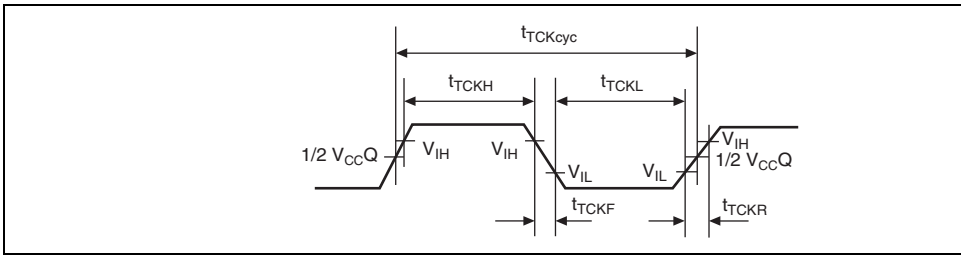


Figure 24.70 TCK Input Timing

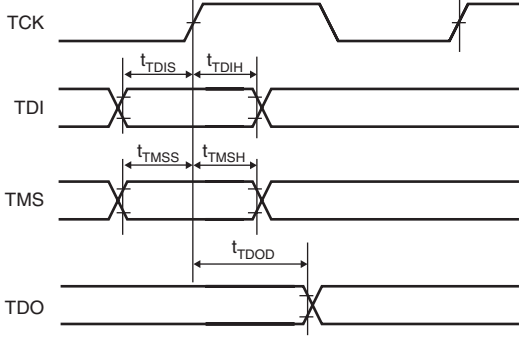


Figure 24.72 H-UDI Data Transfer Timing

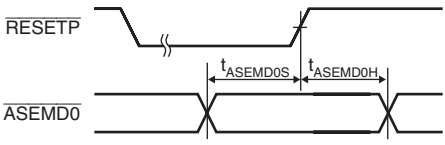


Figure 24.73 $\overline{\text{ASEMD0}}$ Input Timing

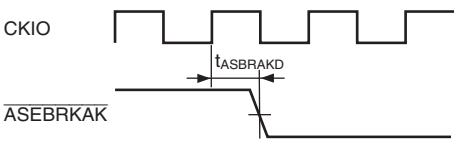
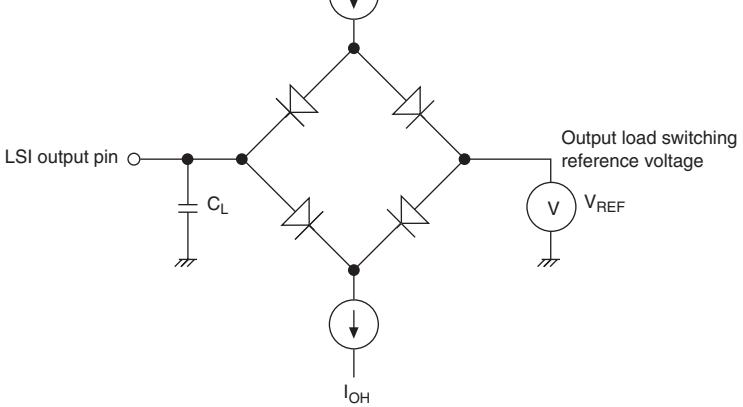


Figure 24.74 $\overline{\text{ASEBRKAK}}$ Delay Time



Notes: 1. C_L is the total value that includes the capacitance of measurement instruments, etc., and is set as follows for each pin.
30 pF: CKIO, $\overline{RAS}$, $\overline{CAS}$, $\overline{CS0}$, $\overline{CS2}$ to $\overline{CS6B}$, $\overline{BACK}$
50 pF: All other pins
2. I_{OL} = 2 mA, I_{OH} = –2 mA

Figure 24.75 Output Load Circuit

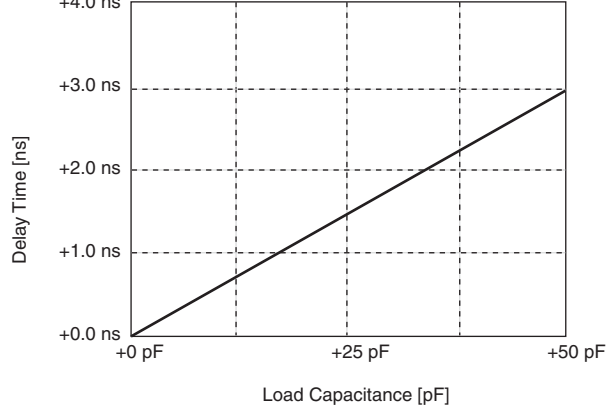


Figure 24.76 Load Capacitance vs. Delay Time

$\overline{BREQ}$	I	I	I	Z	I	I	Pull-
$\overline{BACK}$	O	O	O	Z	O	L	Ope
$\overline{CS0}$	O	H	H	HZ*[4]	O	Z	Ope
$\overline{CS4}$	O	H	H	HZ*[4]	O	Z	Ope
$\overline{CS5A}$	O	H	H	HZ*[4]	O	Z	Ope
$\overline{CS6A}$	O	H	H	HZ*[4]	O	Z	Ope
$\overline{WAIT}$	I	I	I	Z	I	Z	Pull-
$\overline{RD}$	O	H	H	HZ*[4]	O	Z	Ope
$\overline{BS}$	O	H	H	HZ*[4]	O	Z	Ope
D0	IO	Z	Z	Z	IO	Z	Pull-
D1	IO	Z	Z	Z	IO	Z	Pull-
D2	IO	Z	Z	Z	IO	Z	Pull-
D3	IO	Z	Z	Z	IO	Z	Pull-
D4	IO	Z	Z	Z	IO	Z	Pull-
D5	IO	Z	Z	Z	IO	Z	Pull-
D6	IO	Z	Z	Z	IO	Z	Pull-
D7	IO	Z	Z	Z	IO	Z	Pull-
D8	IO	Z	Z	Z	IO	Z	Pull-
D9	IO	Z	Z	Z	IO	Z	Pull-
D10	IO	Z	Z	Z	IO	Z	Pull-
D11	IO	Z	Z	Z	IO	Z	Pull-
D12	IO	Z	Z	Z	IO	Z	Pull-
D13	IO	Z	Z	Z	IO	Z	Pull-

$\overline{CAS}$	O	H	H	HZ*[4]	O	HZ*[4]	Open
CKE	O	H	O	HZ*[4]	O	OZ*[5]	Open
$\overline{RAS}$	O	H	H	HZ*[4]	O	HZ*[4]	Open
$\overline{CS2}$	O	H	H	HZ*[4]	O	Z	Open
$\overline{CS3}$	O	H	H	HZ*[4]	O	Z	Open
A0	O	O	O	OZ*[5]	O	Z	Open
A1	O	O	O	OZ*[5]	O	Z	Open
A2	O	O	O	OZ*[5]	O	Z	Open
A3	O	O	O	OZ*[5]	O	Z	Open
A4	O	O	O	OZ*[5]	O	Z	Open
A5	O	O	O	OZ*[5]	O	Z	Open
A6	O	O	O	OZ*[5]	O	Z	Open
A7	O	O	O	OZ*[5]	O	Z	Open
A8	O	O	O	OZ*[5]	O	Z	Open
A9	O	O	O	OZ*[5]	O	Z	Open
A10	O	O	O	OZ*[5]	O	Z	Open
A11	O	O	O	OZ*[5]	O	Z	Open
A12	O	O	O	OZ*[5]	O	Z	Open
A13	O	O	O	OZ*[5]	O	Z	Open
A14	O	O	O	OZ*[5]	O	Z	Open
A15	O	O	O	OZ*[5]	O	Z	Open
A16	O	O	O	OZ*[5]	O	Z	Open
A17	O	O	O	OZ*[5]	O	Z	Open

D19	IO	Z	Z	Z	IO	Z	Pull-
D20	IO	Z	Z	Z	IO	Z	Pull-
D21	IO	Z	Z	Z	IO	Z	Pull-
D22	IO	Z	Z	Z	IO	Z	Pull-
D23	IO	Z	Z	Z	IO	Z	Pull-
D24	IO	Z	Z	Z	IO	Z	Pull-
D25	IO	Z	Z	Z	IO	Z	Pull-
D26	IO	Z	Z	Z	IO	Z	Pull-
D27	IO	Z	Z	Z	IO	Z	Pull-
D28	IO	Z	Z	Z	IO	Z	Pull-
D29	IO	Z	Z	Z	IO	Z	Pull-
D30	IO	Z	Z	Z	IO	Z	Pull-
D31	IO	Z	Z	Z	IO	Z	Pull-
A18	O	O	O	OZ*[5]	O	Z	Oper
A19	O	O	O	OZ*[5]	O	Z	Oper
A20	O	O	O	OZ*[5]	O	Z	Oper
A21	O	O	O	OZ*[5]	O	Z	Oper
A22	O	O	O	OZ*[5]	O	Z	Oper
A23	O	O	O	OZ*[5]	O	Z	Oper
A24	O	O	O	OZ*[5]	O	Z	Oper
A25	O	O	O	OZ*[5]	O	Z	Oper
PTB0	IO	V	P*[2]Z	K*[3]Z	P*[2]	P*[2]Z	Oper
PTB1/$\overline{\text{CTS1}}$	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Oper

PTA0/RXD0	IO/I	V	P*[2]Z	K*[3]Z	P*[2]I	P*[2]Z	Open
PTA1/TXD0	IO/O	V	P*[2]Z	K*[3]Z	P*[2]O	P*[2]Z	Open
PTA2/SCIF0CK	IO/IO	V	P*[2]Z	K*[3]Z	P	P*[2]Z	Open
PTA3/SCK_SIO0	IO/IO	V	P*[2]I	K*[3]Z	P	P*[2]I	Open
PTA4/SIOMCLK0	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Open
PTA5/RXD_SIO0	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Open
PTA6/TXD_SIO0	IO/O	V	P*[2]O	K*[3]Z	P*[2]O	P*[2]O	Open
PTA7/SIOFSYNC0	IO/IO	V	P*[2]I	K*[3]Z	P	P*[2]I	Open
CRS1	I	I	I	Z	I	I	Pull-dov
COL1	I	I	I	Z	I	I	Pull-dov
ETXD13	O	O	O	O	O	O	Open
ETXD12	O	O	O	O	O	O	Open
ETXD11	O	O	O	O	O	O	Open
ETXD10	O	O	O	O	O	O	Open
TX-EN1	O	O	O	O	O	O	Open
TX-CLK1	I	I	I	Z	I	I	Pull-dov
TX-ER1	O	O	O	O	O	O	Open
RX-ER1	I	I	I	Z	I	I	Pull-dov
RX-CLK1	I	I	I	Z	I	I	Pull-dov
RX-DV1	I	I	I	Z	I	I	Pull-dov
ERXD10	I	I	I	Z	I	I	Pull-dov
ERXD11	I	I	I	Z	I	I	Pull-dov
ERXD12	I	I	I	Z	I	I	Pull-dov
ERXD13	I	I	I	Z	I	I	Pull-dov

CRS0	I	I	I	Z	I	I	Pull-
COL0	I	I	I	Z	I	I	Pull-
ETXD03	O	O	O	O	O	O	Oper
ETXD02	O	O	O	O	O	O	Oper
ETXD01	O	O	O	O	O	O	Oper
ETXD00	O	O	O	O	O	O	Oper
TX-EN0	O	O	O	O	O	O	Oper
TX-CLK0	I	I	I	Z	I	I	Pull-
TX-ER0	O	O	O	O	O	O	Oper
RX-ER0	I	I	I	Z	I	I	Pull-
RX-CLK0	I	I	I	Z	I	I	Pull-
RX-DV0	I	I	I	Z	I	I	Pull-
ERXD00	I	I	I	Z	I	I	Pull-
ERXD01	I	I	I	Z	I	I	Pull-
ERXD02	I	I	I	Z	I	I	Pull-
ERXD03	I	I	I	Z	I	I	Pull-
MDC0	O	O	O	O	O	O	Oper
MDIO0	IO	I	I	Z	I	I	Pull-
WOL0	O	O	O	O	O	O	Oper
LNKSTA0	I	I	I	Z	I	I	Pull-
EXOUT0/TEND0	O/O	O	O	O	O	O	Oper
CAMSEN0/IRQ4	I/I	I	I	ZI*[6]	I	I	Pull-
MD4	I	I	i	Z	I	i	Must

TDO	O	Z	Z	Z	O	Z	Open
$\overline{\text{TRST}}$	I	M	M	V	M	M	Must
TCK	I	M	M	V	M	M	Open
$\overline{\text{ASEBRKAK}}$	O	V	O	O	O	O	Open
$\overline{\text{AUDSYNC}}$	O	Z	O	O	O	O	Open
AUDCK	O	O	O	O	O	O	Open
AUDATA3	O	Z	O	O	O	O	Open
AUDATA2	O	Z	O	O	O	O	Open
AUDATA1	O	Z	O	O	O	O	Open
AUDATA0	O	Z	O	O	O	O	Open
$\overline{\text{RESETM}}$	I	I	I	I	I	I	Pull-u
$\overline{\text{RESETP}}$	I	I	I	I	I	I	Must
NMI	I	I	I	I	I	I	Pull-u
IRQ0/$\overline{\text{IRL0}}$	I	Z	I	I	I	I	Pull-u
IRQ1/$\overline{\text{IRL1}}$	I	Z	I	I	I	I	Pull-u
IRQ2/$\overline{\text{IRL2}}$	I	Z	I	I	I	I	Pull-u
IRQ3/$\overline{\text{IRL3}}$	I	Z	I	I	I	I	Pull-u
STATUS0	O	H	H	H	L	L	Open
STATUS1	O	H	H	L	H	L	Open
CKIO2	O	O	O	OZ*[5]	O	OZ*[5]	Open
DACK0	O	Z	O	Z	O	O	Open

PTC4/SIOFSYNC1	IO/IO	V	P*[2]I	K*[3]Z	P	P*[2]I	Oper
PTC5/$\overline{CE2A}$	IO/O	V	P*[2]O	K*[3]H	P*[2]O	P*[2]Z	Oper
PTC6/$\overline{CE2B}$	IO/O	V	P*[2]O	K*[3]H	P*[2]O	P*[2]Z	Oper
PTC7/$\overline{IOIS16}$	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Oper
CS5B/$\overline{CE1A}$	O/O	H	H	HZ*[4]	O	Z	Oper
CS6B/$\overline{CE1B}$	O/O	H	H	HZ*[4]	O	Z	Oper
MD0	I	I	i	I	I	i	Must
MD1	I	I	i	I	I	i	Must
MD2	I	I	i	I	I	i	Must
MD3	I	I	i	Z	I	i	Must
XTAL	O	O	O	O	O	O	Oper
EXTAL	I	I	I	I	I	I	Pull-
VccQ		—	—	—	—	—	VccQ
VssQ		—	—	—	—	—	VssQ
Vcc		—	—	—	—	—	Vcc
Vss		—	—	—	—	—	Vss
VccQ-RTC		—	—	—	—	—	VccQ
VssQ-RTC		—	—	—	—	—	VssQ
Vcc-PLL1		—	—	—	—	—	Vcc*

L: Low-level output
H: High-level output
Z: High impedance (input or output buffer off)
V: Input/output buffer off, pull-up on
M: Input buffer on, output buffer off, pull-up on
K: Output buffer on or input buffer off (pull-up on or off), depending on register setting
P: Input or output depending on register settings

Notes:

1. Depends on clock mode.
2. The state is P when the port function is used.
3. The state is K when the port function is used.
4. The state is Z or H depending on register settings.
5. The state is Z or O depending on register settings.
6. The state is Z when the Ethernet controller function is used.
7. To avoid the power friction, Vcc-PLL1, Vcc-PLL2, Vss-PLL1, Vss-PLL2, and ot
   and Vss should be wired in three independent patterns from the board power-s
   source.

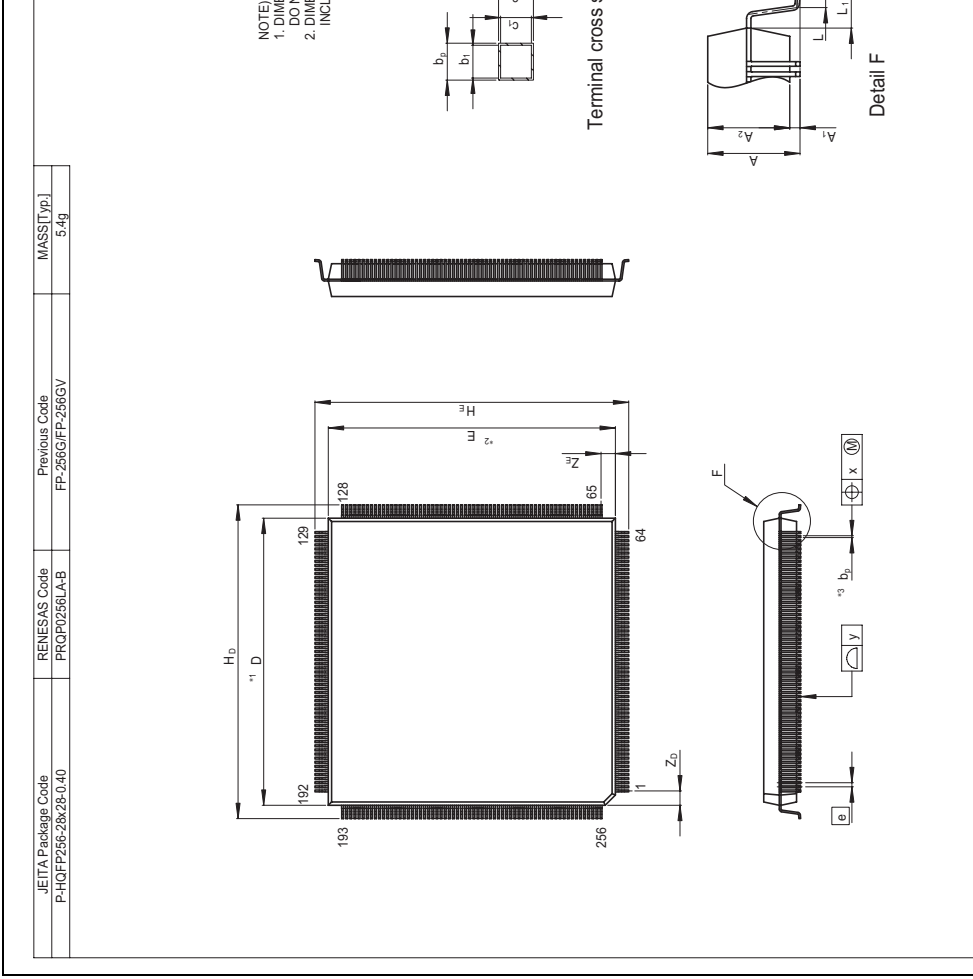
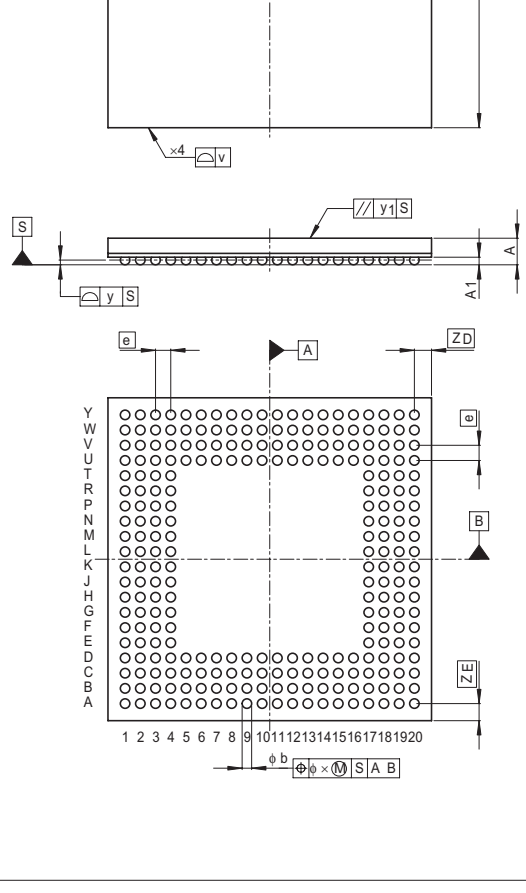


Figure B.1 Package Dimensions (HQFP2828-256 (FP-256G/GV))



Reference Symbol	Dimension in M	
	Min	Nom
D	——	17.0
E	——	17.0
v	——	——
w	——	——
A	——	——
A1	0.35	0.40
e	——	0.80
b	0.45	0.50
x	——	——
y	——	——
y1	——	——
SD	——	——
SE	——	——
ZD	——	0.9
ZE	——	0.9

Figure B.2 Package Dimensions (P-LFBGA1717-256 (BP-256H/HV))

S

T

V

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7712

Publication Date: Rev.1.00, Dec. 27, 2005
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

RENESAS SALES OFFICES

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 205, AZIA Center, No.133 Yincheng Rd (n), Pudong District, Shanghai 200120, China
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,
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7712
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0269-0100

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 8-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

CY8C20524-12PVXIT CY8C28433-24PVXIT MB95F012KPFT-G-SNE2 MB95F013KPMC-G-SNE2 MB95F263KPF-G-SNE2 MB95F264KPFT-G-SNE2 MB95F398KPMC-G-SNE2 MB95F478KPMC2-G-SNE2 MB95F562KPF-G-SNE2 MB95F564KPF-G-SNE2 MB95F634KPMC-G-SNE2 MB95F636KWQN-G-SNE1 MB95F696KPMC-G-SNE2 MB95F698KPMC1-G-SNE2 MB95F698KPMC2-G-SNE2 MB95F698KPMC-G-SNE2 MB95F818KPMC1-G-SNE2 MC908JK1ECDWER MC9S08PA32AVLD MC9S08PT60AVLD R5F1076CMSPV0 R5F5631ECDFBV0 C8051F389-B-GQ C8051F392-A-GMR ISD-ES1600_USB_PROG 901015X SC705C8AE0VFBE STM8TL53G4U6 PIC16F877-04/P-B R5F10Y17ASP#30 CY8C3MFIDOCK-125 403708R MB95F354EPF-G-SNE2 MB95F564KPFT-G-SNE2 MB95F564KWQN-G-SNE1 MB95F636KP-G-SH-SNE2 MB95F636KPMC-G-SNE2 MB95F694KPMC-G-SNE2 MB95F778JPMC1-G-SNE2 MB95F818KPMC-G-SNE2 MC908QY8CDWER MC9S08PT16AVLD MC9S08PT32AVLH MC9S08PT60AVLC MC9S08PT60AVLH C8051F500-IQR LC87F0G08AUJA-AH CP8361BT STM8S207C6T3 CG8421AF