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.

RENESAS

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

SH7615 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer

SuperH™ RISC engine Family/SH7600 Series

SH7615 HD6417615

Renesas Electronics
www.renesas.com

Rev.2.00 2005.03

1. These materials are intended as a reference to assist our customers in the selection of the R
   Technology Corp. product best suited to the customer's application; they do not convey any
   under any intellectual property rights, or any other rights, belonging to Renesas Technology
   a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of ar
   party's rights, originating in the use of any product data, diagrams, charts, programs, algorith
   circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, progra
   algorithms represents information on products at the time of publication of these materials, a
   subject to change by Renesas Technology Corp. without notice due to product improvement
   other reasons. It is therefore recommended that customers contact Renesas Technology C
   an authorized Renesas Technology Corp. product distributor for the latest product informatio
   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.co
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
   system before making a final decision on the applicability of the information and products. R
   Technology Corp. assumes no responsibility for any damage, liability or other loss resulting
   information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a d
   system that is used under circumstances in which human life is potentially at stake. Please
   Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor
   considering the use of a product contained herein for any specific purposes, such as appara
   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 c
   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 produ
   contained therein.

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If
are in their open states, intermediate levels are induced by noise in the vicinity
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 throug
chip and a low level is input on the reset pin. During the period where the stat
undefined, the register settings and the output state of each pin are also undefi
your system so that it does not malfunction because of processing while it is in
undefined state. For those products which have a reset function, reset the LSI
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
may have been be allocated to these addresses. Do not access these registers;
operation is not guaranteed if they are accessed.

5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

     The configuration of the functional description of each module differs accor
     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. I
section includes notes in relation to the descriptions given, and usage notes are given, a
as the 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 v
This does not include all of the revised contents. For details, see the actual locations in
manual.

11. Index

Objective: This manual was written to explain the hardware functions and electrica
characteristics of this LSI to the above users.
Refer to the SH-1/SH-2/SH-DSP Software Manual for a detailed descrip
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
SH7615	HD6417615

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorize
  on the CPU, system control functions, peripheral functions, and electrical characte
- In order to understand the details of the CPU's functions
  Read the SH-1/SH-2/SH-DSP Software 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 w
Please ensure you have the latest versions of all documents you re
(http://www.renesas.com/)

SH7615 manuals:

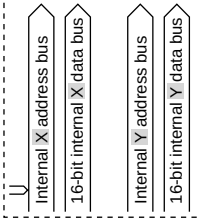
Document Title	Documen
SH7615 Hardware Manual	This manu
SH-1/SH-2/SH-DSP Software Manual	REJ09B0

Users manuals for development tools:

Document Title	Documen
SuperH RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B0
SuperH RISC engine High-performance Embedded Workshop User's Manual	REJ10B0
SuperH RISC engine High-performance Embedded Workshop, Tutorial	REJ10B0
SuperH RISC engine C/C++ Compiler Package Application Note	REJ05B0

bps	bit per second
BSC	Bus State Controller
CCN	Cache memory Controller
CMT	Compare Match Timer
CPG	Clock Pulse Generator
CPU	Central Processing Unit
DMAC	Direct Memory Access Controller
etu	Elementary Time Unit
FIFO	First-In First-Out
Hi-Z	High Impedance
H-UDI	High-performance User Debugging Interface
INTC	Interrupt Controller
IrDA	Infrared Data Association
JTAG	Joint Test Action Group
LQFP	Low Profile QFP
LRU	Least Recently Used
LSB	Least Significant Bit
MMU	Memory Management Unit
MPX	Multiplex
MSB	Most Significant Bit
PC	Program Counter
PFC	Pin Function Controller
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
RTC	Real Time Clock
SCIF	Serial Communication Interface with FIFO
SDRAM	Synchronous DRAM

WDT Watchdog Timer

		MPX
1.2 Block Diagram Figure 1.1 Block Diagram of SH7615	13	Figure 1.1 amended 
1.3.1 Pin Arrangement	14	Description modified Figure 1.2 shows the pin arrangement of the HD64 and HD6417615ARFV, and figure 1.3 shows the p arrangement of the HD6417615ARBP and HD6417615ARBPV.

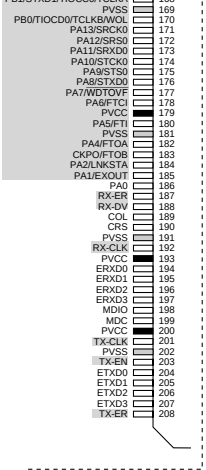


Figure 1.3 HD6417615ARBP and HD6417615ARBPV Pin Arrangement (BP-240A, BP-240AV)	15	Figure 1.3 added
1.3.2 Pin Functions Table 1.2 Pin Functions	18 to 20	Table 1.2 amended

Type	Symbol	I/O	Name	Function
Bus contro	$\overline{\text{BUSHiZ}}$	Input	Bus high impedance	Signal used in combinat signal to place bus and the high-impedance (Hi ending the bus cycle
Ethernet controller (EtherC)	ETXD0 to ETXD3	Output	Transmit data 0 to 3	4-bit transmit data
Serial communi-cation interface with FIFO (SCIF)	RXD1, RXD2	Input	Receive data input channel 1, 2	SCIF channel 1 and 2 r pins

1.3.3 Pin Multiplexing Table 1.3 Pin Multiplexing	22 to 27	Table 1.3 amended The pin numbers for the BP-240A and BP-240AV pa are added.

		accepted. If the $\overline{BRLS}$ signal continues to be as LSI remains in the bus-released state (asserts signal). When the $\overline{BRLS}$ signal is negated in the bus-re state during manual reset signal assertion, this using the bus (negates the $\overline{BGR}$ signal).
2.2.5 DSP Type Instructions and Data Formats Table 2.5 Destination Register Data Formats for DSP Instructions	49	Table 2.5 amended

Register	Instruction		Guard Bits 39 to 32	Re 31 to 16
A0, A1	Data transfer	MOVS.W	Sign extend	16-bit data
		MOVS.L		32-bit data
X0, X1, Y0, Y1, M0, M1	Data transfer	MOVX.W, MOVY.W, MOVS.W		16-bit data
		MOVS.L		32-bit data

2.6 Usage Notes	104 to 106	3. and 4. added
3.2.3 Connecting a Crystal Resonator Figure 3.2 Example of Crystal Oscillator Connection	112	Note 1 amended Note: 1. The CKIO pin is an output or high imped clock modes 0,1, and 2, and is high impedance in 3.
3.2.4 External Clock Input Figure 3.3 External Clock Input Method	113	Description amended The CKIO pin is an output or high impedance in cl 0, 1, and 2, and is high impedance in clock mode 3
3.2.7 Notes on Board Design	123	When Using an External Crystal Oscillator Description added Figure 3.5 shows an example of the oscillator circu sample oscillator circuit and in the actual system, t shown in the figure are affected by the environmen noise, power supply characteristics, or wiring patte values cannot be guaranteed and should be used reference values. To determine the optimum oscill constants for the user system, please consult with resonator manufacturer.

	124	Description added 2. V_{SS}/V_{CC} pairs for BP-240A and BP-240AV
5.3.29 IRQ Control/Status Register (IRQCSR)	189	Bits 15 to 8—IRQ Sense Select Bits (IRQ31S to IRQ

Bit table amended

Bits 15 to 8: IRQn1S	Bits 15 to 8: IRQn0S	Description
0	0	Low-level detection
	1	Falling-edge detection
1	0	Rising-edge detection
	1	Both-edge detection

7.2.1 Bus Control Register 1 (BCR1)	259	Bit 12—Endian Specification for Area 2 (A2ENDIAN

Note added

Note: Data rearrangement into little-endian format r
extra processing time.

	260	Bit 3—Endian Specification for Area 4 (A4ENDIAN)

Note: Data rearrangement into little-endian format r
extra processing time.

7.2.7 Individual Memory Control Register (MCR)	271	• For DRAM interface

Bit table added

Bit 1: TRP1	Bit 15: TRP0	Description
0	0	1 cycle
	1	2 cycles
1	0	Reserved (do not set)
	1	Reserved (do not set)

• For synchronous DRAM interface

Bit table added

Bit 1: TRP1	Bit 15: TRP0	Description
0	0	1 cycle
	1	2 cycles
1	0	3 cycles
	1	4 cycles

	274	• For synchronous DRAM interface

Bit table replaced

Except t_{Ecyc}:t_{Pcyc} 1:1 Figure 7.21 (a) Single Read Timing (Auto-Precharge) Except t_{Ecyc}:t_{Pcyc} 1:1 Figure 7.23 (a) Basic Burst Write Timing (Auto-Precharge) Except t_{Ecyc}:t_{Pcyc} 1:1 Figure 7.24 (a) Burst Read Timing (No Precharge) Except t_{Ecyc}:t_{Pcyc} 1:1 Figure 7.25 (a) Burst Read Timing (Bank Active, Same Row Address) Except t_{Ecyc}:t_{Pcyc} 1:1 Figure 7.26 (a) Burst Read Timing (Bank Active, Different Row Addresses) Except t_{Ecyc}:t_{Pcyc} 1:1		
7.5.11 64-Mbit Synchronous DRAM (2 Mword × 32 Bit) Connection	323	Synchronous DRAM Mode Settings Description amended Synchronous DRAM Mode Settings: To make moc for the synchronous DRAM, write to address X + H or X + H'FFFF8000 from the CPU. (X represents th value.) Whether to use X + H'FFFF0000 or X + H'F determines on the synchronous DRAM used.
Figure 7.35 128-Mbit Synchronous DRAM (4 Mwords × 32 Bits) Connection Example	324	• 128-Mbit Synchronous DRAM (4 Mwords × 32 B Connection Example Figure 7.35 added

Figure 7.37 256-Mbit Synchronous DRAM (8 Mwords × 32 Bits) Connection Example	326	• 256-Mbit Synchronous DRAM (8 Mwords × 32 Bit Connection Example Figure 7.37 added
7.11.3 Preventing Wrong Data Output to Synchronous DRAM	358	Section 7.11.3 added
8.2.1 Cache Control Register (CCR) Bit 4—Cache Purge Bit (CP)	361	Description added ... the CP bit reverts to 0. The CP bit always reads 0 cache to check if initialization is completed.
9.2.1 EtherC Mode Register (ECMR)	384	Bit 1—Duplex Mode (DM) Note added Note: When internal loopback mode is specified (IL full-duplex transfer (DM = 1) must be used. The duplex mode information (half-duplex or full-du detected by the PHY-LSI must be set to the DM bit. setting does not match the duplex mode in the PHY transfer rate may be degraded or a data collision ma
9.2.8 PHY Interface Status Register (PSR)	391	Note added Note: The LMON bit is cleared to 0 when the LNKS at a high level, and is set to 1 when the pin is at a lo
9.3.1 Transmission	405	Description amended 4. After waiting for the frame interval time (9.6 µs fo or 0.96 µs for 100Base), the transmitter enters the i and if there is more transmit data, continues transm
9.5 Usage Notes	416	Section 9.5 added
10.2.2 E-DMAC Transmit Request Register (EDTRR)	422	Note added For details on writing to the register, see section 10. Notes.
10.2.3 E-DMAC Receive Request Register (EDRRR)	423	Note added For details on writing to the register, see section 10. Notes.

Bit 26: TABT	Description
0	Transmit abort not detected
1	Transmit abort detected

This bit will be set when any one or more of the following bits are set.

EESR bit 12: Illegal Transmit Frame (ITF)
EESR bit 11: Carrier Not Detected (CND)
EESR bit 10: Detect Loss of Carrier (DLC)
EESR bit 9: Delayed Collision Detect (CD)

Bit 25— Receive abort detected (RABT): Indicates whether or not a receive abo

Bit 25: RABT	Description
0	Receive abort not detected
1	Receive abort detected

This bit will be set when any one or more of the following bits are set.

EESR bit 4: Receive Residual-Bit Frame (RRF)
EESR bit 3: Receive Too-Long Frame (RTLF)
EESR bit 2: Receive Too-Short Frame (RTSF)
EESR bit 1: PHY-LSI Receive Error (PRE)
EESR bit 0: CRC Error on Received Frame (CERF)

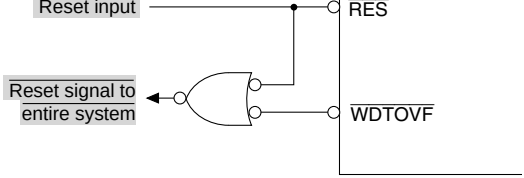
	430	Bit 9—Delayed Collision Detect (CD) Description amended

Bit 9—Delayed Collision Detect (CD): Indicates that a delayed collision has be frame transmission.

Bit 9: CD	Description
0	Delayed Collision not detected
1	Delayed Collision detected (interrupt source)

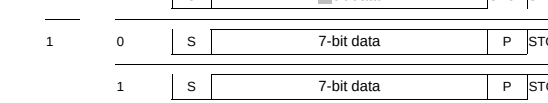
10.2.8 Transmit/Receive Status Copy Enable Register (TRSCER)	437	• General description of register modified • Bits 12 to 8, and 4 to 0 amended to reserved b • Bit 7 Multicast Address Frame Receive (RMAF Enable (RMAFCE) added.
10.2.10 Transmit FIFO Threshold Register (TFTR)	440, 441	Restriction added

		aligned with a 16-byte boundary when SDRAM is co
Transmit Descriptor 0 (TD0)	451	Bit 27—Transmit Frame Error (TFE) Description deleted Indicates that one or other bit of the transmit frame s indicated by bits 26 to 0 is set. ...
		Bits 26 to 0—Transmit Frame Status 26 to 0 (TFS26 • Descriptions of TFS26 to TFS5 replaced • Description of TFS1 amended Delayed Collision Detect in Transmission (corres CD bit in EESR)
Transmit Descriptor 2 (TD2)	452	Note replaced
Receive Descriptor		Notes replaced
	454	Bit 27—Receive Frame Error (RFE) Description amended ... indicated by bits 26 to 0 is set. Whether or not the address frame receive information, which is part of t frame status, is copied into this bit is specified by th transmit/receive status copy enable register.
		Bits 26 to 0—Receive Frame Status 26 to 0 (RFS26 Description of RFS8 amended
Receive Descriptor 2 (RD2)	455	Note replaced
10.4 Usage Notes	461	Newly added
11.2.4 DMA Channel Control Registers 0 and 1 (CHCR0, CHCR1) Bit 4—Transfer Bus Mode Bit (TB)	473	Description amended When 1 (burst mode) is set to bit TB, set 1 (edge de the DREQ select bit (DS).
11.5 Usage Notes	520 to 524	Notes 12, 13, 14, 15, and 16 added

Circuit for System Reset with $\overline{\text{WDTOVF}}$ Signal		
13.4.6 Internal Reset by Watchdog Timer (WDT) in Sleep Mode		Section 13.4.6 added
14.2.6 Serial Control Register (SCSCR)	574	Bit 6—Receive Interrupt Enable (RIE) Description amended Bit 6—Receive Interrupt Enable (RIE): Enables or generation of receive-FIFO-data full interrupt (RXI) error interrupt (ERI), and break interrupt (BRI) requ after serial receive data is transferred from the rec register (SCRSR) to the receive FIFO data registe (SCFRDR), the number of data bytes in SCFRDR exceeds the receive trigger set number, and the R set to 1 in SC1SSR.
14.2.9 Bit Rate Register (SCBRR) Table 14.3 Examples of Bit Rates and SCBRR Settings in Asynchronous Mode	587, 588	Table 14.3 amended

Pϕ (MHz)					
12			30		
n	N	Error (%)	n	N	Error (%)
2	212	0.03	3	132	0.13
2	155	0.16	3	97	–0.35
2	77	0.16	2	194	0.16
1	155	0.16	2	97	–0.35
1	77	0.16	1	194	0.16
0	155	0.16	1	97	–0.35
0	77	0.16	0	194	0.16
0	38	0.16	0	97	–0.35
0	19	–2.34	0	48	–0.35
0	11	0.00	0	29	0.00
0	9	–2.34	0	23	1.73

(Asynchronous Mode)



14.3.3 Multiprocessor Communication Function Figure 14.12 Sample Multiprocessor Serial Transmission Flowchart	613	Figure 14.12 amended

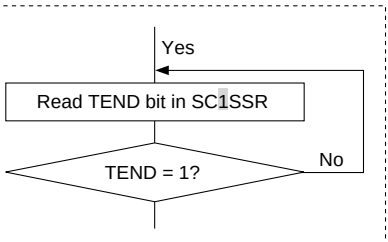
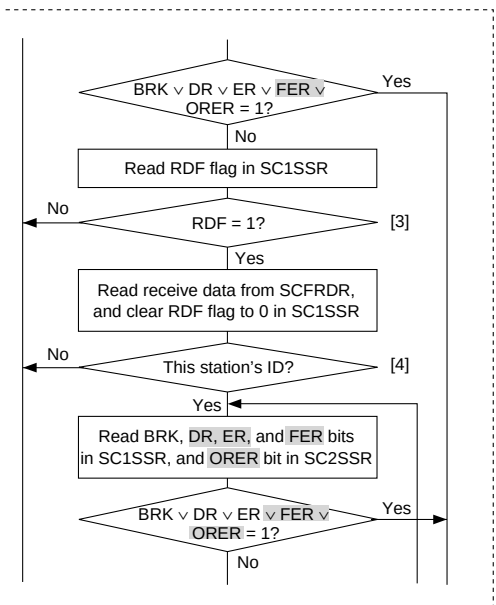
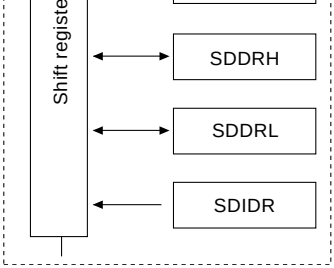


Figure 14.14 Sample Multiprocessor Serial Reception Flowchart (1)	616	Figure 14.14 (1) amended



14.3.4 Operation in Synchronous Mode	619	Description amended In synchronous mode, the SCIF receives data in synchronization with the rising edge of the serial clo



17.1.4 Register Configuration Table 17.2 Register Configuration	733	Table 17.2 amended

Register	Abbreviation	R/W*1	Initial Value*2	Address
Instruction register	SDIR	R	H'E000	H'FFFFFCE
Status register	SDSR	R/W	H'0401	H'FFFFFCE
Data register H	SDDRH	R/W	Undefined	H'FFFFFCE
Data register L	SDDRL	R/W	Undefined	H'FFFFFCE
ID code register	SDIDR	—	H'0101000F	—

17.3.2 Status Register (SDSR)	737	Bit table amended

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R

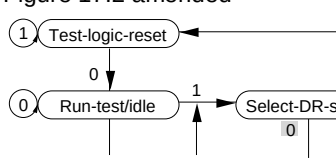
17.3.6 ID Code Register (SDIDR)	750	Description amended

The ID code register (SDIDR) is a 32-bit register. I
IDCODE mode, SDIDR can output H'0101000F, w
fixed code, from TDO.

17.4.1 TAP Controller	751	Description amended

Figure 17.2 shows the internal states of TAP contr
transitions basically conform with the IEEE1149.1

Figure 17.2 TAP Controller State Transitions		Figure 17.2 amended



- The registers are not initialized in standby mode.
  set to 0 in standby mode, IDCODE mode will be

18.1 Overview Table 18.1 Multiplex Pins	762	Table 18.1 amended

	Function 1 [00]*			Function 2 [01]*			Function 3 [10]*			Fu
Port	Signal Name	I/O	Related Module	Signal Name	I/O	Related Module	Signal Name	I/O	Related Module	Signa Name
B	PB3	I/O	Port	STCK1	I	SIO1	TIOCA0	I/O	TPU0	—
B	PB2	I/O	Port	STS1	I/O	SIO1	TIOCB0	I/O	TPU0	—
B	PB1	I/O	Port	STXD1	O	SIO1	TIOCC0	I/O	TPU0	—
B	PB0	I/O	Port	—	—	—	TIOCD0	I/O	TPU0	WOL

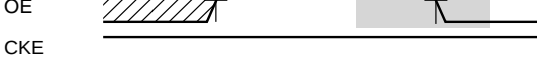
21.2 DC Characteristics Table 21.2 DC Characteristics	794	Table 21.2 amended

Item		Symbol	Min	Typ	Max	Unit	T
Three-state leakage current	All I/O and output pins (off status)	I_{TSI}	—	—	1.0	μA	V V V P
Output high voltage	Both 3.3 V and 5 V	V_{OH}	PV_{CC} – 0.7	—	—	V	I
	Other output pins		V_{CC} – 0.5	—	—	V	I
			V_{CC} – 1.0	—	—	V	I
Output low voltage	Both 3.3 V and 5 V	V_{OL}	—	—	0.6	V	I
	Other output pins		—	—	0.4	V	I

21.3.1 Clock Timing Table 21.5 Clock Timing	797	Table 21.5 amended

Item	Symbol	Min	Max	Un
CKPO clock output cycle time	t_{CKPCYC}	32	1000	ns
CKPO clock output low-level pulse width	t_{CKPOL}	11	—	ns
CKPO clock output high-level pulse width	t_{CKPOH}	11	—	ns
CKPO clock rise time	t_{CKPOr}	—	5	ns
CKPO clock fall time	t_{CKPOf}	—	5	ns
Power-on oscillation stabilization time	t_{OSC1}	10	—	ms
Standby recovery oscillation stabilization time 1	t_{OSC2}	10	—	ms
Standby recovery oscillation stabilization time 2	t_{OSC3}	10	—	ms
PLL synchronization stabilization time	t_{PLL}	1	—	ms

Figure 21.3 CKIO Clock Output Timing	798	Figure 21.3 amended (Before) V_{IH} → (After) V_{OH}
Figure 21.4 CKPO Clock Output Timing	799	Figure 21.4 added


OE
CKE

(Bank Active, Same Row Access, Except t_{Ecyc}:t_{Pcyc} 1:1) Figure 21.45 Interrupt Vector Fetch Cycle (No Wait, Except t_{Ecyc}:t_{Pcyc} 1:1) Figure 21.46 Interrupt Vector Fetch Cycle (External Wait Input, Except t_{Ecyc}:t_{Pcyc} 1:1) Figure 21.50 FRT Input/Output Timing (Except t_{Ecyc}:t_{Pcyc} 1:1) Figure 21.52 FRT Clock Input Timing (Except t_{Ecyc}:t_{Pcyc} 1:1) Figure 21.57 TPU Input/Output Timing (Except t_{Ecyc}:t_{Pcyc} 1:1) Figure 21.60 Watchdog Timer Output Timing (Except t_{Ecyc}:t_{Pcyc} 1:1) Figure 21.69 I/O Port Input/Output Timing (Except t_{Ecyc}:t_{Pcyc} 1:1) Figure 21.70 MII Transmit Timing (Normal Operation) Figure 21.71 MII Transmit Timing (Case of Conflict)	852, 854	
21.3.6 Serial Communication Interface Timing Table 21.10 Serial Communication Interface Timing	844	Table 21.10 amended (Before) t_{cscyc} → (After) t_{scyc}

21.3.12 STATS, BH, and BUSHiZ Signal Timing Table 21.17 STATS, BH, and BUSHiZ Signal Timing	856	Table 21.17 amended

Item	Symbol	Min	Typ	Max
STATS1 and STATS0 output delay time	t_{STAtd}	—	—	16

Figure 21.78 STATS Output Timing Figure 21.79 BH Output Timing	856	Figure 21.78 and Figure 21.79 amended (Before) G-DMAC → (After) DMAC
A.1 Addresses	859	SICTR2 amended

Address	Register Name	Bit Names						
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
H'FFFFFC24	SICTR2	—	—	—	—	—	—	—
H'FFFFFC25		—	TM	SE	DL	TIE	RIE	TE

	862	SCSMR1 amended

Address	Register Name	Bit Names						
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
H'FFFF FCC0	SCSMR1	C/Ā	CHR/ICK3	PE/ICK2	O/Ē/ICK1	STOP/ICK0	MP	CKS1

	864	EESR amended

Address	Register Name	Bit Names						
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
H'FFFF FD14	EESR	—	—	—	—	—	TABT	RABT
H'FFFF FD15		—	ECI	TC	TDE	TFUF	FR	RDE
H'FFFF FD16		—	—	—	ITF	CND	DLC	CD
H'FFFF FD17		RMAF	—	—	RRF	RTLF	RTSF	PRE

TRSCER amended

Address	Register Name	Bit Names						
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
H'FFFF FD1C	TRSCER	—	—	—	—	—	—	—
H'FFFF FD1D		—	—	—	—	—	—	—
H'FFFF FD1E		—	—	—	—	—	—	—
H'FFFF FD1F		RMAFCE	—	—	—	—	—	—

H'FFFFFE72 (Before) DCDR1 → (After) DRCR1

872 BBRC amended

Address	Register Name	Bit Names							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
H'FFFFFF48	BBRC	—	—	—	—	—	—	XYEC	
H'FFFFFF49		CPC1	CPC0	IDC1	IDC0	RWC1	RWC0	SZC1	

876 MCR amended

Address	Register Name	Bit Names							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
H'FFFFFFEC	MCR	TRP0	RCD0	TRWL0	TRAS1	TRAS0	BE	RASD	
H'FFFFFFED		AMX2	SZ	AMX1	AMX0	RFSH	RMODE	TRP1	

B.1 Pin States in Reset, Power-Down State, and Bus-Released State

880 Table amended

Pin Type	Pin Name	Pin State					
			Manual Reset		Power-Down Sta		
		Power-On Reset	Bus Acquired	Bus Released	Standby Mode (HIZ = 0)	Standby Mode (HIZ = 1)	S M
EtherC	ETXD3 to ETXD0	O	O	O	O	O	O
	ERXD3 to ERXD0	I	I	I	I	I	I

Appendix C Product Lineup

Table C.1 SH7615 Product Lineup

881 Table amended

Abbreviation	Voltage	Operating Frequency	Mark Code
SH7615	V_{cc} = $PLLV_{cc}$ = 3.3 V	62.5 MHz	HD6417615ARF
	PV_{cc} = 5.0 V/3.3 V		HD6417615ARFV
			HD6417615ARBP
			HD6417615ARBPV

Appendix D Package Dimensions

882 Description amended

Figure D.1 shows the FP-208C and FP-208CV pack dimensions, and figure D.2 shows the BP-240A and 240AV package dimensions.

Figure D.1 Package Dimensions (FP-208C, FP-208CV)

882 Figure D.1 replaced

Figure D.2 Package Dimensions (BP-240A, BP-240AV)

883 Figure D.2 added

The CPU has a RISC (Reduced Instruction Set Computer) type instruction set. The CP
operates at a rate of one instruction per cycle, offering a great improvement in instruct
execution speed. In addition, the 32-bit internal architecture provides improved data p
power, and DSP functions have also been enhanced with the implementation of extend
architecture DSP data bus functions. With this CPU, it has become possible to assemb
high-performance/high-functionality systems even for applications such as realtime co
could not previously be handled by microcomputers because of their high-speed proce
requirements. The SH7615 also includes a maximum 4-kbyte cache, for greater CPU p
power when accessing external memory.

The SH7615 is equipped with a media access controller (MAC) conforming to the IEE
standard, and an Ethernet controller that includes a media independent interface (MII)
unit, enabling 10/100 Mbps LAN connection. Supporting functions necessary for syste
configuration are also provided, including RAM, timers, a serial communication interf
FIFO (SCIF), interrupt controller (INTC), and I/O ports.

— Ten 32-bit system registers

- RISC (Reduced Instruction Set Computer) type instruction set
  - — Fixed 16-bit instruction length for improved code efficiency
  - — Load-store architecture (basic operations are executed betw registers)
  - — Delayed branch instructions reduce pipeline disruption duri branches
  - — C-oriented instruction set
- Instruction execution time: One instruction per cycle (16.0 ns/in 62.5 MHz operation)
- Address space: Architecture supports 4 Gbytes
- On-chip multiplier: Multiply operations (32 bits × 32 bits → 64 b multiply-and-accumulate operations (32 bits × 32 bits + 64 bits executed in two to four cycles
- Five-stage pipeline

— Single-cycle multiplier

- DSP registers
  - — Two 40-bit data registers
  - — Six 32-bit data registers
  - — Modulo register (MOD, 32 bits) added to control registers
  - — Repeat counter (RC) added to status register (SR)
  - — Repeat start register (RS, 32 bits) and repeat end register added to control registers
- DSP data bus
  - — Extended Harvard architecture
  - — Simultaneous access to two data buses and one instructio
- Parallel processing
  - — Maximum of four parallel processes
  - — ALU operations, multiplication, and two loads or stores
- Address processors
  - — Two address processors
  - — Address operations to access two memories
- DSP data addressing modes
  - — Increment and index
  - — Each with or without modulo addressing
- Repeat control: Zero-overhead repeat (loop) control
- Instruction set
  - — 16-bit length (in case of load or store only)
  - — 32-bit length (including ALU operations and multiplication)
  - — Added SuperH microcomputer instructions for accessing D registers
- Fifth and last pipeline stage is DSP stage

	• LRU replacement algorithm • Can also be used as 2-kbyte cache and 2-kbyte RAM (2-way c • Mixed instruction/data cache, instruction cache, or data cache be set • 1-cycle reads, 2-cycle writes (in write-back mode)
Interrupt controller (INTC)	• 16 priority levels can be set • On-chip supporting module interrupt vector numbers can be se • 41 internal interrupt sources • The E-DMAC interrupt (EINT) is input to the INTC as the OR of and E-DMAC interrupt sources (max.). Thus, from the viewpoin INTC, there is one EtherC/E-DMAC interrupt source. • Five external interrupt pins (NMI, $\overline{\text{IRL0}}$ to $\overline{\text{IRL3}}$) 15 external interrupt sources (encoded input) can also be selec $\overline{\text{IRL0}}$ to $\overline{\text{IRL3}}$ (IRL interrupts) • IRL interrupt vector number setting can also be selected (selec vector or external vector) • Provision for IRQ interrupt setting (low-level, rising-edge, falling both-edge detection)

- User break interrupt generated on occurrence of break condit
- Processing can be stopped before or after instruction executio
  instruction fetch cycle
- Break with specification of number of executions (channels C
  Settable number of executions: max. 2^{12} – 1 (4095)
- PC trace function
  Branch source/branch destination can be traced in branch ins
  (max. 8 addresses (4 pairs))

— Wait state insertion control for each area
— Control signal output for each area
— Endian can be set for CS2 and CS4

- Cache
  — Cache area/cache-through area selection by access addres
  — Selection of write-through or write-back mode
- Refresh functions
  — CAS-before-RAS refreshing (auto refreshing) or self-refresh
  — Refresh interval settable by means of refresh counter and c setting
  — Concentrated refreshing according to refresh count setting (1, 2, 4, 6, 8)
  — Refresh request output possible (REFOUT)
- Direct DRAM interface
  — Multiplexed row address/column address output
  — Fast page mode burst transfer and continuous access wher
  — EDO mode
  — TP cycle generation to secure RAS precharge time
- Direct synchronous DRAM interface
  — Multiplexed row address/column address output
  — Bank-active mode (valid for CS3 only)
  — Selection of burst read/single write mode or burst read/burs mode
- Bus arbitration ($\overline{\text{BRLS}}$, $\overline{\text{BGR}}$)
- Refresh counter can be used as interval timer
  — Interrupt request generated on compare match (CMI interru signal)

	— When synchronous DRAM is connected, 16-byte continuo continuous write transfer is available (dual) — When synchronous DRAM is connected, single-address tr available in a single clock cycle at maximum 31.25 MHz • Cycle stealing or burst transfer • Relative channel priorities can be set (fixed mode/round robin • DMA transfer is possible for the following devices: — External memory, on-chip memory, on-chip supporting mo (excluding DMAC, BSC, UBC, cache, E-DMAC, EtherC) • External requests, DMA transfer requests from on-chip suppo modules, auto requests • Interrupt request (DEIn) can be issued to CPU at end of data • DACK used for DREQ sampling (however, there is always on there is one acceptance before first DACK)
On-chip RAM	• 4-kbyte X-RAM • 4-kbyte Y-RAM
Ethernet controller direct memory access controller (E-DMAC), 2 channels	• Transfer possible between EtherC and external memory/on-cl • 16-byte burst transfer possible • Single address transfer • Chain block transfer • 32-bit transfer data width • 4-Gbyte address space

	— Transmitting and receiving short and long packets • Compatible with MII (Media Independent Interface) standard — Converts 8-bit stream data from MAC level to MII nibble str — Station management (STA) functions — 18 TTL-level signals — Variable transfer rate: 10/100 Mbps • Magic Packet™* (with WOL (Wake On LAN) output)
Serial communi-cation interface with FIFO (SCIF), 2 channels	• Asynchronous mode — Data length: 7 or 8 bits — Stop bit length: 1 or 2 — Parity: Even, odd, or none — Receive error detection: Parity errors, framing errors, overr — Break detection • Synchronous mode — One serial communication format (8-bit data length) — Receive error detection: Overrun errors • IrDA mode (conforming to IrDA 1.0) • Simultaneous transmission/reception (full-duplex) capability — Half-duplex communication used for IrDA communication • Built-in dedicated baud rate generator allows selection of bit ra • Built-in 16-stage transmit and receive FIFOs enable high-speed continuous communication • Internal or external (SCK) transmit/receive clock source

Note: * Magic Packet is a registered trademark of Advanced Micro Devices, Inc.

	of receive FIFO register transmit data errors • Timeout error (DR) can be detected during reception
Serial I/O (SIO), 3 channels	• Full-duplex operation (independent transmit and receive regis independent transmit and receive clocks) • Transmit/receive ports with double-buffer structure (enabling c transmission/reception) • Interval transfer mode and continuous transfer mode • Choice of 8- or 16-bit data length • Data transfer communication by means of polling or interrupts • MSB-first transfer between SIO and data I/O
High-performance user debugging interface (H-UDI)	• Conforms to IEEE1149.1 standard — Five test signals (TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$) — TAP controller — Instruction register — Data register — Bypass register • Test mode that conforms to the IEEE1149.1 standard — Standard instructions: BYPASS, SAMPLE/PRELOAD, and — Optional instructions: CLAMP, HIGHZ, and IDCODE • H-UDI interrupt — H-UDI interrupt request to INTC • Reset hold

or input capture

— Synchronous operation: Multiple timer counters (TCNT) car
to simultaneously; simultaneous clearing by compare match
capture possible; simultaneous register input/output possibl
counter synchronous operation

— PWM mode: Any PWM output duty can be set; maximum 7
PWM output possible by combination with synchronous ope

- Buffer operation settable for channel 0
  — Input capture register double-buffering possible
  — Automatic rewriting of output compare register possible
- Phase counting mode settable independently for channels 1 ar
  — Two-phase encoder pulse up/down-count possible
- 13 interrupt sources
  — For channel 0, four compare match/input capture dual-func
  interrupts and one overflow interrupt can be requested inde
  — For channels 1 and 2, two compare match/input capture du
  interrupts, one overflow interrupt, and one underflow interru
  requested independently

	— Counter value can be cleared by compare match A • Four interrupt sources — Two compare match sources (OCIA, OCIB) — One input capture source (ICI) — One overflow source (OVI)
Watchdog timer (WDT), 1 channel	• Can be switched between watchdog timer mode and interval t • Internal reset, external signal ($\overline{\text{WDTOVF}}$), or interrupt generat overflow • Used when standby mode is cleared or the clock frequency is and in clock pause mode • Selection of eight counter input clocks
Clock pulse generator (CPG)	• Built-in clock pulse generator • Selection of crystal or external clock as clock source • Built-in clock-multiplication PLL circuits • Built-in PLL circuit for phase synchronization between externa internal clock • CPU/DSP core clock (Iϕ), peripheral module clock (Pϕ), and e interface clock (Eϕ) frequencies can be scaled independently

	— Module standby function: Operation of FRT, SCIF, DMAC, TPU, and SIO on-chip supporting modules is halted selecti
I/O ports	• 29 input/output ports

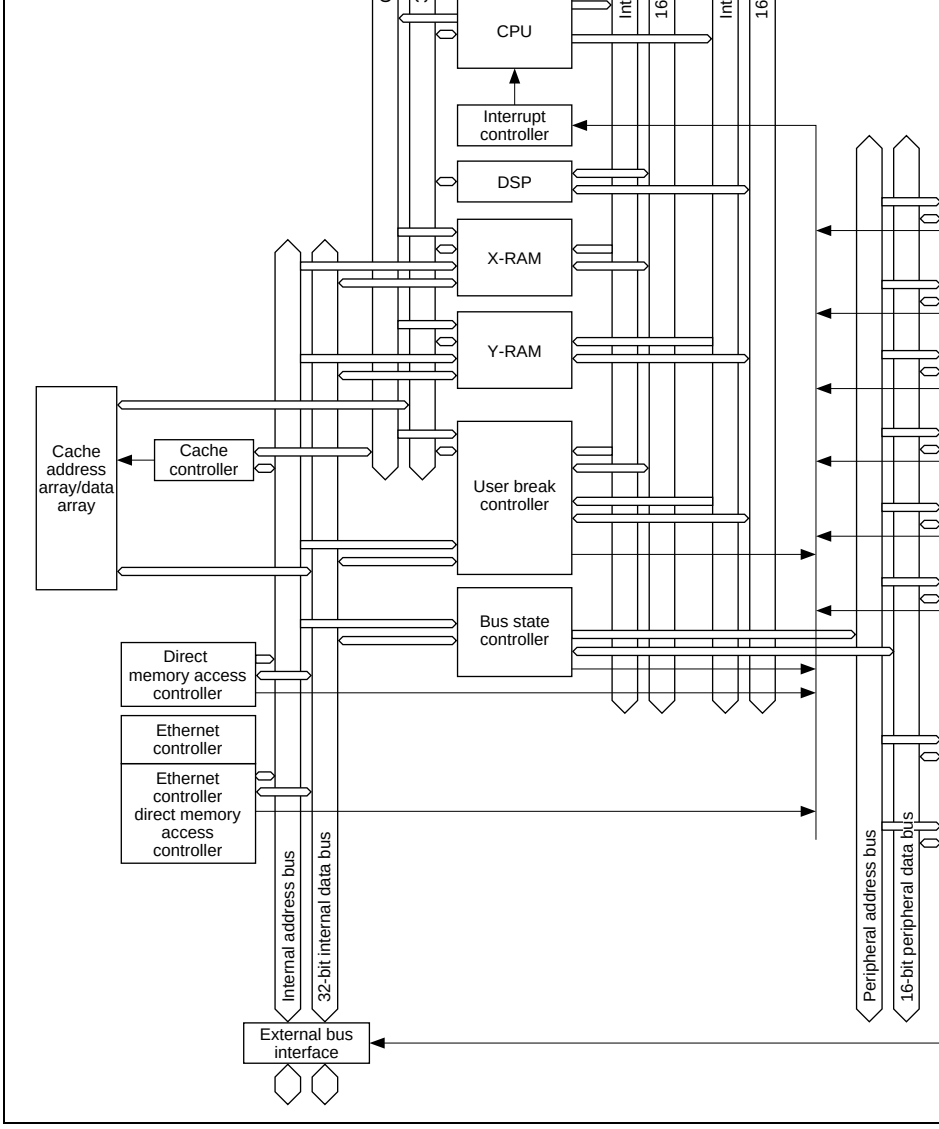
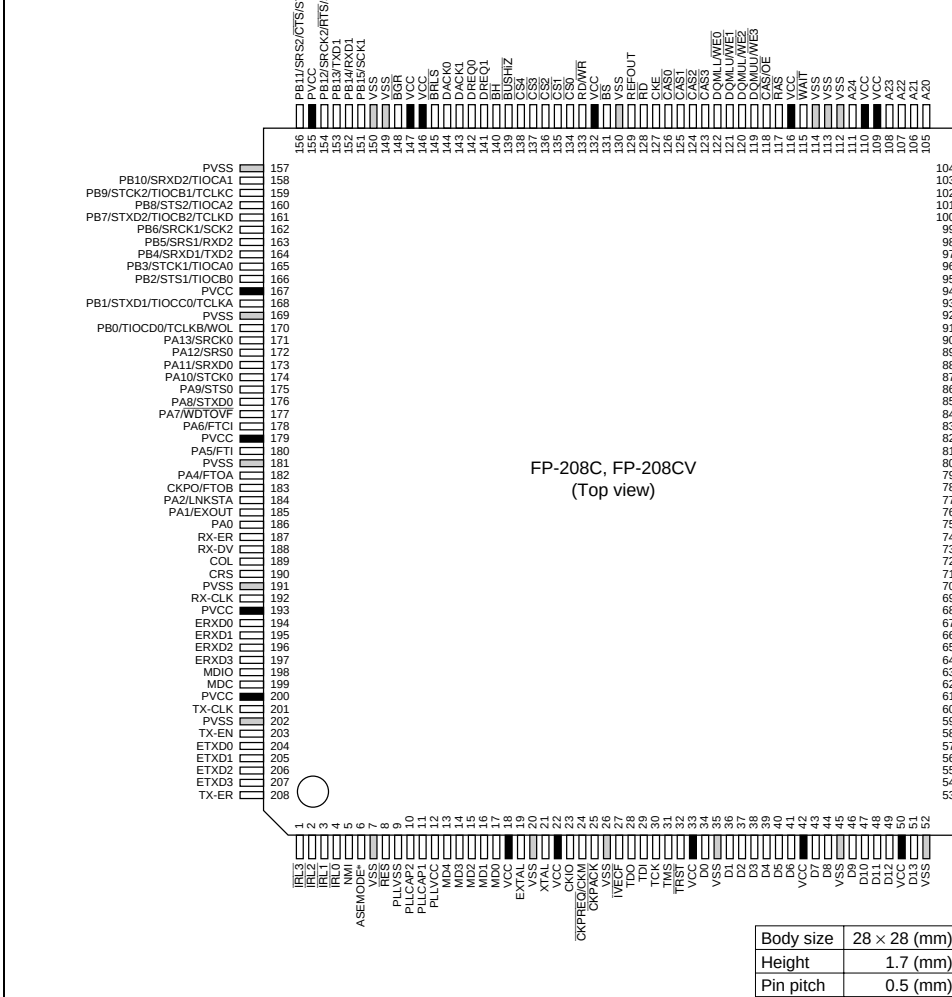


Figure 1.1 Block Diagram of SH7615



Body size	28 × 28 (mm)
Height	1.7 (mm)
Pin pitch	0.5 (mm)

Note: * When doing debugging using the E10A emulator, this pin is used for mode switching. It should be connected
when using the E10A emulator and connected to Vcc when using a normal user system.

Figure 1.2 HD6417615ARF and HD6417615ARFV Pin Arrangement (FP-208C, FP-208CV)

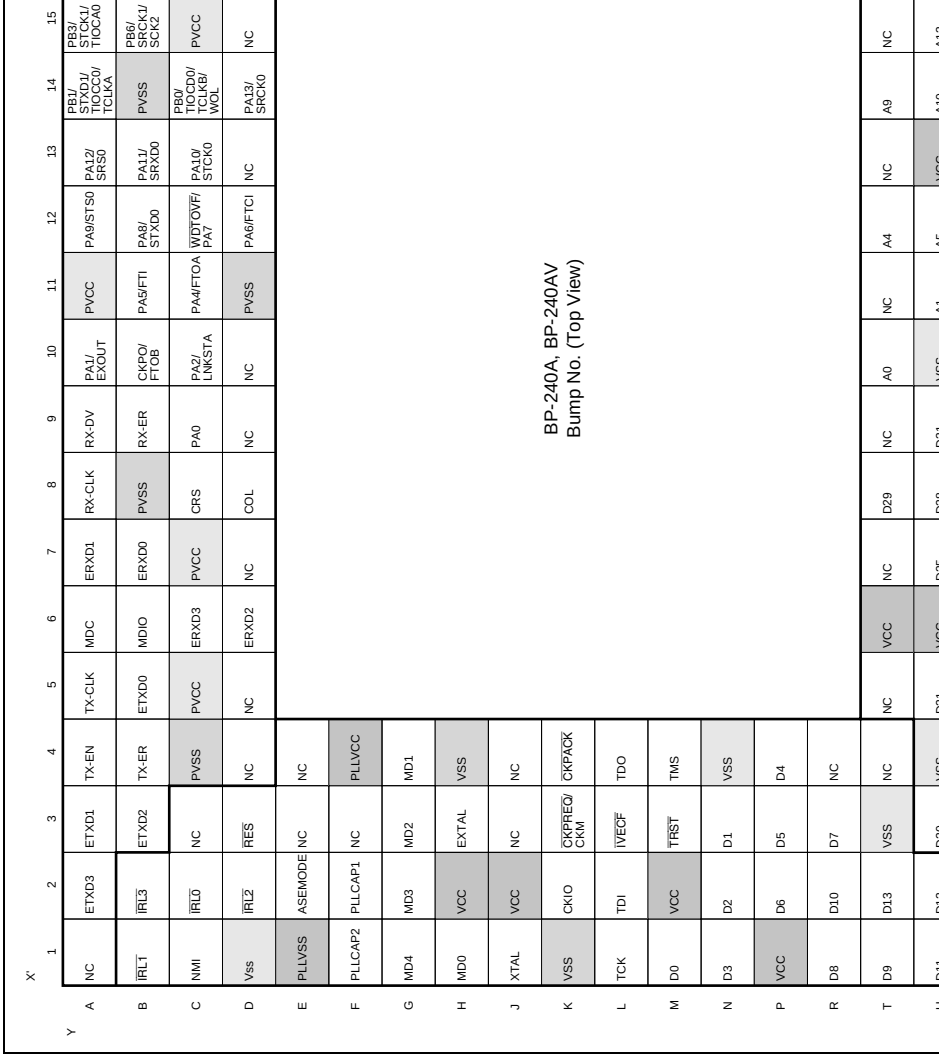


Figure 1.3 HD6417615ARBP and HD6417615ARBPV Pin Arrangeme
(BP-240A, BP-240AV)

	V_{SS}	Input	Ground	For connection to ground. Co V_{SS} pins to the system groun will not operate if there are a
	PV_{CC}	Input	I/O circuit power	Power supply for the I/O circu
	PV_{SS}	Input	I/O circuit ground	Ground for the I/O circuits
Clock	XTAL	Output	Crystal input/ output pin	For connection to a crystal re
	EXTAL	Input		For connection to a crystal re used as external clock input
	CKIO	I/O	System clock input/output pin	Used as the external clock in internal clock output pin
	$\overline{\text{CKPREQ}}$/ CKM	Input	Clock pause request input	Used as the clock pause req changing the frequency of th from the CKIO pin, or halting
	$\overline{\text{CKPACK}}$	Output	Clock pause acknowledge signal	Indicates that the chip is in th pause state (standby state) i
	CKPO	Output	On-chip peripheral clock (Pϕ) output	Outputs the on-chip peripher
	PLLCAP1	Input	PLL capacitance connection pins	Connects capacitance for op PLL circuit 1
	PLLCAP2	Input		Connects capacitance for op PLL circuit 2
	$PLLV_{CC}$	Input	PLL power	PLL oscillator power supply
	$PLLV_{SS}$	Input	PLL ground	PLL oscillator ground

				output the $\overline{BRLS}$ signal reco the bus has been acquired receives the $\overline{BGR}$ signal
	$\overline{BRLS}$	Input	Bus release	Driven low when an externa requests release of the bus
Operating mode	MD0 to MD4	Input	Mode setting	The operating mode is spec levels at these pins
Interrupts	NMI	Input	Nonmaskable interrupt	Inputs the nonmaskable inte signal
	$\overline{IRL3}$ to $\overline{IRL0}$	Input	External interrupt request input 0 to 3	These pins input maskable request signals
	$\overline{IVECF}$	Output	Interrupt vector fetch cycle	Indicates an external vector
Bus control	$\overline{BS}$	Output	Bus cycle start	Signal indicating the start of Asserted every data cycle i transfer
	$\overline{CS4}$ to $\overline{CS0}$	Output	Chip select 0 to 4	Chip select signals indicatin being accessed
	$\overline{WAIT}$	Input	Wait	Wait state request signal
	$\overline{RD}$	Output	Read	Strobe signal indicating a re
	$\overline{RAS}$	Output	Row address strobe	DRAM/synchronous DRAM

DQMLU/ $\overline{WE1}$	Output	Third byte access	SRAM/synchronous DRAM t select signal
DQMLL/ $\overline{WE0}$	Output	Lowest byte access	SRAM/synchronous DRAM l select signal
$\overline{CAS3}$	Output	Column address strobe 3	DRAM highest byte select si
$\overline{CAS2}$	Output	Column address strobe 2	DRAM second byte select si
$\overline{CAS1}$	Output	Column address strobe 1	DRAM third byte select signa
$\overline{CAS0}$	Output	Column address strobe 0	DRAM lowest byte select sig
CKE	Output	Clock enable	Synchronous DRAM clock e
REFOUT	Output	Refresh out	Signal requesting refresh exe when the bus is released
RD/$\overline{WR}$	Output	Read/write	DRAM/synchronous DRAM w
$\overline{BUSHiZ}$	Input	Bus high impedance	Signal used in combination w signal to place bus and strob the high-impedance (HiZ) sta ending the bus cycle
$\overline{BH}$	Output	Burst hint	Asserted at the start of a DM negated one bus cycle befor the burst
STATS0, STATS1	Output	Status	CPU, DMAC, and E-DMAC s information

	$\overline{\text{TRST}}$	Input	Test reset	Test reset input signal
	ASEMODE*1	Input	ASE mode input	ASE mode/user mode selec
Ethernet controller (EtherC)	TX-CLK	Input	Transmitter clock	TX-EN, ETXD0 to ETXD3, T reference signal
	RX-CLK	Input	Receive clock	RX-DV, ERXD0 to ERXD3, timing reference signal
	TX-EN	Output	Transmit enable	Signal indicating that transm ETXD0 to ETXD3 is ready
	ETXD0 to ETXD3	Output	Transmit data 0 to 3	4-bit transmit data
	TX-ER	Output	Transmit error	Signal sending error status port
	RX-DV	Input	Receive data enable	Indicates that enable receiv ERXD0 to ERXD3 exist
	ERXD0 to ERXD3	Input	Receive data 0 to 3	4-bit receive data
	RX-ER	Input	Receive error	Reports error state that occ transfer of frame data
	CRS	Input	Carrier sense	Carrier detection notificatior
	COL	Input	Collision	Collision detection signal
	MDC	Output	Management data clock	Reference clock signal for i transfer by MDIO
	MDIO	I/O	Management data input/output	Bidirectional signal for exch management information be and PHY

controller (DMAC)	DREQ0, DREQ1	Input	DMAC channel 0, 1 request	Pins that input DMA transfer from an external device
Serial communi-cation interface with FIFO (SCIF)	TXD1, TXD2	Output	Transmit data output channel 1, 2	SCIF channel 1 and 2 transm output pins
	RXD1, RXD2	Input	Receive data input channel 1, 2	SCIF channel 1 and 2 receiv pins
	SCK1, SCK2	I/O	Serial clock input/output channel 1, 2	SCIF clock input/output pins
	$\overline{\text{RTS}}$	Output	Transmit request	SCIF channel 1 transmit requ pin
	$\overline{\text{CTS}}$	Input	Transmit enable	SCIF channel 1 transmit ena
Timer pulse unit (TPU)	TCLKA TCLKB TCLKC TCLKD	Input	TPU timer clock input A, B, C, D	Pins that input an external cl TPU counter
	TIOCA0 TIOCB0 TIOCC0 TIOCD0	I/O	TPU input capture/output compare (channel 0)	Channel 0 input capture inpu output compare output/PWM
	TIOCA1 TIOCB1	I/O	TPU input capture/output compare (channel 1)	Channel 1 input capture inpu output compare output/PWM

	FTOB	Output	Output compare B output	Output compare B output pi
	FTI	Input	Input capture input	Input capture input pin
Serial I/O (SIO)	SRXD0 to SRXD2	Input	Serial receive data input 0 to 2	Serial receive data input po
	SRCK0 to SRCK2	Input	Serial receive clock input 0 to 2	Serial receive clock ports
	SRS0 to SRS2	Input	Serial receive synchronization input 0 to 2	Serial receive synchronizati
	STXD0 to STXD2	Output	Serial transmit data 0 to 2	Serial data output ports
	STCK0 to STCK2	Input	Serial transmit clock input 0 to 2	Serial transmit clock ports
	STS0 to STS2	I/O	Serial transmit synchronization input/output 0 to 2	Serial transmit synchronizat input/output ports
I/O ports	PA0 to PA13*2	I/O	General port	General input/output port pi Input or output can be speci
	PB0 to PB15	I/O	General port	General input/output port pi Input or output can be speci

Notes: 1. When carrying out debugging using the E10A emulator, this pin is used for switching. It should be connected to V_{SS} when using the E10A emulator an to V_{CC} when using a normal user system. When a boundary scan test is pe the H-UDI, user mode must be used. A boundary scan test cannot be perfo mode.

2. PA3 cannot be used; CKPO is valid instead.

F2	11	PLLCAP1	
F1	10	PLLCAP2	
H3	19	EXTAL	
J1	21	XTAL	
K2	23	CKIO	
K3	24	$\overline{\text{CKPREQ}}$/CKM	
K4	25	$\overline{\text{CKPACK}}$	9
D3	8	$\overline{\text{RES}}$	S
G1	13	MD4	c
G2	14	MD3	
G3	15	MD2	
G4	16	MD1	
H1	17	MD0	6
C1	5	NMI	I
B2	1	$\overline{\text{IRL3}}$	
D2	2	$\overline{\text{IRL2}}$	
B1	3	$\overline{\text{IRL1}}$	
C2	4	$\overline{\text{IRL0}}$	
L3	27	$\overline{\text{IVECF}}$	6

J17	134	$\overline{CS0}$
F19	148	$\overline{BGR}$
G19	145	$\overline{BRLS}$
R17	115	$\overline{WAIT}$
L18	128	$\overline{RD}$
P19	117	$\overline{RAS}$
P18	118	$\overline{CAS}$/$\overline{OE}$
N19	119	DQMUU/$\overline{WE3}$
N18	120	DQMUL/$\overline{WE2}$
N17	121	DQMLU/$\overline{WE1}$
N16	122	DQMLL/$\overline{WE0}$
M19	123	$\overline{CAS3}$
M18	124	$\overline{CAS2}$
M17	125	$\overline{CAS1}$
M16	126	$\overline{CAS0}$
L19	127	CKE
L16	129	REFOUT
K19	133	RD/$\overline{WR}$
H17	139	$\overline{BUSHiZ}$
H18	140	$\overline{BH}$

V16	104	A19	
W18	103	A18	
V17	102	A17	
V15	100	A16	
U16	98	A15	
W15	97	A14	
U15	96	A13	
W14	95	A12	
V14	94	A11	
U14	93	A10	
T14	92	A9	
V13	90	A8	
W12	88	A7	
V12	87	A6	
U12	86	A5	
T12	85	A4	
W11	84	A3	
V11	83	A2	
U11	82	A1	
T10	80	A0	2

W8	71	D26
U7	70	D25
W7	68	D24
V6	65	D23
W6	64	D22
U5	63	D21
U3	62	D20
W4	59	D19
W3	57	D18
V3	56	D17
W2	55	D16
V4	54	D15
V2	53	D14
T2	51	D13
U2	49	D12
U1	48	D11
R2	47	D10
T1	46	D9
R1	44	D8
R3	43	D7
P2	41	D6
P3	40	D5
P4	39	D4
N1	38	D3
N2	37	D2
N3	36	D1
M1	34	D0

E2	6	ASEMODE*	6
A5	201	TX-CLK	E
A8	192	RX-CLK	
A4	203	TX-EN	5
A2	207	ETXD3	c
B3	206	ETXD2	
A3	205	ETXD1	
B5	204	ETXD0	
B4	208	TX-ER	
A9	188	RX-DV	
C6	197	ERXD3	
D6	196	ERXD2	
A7	195	ERXD1	
B7	194	ERXD0	
B9	187	RX-ER	
C8	190	CRS	
D8	189	COL	
A6	199	MDC	
B6	198	MDIO	1
G17	143	DACK1	D
G18	144	DACK0	
H19	141	DREQ1	
G16	142	DREQ0	4

Note: * When carrying out debugging using the E10A emulator, this pin is used for m
switching. It should be connected to V_{SS} when using the E10A emulator (ASE
When using the chip in the normal user system, and not using the E10A emu
mode), connect this pin to V_{CC}. When a boundary scan test is performed with
UDI, user mode must be used. A boundary scan test cannot be performed in
mode.

C18	154	PB12	SRCK2	RTS	STATS1
D18	156	PB11	SRS2	$\overline{\text{CTS}}$	STATS0
B16	158	PB10	SRXD2	TIOCA1	
A18	159	PB9	STCK2	TIOCB1/TCLKC	
B17	160	PB8	STS2	TIOCA2	
A17	161	PB7	STXD2	TIOCB2/TCLKD	
B15	162	PB6	SRCK1	SCK2	
A16	163	PB5	SRS1	RXD2	
C16	164	PB4	SRXD1	TXD2	
A15	165	PB3	STCK1	TIOCA0	
C17	166	PB2	STS1	TIOCB0	
A14	168	PB1	STXD1	TIOCC0/TCLKA	
C14	170	PB0		TIOCD0/TCLKB	WOL
D14	171	PA13	SRCK0		
A13	172	PA12	SRS0		
B13	173	PA11	SRXD0		
C13	174	PA10	STCK0		
A12	175	PA9	STS0		
B12	176	PA8	STXD0		
C12	177	$\overline{\text{WDTOVF}}$	PA7		
D12	178	PA6	FTCI		
B11	180	PA5	FTI		
C11	182	PA4	FTOA		
B10	183	CKPO	FTOB		
C10	184	PA2	LNKSTA		
A10	185	PA1	EXOUT		
C9	186	PA0			

bus-released state, program execution state, and power-down state. Figure 1.4 shows th
transitions.

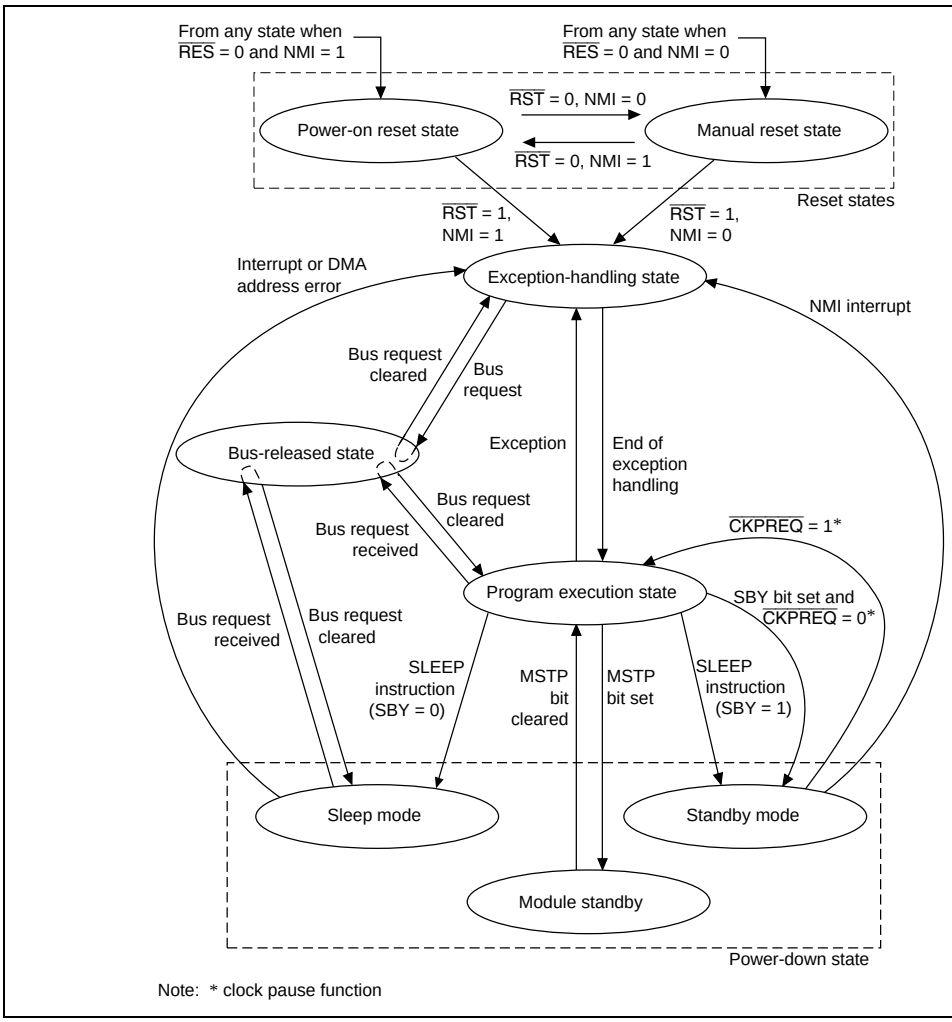


Figure 1.4 Processing State Transitions

program counter (PC) from the exception vector table, and the initial value of the s
(SP), stores these values, branches to the start address, and begins program executi
address.

In the case of an interrupt, etc., the CPU references the SP and saves the PC and st
(SR) in the stack area. It fetches the start address of the exception service routine f
exception vector table, branches to that address, and begins program execution.

Subsequently, the processing state is the program execution state.

- Program Execution State

  In the program execution state the CPU executes program instructions in normal s

- Power-Down State

  In the power-down state the CPU stops operating to conserve power. The power-d
  entered by executing a SLEEP instruction. The power-down state includes two mo
  mode and standby mode—and a module standby function.

- Bus-Released State

  1. In the bus-released state, the CPU releases the bus to a device that has requeste
  2. Bus-released state during manual reset signal assertion

     While the manual reset signal is being asserted ($\overline{\text{RES}}$ = low and NMI = low), n
     request ($\overline{\text{BRLS}}$ input) is accepted. If the $\overline{\text{BRLS}}$ signal continues to be asserted,
     remains in the bus-released state (asserts the $\overline{\text{BGR}}$ signal).

     When the $\overline{\text{BRLS}}$ signal is negated in the bus-released state during manual reset
     assertion, this LSI starts using the bus (negates the $\overline{\text{BGR}}$ signal).

Power-Down State: In addition to the normal program execution state, another CPU
state called the power-down state is provided. In this state, CPU operation is halted an
consumption is reduced. The power-down state includes two modes—sleep mode and
mode—and a module standby function.

When entering standby mode, the DMAC's DMA master enable bit should be clear
Also, the cache should be turned off before entering this mode. The contents of the
on-chip RAM are not retained in this mode.

Standby mode is exited by means of a reset or an external NMI interrupt. When star
is exited, the normal program execution state is entered via the exception handling s
the elapse of the oscillation settling time.

If a transition is made to standby mode using the clock pause function, it is possible
the frequency of the CKIO pin input clock, or to stop the clock itself. When SBY in
is set to 1 and a low level is applied to the $\overline{\text{CKPREQ}}$/CKM pin, a transition is made
mode and a low level is output from the $\overline{\text{CKPACK}}$ pin. The clock can then be stopp
frequency changed.

On-chip supporting module states and pin states are the same as in the normal stand
entered by means of the SLEEP instruction. A transition to the program execution s
made by applying a high level to the $\overline{\text{CKPREQ}}$/CKM pin.

In this mode the oscillator is halted, greatly reducing power consumption.

- Module Standby Function

  A module standby function is provided for the following on-chip supporting modul
  direct memory access controller (DMAC), DSP, 16-bit free-running timer (FRT), se
  communication interface with FIFO (SCIF), serial I/O (SIO), user break controller (
  timer pulse unit (TPU). A module standby function is not supported for the Etherne
  (EtherC) or the Ethernet direct memory access controller (E-DMAC).

  Setting one of module stop bits 11 to 3 and 1 (MSTP11 to MSTP3, MSTP1) to 1 in
  control register (SBYCR1/2) stops the clock supply to the corresponding on-chip su
  module. Use of this function enables power consumption to be reduced.

  The module standby function is cleared by clearing the corresponding MSTP bit to

  DSP instructions must not be used when the DSP has been placed in the module sta

  When using the DMAC module standby function, the direct memory access control
  master enable bit should be cleared to 0.

	while SBY bit is cleared in SBYCR1						3. P 4. M
Standby mode	Executing SLEEP instruction while SBY bit is set in SBYCR1	Halted	Halted	Halted and initialized*1	Held	Undefined	1. N 2. P 3. M
Module standby function	Setting MSTP bit corresponding to individual module	Operating	Operating (DSP halted)	Clock supply to specified module halted, module initialized*2	Held	Held	1. C b 2. P 3. M

Notes: 1. Depends on individual supporting module or pin.
2. DMAC and DSP registers and specified module interrupt vectors retain thei

reason, several registers have been added to the previous SuperH microcomputer regis
added registers are the three control registers: repeat start register (RS), repeat end reg
and modulo register (MOD), and the six system registers: DSP status register (DSR), a
X0, X1, Y0 and Y1 among the DSP data registers.

The general registers are used in the same manner as the SH-1 or SH-2 with regard to
microcomputer-type instructions. With regard to DSP type instructions, they are used
and index registers for accessing memory.

2.1.1 General Registers

There are 16 general registers (Rn) numbered R0 to R15, which are 32 bits in length. G
registers are used for data processing and address calculation.

With SuperH microcomputer type instructions, R0 is also used as an index register. Se
instructions are limited to use of R0 only. R15 is used as the hardware stack pointer (S
and recovering the status register (SR) and program counter (PC) in exception process
accomplished by referencing the stack using R15.

With DSP type instructions, eight of the 16 general registers are used for the addressin
Y data memory and data memory (single data) using the I bus.

R4 and R5 are used as an X address register (Ax) for X memory accesses, and R8 is u
index register (Ix). R6 and R7 are used as a Y address register (Ay) for Y memory acc
R9 is used as a Y index register (Iy). R2, R3, R4, and R5 are used as a single data add
(As) for accessing single data using the I bus, and R8 is used as a single data index reg

DSP type instructions can simultaneously access X and Y data memory. There are two
address pointers for designating X and Y data memory addresses.

Figure 2.1 shows the general registers.

R7, [Ay] *3
R8, [Ix, Is]*3
R9, [Iy]*3
R10
R11
R12
R13
R14
R15, SP *2

Notes:
1. R0 also functions as an index register in the indirect indexed register addressing mode and indirect indexed GBR addressing mode. In some instructions, only the R0 functions as a source register or destination re
2. R15 functions as a hardware stack pointer (SP) during exception proce
3. Used as memory address registers, memory index registers with DSP instructions.

Figure 2.1 General Register Configuration

With the assembler, symbol names are used for R2, R3 ... R9. If it is wished to use a na makes clear the role of a register for DSP type instructions, a different register name (al used. This is written in the following manner for the assembler.

```
Ix:    .REG (R8)
```

```
As0:    .REG (R4)  defined when an alias is required for single data transfer
As1:    .REG (R5)  defined when an alias is required for single data transfer
As2:    .REG (R2)  defined when an alias is required for single data transfer
As3:    .REG (R3)  defined when an alias is required for single data transfer
Is:     .REG (R8)  defined when an alias is required for single data transfer
```

2.1.2 Control Registers

The six 32-bit control registers consist of the status register (SR), repeat start register (
end register (RE), global base register (GBR), vector base register (VBR), and modulo
(MOD).

The SR register indicates processing states.

The GBR register functions as a base address for the indirect GBR addressing mode, a
for such as on-chip peripheral module register data transfers.

The VBR register functions as the base address of the exception processing vector are
interrupts).

The RS and RE registers are used for program repeat (loop) control. The repeat count
designated in the SR register repeat counter (RC), the repeat start address in the RS re
the repeat end address in the RE register. However, note that the address values stored
and RE registers are not necessarily always the same as the physical start and end addr
of the repeat.

The MOD register is used for modulo addressing to buffer the repeat data. The modul
designation is made by DMX or DMY, the modulo end address (ME) is designated in
bits of the MOD register, and the modulo start address (MS) is designated in the lower
Note that the DMX and DMY bits cannot simultaneously designate modulo addressing
addressing is possible with X and Y data transfer instructions (MOVX, MOVY). It is
with single data transfer instructions (MOVS).

Repeat end register (RE)

31	0
RE	

Global base register (GBR)

31	0
GBR	

Vector base register (VBR)

31	0
VBR	

Modulo register (MOD)

31	16 15	0
ME	MS	

ME: Modulo end address
MS: Modulo start address

Figure 2.2 Control Register Configuration

	addressing designation (DMX)	memory address pointer, Ax (R4, R5)
9	M bit	Used by the DIV0S/U, DIV1 instructions
8	Q bit	Used by the DIV0S/U, DIV1 instructions
7 to 4	Interrupt request mask (I3 to I0)	Indicate the receive level of an interrupt r 15)
3 to 2	Repeat flags (RF1, RF0)	Used in zero overhead repeat (loop) cont below for an SETRC instruction For 1 step repeat 00 RE — RS = –4 For 2 step repeat 01 RE — RS = –2 For 3 step repeat 11 RE — RS = 0 For 4 steps or more 10 RE — RS > 0
1	Saturation arithmetic bit (S)	Used with MAC instructions and DSP inst 1: Designates saturation arithmetic (preve overflows)
0	T bit	For MOVT, CMP/cond, TAS, TST, BT, BT BF/S, SETT, CLRT and DT instructions, 0: represents false 1: represents true
		For ADDV/ADDC, SUBV/SUBC, DIV0U/D NEGC, SHAR/SHAL, SHLR/SHLL, ROTR ROTCR/ROTCL instructions, 1: represents occurrence of carry, borrow underflow
31 to 28 15 to 12	0 bit	0: 0 is always read out; write a 0

```
LDRS   @(disp,PC);   disp×2 + PC→RS
LDRE   @(disp,PC);   disp×2 + PC→RE
```

The GBR register and VBR register are the same as the previous SuperH microprocesso
An RC counter and four control bits (DMX bit, DMY bit, RF1 bit, RF0 bit) have been a
the SR register. The RS, RE and MOD registers are new registers.

2.1.3 System Registers

System registers consist of four 32-bit registers: high and low multiply and accumulate
(MACH and MACL), the procedure register (PR), and the program counter (PC). The M
MACL store the results of multiplication or multiply and accumulate operations*. The
the return address from the subroutine procedure. The PC indicates the address of the p
execution; it controls the flow of the processing. The PC indicates the fourth byte after
instruction currently being executed. These registers are the same as those in the Superl
microprocessor.

Note: * These are used only when executing an instruction that was supported by SH
SH-2. They are not used for newly added multiplication instructions (PMUL

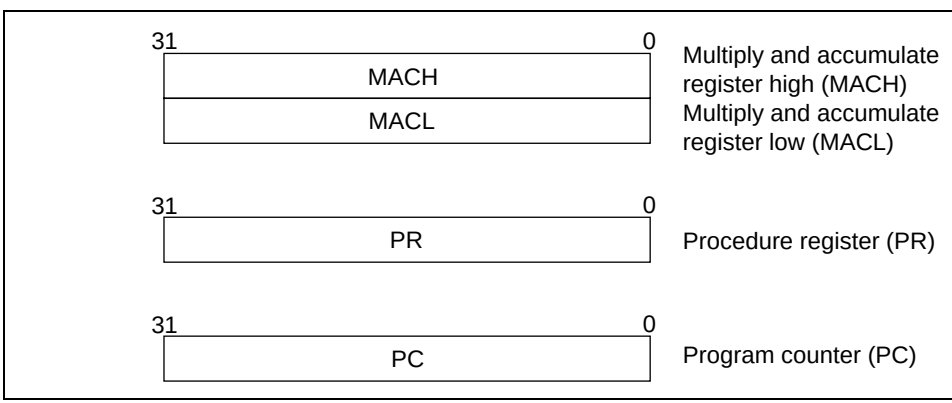


Figure 2.3 System Register Configuration

The DSP data registers are comprised of the two 40-bit registers A0 and A1, and the s
registers M0, M1, X0, X1, Y0 and Y1. The A0 and A1 registers have the 8-bit guard b
A1G, respectively.

The DSP data registers are used for the transfer and processing of the DSP data of DS
operands. There are three types of instructions that access DSP data registers: those fo
processing, and those for X or Y data transfer processing.

The control register is the 32-bit DSP status register (DSR) that represents operation r
DSR register has bits that represent operation results, a signed greater than bit (GT), a
a negative value bit (N), an overflow bit (V), a DSP status bit (DC: DSP condition), ar
selection bit (CS: condition select) for controlling DC bit setting.

The DC bit represents one status flag and is very similar to the SuperH microprocesso
T bit. For conditional DSP type instructions, DSP data processing execution is control
accordance with the DC bit. This control is related to execution in the DSP unit only, a
DSP registers are updated. It bears no relation to address calculation or such SuperH
microprocessor CPU core execution instructions as load/store instructions. The contro
(bits 2 to 0) designate the status for setting the DC bit.

DSP type instructions are comprised of unconditional DSP type instructions and cond
type instructions. The status and DC bits are updated in unconditional DSP type data p
with the exception of the PMULS, MOVX, MOVY and MOVS instructions. Conditio
type instructions are executed according to the status of the DC bit, but regardless of w
not they are executed, the DSR register is not updated.

Figure 2.4 shows the DSP registers. The DSR register bit functions are shown in table

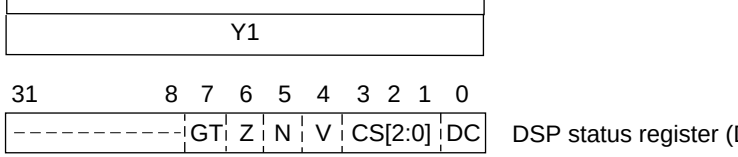


Figure 2.4 DSP Register Configuration

6	Zero bit (Z)	Indicates that the operation result is zero operand 1 is equal to operand 2 1: Operation result is zero (0), or equivale
5	Negative bit (N)	Indicates that the operation result is nega operand 1 is smaller than operand 2 1: Operation result is negative, or operan smaller
4	Overflow bit (V)	Indicates that the operation result has ove 1: Operation result has overflowed
3 to 1	Status selection bits (CS)	Designate the mode for selecting the ope status set in the DC bit Do not set either 110 or 111 000: Carry/borrow mode 001: Negative value mode 010: Zero mode 011: Overflow mode 100: Signed greater mode 101: Signed above mode
0	DSP status bit (DC)	Sets the status of the operation result in t designated by the CS bits 0: Designated mode status not realized (u 1: Designated mode status realized

When place overflows occur so that the correct result cannot be displayed even when th
bits are used, the N flag cannot indicate the correct status.

2.1.6 Initial Values of Registers

Table 2.3 lists the values of the registers after reset.

Table 2.3 Initial Values of Registers

Classification	Register	Initial Value
General registers	R0 to R14	Undefined
	R15 (SP)	Value of the SP in the vector address table
Control registers	SR	Bits I3 to I0 are 1111 (H'F), the reserved bi DMY, and DMX are 0, and other bits are u
	RS	Undefined
	RE	
	GBR	Undefined
	VBR	H'00000000
	MOD	Undefined
System registers	MACH, MACL, PR	Undefined
	PC	Value of the PC in the vector address table
DSP registers	A0, A0G, A1, A1G, M0, M1, X0, X1, Y0, Y1	Undefined
	DSR	H'00000000

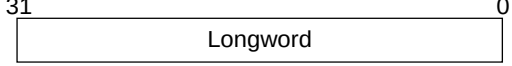


Figure 2.5 Register Data Format

2.2.2 Data Formats in Memory

These formats are classified into bytes, words, and longwords.

Place byte data in any address, word data from 2n addresses, and longword data from
addresses. An address error will occur if accesses are made from any other boundary.
cases, the access results cannot be guaranteed. In particular, the stack area referred to
hardware stack pointer (SP, R15) stores the program counter (PC) and status register (
longwords, so establish the hardware stack pointer so that a 4n value will always resul

To enable sharing of the processor accessing memory in little-endian mode and memo
4 space (area 2, 4) has a function that allows access in little-endian mode. The order o
differs between little-endian mode and normal big-endian mode.

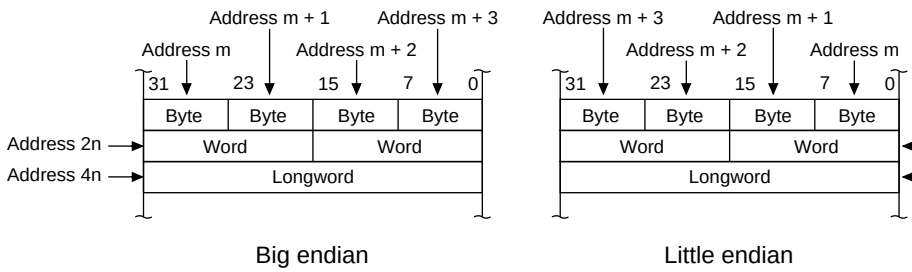


Figure 2.6 Data Formats in Memory

Word or longword immediate data is not located in the instruction code, it should be pl
memory table. Use an immediate data transfer instruction (MOV) to refer the memory t
the PC relative addressing mode with displacement.

2.2.4 DSP Type Data Formats

This chip has three different types of data format that correspond to various instructions
the fixed-point data format, the integer data format, and the logical data format.

The DSP type fixed-point data format has a binary point fixed between bits 31 and 30. T
three types: with guard bits, without guard bits, and multiplication input; each with diff
bit lengths and value ranges.

The DSP type integer data format has a binary point fixed between bits 16 and 15. Ther
types: with guard bits, without guard bits, and shift amount; each with different valid bi
and value ranges. The shift amount of the arithmetic shift (PSHA) has a 7-bit range and
express values from –64 to +63, but the actual valid values are from –32 to +32. In the
manner, the shift amount of the logical shift has a 6-bit range, but the actual valid value
–16 to +16.

The DSP type logical data format does not have a decimal point.

The data format and valid data length are determined by the instructions and DSP regis

Figure 2.7 shows the three DSP type data formats and binary point positions. The Supe
data format is also shown for reference.

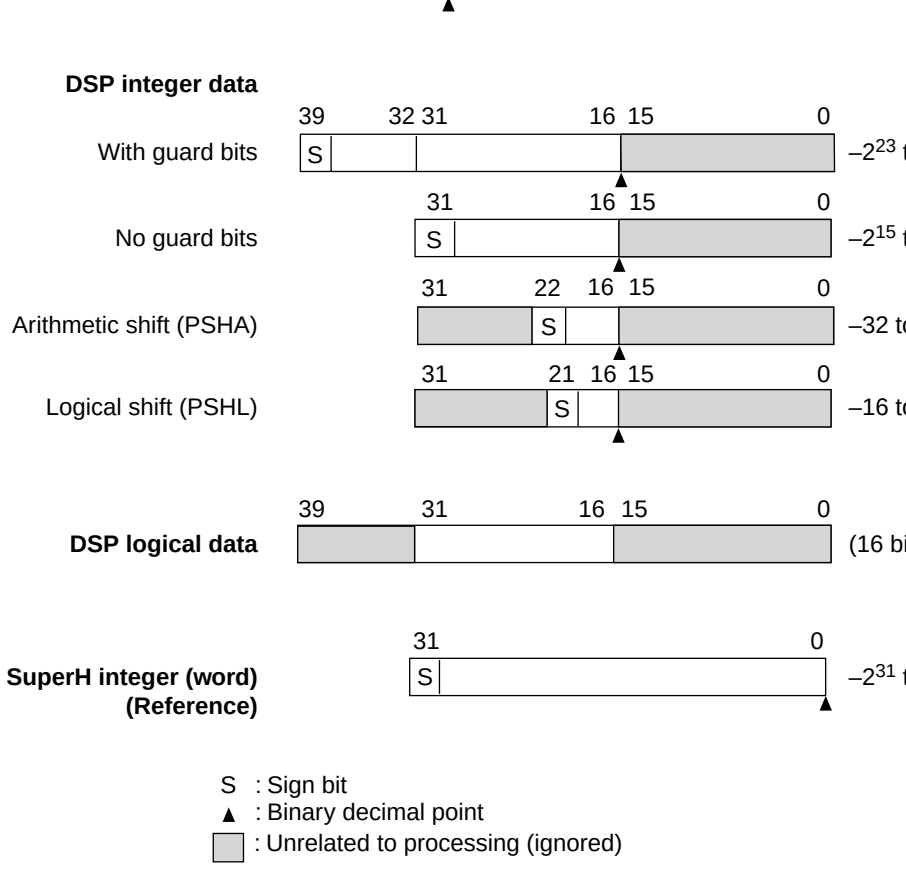


Figure 2.7 DSP Type Data Formats

register data becomes the bits 39 to 32 data. When the A0 and A1 registers are used as
registers, the guard bits (bits 39 to 32) are valid. When any registers other than A0, A1
destination registers, bits 39 to 32 of the result data are disregarded.

Processing for DSP integer data is the same as the DSP fixed-point data processing. Ho
lower word (the lower 16 bits, bits 15 to 0) of the source register is disregarded. The lo
of the destination register is cleared to 0.

In DSP logical data processing, the upper word (the upper 16 bits, bits 31 to 16) of the
register is valid. The lower word and the guard bits of the A0, A1 registers are disregar
upper word of the destination register is valid. The lower word and the guard bits of the
registers are cleared to 0.

X, Y Data Transfers: The MOVX.W and MOVY.W instructions access X, Y memory
16-bit X, Y data buses. The data loaded into registers and data stored from registers is a
upper word (the upper 16 bits, bits 31 to 16).

When loading, the MOVX.W instruction loads X memory, with the X0 and X1 register
destination registers. The MOVY.W instruction loads Y memory, with the Y0 and Y1 r
the destination registers. Data is stored in the upper word of the register; the lower word
to 0.

The upper word data of the A0, A1 registers can be stored in X or Y memory with these
transfer instructions, but storing is not possible from any other registers. The guard bits
lower word of the A0, A1 registers are disregarded.

Single Data Transfers: The MOVS.W and MOVS.L instructions can access any memo
data bus (CDB). All DSP registers are connected to the CDB bus, and they can become
destination registers during data transfers. The two data transfer modes are word and lo

In word mode, data is loaded to and stored in the upper word of the DSP register, with
exception of the A0G, A1G registers. In longword mode, data is loaded to and stored in
of the DSP register, with the exception of the A0G, A1G registers. The A0G, A1G regi
treated as independent registers during single data transfers. The load/store data length
A0G, A1G registers is 8 bits.

sign is extended and stored in the guard bits; the lower word is cleared to 0. When the
registers are the destination registers in word mode, the least significant 8 bits of the d
loaded into the registers; the A0, A1 registers are not zero cleared but retain their prev

If the DSP registers are used as source registers in longword mode, when data is store
registers other than A0G, A1G, the 32 bits (data) of the register are transferred. When
registers are used as the source registers the guard bits are disregarded. When the A0G
registers are the source registers in longword mode, only 8 bits of the data are stored f
registers; the upper bits are sign-extended.

If the DSP registers are used as destination registers in longword mode, the load is to t
the register, with the exception of A0G, A1G. In the case of the A0, A1 registers, the
extended and stored in the guard bits. When the A0G, A1G registers are the destinatio
longword mode, the least significant 8 bits of the data are loaded into the registers; the
registers are not zero cleared but retain their previous values.

Tables 2.4 and 2.5 indicate the register data formats for DSP instructions. Some regist
be accessed by certain instructions. For example, the PMULS instruction can designat
register as a source register but cannot designate A0 as such. Refer to the instruction e
for details.

Figure 2.8 shows the relationship between the buses and the DSP registers during tran

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

Note: * The sign is extended and stored in the ALU's guard bits.

		PDMSB			
		Logic, PSHL	Clear to 0	16-bit result	
	Data transfer	MOVS.W	Sign extend	16-bit data	
		MOVS.L		32-bit data	
A0G, A1G	Data transfer	MOVS.W	Data	Not updated	N
		MOVS.L			
X0, X1, Y0, Y1, M0, M1	DSP operation	Fixed decimal, PSHA, PMULS	—	32-bit result	
		Integer, logic, PDMSB, PSHL		16-bit result	C
	Data transfer	MOVX.W, MOVY.W, MOVS.W		16-bit data	
		MOVS.L		32-bit data	

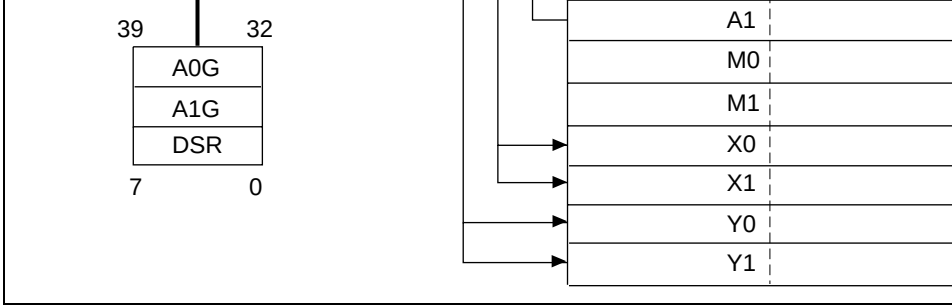


Figure 2.8 DSP Register-Bus Relationship during Data Transfers

2.3 CPU Core Instruction Features

The CPU core instructions are RISC type. The characteristics are as follows.

16-Bit Fixed Length: All instructions are 16 bits long, increasing program code efficie

One Instruction per Cycle: The microprocessor can execute basic instructions in one
the pipeline system. One state equals 16.0 ns when operating at 62.5 MHz.

Data Length: Longword is the basic data length for all operations. Memory can be acc
bytes, words, or longwords. Byte or word data accessed from memory is sign-extended
handled as longword data. Immediate data is sign-extended for arithmetic operations or
extended for logic operations. It also is handled as longword data.

Load-Store Architecture: Basic operations are executed between registers. For opera
involve memory access, data is loaded to the registers and executed (load-store archite
However, Instructions such as AND manipulating bits, are executed directly in memo

Delayed Branches: Such instructions as unconditional branches are delayed branch i
In the case of delayed branch instructions, the branch occurs after execution of the ins
immediately following the delayed branch instruction (slot instruction). This reduces
disruption during branching.

The branching operation of the delayed branch occurs after execution of the slot instru
However, with the exception of such branch operations as register updating, execution
instructions is performed with the order of delayed branch instruction, then delayed sl
instruction.

For example, even if the contents of a register storing a branch destination address are
a delayed slot, the branch destination address will still be the contents of the register b
modification.

Table 2.7 Delayed Branch Instructions

SH7615 CPU	Description	Example of Conventio
BRA TRGET ADD R1,R0	Executes an ADD before branching to TRGET	ADD.W R1,R0 BRA TRGET

Multiplication/Multiply-Accumulate Operation: $16 \times 16 \rightarrow 32$ multiplications exec
to three cycles, and $16 \times 16 + 64 \rightarrow 64$ multiply-accumulate operations execute in two
cycles. $32 \times 32 \rightarrow 64$ multiplications and $32 \times 32 + 64 \rightarrow 64$ multiply-accumulate ope
execute in two to four cycles.

		when R0 ≥ R1.		
BF	TRGET1	The program branches to TRGET1 when R0 < R1	BLT	TRGET1
ADD CMP/EQ BT	#–1,R0 #0,R0 TRGET	T bit is not changed by ADD. T bit is set when R0 = 0. The program branches when R0 = 0	SUB.W BEQ	#1,R0 TRGET

Immediate Data: Byte immediate data resides in instruction code. Word or longword i
data is not input in instruction codes but is stored in a memory table. An immediate dat
instruction (MOV) accesses the memory table using the PC relative addressing mode w
displacement.

Table 2.9 Immediate Data Accessing

Classification	SH7615 CPU		Example of Conventiona	
8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOV.W ..... .DATA.W	@(disp,PC),R0 H'1234	MOV.W	#H'1234,R0
32-bit immediate	MOV.L ..... .DATA.L	@(disp,PC),R0 H'12345678	MOV.L	#H'12345678,R0

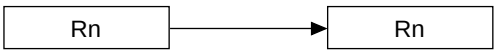
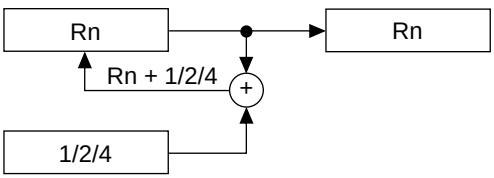
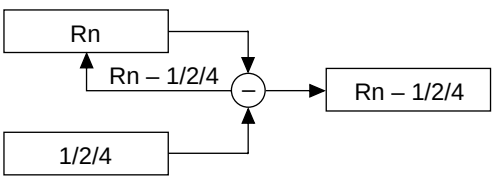
Note: @(disp, PC) accesses the immediate data.

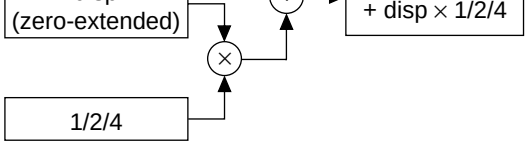
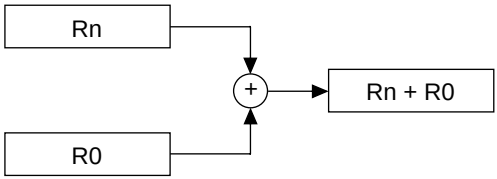
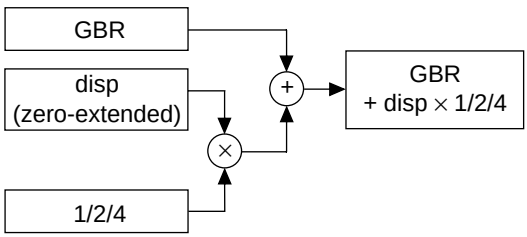
	MOV.B @R1,R0	
	.....	
	.DATA.L H'12345678	

16-Bit/32-Bit Displacement: When data is accessed by 16-bit or 32-bit displacement, existing displacement value is placed in the memory table. Loading the immediate dat instruction is executed transfers that value to the register and the data is accessed in th indexed register addressing mode.

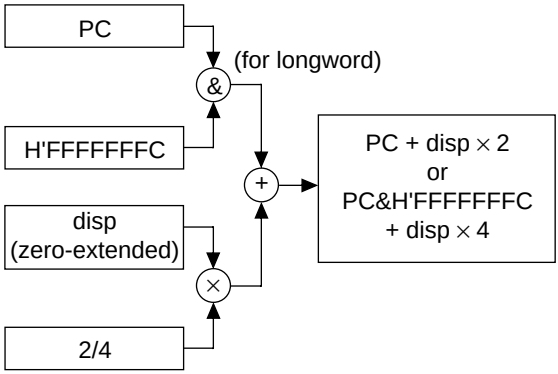
Table 2.11 Displacement Accessing

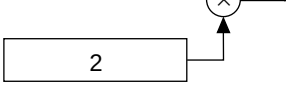
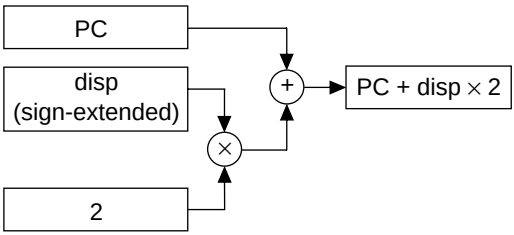
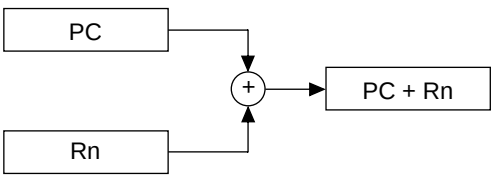
Classification	SH7615 CPU	Example of Convention
16-bit displacement	MOV.W @(disp,PC),R0 MOV.W @(R0,R1),R2 ..... .DATA.W H'1234	MOV.W @(H'1234,R1),

Addressing Mode	Instruction Format	Effective Addresses Calculation	Equatio
Direct register addressing	Rn	The effective address is register Rn (The operand is the contents of register Rn)	—
Indirect register addressing	@Rn	The effective address is the content of register Rn 	Rn
Post-increment indirect register addressing	@Rn+	The effective address is the content of register Rn. A constant is added to the content of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation 	Rn (After th instructi executes Byte: Rn Word: R Longwo → Rn
Pre-decrement indirect register addressing	@–Rn	The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation 	Byte: Rn Word: R Rn Longwo → Rn (I executed after cal

			
Indirect indexed register addressing	@(R0, Rn)	The effective address is the Rn value plus R0 	Rn + R
Indirect GBR addressing with displacement	@(disp:8, GBR)	The effective address is the GBR value plus an 8-bit displacement (disp). The value of disp is zero-extended, and remains the same for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation 	Byte: G Word: × 2 Longw disp ×

addressing with displacement	@(disp, PC)	8-bit displacement (disp). The value of disp is zero-extended, is doubled for a word operation, and is quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked	× 2 Long H'FF disp



			
	disp:12	The effective address is the PC value sign-extended with a 12-bit displacement (disp), doubled, and added to the PC value 	PC
	Rn	The effective address is the register PC value plus Rn 	PC
Immediate addressing	#imm:8	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions are zero-extended	—
	#imm:8	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions are sign-extended	—
	#imm:8	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and is quadrupled	—

Classification	(MOVX.W, MOVY.W)	(MOVS.W, MOVS.L)
Address registers	Ax: R4, R5; Ay: R6, R7	As: R2, R3, R4, R5
Index registers	Ix: R8, Iy: R9	Is: R8
Addressing	Nop/Inc(+2)/index addition: post-update	Nop/Inc(+2,+4)/index ad update
	—	Dec(–2,–4): pre-update
Modulo addressing	Possible	Not possible
Data bus	XDB, YDB	CDB
Data length	16 bits (word)	16 bits/32 bits (word/lon
Bus contention	None	Yes
Memory	X, Y data memory	All memory spaces
Source registers	Dx, Dy: A0, A1	Ds: A0/A1, M0/M1, X0/X A0G, A1G
Destination registers	Dx: X0/X1; Dy: Y0/Y1	Ds: A0/A1, M0/M1, X0/X A0G, A1G

X, Y Data Addressing: Among the DSP instructions, the MOVX.W and MOVY.W in can be used to simultaneously access X, Y data memory. The DSP instructions have tw pointers for simultaneous accessing of X, Y data memory. Only pointer addressing is p DSP instructions; there is no immediate addressing. The address registers are divided i R4, R5 registers become the X memory address register (Ax), and the R6, R7 registers Y memory address register (Ay). The following three types of addressing exist with X, transfer instructions.

1. Non-updated address registers: The Ax, Ay registers are address pointers. They are updated.
2. Add index registers: The Ax, Ay registers are address pointers. The Ix, Iy register v added to them, respectively, after the data transfer (post-update).
3. Increment address registers: The Ax, Ay registers are address pointers. The value + to each of them after the data transfer (post-update).

Figure 2.9 shows the X, Y data transfer addressing. When X memory and Y memory a
using the X, Y bus, the upper word of Ax (R4 or R5) and Ay (R6 or R7) is ignored. Th
@Ay+ and @Ay+Iy is stored in the lower word of Ay, and the upper word retains its o
value.

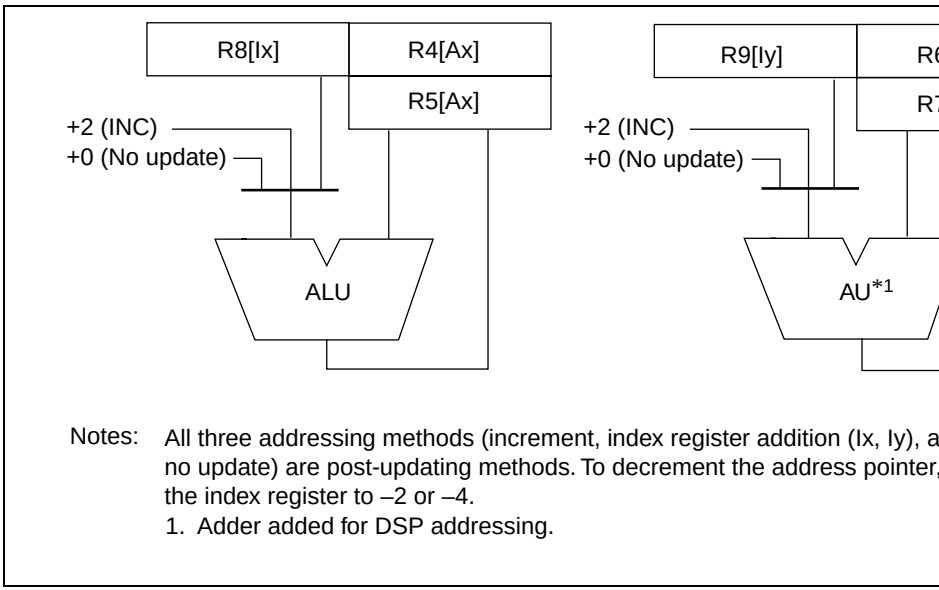


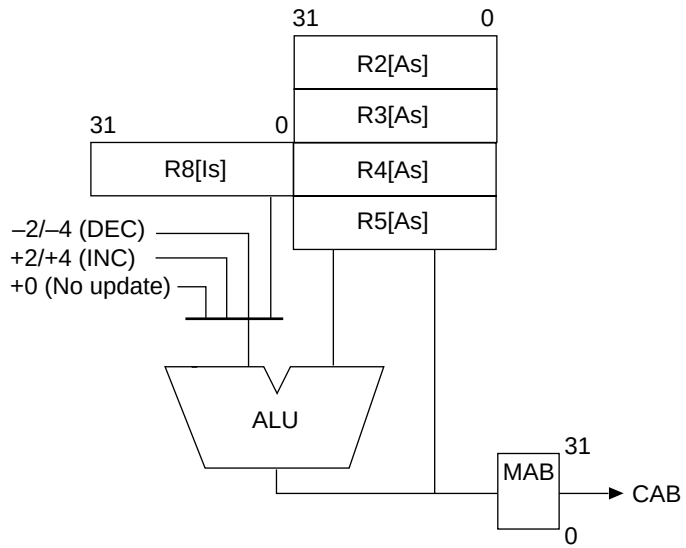
Figure 2.9 X, Y Data Transfer Addressing

them after the data transfer (post-update).

3. Increment address registers: The As registers are address pointers. The value +2 or
after the data transfer (post-update).
4. Decrement address registers: The As registers are address pointers. The value –2 or
(+2 or +4 is subtracted) before the data transfer (pre-update).

The address pointer (As) uses the R8 register as an index register (Is).

Figure 2.10 shows the single data transfer addressing.



Note: There are four addressing methods (no update, index register addition (
increment, and decrement). Index register addition and increment are
post-updating methods. Decrement is a pre-updating method.

Figure 2.10 Single Data Transfer Addressing

both modulo addressing mode at the same time. Therefore, do not simultaneously set t
and DMY. If they happen to be set at the same time, only the DMY side is valid.

The MOD register is used to designate the start and end addresses of the modulo addr
stores the MS (modulo start) and ME (modulo end). An example of MOD register (M
usage is indicated below.

```
           MOV.L     ModAddr,Rn;        Rn=ModEnd, ModStart
           LDC       Rn,MOD;            ME=ModEnd, MS=ModStart
ModAddr:   .DATA.W   mEnd;              ModEnd
           .DATA.W   mStart;            ModStart

ModStart:  .DATA
             :
ModEnd:    .DATA
```

Designate the start and end addresses in MS and ME, and then set the DMX or DMY
contents of the address register are compared with ME. If they match ME, the start ad
stored in the address register. The lower 16 bits of the address register are compared w
maximum modulo size is 64 kbytes. This is sufficient for X, Y data memory accesses.
shows a block diagram of modulo addressing.

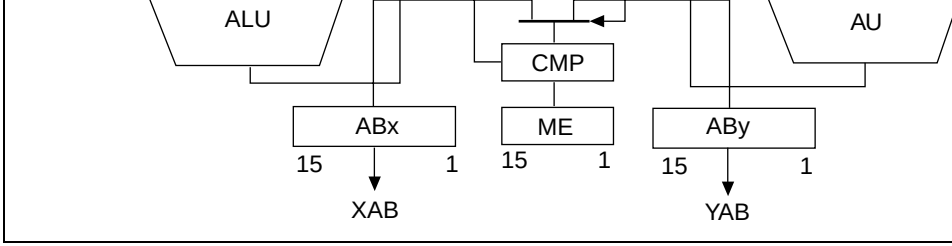


Figure 2.11 Modulo Addressing

An example of modulo addressing is indicated below:

```
MS=H'E008; ME=H'E00C; R4=H'1000E008;
```

DMX=1; DMY=0; (sets modulo addressing for address register Ax (R4, R5))

The R4 register changes as follows due to the above settings.

R4: H'1000E008

```
Inc.   R4: H'1000E00A
Inc.   R4: H'1000E00C
Inc.   R4: H'1000E008
```

(becomes the modulo start address because the mod
address occurred)

Data is placed so that the upper 16 bits of the modulo start and end addresses become id
This is so because the modulo start address replaces only the lower 15 bits of the addres
excepting bit 0.

Note: When using add index with DSP data addressing, there are cases where the valu
exceeded without the address pointer matching the ME. In such cases, the addr
does not return to the modulo start address. Bit 0 is disregarded not only for mo
addressing, but also during X, Y data addressing, so always write 0 to the 0 bits
address pointer, index register, MS, and ME.

```
    if ( DMX==0 || DMX==1 && DMY==1 )} Ax=Ax+(+2 or R8[Ix} or +
    /* Inc,Index,Not-Update */
    else if (!not-update) Ax=modulo( Ax, (+2 or R8[Ix]) );

    /* Ay is one of R6,7 */
    if ( DMY==0 ) Ay=Ay+(+2 or R9[Iy] or +0; /* Inc,Index,Not-U
    else if (! not-update) Ay=modulo( Ay, (+2 or R9[Iy]) );
}
else if ( Operation is MOVS.W or MOVS.L ) {
    if ( Addressing is Nop, Inc, Add-index-reg ) {
        MAB=As;
        /* memory access cycle uses MAB. The address to be used
been updated */
        /* As is one of R2–5 */
        As=As+(+2 or +4 or R8[Is] or +0); /* Inc.Index,Not-Updat
    else { /* Decrement, Pre-update */
    /* As is one of R2–5 */
    As=As+(–2 or –4);
    MAB=As;
    /* memory access cycle uses MAB. The address to be used has
updated */
}

/* The value to be added to the address register depends on ad
operations.
For example, (+2 or R8[Ix] or +0) means that
        +2:       if operation is increment
        R8[Ix}:       if operation is add-index-reg
        +0:       if operation is not-update
*/
```

The instruction format of instructions executed by the CPU core and the meanings of th
and destination operands are indicated below. The meaning of the operand depends on
instruction code. The symbols are used as follows:

xxxx: Instruction code
mmmm: Source register
nnnn: Destination register
iiii: Immediate data
dddd: Displacement

Table 2.14 Instruction Formats for CPU Instructions

Instruction Formats	Source Operand	Destination Operand	Exampl
0 format 15 0 xxxx xxxx xxxx xxxx	—	—	NOP
n format 15 0 xxxx nnnn xxxx xxxx	—	nnnn: Direct register	MOVT
	Control register or system register	nnnn: Direct register	STS M
	Control register or system register	nnnn: Indirect pre-decrement register	STC.L
m format 15 0 xxxx mmmm xxxx xxxx	mmmm: Direct register	Control register or system register	LDC R
	mmmm: Indirect post-increment register	Control register or system register	LDC.L
	mmmm: Indirect register	—	JMP @
	mmmm: PC relative using Rm	—	BRAF

	accumulate)		
	nnnn: Indirect post-increment register (multiply/ accumulate)*		
	mmmm: Indirect post-increment register	nnnn: Direct register	MOV.L
	mmmm: Direct register	nnnn: Indirect pre-decrement register	MOV.L
	mmmm: Direct register	nnnn: Indirect indexed register	MOV.L Rm,@(
md format 15 0 xxxx xxxx mmmm dddd	mmmmdddd: indirect register with displacement	R0 (Direct register)	MOV.B @(disp
nd4 format 15 0 xxxx xxxx nnnn dddd	R0 (Direct register)	nnnndddd: Indirect register with displacement	MOV.B R0,@(
nmd format 15 0 xxxx nnnn mmmm dddd	mmmm: Direct register	nnnndddd: Indirect register with displacement	MOV.L Rm,@(
	mmmmdddd: Indirect register with displacement	nnnn: Direct register	MOV.L @(disp

	relative with displacement		@(disp,
	dddddddd: PC relative	—	BF lab
d12 format 15 0 xxxx dddd dddd dddd	dddddddddddd: PC relative	—	BRA la (label=di
nd8 format 15 0 xxxx nnnn dddd dddd	dddddddd: PC relative with displacement	nnnn: Direct register	MOV.L @(disp,
i format 15 0 xxxx xxxx iiii iiii	iiiiiiii: Immediate	Indirect indexed GBR	AND.B #imm,@
	iiiiiiii: Immediate	R0 (Direct register)	AND #
	iiiiiiii: Immediate	—	TRAPA
ni format 15 0 xxxx nnnn iiii iiii	iiiiiiii: Immediate	nnnn: Direct register	ADD #

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

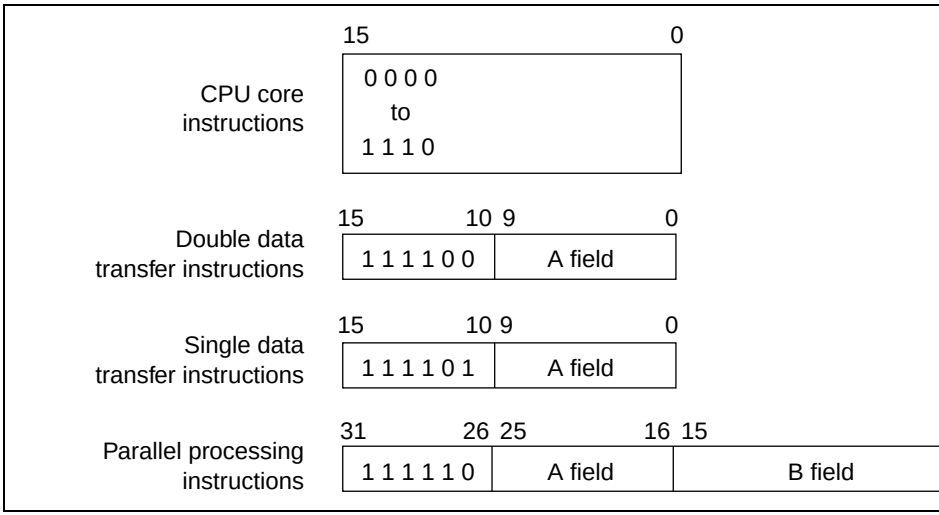


Figure 2.12 Instruction Formats for DSP Instructions

Double, Single Data Transfer Instructions: Table 2.15 indicates the data formats fo
transfer instructions, and table 2.16 indicates the data formats for single data transfer i

	MOVX.W Da,@Ax+Ix						
Y memory data transfers	NOPY	1	1	1	1	0	0
	MOVY.W @Ay,Dy MOVY.W @Ay+,Dy MOVY.W @Ay+Iy,Dy						
	MOVY.W Da,@Ay MOVY.W Da,@Ay+ MOVY.W Da,@Ay+Iy						

Category	Mnemonic	7	6	5	4	3	2
X memory data transfers	NOPX	0		0		0	0
	MOVX.W @Ax,Dx MOVX.W @Ax+,Dx MOVX.W @Ax+Ix,Dx	Dx		0		0 1 1	1 0 1
	MOVX.W Da,@Ax MOVX.W Da,@Ax+ MOVX.W Da,@Ax+Ix	Da		1		0 1 1	1 0 1
Y memory data transfers	NOPY		0		0		
	MOVY.W @Ay,Dy MOVY.W @Ay+,Dy MOVY.W @Ay+Iy,Dy		Dy		0		
	MOVY.W Da,@Ay MOVY.W Da,@Ay+ MOVY.W Da,@Ay+Iy		Da		1		

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	
	MOVS.L @–As,Ds MOVS.L @As,Ds MOVS.L @As+,Ds MOVS.L @As+Is,Ds	
	MOVS.L Ds,@-As MOVS.L Ds,@As MOVS.L Ds,@As+ MOVS.L Ds,@As+Is	

Category	Mnemonic	7	6	5	4	3	2
Single data transfer	MOVS.W @–As,Ds MOVS.W @As,Ds MOVS.W @As+,Ds MOVS.W @As+Is,Ds	Ds		0: (*) 1: (*) 2: (*) 3: (*)		0 0 1 1	0 1 0 1
	MOVS.W Ds,@-As MOVS.W Ds,@As MOVS.W Ds,@As+ MOVS.W Ds,@As+Is			4: (*) 5: A1 6: (*) 7: A0		0 0 1 1	0 1 0 1
	MOVS.L @–As,Ds MOVS.L @As,Ds MOVS.L @As+,Ds MOVS.L @As+Is,Ds			8: X0 9: X1 A: Y0 B: Y1		0 0 1 1	0 1 0 1
	MOVS.L Ds,@-As MOVS.L Ds,@As MOVS.L Ds,@As+ MOVS.L Ds,@As+Is			C: M0 D: A1G E: M1 F: A0G		0 0 1 1	0 1 0 1

Note: * System reserved code

and multiplication instructions. A fields instruction is the same as double data transfers
2.15.

Table 2.17 A Field Parallel Data Transfer Instructions

Category	Mnemonic	31	30	29	28	27	26	25
X memory data transfers	NOPX	1	1	1	1	1	0	0
	MOVX.W @Ax,Dx MOVX.W @Ax+,Dx MOVX.W @Ax+Ix,Dx							Ax
	MOVX.W Da,@Ax MOVX.W Da,@Ax+ MOVX.W Da,@Ax+Ix							
Y memory data transfers	NOPY							
	MOVY.W @Ay,Dy MOVY.W @Ay+,Dy MOVY.W @Ay+Iy,Dy							
	MOVY.W Da,@Ay MOVY.W Da,@Ay+ MOVY.W Da,@Ay+Iy							

data transfers	MOVY.W @Ay,Dy MOVY.W @Ay+,Dy MOVY.W @Ay+Iy,Dy	Dy		0		0 1 1	1 0 1
	MOVY.W Da,@Ay MOVY.W Da,@Ay+ MOVY.W Da,@Ay+Iy	Da		1		0 1 1	1 0 1

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

parallel instruction					1:X1	1:Y1	1:X1	1:Y1	
	PSUB Sx, Sy, Du PMULS Se, Sf, Dg		0 1	1 0	2:Y0 3:A1	2:X0 3:A1	2:A0 3:A1	2:M0 3:M1	
	PADD Sx, Sy, Du PMULS Se, Sf, Dg		0 1	1 1					
Three operand instructions	Reserved		1 0	0 0 0 1	0 0	0 0			
	PSUBC Sx, Sy, Dz			1 0					
	PADDC Sx, Sy, Dz			1 1					
	PCMP Sx, Sy			0 0	0 1				
	Reserved			0 1					
	Reserved			1 0					
	Reserved			1 1					
	PABS Sx, Dz			0 0	1 0				
	PRND Sx, Dz			0 1					
	PABS Sy, Dz			1 0					
	PRND Sy, Dz			1 1					
	Reserved			0 0 0 1 1 0 1 1	1 1				

	(if cc) PINC Sx, Dz				0 1						
	(if cc) PDEC Sy, Dz				1 0						
	(if cc) PINC Sy, Dz				1 1						
	(if cc) PCLR Dz				0 0	1 1	11:DCF				
	(if cc) PDMSB Sx, Dz				0 1						
	Reserved				1 0						
	(if cc) PDMSB Sy, Dz				1 1						
	(if cc) PNEG Sx, Dz			1 1	0 0	1 0					
	(if cc) PCOPY Sx, Dz				0 1						
	(if cc) PNEG Sy, Dz				1 0						
	(if cc) PCOPY Sy, Dz				1 1						
	Reserved						0 0				
	(if cc) PSTS MACH, Dz				0 0	1 1	if cc				
	(if cc) PSTS MACL, Dz				0 1						
	(if cc) PLDS Dz, MACH				1 0						
	(if cc) PLDS Dz, MACL				1 1						
	Reserved*2						0 0				
						0 *					
	Reserved	1									

Notes: 1. System reserved code
2. (if cc): DCT (DC bit true), DCF (DC bit false), or none (unconditional instruc

2.5 Instruction Set

The instructions are divided into three groups: CPU instructions executed by the CPU
data transfer instructions executed by the DSP unit, and DSP operation instructions. T
number of CPU instructions for supporting the DSP functions. The instruction set is e
below in terms of each of the three groups.

		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	
Arithmetic operations	21	ADD	Binary addition	3
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Initialization of signed division	
		DIV0U	Initialization of unsigned division	
		DMULS	Signed double-length multiplication	
		DMULU	Unsigned double-length multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply/accumulate, double-length multiply/accumulate operation	
		MUL	Double-length multiply operation	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	

Shift	10	ROTCL	One-bit left rotation with T bit	1
		ROTCR	One-bit right rotation with T bit	
		ROTL	One-bit left rotation	
		ROTR	One-bit right rotation	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
Branch	9	BF	Conditional branch, conditional branch with delay (Branch when T = 0)	1
		BT	Conditional branch, conditional branch with delay (Branch when T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	

	NOP	No operation
	RTE	Return from exception processing
	SETRC	Repeat count setting
	SETT	T bit set
	SLEEP	Shift into power-down mode
	STC	Storing control register data
	STS	Storing system register data
	TRAPA	Trap exception handling
Total: 65		18

OP.Sz SRC, DEST
OP: Operation code
Sz: Size
SRC: Source
DEST: Destination

Rm: Source register

Rn: Destination register

imm: Immediate data

disp: Displacement[*2]

mmmm: Source register

nnnn: Destination register
0000: R0
0001: R1
.....
1111: R15

iiii: Immediate data

dddd: Displacement

→, ←: Transfer direction

(xx): Memory operand

M/Q/T: Flag bits in the SR

&: Logical AND of each bit

|: Logical OR of each bit

^: Exclusive OR of each bit

~: Logical NOT of each bit

<<n: n-bit left shift

>>n: n-bit right shift

Notes: 1. Instruction execution cycles: The execution cycles shown in the table are m
The actual number of cycles may be increased when (1) contention occurs
instruction fetches and data access, or (2) when the destination register of
instruction (memory → register) and the register used by the next instructio
same.

2. Depending on the instruction's operand size, scaling is ×1, ×2, or ×4. For de
the SH-1/SH-2/SH-DSP Programming Manual.

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 × 2 + Rn)	1
MOV.L	Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm → (disp × 4 + Rn)	1
MOV.B	@(disp,Rm),R0	10000100mmmmdddd	(disp + Rm) → Sign extension → R0	1
MOV.W	@(disp,Rm),R0	10000101mmmmdddd	(disp × 2 + Rm) → Sign extension → R0	1
MOV.L	@(disp,Rm),Rn	0101nnnnmmmmdddd	(disp × 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	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0 + Rm) → Sign extension → Rn	1
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0 + Rm) → Sign extension → Rn	1

			extension → R0	
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1
MOVA	@(disp,PC),R0	11000111dddddddd	disp × 4 + PC → R0	1
MOVT	Rn	0000nnnn00101001	T → Rn	1
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm → Swap the bottom two bytes → Rn	1
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm → Swap upper and lower words → Rn	1
XTRCT	Rm,Rn	0010nnnnmmmm1101	Rm: Middle 32 bits of Rn → Rn	1

CMP/EQ	#imm,R0	10001000iiiiiiii	If R0 = imm, 1 → T	1	Co res
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	If Rn = Rm, 1 → T	1	Co res
CMP/HS	Rm,Rn	0011nnnnmmmm0010	If Rn ≥ Rm with unsigned data, 1 → T	1	Co res
CMP/GE	Rm,Rn	0011nnnnmmmm0011	If Rn ≥ Rm with signed data, 1 → T	1	Co res
CMP/HI	Rm,Rn	0011nnnnmmmm0110	If Rn > Rm with unsigned data, 1 → T	1	Co res
CMP/GT	Rm,Rn	0011nnnnmmmm0111	If Rn > Rm with signed data, 1 → T	1	Co res
CMP/PL	Rn	0100nnnn00010101	If Rn > 0, 1 → T	1	Co res
CMP/PZ	Rn	0100nnnn00010001	If Rn ≥ 0, 1 → T	1	Co res
CMP/STR	Rm,Rn	0010nnnnmmmm1100	If Rn and Rm contain an identical byte, 1 → T	1	Co res
DIV1	Rm,Rn	0011nnnnmmmm0100	Single-step division (Rn/Rm)	1	Ca res
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Ca res
DIV0U		0000000000011001	0 → M/Q/T	1	0
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2 to 4*	—
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2 to 4*	—

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

Note: * The normal number of execution cycles. The number in parentheses is the
execution cycles in the case of contention with preceding or following instru

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

SHAR	Rn	0100nnnn00100001	MSB → Rn → T	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	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	2
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	2
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	2
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	2
RTS		0000000000001011	Delayed branch, PR → PC	2

Note: * One state when it does not branch.

LDC	Rm,MOD	0100mmmm01011110	Rm → MOD	1
LDC	Rm,RE	0100mmmm01111110	Rm → RE	1
LDC	Rm,RS	0100mmmm01101110	Rm → RS	1
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	3
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	3
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	3
LDC.L	@Rm+,MOD	0100mmmm01010111	(Rm) → MOD, Rm + 4 → Rm	3
LDC.L	@Rm+,RE	0100mmmm01110111	(Rm) → RE, Rm + 4 → Rm	3
LDC.L	@Rm+,RS	0100mmmm01100111	(Rm) → RS, Rm + 4 → Rm	3
LDRE	@(disp,PC)	10001110dddddddd	disp × 2 + PC → RE	1
LDRS	@(disp,PC)	10001100dddddddd	disp × 2 + PC → RS	1
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1
LDS	Rm,DSR	0100mmmm01101010	Rm → DSR	1
LDS	Rm,A0	0100mmmm01111010	Rm → A0	1
LDS	Rm,X0	0100mmmm10001010	Rm → X0	1
LDS	Rm,X1	0100mmmm10011010	Rm → X1	1
LDS	Rm,Y0	0100mmmm10101010	Rm → Y0	1
LDS	Rm,Y1	0100mmmm10111010	Rm → Y1	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
LDS.L	@Rm+,DSR	0100mmmm01100110	(Rm) → DSR, Rm + 4 → Rm	1
LDS.L	@Rm+,A0	0100mmmm01110110	(Rm) → A0, Rm + 4 → Rm	1
LDS.L	@Rm+,X0	0100mmmm10000110	(Rm) → X0, Rm + 4 → Rm	1

			(repeat status) → RF1, RF0 Rm[11:0] → RC (SR[27:16])	
SETRC	#imm	10000010iiiiiiii	RE–RS operation result (repeat status) → RF1, RF0 imm → RC (SR[23:16]), 0 → SR[27:24]	1
SETT		0000000000011000	1 → T	1
SLEEP		0000000000011011	Sleep	3*
STC	SR,Rn	0000nnnn00000010	SR → Rn	1
STC	GBR,Rn	0000nnnn00010010	GBR → Rn	1
STC	VBR,Rn	0000nnnn00100010	VBR → Rn	1
STC	MOD,Rn	0000nnnn01010010	MOD → Rn	1
STC	RE,Rn	0000nnnn01110010	RE → Rn	1
STC	RS,Rn	0000nnnn01100010	RS → Rn	1
STC.L	SR,@–Rn	0100nnnn00000011	Rn–4 → Rn, SR → (Rn)	2
STC.L	GBR,@–Rn	0100nnnn00010011	Rn–4 → Rn, GBR → (Rn)	2
STC.L	VBR,@–Rn	0100nnnn00100011	Rn–4 → Rn, VBR → (Rn)	2
STC.L	MOD,@–Rn	0100nnnn01010011	Rn–4 → Rn, MOD → (Rn)	2
STC.L	RE,@–Rn	0100nnnn01110011	Rn–4 → Rn, RE → (Rn)	2
STC.L	RS,@–Rn	0100nnnn01100011	Rn–4 → Rn, RS → (Rn)	2
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1
STS	PR,Rn	0000nnnn00101010	PR → 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	Y0,@–Rn	0100nnnn10100010	Rn–4 → Rn, Y0 → (Rn)	1
STS.L	Y1,@–Rn	0100nnnn10110010	Rn–4 → Rn, Y1 → (Rn)	1
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	8

Note: * The number of execution cycles before the chip enters sleep mode.

Precautions Concerning the Number of Instruction Execution Cycles: The execut
listed in the tables are minimum values. In practice, the number of execution cycles in
under such conditions as 1) when the instruction fetch is in contention with a data acc
the destination register of a load instruction (memory → register) is the same as the re
by the next instruction, 3) when the branch destination address of a branch instruction
address.

CPU Instructions That Support DSP Functions: A number of system control instru
been added to the CPU core instructions to support DSP functions. The RS, RE and M
registers have been added to support repeat control and modulo addressing, and the re
(RC) has been added to the status register (SR). The LDC and STC instructions have b
in order to access the aforementioned. The LDS and STS instructions have been added
access the DSP registers DSR, A0, X0, X1, Y0 and Y1.

The SETRC instruction has been added to set the repeat counter (RC, bits 27 to 16) an
flags (RF1, RF0, bits 3 and 2) of the SR register. When the SETRC instruction operan
immediate, the 8-bit immediate data is stored in bits 23 to 16 of the SR register and bi
are cleared to 0. When the operand is a register, bits 11 to 0 (12 bits) of the register are
bits 27 to 16 of the SR register. Additionally, the status of 1 instruction repeat (00), 2 i
repeat (01), 3 instruction repeat (11) or 4 instruction or greater repeat (10) is set from t
RE set values.

In addition to the LDC instruction, the LDRS and LDRE instructions have been added
establishing the repeat start and repeat end addresses in the RS and RE registers.

The added instructions are listed in table 2.26.

LDC.L	@Rm+,RS	0100mmmm01100111	(Rm)→RS,Rm+4→Rm	3
STC	MOD,Rn	0000nnnn01010010	MOD→Rn	1
STC	RE,Rn	0000nnnn01110010	RE→Rn	1
STC	RS,Rn	0000nnnn01100010	RS→Rn	1
STC.L	MOD,@-Rn	0100nnnn01010011	Rn–4→Rn,MOD→(Rn)	2
STC.L	RE,@-Rn	0100nnnn01110011	Rn–4→Rn,RE→(Rn)	2
STC.L	RS,@-Rn	0100nnnn01100011	Rn–4→Rn,RS→(Rn)	2
LDS	Rm,DSR	0100mmmm01101010	Rm→DSR	1
LDS.L	@Rm+,DSR	0100mmmm01100110	(Rm)→DSR,Rm+4→Rm	1
LDS	Rm,A0	0100mmmm01111010	Rm→A0	1
LDS.L	@Rm+,A0	0100mmmm01110110	(Rm)→A0,Rm+4→Rm	1
LDS	Rm,X0	0100mmmm10001010	Rm→X0	1
LDS.L	@Rm+,X0	0100mmmm10000110	(Rm)→X0,Rm+4→Rm	1
LDS	Rm,X1	0100mmmm10011010	Rm→X1	1
LDS.L	@Rm+,X1	0100mmmm10010110	(Rm)→X1,Rm+4→Rm	1
LDS	Rm,Y0	0100mmmm10101010	Rm→Y0	1
LDS.L	@Rm+,Y0	0100mmmm10100110	(Rm)→Y0,Rm+4→Rm	1
LDS	Rm,Y1	0100mmmm10111010	Rm→Y1	1
LDS.L	@Rm+,Y1	0100mmmm10110110	(Rm)→Y1,Rm+4→Rm	1
STS	DSR,Rn	0000nnnn01101010	DSR→Rn	1
STS.L	DSR,@-Rn	0100nnnn01100010	Rn–4→Rn,DSR→(Rn)	1
STS	A0,Rn	0000nnnn01111010	A0→Rn	1
STS.L	A0,@-Rn	0100nnnn01110010	Rn–4→Rn,A0→(Rn)	1
STS	X0,Rn	0000nnnn10001010	X0→Rn	1
STS.L	X0,@-Rn	0100nnnn10000010	Rn–4→Rn,X0→(Rn)	1
STS	X1,Rn	0000nnnn10011010	X1→Rn	1

			0→SR[27:24]	
LDRS	@(disp,PC)	10001100dddddddd	disp × 2+PC→RS	1
LDRE	@(disp,PC)	10001110dddddddd	disp × 2+PC→RE	1

2.5.2 DSP Data Transfer Instruction Set

Table 2.27 lists the DSP data transfer instructions by classification.

Table 2.27 Classification of DSP Data Transfer Instructions

Classification	Types	Operation Code	Function	No. In
Double data transfer instructions	4	NOPX	X memory no operation	14
		MOVX	X memory data transfer	
		NOPY	Y memory no operation	
		MOVY	Y memory data transfer	
Single data transfer instructions	1	MOVS	Single data transfer	16
	Total: 5			To

The data transfer instructions are divided into two groups, double data transfers and si
transfers. Double data transfers can be combined with DSP operation instructions to p
parallel processing. The parallel processing instructions are 32 bits in length, and the d
transfer instructions are incorporated into their A fields. Double data transfers that are
processing instructions are 16 bits in length, as are the single data transfer instructions

The X memory and Y memory can be accessed simultaneously in parallel in double da
One instruction each is designated from among the X and Y memory data accesses. Th
pointer is used to access X memory; the Ay pointer is used to access Y memory. Doub
transfers can only access X, Y memory.

	Ax+2→Ax		
MOVX.W @Ax+Ix,Dx	(Ax)→MSW of Dx,0→LSW of Dx, Ax+Ix→Ax	111100A*D*0*11**	1
MOVX.W Da,@Ax	MSW of Da→(Ax)	111100A*D*1*01**	1
MOVX.W Da,@Ax+	MSW of Da→(Ax),Ax+2→Ax	111100A*D*1*10**	1
MOVX.W Da,@Ax+Ix	MSW of Da→(Ax),Ax+Ix→Ax	111100A*D*1*11**	1

Table 2.29 Double Data Transfer Instructions (Y Memory Data)

Instruction	Operation	Code	Cycl
NOPY	No Operation	111100*0*0*0**00	1
MOVY.W @Ay,Dy	(Ay)→MSW of Dy,0→LSW of Dy	111100*A*D*0**01	1
MOVY.W @Ay+,Dy	(Ay)→MSW of Dy,0→LSW of Dy, Ay+2→Ay	111100*A*D*0**10	1
MOVY.W @Ay+Iy,Dy	(Ay)→MSW of Dy,0→LSW of Dy, Ay+Iy→Ay	111100*A*D*0**11	1
MOVY.W Da,@Ay	MSW of Da→(Ay)	111100*A*D*1**01	1
MOVY.W Da,@Ay+	MSW of Da→(Ay),Ay+2→Ay	111100*A*D*1**10	1
MOVY.W Da,@Ay+Iy	MSW of Da→(Ay),Ay+Iy→Ay	111100*A*D*1**11	1

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

Note: * When guard bit registers A0G and A1G are specified for the source operan
sign-extended before being transferred.

Ax					Yes	Yes			
Ix (Is)	—	—	—	—	—	—	—	—	Yes
Dx	—	—	—	—	—	—	—	—	—
Ay	—	—	—	—	—	—	Yes	Yes	—
Iy	—	—	—	—	—	—	—	—	—
Dy	—	—	—	—	—	—	—	—	—
Da	—	—	—	—	—	—	—	—	—
As	—	—	Yes	Yes	Yes	Yes	—	—	—
Ds	—	—	—	—	—	—	—	—	—

Oper-and	DSP Registers								
	X0	**X1**	**Y0**	**Y1**	**M0**	**M1**	**A0**	**A1**	**A0G**
Ax	—	—	—	—	—	—	—	—	—
Ix (Is)	—	—	—	—	—	—	—	—	—
Dx	Yes	Yes	—	—	—	—	—	—	—
Ay	—	—	—	—	—	—	—	—	—
Iy	—	—	—	—	—	—	—	—	—
Dy	—	—	Yes	Yes	—	—	—	—	—
Da	—	—	—	—	—	—	Yes	Yes	—
As	—	—	—	—	—	—	—	—	—
Ds	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note: Yes indicates that the register can be set.

The B field data operation instructions are divided into three groups: double data oper
instructions, conditional single data operation instructions, and unconditional single da
instructions. Table 2.32 lists the instruction formats of the DSP operation instructions.
operands can be independently selected from the DSP registers. Table 2.33 shows the
correspondence between the DSP operation instruction operands and registers.

Table 2.32 DSP Operation Instruction Formats

Classification		Instruction Forms	Instruction
Double data operation instructions (6 operands)		ALUop. Sx, Sy, Du MLTop. Se, Sf, Dg	PADD PMU PSUB PMU
Conditional single data operation instructions	3 operands	ALUop. Sx, Sy, Dz DCT ALUop. Sx, Sy, Dz DCF ALUop. Sx, Sy, Dz	PADD, PAN PSHA, PSH PXOR
	2 operands	ALUop. Sx, Dz DCT ALUop. Sx, Dz DCF ALUop. Sx, Dz ALUop. Sy, Dz DCT ALUop. Sy, Dz DCF ALUop. Sy, Dz	PCOPY, PD PDMSB, PI PLDS, PST
	1 operand	ALUop. Dz DCT ALUop. Dz DCF ALUop. Dz	PCLR, PSH PSHL #imm
Unconditional single data operation instructions	3 operands	ALUop. Sx, Sy, Du MLTop. Se, Sf, Dg	PADDC, PS PMULS
	2 operands	ALUop. Sx, Dz ALUop. Sy, Dz ALUop. Sx, Sy	PCMP, PAE
	1 operand	ALUop. Dz	PSHA #imm PSHL #imm

X0	Yes	—	Yes	Yes	Yes	Yes
X1	Yes	—	Yes	—	Yes	—
Y0	—	Yes	Yes	Yes	Yes	Yes
Y1	—	Yes	Yes	—	—	Yes

When writing parallel instructions, write the B field instructions first, then write the A f
instructions:

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

Text in brackets ([]) can be omitted. The no operation instructions NOPX and NOPY ca
omitted. Semicolons (;) are used to demarcate instruction lines, but can be omitted. If se
are used, the space after the semicolon can be used for comments.

The individual status codes (DC, N, Z, V, GT) of the DSR register are always updated b
unconditional ALU operation instructions and shift operation instructions. Conditional
do not update the status codes, even if the conditions have been met. Multiplication inst
also do not update the status codes. DC bit definitions are determined by the specificati
CS bits in the DSR register.

Table 2.34 lists the DSP operation instructions by classification.

			PADDC	Addition with carry
			PCLR	Clear
			PCMP	Compare
			PCOPY	Copy
			PNEG	Invert sign
			PSUB	Subtraction
			PSUB PMULS	Subtraction and signed multiplication
			PSUBC	Subtraction with borrow
	ALU integer operation instructions	2	PDEC	Decrement
			PINC	Increment
	MSB detection instruction	1	PDMSB	MSB detection
	Rounding operation instruction	1	PRND	Rounding
ALU logical operation instructions		3	PAND	Logical AND
			POR	Logical OR
			PXOR	Logical exclusive OR
Fixed decimal point multiplication instruction		1	PMULS	Signed multiplication
Shift	Arithmetic shift operation instruction	1	PSHA	Arithmetic shift
	Logical shift operation instruction	1	PSHL	Logical shift
System control instructions		2	PLDS	System register load
			PSTS	Store from system register
		Total 23		

PABS Sy,Dz	If Sy≥0,Sy→Dz	111110*********	1
	If Sy<0,0–Sy→Dz	1010100000yyzzzz	
PADD Sx,Sy,Dz	Sx+Sy→Dz	111110*********	1
		10110001xxyyzzzz	
DCT PADD Sx,Sy,Dz	if DC=1,Sx+Sy→Dz if 0,nop	111110*********	1
		10110010xxyyzzzz	
DCF PADD Sx,Sy,Dz	if DC=0,Sx+Sy→Dz if 1,nop	111110*********	1
		10110011xxyyzzzz	
PADD Sx,Sy,Du	Sx+Sy→Du	111110*********	1
PMULS Se,Sf,Dg	MSW of Se × MSW of Sf→Dg	0111eeffxxyygguu	
PADDC Sx,Sy,Dz	Sx+Sy+DC→Dz	111110*********	1
		10110000xxyyzzzz	
PCLR Dz	H'00000000→Dz	111110*********	1
		100011010000zzzz	
DCT PCLR Dz	if DC=1,H'00000000→Dz	111110*********	1
	if 0,nop	100011100000zzzz	
DCF PCLR Dz	if DC=0,H'00000000→Dz	111110*********	1
	if 1,nop	100011110000zzzz	
PCMP Sx,Sy	Sx–Sy	111110*********	1
		10000100xxyy0000	
PCOPY Sx,Dz	Sx→Dz	111110*********	1
		11011001xx00zzzz	
PCOPY Sy,Dz	Sy→Dz	111110*********	1
		1111100100yyzzzz	
DCT PCOPY Sx,Dz	if DC=1,Sx→Dz if 0,nop	111110*********	1
		11011010xx00zzzz	

		11001001xx00zzzz	
PNEG Sy,Dz	0–Sy→Dz	111110**********	1
		1110100100yyzzzz	
DCT PNEG Sx,Dz	if DC=1,0–Sx→Dz	111110**********	1
	if 0,nop	11001010xx00zzzz	
DCT PNEG Sy,Dz	if DC=1,0–Sy→Dz	111110**********	1
	if 0,nop	1110101000yyzzzz	
DCF PNEG Sx,Dz	if DC=0,0–Sx→Dz	111110**********	1
	if 1,nop	11001011xx00zzzz	
DCF PNEG Sy,Dz	if DC=0,0–Sy→Dz	111110**********	1
	if 1,nop	1110101100yyzzzz	
PSUB Sx,Sy,Dz	Sx–Sy→Dz	111110**********	1
		10100001xxyyzzzz	
DCT PSUB Sx,Sy,Dz	if DC=1,Sx–Sy→Dz if 0,nop	111110**********	1
		10100010xxyyzzzz	
DCF PSUB Sx,Sy,Dz	if DC=0,Sx–Sy→Dz if 1,nop	111110**********	1
		10100011xxyyzzzz	
PSUB Sx,Sy,Du	Sx–Sy→Du	111110**********	1
PMULS Se,Sf,Dg	MSW of Se × MSW of Sf→Dg	0110eeffxxyygguu	
PSUBC Sx,Sy,Dz	Sx–Sy–DC→Dz	111110**********	1
		10100000xxyyzzzz	

	if 0, nop	10001010xx00zzzz	
DCT PDEC Sy,Dz	If DC=1, MSW of Sy – 1 → MSW of Dz, clear LSW of Dz; if 0, nop	111110********** 1010101000yyzzzz	1
DCF PDEC Sx,Dz	If DC=0, MSW of Sx – 1 → MSW of Dz, clear LSW of Dz; if 1, nop	111110********** 10001011xx00zzzz	1
DCF PDEC Sy,Dz	If DC=0, MSW of Sy – 1 → MSW of Dz, clear LSW of Dz; if 1, nop	111110********** 1010101100yyzzzz	1
PINC Sx,Dz	MSW of Sx + 1 → MSW of Dz, clear LSW of Dz	111110********** 10011001xx00zzzz	1
PINC Sy,Dz	MSW of Sy + 1 → MSW of Dz, clear LSW of Dz	111110********** 1011100100yyzzzz	1
DCT PINC Sx,Dz	If DC=1, MSW of Sx + 1 → MSW of Dz, clear LSW of Dz; if 0, nop	111110********** 10011010xx00zzzz	1
DCT PINC Sy,Dz	If DC=1, MSW of Sy + 1 → MSW of Dz, clear LSW of Dz; if 0, nop	111110********** 1011101000yyzzzz	1
DCF PINC Sx,Dz	If DC=0, MSW of Sx + 1 → MSW of Dz, clear LSW of Dz; if 1, nop	111110********** 10011011xx00zzzz	1
DCF PINC Sy,Dz	If DC=0, MSW of Sy + 1 → MSW of Dz, clear LSW of Dz; if 1, nop	111110********** 1011101100yyzzzz	1

	Dz; if 0, nop	10011110xx00zzzz	
DCT PDMSB Sy,Dz	If DC=1, Sy data MSB position → MSW of Dz, clear LSW of Dz; if 0, nop	111110********** 1011111000yyzzzz	1
DCF PDMSB Sx,Dz	If DC=0, Sx data MSB position → MSW of Dz, clear LSW of Dz; if 1, nop	111110********** 10011111xx00zzzz	1
DCF PDMSB Sy,Dz	If DC=0, Sy data MSB position → MSW of Dz, clear LSW of Dz; if 1, nop	111110********** 1011111100yyzzzz	1

Table 2.38 Rounding Operation Instructions

Instruction	Operation	Code	Cycle
PRND Sx,Dz	Sx+H'00008000→Dz clear LSW of Dz	111110********** 10011000xx00zzzz	1
PRND Sy,Dz	Sy+H'00008000→Dz clear LSW of Dz	111110********** 1011100000yyzzzz	1

		10010111xxyyzzzz	
POR Sx,Sy,Dz	Sx | Sy → Dz, clear LSW of Dz	111110********** 10110101xxyyzzzz	1
DCT POR Sx,Sy,Dz	If DC=1, Sx | Sy → Dz, clear LSW of Dz; if 0, nop	111110********** 10110110xxyyzzzz	1
DCF POR Sx,Sy,Dz	If DC=0, Sx | Sy → Dz, clear LSW of Dz; if 1, nop	111110********** 10110111xxyyzzzz	1
PXOR Sx,Sy,Dz	Sx ^ Sy → Dz, clear LSW of Dz	111110********** 10100101xxyyzzzz	1
DCT PXOR Sx,Sy,Dz	If DC=1, Sx ^ Sy → Dz, clear LSW of Dz; if 0, nop	111110********** 10100110xxyyzzzz	1
DCF PXOR Sx,Sy,Dz	If DC=0, Sx ^ Sy → Dz, clear LSW of Dz; if 1, nop	111110********** 10100111xxyyzzzz	1

Table 2.40 Fixed Point Multiplication Instructions

Instruction	Operation	Code	Cycles
PMULS Se,Sf,Dg	MSW of Se × MSW of Sf→Dg	111110********** 0100eeff0000gg00	1

DCF PSHA Sx,Sy,Dz	if DC=0 & Sy≥0,Sx<<Sy→Dz if DC=0 & Sy<0,Sx>>Sy→Dz if DC=1,nop	111110 10010011xxyyzzzz	1
PSHA #imm,Dz	if imm≥0,Dz<<imm→Dz if imm<0,Dz>>imm→Dz	111110********** 00010iiiiiiizzzz	1

Table 2.42 Logical Shift Operation Instructions

Instruction	Operation	Code	Cycle
PSHL Sx,Sy,Dz	if Sy≥0,Sx<<Sy→Dz, clear LSW of Dz if Sy<0,Sx>>Sy→Dz, clear LSW of Dz	111110********** 10000001xxyyzzzz	1
DCT PSHL Sx,Sy,Dz	if DC=1 & Sy≥0,Sx<<Sy→Dz, clear LSW of Dz if DC=1 & Sy<0,Sx>>Sy→Dz, clear LSW of Dz if DC=0,nop	111110********** 10000010xxyyzzzz	1
DCF PSHL Sx,Sy,Dz	if DC=0 & Sy≥0,Sx<<Sy→Dz, clear LSW of Dz if DC=0 & Sy<0,Sx>>Sy→Dz, clear LSW of Dz if DC=1,nop	111110********** 10000011xxyyzzzz	1
PSHL #imm,Dz	if imm≥0,Dz<<imm→Dz, clear LSW of Dz if imm<0,Dz>>imm→Dz, clear LSW of Dz	111110********** 00000iiiiiiizzzz	1

	if 0,nop	110111100000zzzz	
DCT PLDS Dz,MACL	if DC=1,Dz→MACL	111110**********	1
	if 0,nop	111111100000zzzz	
DCF PLDS Dz,MACH	if DC=0,Dz→MACH	111110**********	1
	if 1,nop	111011110000zzzz	
DCF PLDS Dz,MACL	if DC=0,Dz→MACL	111110**********	1
	if 1,nop	111111110000zzzz	
PSTS MACH,Dz	MACH→Dz	111110**********	1
		110011010000zzzz	
PSTS MACL,Dz	MACL→Dz	111110**********	1
		110111010000zzzz	
DCT PSTS MACH,Dz	if DC=1,MACH→Dz	111110**********	1
	if 0,nop	110011100000zzzz	
DCT PSTS MACL,Dz	if DC=1,MACL→Dz	111110**********	1
	if 0,nop	110111100000zzzz	
DCF PSTS MACH,Dz	if DC=0,MACH→Dz	111110**********	1
	if 1,nop	110011110000zzzz	
DCF PSTS MACL,Dz	if DC=0,MACL→Dz	111110**********	1
	if 1,nop	110111110000zzzz	

PADD X0, Y0, A0	MOVX.W @R4+, X0	MOVY.W @R6+R9, Y0	1111100000 1011000100
PADD X0, Y0, A0	NOPX	MOVY.W @R6+R9, Y0	1111100000 1011000100
PADD X0, Y0, A0	NOPX	NOPY	1111100000 1011000100
PADD X0, Y0, A0	NOPX		1111100000 1011000100
PADD X0, Y0, A0			1111100000 1011000100
	MOVX.W @R4+, X0	MOVY.W @R6+R9, Y0	1111000000
	MOVX.W @R4+, X0	NOPY	1111000000
	MOVS.W @R4+, X0		1111010010
	NOPX	MOVY.W @R6+R9, Y0	1111000000
		MOVY.W @R6+R9, Y0	1111000000
	NOPX	NOPY	1111000000
NOP			0000000000

2. When the S bit of SR is changed after the DSP instructions are executed, pipeline is executed exactly.

   Execute the processing as described in either A or B below.

   A. After the DSP instructions are executed, don't change the S bit of SR register.

   B. Insert the NOP instruction before the LDC Rn, SR instruction.

      Example:

```
PSHA  #1,A0
PINC  X0,A0   MOVX.W  A1,@R5
NOP
LDC   R0,SR
```

3. When a double-length multiply instruction (MUL.L, DMULU.L, or DMULS.L) or a length multiply-and-accumulate instruction (MAC.L) is executed in combination w operation instruction, a malfunction may occur. See the following conditions and countermeasures.

   Conditions:

   When the following A and B conditions are both satisfied, the instruction shown in below may be executed incorrectly.

   A. An instruction in the on-chip memory or the cache is executed.

   B. The following instructions are executed in the order of a, b, and c.

      a. Double-length multiply instruction (MUL.L, DMULU.L, or DMULS.L) or d length multiply-and-accumulate instruction (MAC.L)

      b. DSP operation instruction other than PMULS, PSTS, and PLDS

         Note: The following instructions are DSP operation instructions other than PSTS, and PLDS:
         PABS, PADD, PADDC, PAND, PCLR, PCMP, PCOPY, PDEC, PDI
         PINC, PNEG, POR, PRND, PSHA, PSHL, PSUB, PSUBC, and PXO

      c. PMULS, PSTS, or PLDS

Countermeasures:

This problem is avoided by any of the following countermeasures.

A. Do not execute the instruction sequence shown in B above.

B. Replace instructions b and c above if the program code includes the instruction shown in B above and replacing instructions b and c does not affect the executi

C. Insert one or more NOP instructions or instructions that are not related to the m between instructions a and b if the program code includes the instruction seque in B above and replacing instructions b and c does affect the execution results.

Supplementary information:

This usage note is also applicable when a delayed branch instruction comes immed before instruction a in B above, the a instruction is placed in the delay slot, and ins and c in B are executed in sequence at the branch destination.

4. This section presents examples of and methods for preventing the instruction exec phenomenon due to multiplier contention caused by multiply and multiply-and-acc instructions.

If the SR (status register) S bit (saturated arithmetic bit) is changed immediately af multiply or multiply-and-accumulate instruction in the state where multiplier conte occurred due to multiply and/or multiply-and-accumulate instructions and instructi execution has stalled, the instruction execution order will be reversed. As a result, instruction that should have been executed before the S bit was changed will be ex the S bit has changed. This can result in an incorrect arithmetic result being produc

Instructions affected by S bit modification:

Multiply-and-accumulate instructions: MAC.W and MAC.L

Conditions:

The following shows an example of error conditions.

A. Multiply instruction and multiply-and-accumulate instruction

a. DMULU.L R4,R10 MUL.L, DMULS.L, DMULU.L, or MAC.L can b a.

Countermeasures:

This problem is avoided by any of the following countermeasures.

A. Do not access SR immediately after the multiply-and-accumulate instruction.

B. Insert a NOP instruction before the LDC Rn,SR instruction.

C. Prevent multiplier access conflict (not to produce stall cycles).

3.2 On-Chip Clock Pulse Generator and Operating Modes

3.2.1 Clock Pulse Generator

A block diagram of the on-chip clock pulse generator circuit is shown in figure 3.1.

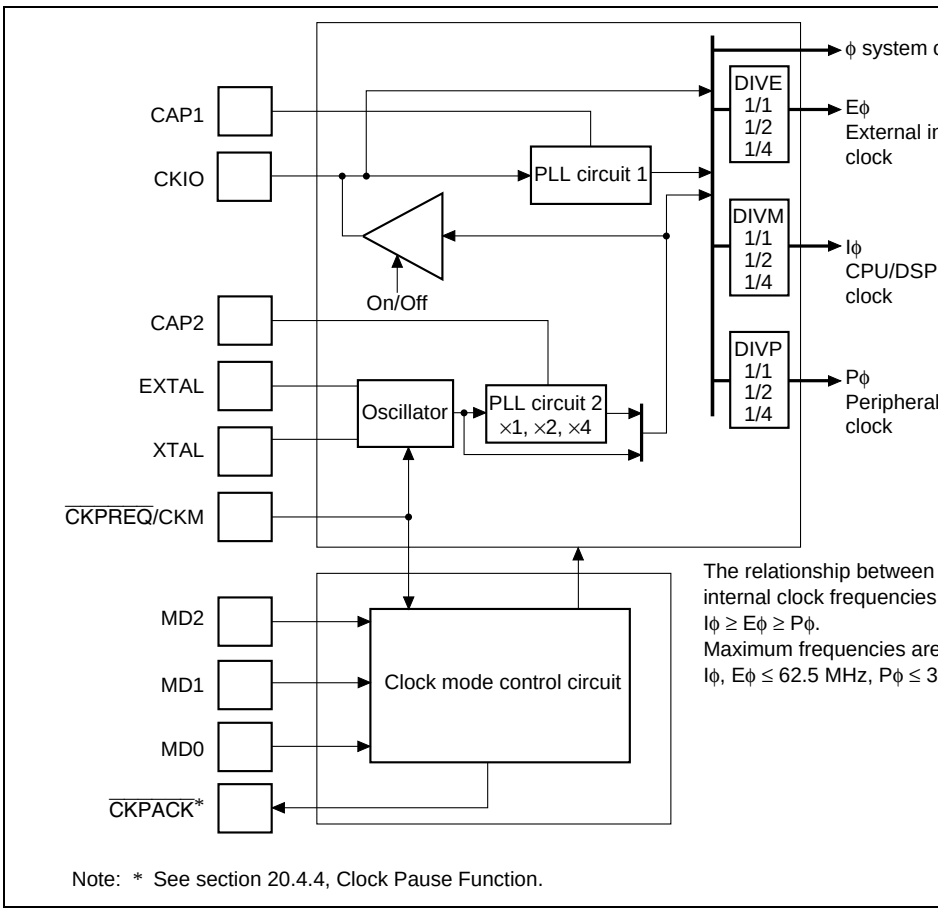


Figure 3.1 Block Diagram of Clock Pulse Generator Circuit

		using PLL circuit 2
CAP1	Input	Connects to capacitance for operating PLL circuit 1
CAP2	Input	Connects to capacitance for operating PLL circuit 2
MD0	Input	The level applied to these pins specifies the clock mode
MD1	Input	
MD2	Input	
$\overline{\text{CKPREQ}}$/CKM	Input	Used as the clock pause request pin, or specifies operation c crystal resonator
$\overline{\text{CKPACK}}$	Output	Clock pause function

PLL Circuit 1: PLL circuit 1 eliminates phase differences between external clocks and supplied internally within the chip. In high-speed operation, the phase difference betwe reference clocks and operating clocks in the chip directly affects the interface margin w peripheral devices. On-chip PLL circuit 1 is provided to eliminate this effect.

PLL Circuit 2: PLL circuit 2 either leaves unchanged, doubles, or quadruples the frequ clocks provided from the crystal resonator or the EXTAL pin external clock input for th operating frequency. The frequency modification register sets the clock frequency mult factor.

	PLL circuits 1 and 2 can be switched between the operating and halted states by means of control bits in the frequency modification register (FMR). The CKIO pin can also be placed in the high-impedance state Normally, mode 0 should be used.	
1	PLL circuits 1 and 2 operate. A clock (with the same frequency as Eϕ) 1/4 ϕ cycle in advance of the chip's internal system clock ϕ is output from the CKIO pin. PLL circuits 1 and 2 can be switched between the operating and halted states by means of control bits in the frequency modification register (FMR). The CKIO pin can also be placed in the high-impedance state. However, clock phase shifting is not performed when PLL circuit 1 is halted. Normally, mode 0 should be used.	
2	Only PLL circuit 2 operates. The clock from PLL circuit 2 is output from the CKIO pin (having the same frequency as the Eϕ). As PLL circuit 1 does not operate, phases are not matched in this mode PLL circuit 2 can be switched between the operating and halted states by means of a control bit in the frequency modification register (FMR). The CKIO pin can also be placed in the high-impedance state	
3	Only PLL circuit 2 operates. The CKIO pin is high-impedance PLL circuit 2 can be switched between the operating and halted states by means of a control bit in the frequency modification register (FMR)	
4	Only PLL circuit 1 operates. Operate PLL circuit 1 when operating with synchronization of the phases of the clock input from the CKIO pin and the internal clocks (Iϕ, Eϕ, Pϕ). PLL circuit 2 does not operate in this mode PLL circuit 1 can be switched between the operating and halted states by means of a control bit in the frequency modification register (FMR)	Externa

	Normally, mode 4 should be used.
6	PLL circuits 1 and 2 do not operate. Set this mode when a clock having a frequency equal to that of clocks the clock input from the CKIO pin is used

The internal clock frequency can be changed in each clock mode (see section 3.2.5, Op
Frequency Selection by Register).

In clock modes 4 to 6, the frequency of the clock input from the CKIO pin can be chang
clock can be stopped (see section 20.4.4, Clock Pause Function).

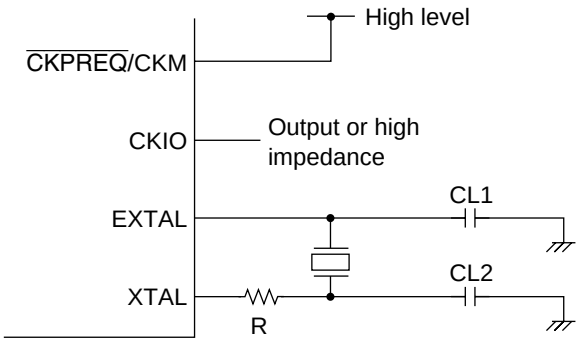
Table 3.3 lists the relationship between pins MD2 to MD0 and the clock operating mod
switch the MD2 to MD0 pins while they are operating. Switching will cause operating

				1	Crystal oscillation	Crystal oscillation	imp
2	0	1	0	0	Clock input	Open	Out imp
				1	Crystal oscillation	Crystal oscillation	
3	0	1	1	0	Clock input	Open	High imp
				1	Crystal oscillation	Crystal oscillation	
4	1	0	0	*	Open	Open	Cloc
5	1	0	1		Open	Open	Cloc
6	1	1	0		Open	Open	Cloc

Notes: Do not use in combinations other than those listed.

* In clock modes 4, 5, and 6, $\overline{\text{CKPREQ}}$/CKM functions as the clock pause re

interfering with correct oscillation.



Notes: 1. The CKIO pin is an output or high impedance in clock modes 0, 1, and 2, and is high impedance in clock mode 3.
2. The values for CL1, CL2, and the damping resistance should be determin consultation with the crystal resonator manufacturer.

Figure 3.2 Example of Crystal Oscillator Connection

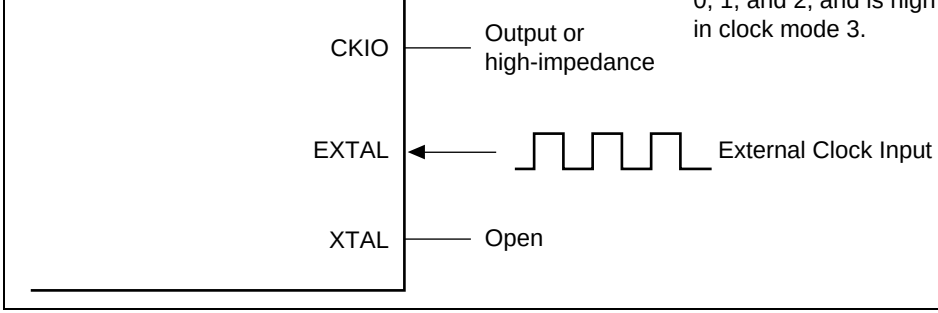


Figure 3.3 External Clock Input Method

Clock Input from CKIO Pin: This method can be used in clock modes 4, 5, and 6.

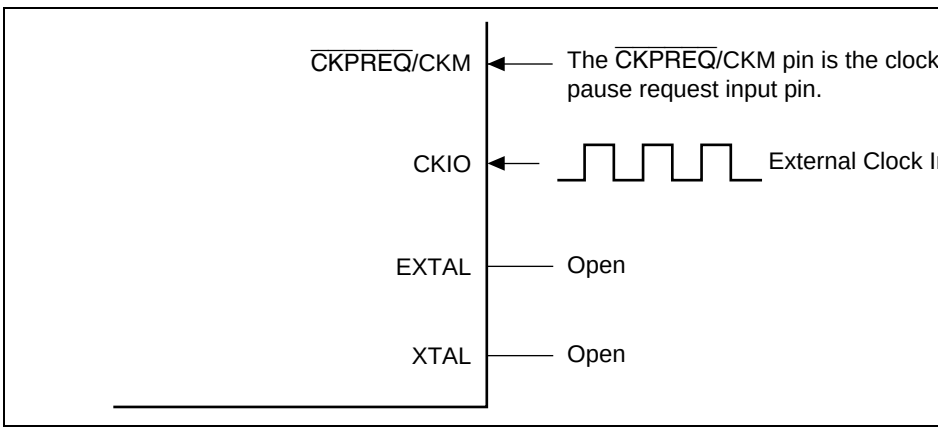


Figure 3.4 External Clock Input Method

between the MD2 to MD0 pin combinations and the initial value of the frequency modi
register.

**Table 3.4 Relationship between Clock Mode Pin Settings and Initial Value of Fr
Modification Register**

Clock Mode	MD2	MD1	MD0	Initial Value
Mode 0	0	0	0	H'00
Mode 1	0	0	1	
Mode 2	0	1	0	H'40
Mode 3	0	1	1	H'60
Mode 4	1	0	0	H'A6
Mode 5	1	0	1	
Mode 6	1	1	0	H'E0

The register configuration is shown in table 3.5.

Table 3.5 Register Configuration

Name	Abbreviation	R/W	Initial Value	Addre
Frequency modification register	FMR	R/W	See table 3.4*	H'FFF

Note: * The initial value depends on the clock mode.

Bit 7: PLL2ST	Description
0	PLL circuit 2 used
1	PLL circuit 2 not used

Bit 6—PLL1ST: Switching is possible in modes 0, 1, 4, and 5. In modes 2, 3, and 6, P
cannot be used. In these modes, this bit always reads 1.

Bit 6: PLL1ST	Description
0	PLL circuit 1 is used
1	PLL circuit 1 is not used

Bit 5—CKIOST: Setting is possible in modes 0 to 3. In modes 4 to 6, the CKIO pin is
pin. In these modes, this bit always reads 1.

Bit 5: CKIOST	Description
0	The CKIO pin outputs Eϕ
1	The CKIO pin is in the high-impedance state (Do not place CKIO i impedance state when PLL circuit 1 is operating)

Bit 4—Reserved: This bit is always read as 0. The write value should always be 0.

Bits 3 to 0—FR3 to FR0: The internal clock frequency and CKIO output frequency (n
can be set by frequency setting bits FR3 to FR0. The values that can be set in bits FR3
depend on the mode and whether PLL circuit 1 and PLL circuit 2 are operating or halt
following tables show the values that can be set in FR3 to FR0, and the internal clock
output frequency ratios, taking the external input clock frequency as 1.

0	1	1	0	×4	×2	×2	×2
1	0	0	0	×4	×4	×1	×1
1	0	0	1	×4	×4	×2	×1
1	0	1	0	×4	×4	×2	×2
1	1	0	0	×4	×4	×4	×1
1	1	1	0	×4	×4	×4	×2

Note: Do not use combinations other than those shown above.

- Modes 0 to 3
  PLL circuit 1 halted, PLL circuit 2 operating
  EXTAL input or crystal resonator used

FR3	FR2	FR1	FR0	ϕ	Iϕ	Eϕ	Pϕ
0	0	0	0	×1	×1	×1	×1
0	1	0	1	×2	×2	×2	×1
0	1	1	0	×2	×2	×2	×2
1	1	0	0	×4	×4	×4	×1
1	1	1	0	×4	×4	×4	×2

Note: Do not use combinations other than those shown above.

Note: Do not use combinations other than those shown above.

- Modes 4 and 5
  PLL circuit 1 operating, PLL circuit 2 halted
  CKIO input

FR3	FR2	FR1	FR0	ϕ	Iϕ	Eϕ	Pϕ
0	1	0	1	×2	×1	×1	×1/2
0	1	1	0	×2	×1	×1	×1
1	0	0	1	×2	×2	×1	×1/2
1	0	1	0	×2	×2	×1	×1

Note: Do not use combinations other than those shown above.

0	1	0	1	×1	×1/2	×1/2	×1/4
0	1	1	0	×1	×1/2	×1/2	×1/2
1	0	0	0	×1	×1	×1/4	×1/4
1	0	0	1	×1	×1	×1/2	×1/4
1	0	1	0	×1	×1	×1/2	×1/2
1	1	0	0	×1	×1	×1	×1/4
1	1	1	0	×1	×1	×1	×1/2
1	1	1	1	×1	×1	×1	×1

Note: Do not use combinations other than those shown above.

Frequency Change: When PLL circuit 1 or PLL circuit 2 becomes operational after m
the frequency modification register (including modification the frequency modification
the operating state), access the frequency modification register using the following proc
noting the cautions listed below.

Frequency change procedure

- Set the on-chip watchdog timer (WDT) overflow time to secure the PLL circuit osc
settling time (CKS2 to CKS0 bits in WTCSR).
- Clear the WT/$\overline{\text{IT}}$ and TME bit to 0 in WTCSR.
- Perform a read anywhere in an external memory area 0 to 4 cache-through area.
- Change the frequency modification register to the target frequency, or change the
operating/halted state of the PLL circuits 1 and 2 (the clocks will stop temporarily i
chip).
- The oscillation circuits operate, and the clock is supplied to the WDT. This clock in
the WDT.
- On WDT overflow, supply of a clock with the frequency set in frequency setting bit
FR0 begins. In this case, the OVF bit in WTSCR and the WOVF bit in RSTCSR are
interval timer interrupt (ITI) is not requested, and the $\overline{\text{WDTOVF}}$ signal is not assert

Sample code for changing the frequency is shown below.

```
XRAM   .equ   h'1000e000

       .export _init_FMR

_init_FMR:
       mov.l  #XRAM,r1
       mov.l  r1,r5
       mov.l  #FREQUENCY,r2
       mov.l  #FREQUENCY_END,r3

program_move:
       mov.w  @r2,r0
       mov.w  r0,@r1
       add    #2,r1
       add    #2,r2
       cmp/eq r2,r3
       bf     program_move
       nop

       mov.l  #PACR,r1
       mov.w  #h'0008,r0
       mov.w  r0,@r1

       MOV.L  #WTCSR,R1
       MOV.W  #H'A51F,R2
       MOV.L  #H'26200000,R3
       MOV.L  #FMR,R4
```

```
        nop
        nop
        nop
        nop
;
;       Main portion of frequency change code.
;       First copy this to XRAM and then run it in XRAM.
FREQUENCY:
        ; <Watchdog timer control and status register setting>
        ; Clear TME bit.
        ; Clock input to WTCNT is ϕ/16384
        ; (Overflow frequency = 262.144 ms)
          MOV.W  R2,@R1

        ; <External cache through area read>
        ; Cache through area of external member space 3: H'26200000
          MOV.L  @R3,R0

        ; <Frequency change register setting>
        ; PLL circuit 1 → Disabled.
        ; PLL circuit 2 → Enabled.
        ; Iϕ (×4) = 62.5 MHz, Eϕ (×4) = 62.5 MHz,
        ; Pϕ (×2) = 31.25 MHz, CKIO (Eϕ) = 62.5 MHz,
        ;   MOV     #H'4E,R0

        ; PLL circuits 1 and 2 → Enabled.
        ; Iϕ (×4) = 62.5 MHz, Eϕ (×2) = 31.25 MHz,
        ; Pϕ (×2 ) = 31.25 MHz, CKIO (Eϕ) = 31.25 MHz,
          MOV     #H'0A,R0
```

```
        nop
FREQUENCY_END:
        NOP


    .END
```

Cautions

- The read from the external memory space 0–4 cache-through area and the write to frequency modification register should be performed in on-chip X/Y memory. Aft from the external memory space 0–4 cache-through area, do not perform any write in external memory spaces 0–4 until the write to the frequency modification regist
- When the write access to the frequency modification register is executed, the WDT automatically.
- Do not turn off the CKIO output when PLL circuit 1 is in the operating state.
- The CKIO output will be unstable until the PLL circuit stabilizes.
- When a frequency is modified, halt the on-chip DMAC (E-DMAC and DMAC) op before the frequency modification.

If PLL circuit 1 or PLL circuit 2 does not become operational after modifying the freq modification register (including modification in the operating state), it means that the procedure or cautions have not been properly observed. In this case, the WDT will not even though the frequency modification register is modified.

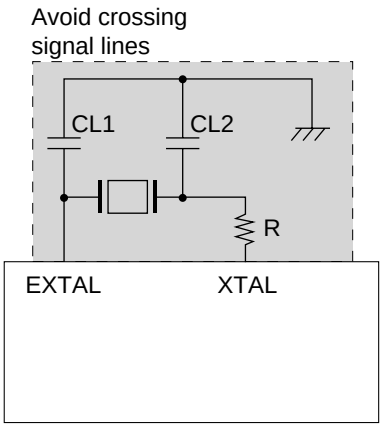
0, 1	EXTAL or crystal resonator*1	8–15.625	On	On	8–62.5	8–62.5	8–31.25	8
			Off	On	8–62.5	8–62.5	8–31.25	8
			On	Off	8–62.5	8–15.625	8–15.625	8
		1–31.25	Off	Off	1–31.25	1–31.25	1–31.25	1
2		8–15.625	Off	On	8–62.5	8–62.5	8–31.25	8
		1–31.25		Off	1–31.25	1–31.25	1–31.25	1
3		8–15.625		On	8–62.5	8–62.5	8–31.25	–
		1–31.25		Off	1–31.25	1–31.25	1–31.25	
4, 5	CKIO	16–31.25	On	Off	16–62.5	16–31.25	16–31.25	–
		1–31.25	Off		1–31.25	1–31.25	1–31.25	
6		1–31.25	Off		1–31.25	1–31.25	1–31.25	

Notes: 1. When a crystal resonator is used, set the frequency in the range of 8 to 15.6

2. Set the frequency modification register so that the frequency of all internal cl MHz or higher.

3. Use internal clock frequencies such that $I\phi \geq E\phi \geq P\phi$.

supply characteristics, or wiring patterns. These values cannot be guaranteed and shou
as reference values. To determine the optimum oscillator circuit constants for the user
please consult with the crystal resonator manufacturer.



CL1, CL2 = 20 to 22 p
R = 0 Ω
Oscillation frequency:
15.625 MHz
Crystal resonator:
HC-49/u manufact
KYUSHU DENTSU

Note: The values for CL1, CL2, and the damping resistance should be determined a
consultation with the crystal resonator manufacturer.

Figure 3.5 Points for Attention when Using Crystal Resonator

Bypass Capacitors: As far as possible, insert a laminated ceramic capacitor of 0.01 to
bypass capacitor for each V_{SS}/V_{CC} pair. Mount the bypass capacitors as close as possil
LSI power supply pins, and use components with a frequency characteristic suitable fo
operating frequency, as well as a suitable capacitance value.

1. V_{SS}/V_{CC} pairs for FP-208C and FP-208CV
   a. PLL system: 9-12
   b. 3 V digital system: 20-18, 26-22, 35-33, 45-42, 52-50, 60-58, 61-67, 69-66, 7
   79-81, 91-89, 101-99, 112-109, 113-110, 114-116, 130-1
   150-147
   c. 5 V digital system: 157-155, 169-167, 181-179, 191-193, 202-200

inductance component. Ground the oscillation stabilization capacitors C1 and C2 to V_{S}
and V_{SS} (PLL2), respectively. Place C1 and C2 close to the CAP1 and CAP2 pins and c
locate a wiring pattern in the vicinity.

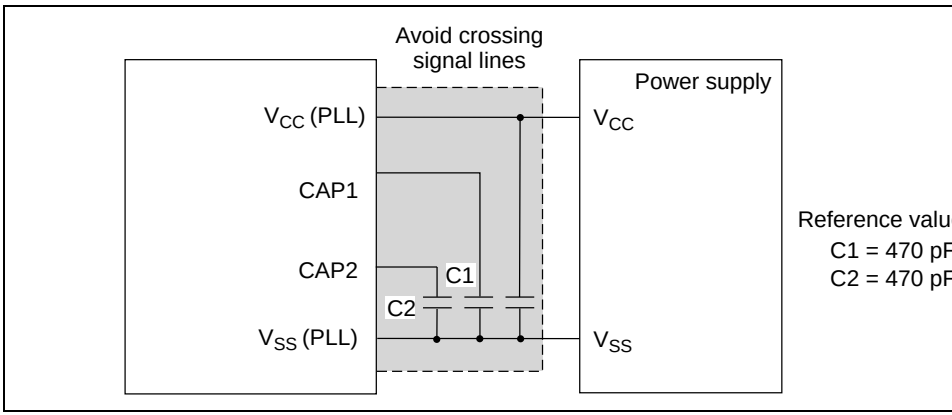


Figure 3.6 Points for Attention when Using PLL Oscillator Circuit

3.3 Bus Width of the CS0 Area

Pins MD3 and MD4 are used to specify the bus width of the CS0 area. The pin combina
functions are listed in table 3.6. Do not switch the MD4 and MD3 pins while they are o
Switching them will cause operating errors.

Table 3.6 Bus Width of the CS0 Area

Pin		
MD4	**MD3**	**Function**
0	0	8-bit bus width selected
0	1	16-bit bus width selected
1	0	32-bit bus width selected
1	1	Setting prohibited

(table 4.2). When several exception sources occur simultaneously, they are accepted according to the priority order shown in table 4.1.

	User break	
	High-performance user debugging interface (H-UDI)	
	External interrupts (IRL1 to IRL15, IRQ0 to IRQ3 (set with $\overline{\text{IRL3}}$, $\overline{\text{IRL2}}$, $\overline{\text{IRL1}}$, $\overline{\text{IRL0}}$ pins))	
	On-chip peripheral modules	Direct memory access controller (DMAC)
		Watchdog timer (WDT)
		Compare match interrupt (part of the bus state controller)
		Ethernet controller (EtherC) and Ethernet controller direct memory access controller (E-DMAC)
		16-bit free-running timer (FRT)
		Serial communication interface with FIFO (SCIF)
		16-bit timer pulse unit (TPU)
		Serial I/O (SIO)
Instructions	Trap instruction (TRAPA)	
	General illegal instructions (undefined code)	
	Illegal slot instructions (undefined code placed directly following a delayed branch instruction*1 or instructions that rewrite the PC*2)	

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BRAF

2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, T BF/S, BT/S, BSRF, BRAF

	Manual reset	Starts when the NMI pin is low and the $\overline{\text{RES}}$ pin cha low to high
Address error		Detected when instruction is decoded and starts wh previous executing instruction finishes executing
Interrupts		Detected when instruction is decoded and starts wh previous executing instruction finishes executing
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction
	General illegal instructions	Starts from the decoding of undefined code anytime a delayed branch instruction (delay slot)
	Illegal slot instructions	Starts from the decoding of undefined code placed following a delayed branch instruction (delay slot) o instruction that rewrites the PC

When exception handling starts, the CPU operates as follows:

1. Exception handling triggered by reset

   The initial values of the program counter (PC) and stack pointer (SP) are fetched f exception vector table (PC and SP are respectively addresses H'00000000 and H'00 a power-on reset and addresses H'00000008 and H'0000000C addresses for a manu See section 4.1.3, Exception Vector Table, for more information. 0 is then written base register (VBR) and 1111 is written to the interrupt mask bits (I3 to I0) of the s register (SR). The program begins running from the PC address fetched from the e vector table.

2. Exception handling triggered by address errors, interrupts, and instructions

   SR and PC are saved to the stack address indicated by R15. For interrupt exceptio the interrupt priority level is written to the SR's interrupt mask bits (I3 to I0). For a and instruction exception handling, the I3 to I0 bits are not affected. The start addr fetched from the exception vector table and the program begins running from that

address.

Table 4.3 lists the vector numbers and vector table address offsets. Table 4.4 shows vec
address calculations.

Table 4.3 (a) Exception Vector Table

Exception Source		Vector Number	Vector Table Address Offset	Vector A
Power-on reset	PC	0	H'00000000–H'00000003	Vector nu
	SP	1	H'00000004–H'00000007	
Manual reset	PC	2	H'00000008–H'0000000B	
	SP	3	H'0000000C–H'0000000F	
General illegal instruction		4	H'00000010–H'00000013	VBR + (v
(Reserved by system)		5	H'00000014–H'00000017	number ×
Slot illegal instruction		6	H'00000018–H'0000001B	
(Reserved by system)		7	H'0000001C–H'0000001F	
		8	H'00000020–H'00000023	
CPU address error		9	H'00000024–H'00000027	
DMA address error (DMAC and E-DMAC)		10*5	H'00000028–H'0000002B	
Interrupt	NMI	11	H'0000002C–H'0000002F	
	User break	12	H'00000030–H'00000033	
	H-UDI	13	H'00000034–H'00000037	
(Reserved by system)		14	H'00000038–H'0000003B	
		:	:	
		31	H'0000007C–H'0000007F	
Trap instruction (user vector)		32	H'00000080–H'00000083	
		:	:	
		63	H'000000FC–H'000000FF	

	peripheral module*3	:	:
		127*4	H'000001FC–H'000001FF

Table 4.3 (c) Exception Processing Vector Table (IRL Mode)

Exception Source		Vector Number	Vector Table Address Offset	Vector
Interrupt	IRL1*1	64*2	H'00000100–H'00000103	VBR + (
	IRL2*1	65*2	H'00000104–H'00000107	number
	IRL3*1			
	IRL4*1	66*2	H'00000108–H'0000010B	
	IRL5*1			
	IRL6*1	67*2	H'0000010C–H'0000010F	
	IRL7*1			
	IRL8*1	68*2	H'00000110–H'00000113	
	IRL9*1			
	IRL10*1	69*2	H'00000114–H'00000117	
	IRL11*1			
	IRL12*1	70*2	H'00000118–H'0000011B	
	IRL13*1			
	IRL14*1	71*2	H'0000011C–H'0000011F	
	IRL15*1			
	On-chip peripheral module*3	0*4	H'00000000–H'00000003	
		:	:	
		127*4	H'000001FC–H'000001FF	

Notes: 1. When 1110 is input to the $\overline{\text{IRL3}}$, $\overline{\text{IRL2}}$, $\overline{\text{IRL1}}$, and $\overline{\text{IRL0}}$ pins, an IRL1 interru
When 0000 is input, an IRL15 interrupt results.

2. External vector number fetches can be performed without using the auto-ve
numbers in this table.

(EDOCR) must therefore be read in the exception service routine to determi
DMA address error has occurred.

Table 4.4 Calculating Exception Vector Table Addresses

Exception Source	Vector Table Address Calculation	
Power-on reset Manual reset	(Vector table address)	= (vector table address offset) = (vector number) × 4
Other exception handling	(Vector table address)	= VBR + (vector table address o = VBR + (vector number) × 4

Note: VBR: Vector base register
Vector table address offset: See table 4.3.
Vector number: See table 4.3.

(FMR) are initialized. (Use the power-on reset when turning the power on.)

Table 4.5 Types of Resets

	Conditions for Transition to Reset Status		Internal Status	
Type	**NMI Pin**	**$\overline{RES}$ Pin**	**CPU**	**On-Chip Peripheral**
Power-on reset	High	Low	Initialized	Initialized
Manual reset	Low	Low	Initialized	Initialized except for E PFC, and frequency r register (FMR)

4.2.2 Power-On Reset

When the NMI pin is high and the $\overline{RES}$ pin is driven low, the device performs a powe
For a reliable reset, the $\overline{RES}$ pin should be kept low for at least the duration of the osc
settling time (when the PLL circuit is halted) or for $20t_{pcyc}$ (when the PLL circuit is ru
During a power-on reset, the CPU's internal state and all on-chip peripheral module re
initialized. See Appendix B, Pin States, for the state of individual pins in the power-o

In a power-on reset, power-on reset exception handling starts when the NMI pin is kep
the $\overline{RES}$ pin is first driven low for a set period of time and then returned to high. The C
then operate as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched f
exception vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception vector tabl
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bi
of the status register (SR) are set to H'F (1111).
4. The values fetched from the exception vector table are set in the program counter (
stack pointer (SP), and the program begins executing.

state of individual pins in the manual reset state.

In a manual reset, manual reset exception handling starts when the NMI pin is kept low
$\overline{\text{RES}}$ pin is first kept low for a set period of time and then returned to high. The CPU wi
operate in the same way as for a power-on reset.

4.3 Address Errors

4.3.1 Sources of Address Errors

Address errors occur when instructions are fetched or data read or written, as shown in

		module space	
		Instruction fetched from on-chip peripheral module space	Addre occur
Data read/write	CPU or DMAC, E-DMAC	Word data accessed from even address	None
		Word data accessed from odd address	Addre occur
		Longword data accessed from a longword boundary	None
		Longword data accessed from other than a longword boundary	Addre occur
		Access of cache purge space, address array read/write space, on-chip peripheral module space, or synchronous DRAM mode setting space by PC-relative addressing	Addre occur
		Access of cache purge space, address array read/write space, data array read/write space, on-chip peripheral module space, or synchronous DRAM mode setting space by a TAS.B instruction	Addre occur
		Byte, word, or longword data accessed in on-chip peripheral module space at addresses H'FFFFFC00 to H'FFFFFCFF	None
		Longword data accessed in on-chip peripheral module space at addresses H'FFFFFE00 to H'FFFFFEFF	Addre occur
		Word or byte data accessed in on-chip peripheral module space at addresses H'FFFFFE00 to H'FFFFFEFF	None
		Byte data accessed in on-chip peripheral module space at addresses H'FFFF0000 to H'FFFFF0FF or H'FFFFFF00 to H'FFFFFFFF	Addre occur
		Word or longword data accessed in on-chip peripheral module space at addresses H'FFFF0000 to H'FFFFF0FF or H'FFFFFF00 to H'FFFFFFFF	None

Notes: 1. Address errors do not occur during the synchronous DRAM mode register w
2. 16-byte DMAC transfers use longword accesses.

3. The exception service routine start address is fetched from the exception vector tabl
corresponds to the address error that occurred, and the program starts executing fro
address. The jump that occurs is not a delayed branch.

Note: The same vector number, 10, is generated for a DMAC DMA address error and
DMAC DMA address error. (See table 4.3 (a).)

Both the address error flag (AE) in the DMAC's DMA operation register (DMA
the address error control bit (AEC) in the E-DMAC's E-DMAC operation cont
(EDOCR) must therefore be read in the exception service routine to determine
DMA address error has occurred.

Type	Request Source	Number
NMI	NMI pin (external input)	1
User break	User break controller (UBC)	1
H-UDI	High-performance user debugging interface (H-UDI)	1
IRL	IRL1 to IRL15 (external input)	15
IRQ	IRQ0 to IRQ3 (external input)	4
On-chip peripheral module	Direct memory access controller (DMAC)	2
	Ethernet controller (EtherC) and Ethernet controller direct memory access controller (E-DMAC)	1
	16-bit free-running timer (FRT)	3
	Watchdog timer (WDT)	1
	Bus state controller (BSC)	1
	Serial I/O (SIO)	4
	Serial communication interface with FIFO (SCIF)	4
	16-bit timer pulse unit (TPU)	13

Each interrupt source is allocated a different vector number and vector table address o
table 5.4, Interrupt Exception Vectors and Priority Order, in section 5, Interrupt Contr
more information.

1 to 15. On-chip peripheral module interrupt priority levels can be set freely using the I
interrupt priority level setting registers A to E (IPRA to IPRE) as shown in table 4.8. Th
levels that can be set are 0 to 15. Level 16 cannot be set. For more information on IPRA
see sections 5.3.1, Interrupt Priority Level Setting Register A (IPRA), to 5.3.5, Interrup
Level Setting Register E (IPRE).

Table 4.8 Interrupt Priority Order

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked
User break	15	Fixed priority level
H-UDI	15	Fixed priority level
IRL	1 to 15	Set with $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins
IRQ	0 to 15	Set with interrupt priority level setting regis (IPRC)
On-chip peripheral module	0 to 15	Set with interrupt priority level setting regis D, and E (IPRA, IPRB, IPRD, IPRE)

4.4.3 Interrupt Exception Handling

When an interrupt occurs, its priority level is ascertained by the interrupt controller (IN
is always accepted, but other interrupts are only accepted if they have a priority level hi
the priority level set in the interrupt mask bits (I3 to I0) of the status register (SR).

When an interrupt is accepted, exception handling begins. In interrupt exception handli
CPU saves SR and the program counter (PC) to the stack. The priority level value of th
interrupt is written to SR bits I3 to I0. For NMI, however, the priority level is 16, but th
in I3 to I0 is H'F (level 15). Next, the start address of the exception service routine is fe
the exception vector table for the accepted interrupt, that address is jumped to and exec
begins. For more information about interrupt exception handling, see section 5.4, Interr
Operation.

Type	Source Instruction	Comment
Trap instruction	TRAPA	—
Illegal slot instruction	Undefined code placed immediately after a delayed branch instruction (delay slot) and instructions that rewrite the PC	Delayed branch instructions: JMP BRA, BSR, RTS, RTE, BF/S, BT/ BRAF Instructions that rewrite the PC: J BRA, BSR, RTS, RTE, BT, BF, T BF/S, BT/S, BSRF, BRAF
General illegal instruction	Undefined code anywhere besides in a delay slot	—

4.5.2 Trap Instructions

When a TRAPA instruction is executed, trap instruction exception handling starts. Th operates as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the start ad instruction to be executed after the TRAPA instruction.
3. The exception service routine start address is fetched from the exception vector ta corresponds to the vector number specified by the TRAPA instruction. That addres to and the program starts executing. The jump that occurs is not a delayed branch.

2. The program counter (PC) is saved to the stack. The PC value saved is the jump add
delayed branch instruction immediately before the undefined code or the instruction
rewrites the PC.
3. The exception service routine start address is fetched from the exception vector tabl
corresponds to the exception that occurred. That address is jumped to and the progra
executing. The jump that occurs is not a delayed branch.

4.5.4 General Illegal Instructions

When undefined code placed anywhere other than immediately after a delayed branch i
(i.e., in a delay slot) is decoded, general illegal instruction exception handling starts. Th
handles general illegal instructions in the same way as illegal slot instructions. Unlike p
of illegal slot instructions, however, the program counter value saved is the start addres
undefined code.

	Exception Sou	
Point of Occurrence	**Address Error**	**Int**
Immediately after a delayed branch instruction*1	Not accepted	No
Immediately after an interrupt-disabled instruction*2	Accepted	No
A repeat loop comprising up to three instructions (instruction fetch cycle not generated) First instruction or last three instructions in a repeat loop containing four or more instructions	Not accepted	No
Fourth from last instruction in a repeat loop containing four or more instructions	Accepted	No

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S
BRAF
2. Interrupt-disabled instructions: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS

4.6.1 Immediately after a Delayed Branch Instruction

When an instruction placed immediately after a delayed branch instruction (delay slot) neither address errors nor interrupts are accepted. The delayed branch instruction and instruction located immediately after it (delay slot) are always executed consecutively exception handling occurs between the two.

4.6.2 Immediately after an Interrupt-Disabled Instruction

When an instruction immediately following an interrupt-disabled instruction is decode are not accepted. Address errors are accepted.

A. All interrupts and address errors are accepted.
B. Address errors only are accepted.
C. No interrupts or address errors are accepted.

When RC ≥ 1

(1) One instruction

```
                 ← A
        instr0
                 ← B
Start (End):instr1
                 ← C
        instr2
                 ← A
```

(2) Two instructions

```
                 ← A
        instr0
                 ← B
Start:  instr1
                 ← C
End:    instr2
                 ← C
        instr3
                 ← A
```

(3) Three instruc

```
        instr(
Start:  instr1
        instr2
End:    instr3
        instr4
```

(4) Four or more instructions

```
                 ← A
        instr0
                 ← A or C (on return from instr n)
Start:instr1
                 ← A
        :
                 :
        :
                 ← A
        instr n-3
                 ← B
        instr n-2
                 ← C
        instr n-1
                 ← C
End:    instr n
                 ← C
        instr n+1
                 ← A
```

When RC = 0

All interrupts and address errors are accepted.

Figure 4.1 Interrupt Acceptance Restrictions in Repeat Mode

Trap instruction	SP →	Address of instruction after TRAPA instruction
		SR
General illegal instruction	SP →	Start address of illegal instruction
		SR
Interrupt	SP →	Address of instruction after executed instruction
		SR
Illegal slot instruction	SP →	Jump destination address of delayed branch instruc
		SR

The value of the vector base register must always be a multiple of four, otherwise an ad
will occur when the vector table is accessed during exception handling.

4.8.3 Address Errors Caused by Stacking of Address Error Exception Handlin

If the stack pointer value is not a multiple of four, an address error will occur during sta
the exception handling (interrupts, etc.). Address error exception handling will begin af
original exception handling ends, but address errors will continue to occur. To ensure th
error exception handling does not go into an endless loop, no address errors are accepte
point. This allows program control to be shifted to the address error exception service r
enables error handling to be carried out.

When an address error occurs during exception handling stacking, the stacking bus cycl
executed. In stacking of the status register (SR) and program counter (PC), the SP is de
by 4 for both, so the value of SP will not be a multiple of four after the stacking either.
address value output during stacking is the SP value, so the address where the error occ
itself output. This means that the write data stacked will be undefined.

4.8.4 Manual Reset during Register Access

Do not initiate a manual reset during access of a bus state controller (BSC), user break c
(UBC), or pin function controller (PFC) register, or the frequency modification register
otherwise a write error may result.

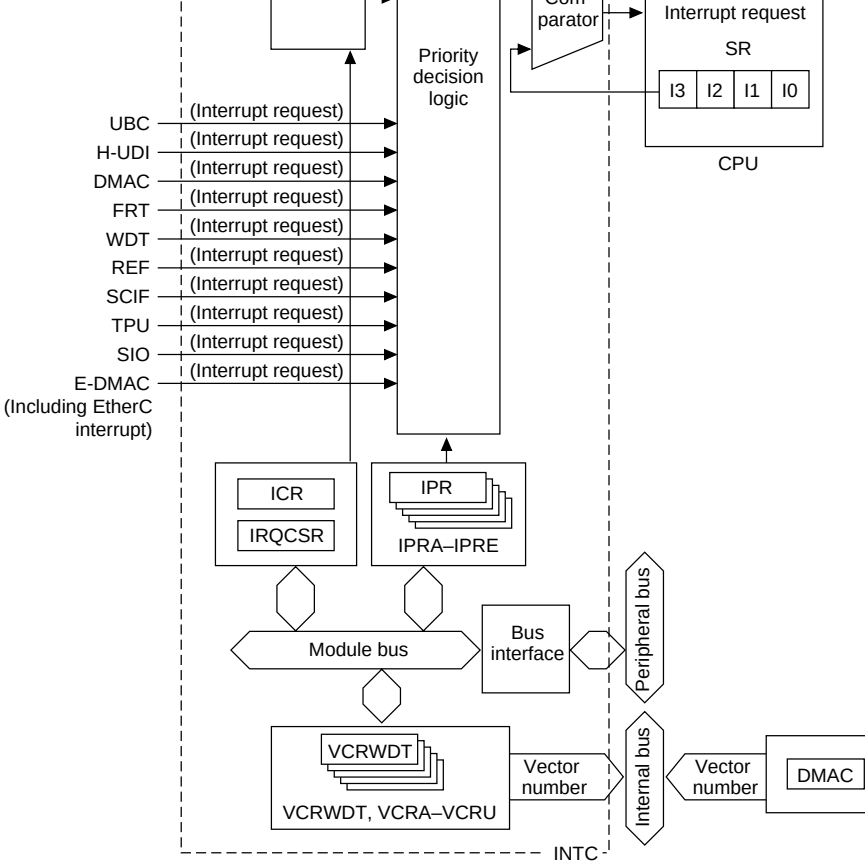
5.1.1 Features

The INTC has the following features:

- Sixteen interrupt priority levels can be set

  By setting the five interrupt priority registers, the priorities of on-chip peripheral m
  interrupts can be selected at 16 levels for different request sources.
- Vector numbers for on-chip peripheral module interrupt can be set

  By setting the 24 vector number setting registers, the vector numbers of on-chip pe
  module interrupts can be set to values from 0 to 127 for different request sources.
- The IRL interrupt vector number setting method can be selected: Either of two mo
  selected by a register setting: auto-vector mode in which vector numbers are deter
  internally, and external vector mode in which vector numbers are set externally.
- IRQ interrupt settings can be made (low level, rising-, falling-, or both-edge detect



UBC: User break controller
H-UDI: High-performance user debugging interface
DMAC: Direct memory access controller
FRT: 16-bit free-running timer
WDT: Watchdog timer
REF: Refresh request within bus state controller
SCIF: Serial communication interface with FIFO
TPU: 16-bit timer pulse unit
SIO: Serial I/O
E-DMAC: Ethernet controller direct mem
access controller
EtherC: Ethernet controller
ICR: Interrupt control register
IRQCSR: IRQ control/status register
IPRA–IPRE: Interrupt priority level setting
registers A–E
VCRWDT: Vector number setting registe
VCRA–VCRU: Vector number setting registe
SR: Status register

Figure 5.1 INTC Block Diagram

Level request interrupt input pins	IRL3 to IRL0	Input	Input of maskable interrup signals
Interrupt acceptance level output pins	A3 to A0	Output	In external vector mode, o interrupt level signal when interrupt is accepted
External vector fetch pin	$\overline{\text{IVECF}}$	Output	Indicates external vector r
External vector number input pins	D7 to D0	Input	Input external vector numb

5.1.4 Register Configuration

The INTC has the 31 registers shown in table 5.2. These registers perform various INT including setting interrupt priority, and controlling external interrupt input signal dete

Table 5.2 Register Configuration

Name	Abbr.	R/W	Initial Value	Address
Interrupt priority register setting register A	IPRA	R/W	H'0000	H'FFFFFE
Interrupt priority register setting register B	IPRB	R/W	H'0000	H'FFFFFE
Interrupt priority register setting register C	IPRC	R/W	H'0000	H'FFFFFE
Interrupt priority register setting register D	IPRD	R/W	H'0000	H'FFFFFE
Interrupt priority register setting register E	IPRE	R/W	H'0000	H'FFFFFE
Vector number setting register A	VCRA	R/W	H'0000	H'FFFFFE
Vector number setting register B*3	VCRB	R/W	H'0000	H'FFFFFE
Vector number setting register C	VCRC	R/W	H'0000	H'FFFFFE
Vector number setting register D	VCRD	R/W	H'0000	H'FFFFFE
Vector number setting register E	VCRE	R/W	H'0000	H'FFFFFE
Vector number setting register F	VCRF	R/W	H'0000	H'FFFFFE
Vector number setting register G	VCRG	R/W	H'0000	H'FFFFFE

Vector number setting register N	VCRN	R/W	H'0000	H'FFFFFE5
Vector number setting register O	VCRO	R/W	H'0000	H'FFFFFE5
Vector number setting register P	VCRP	R/W	H'0000	H'FFFFFEC
Vector number setting register Q	VCRQ	R/W	H'0000	H'FFFFFEC
Vector number setting register R	VCRR	R/W	H'0000	H'FFFFFEC
Vector number setting register S	VCRS	R/W	H'0000	H'FFFFFEC
Vector number setting register T	VCRT	R/W	H'0000	H'FFFFFEC
Vector number setting register U	VCRU	R/W	H'0000	H'FFFFFEC
Vector number setting register WDT	VCRWDT	R/W	H'0000	H'FFFFFEE
Vector number setting register DMA0*4	VCRDMA0	R/W	Undefined	H'FFFFFFA
Vector number setting register DMA1*4	VCRDMA1	R/W	Undefined	H'FFFFFFA
Interrupt control register	ICR	R/W	H'8000/ H'0000*1	H'FFFFFEE
IRQ control/status register	IRQCSR	R/W	*2	H'FFFFFEE

Notes: 1. The value when the NMI pin is high is H'8000; when the NMI pin is low, it is

2. When pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ are high, bits 7 to 4 in IRQCSR are set to 1. When p $\overline{\text{IRL0}}$ are low, bits 7 to 4 in IRQCSR are cleared to 0. The initial value of bits 7 to 4 is 0.

3. In the SH7615, VCRB is a reserved register and must not be accessed.

4. See section 11, Direct Memory Access Controller (DMAC), for more informa VCRDMA0, and VCRDMA1.

5.2 Interrupt Sources

There are five types of interrupt sources: NMI, user breaks, H-UDI, IRL/IRQ and on-cl peripheral modules. Each interrupt has a priority expressed as a priority level (0 to 16, lowest and 16 the highest). Giving an interrupt a priority level of 0 masks it.

A user break interrupt has priority level 15 and occurs when the break condition set in
break controller (UBC) is satisfied. User break interrupt exception handling sets the in
level bits (I3 to I0) in the status register (SR) to level 15. For more information about
break interrupt, see section 6, User Break Controller.

5.2.3 H-UDI Interrupt

The H-UDI interrupt has a priority level of 15, and is generated when an H-UDI intern
instruction is serially input. H-UDI interrupt exception processing sets the interrupt m
to I0) in the status register (SR) to level 15. See section 17, High-Performance User D
Interface (H-UDI), for details of the H-UDI interrupt.

5.2.4 IRL Interrupts

IRL interrupts are requested by input from pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$. Fifteen interrupts, IRL1
can be input externally via pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$. The priority levels of interrupts IRL15
15 to 1, respectively, and their vector numbers are 71 to 64. Set the vector numbers wi
interrupt vector mode select (VECMD) bit of the interrupt control register (ICR) to en
input. External input of vector numbers consists of vector numbers 0 to 127 from the e
vector input pins (D7 to D0). When an external vector is used, 0 is input to D7. Intern
are called auto-vectors and vectors input externally are called external vectors. Table 5
priority levels and auto vector numbers.

When an IRL interrupt is accepted in external vector mode, the IRL interrupt level is o
the interrupt acceptance level output pins (A3 to A0). The external vector fetch pin ($\overline{\text{IV}}$
also asserted. The external vector number is read from pins D7 to D0 at this time.

IRL interrupt exception processing sets the interrupt mask level bits (I3 to I0) in the st
(SR) to the priority level value of the IRL interrupt that was accepted.

External vector numbers are 0 to 127, and are input to the external vector input pins (D
during the interrupt vector fetch bus cycle. When an external vector is used, 0 is input t

When an IRQ interrupt is accepted in external vector mode, the IRQ interrupt priority l
output from the interrupt acceptance level output pins (A3 to A0). The external vector f
($\overline{\text{IVECF}}$) is also asserted. The external vector number is read from signals D7 to D0 at t

IRQ interrupt exception processing sets the interrupt mask bits (I3 to I0) in the status re
to the priority level value of the IRQ interrupt that was accepted.

Table 5.3 IRL Interrupt Priority Levels and Auto-Vector Numbers

Pin				Priority Level	Vect Num
$\overline{\text{IRL3}}$	$\overline{\text{IRL2}}$	$\overline{\text{IRL1}}$	$\overline{\text{IRL0}}$		
0	0	0	0	15	71
			1	14	
		1	0	13	70
			1	12	
	1	0	0	11	69
			1	10	
		1	0	9	68
			1	8	
1	0	0	0	7	67
			1	6	
		1	0	5	66
			1	4	
	1	0	0	3	65
			1	2	
		1	0	1	64

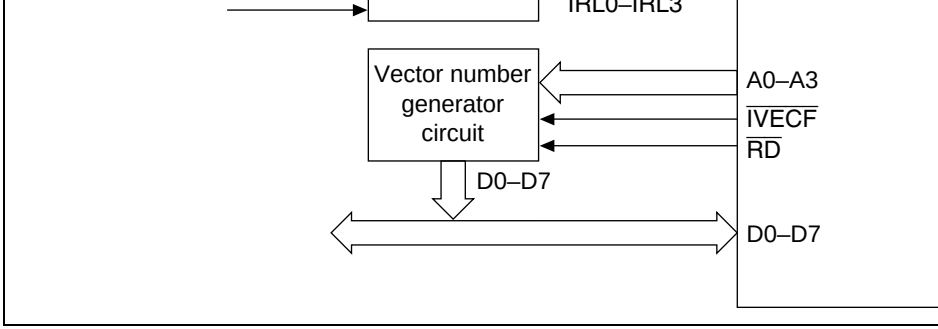


Figure 5.2 Example of Connections for External Vector Mode Interru

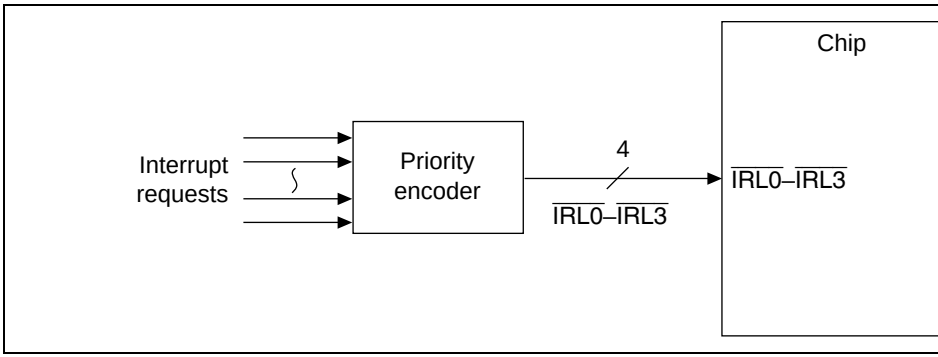


Figure 5.3 Example of Connections for Auto-Vector Mode Interrupt

Figures 5.4 to 5.7 show the interrupt vector fetch cycle for the external vector mode. D
cycle, $\overline{CS0}$ to $\overline{CS4}$ stay high. A24 to A4 output undefined values. The $\overline{WAIT}$ pin is sa
programmable waits are not valid.

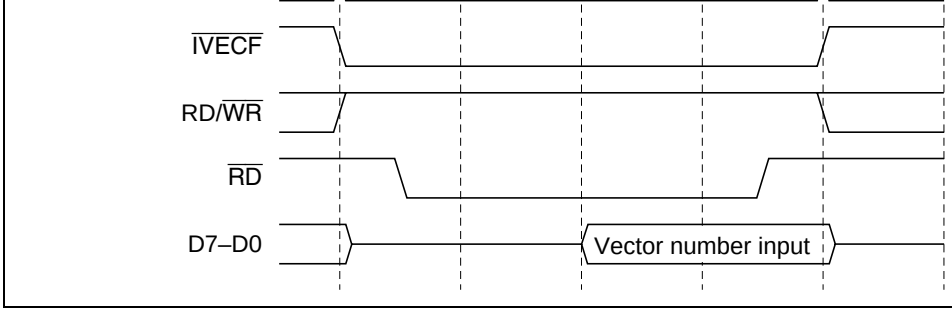


Figure 5.4 External Vector Fetch (Iϕ:Eϕ = 1:1)

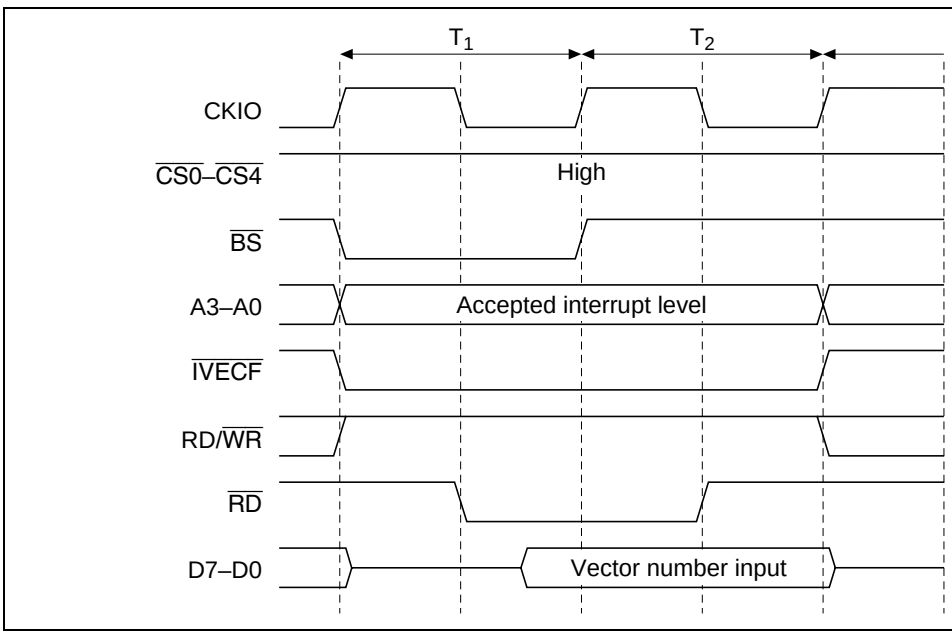


Figure 5.5 External Vector Fetch (Iϕ:Eϕ ≠ 1:1)

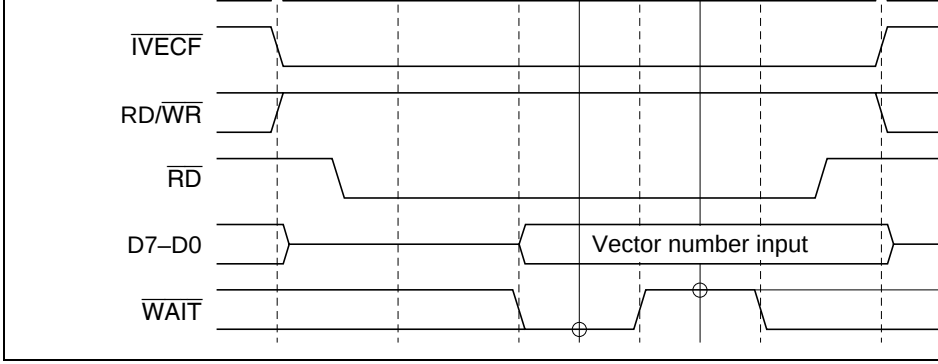


Figure 5.6 External Vector Fetch (Iϕ:Eϕ = 1:1 (WAIT Input))

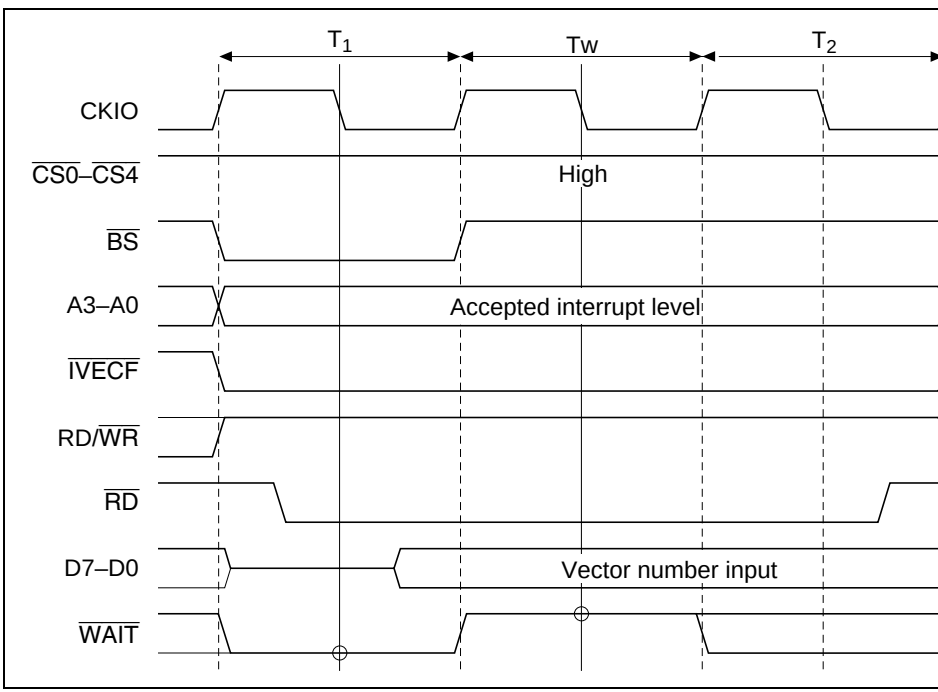


Figure 5.7 External Vector Fetch (Iϕ:Eϕ ≠ 1:1 (WAIT Input))

- Ethernet controller direct memory access controller (E-DMAC) (Including EtherC i
- 16-bit timer pulse unit (TPU)
- Serial communication interface with FIFO (SCIF)
- Serial I/O (SIO)

A different interrupt vector is assigned to each interrupt source, so the exception servic does not have to decide which interrupt has occurred. Priority levels between 0 and 15 assigned to individual on-chip peripheral modules in interrupt priority registers A, B, D (IPRA, IPRB, IPRD, IPRE). On-chip peripheral module interrupt exception handling se interrupt mask level bits (I3 to I0) in the status register (SR) to the priority level value c chip peripheral module interrupt that was accepted.

5.2.7 Interrupt Exception Vectors and Priority Order

Table 5.4 lists interrupt sources and their vector numbers, vector table address offsets a priorities.

Each interrupt source is allocated a different vector number and vector table address of table addresses are calculated from vector numbers and vector table address offsets. In exception handling, the exception service routine start address is fetched from the vecto entry indicated by the vector table address. See table 4.4, Calculating Exception Vector Addresses, in section 4, Exception Handling, for more information on this calculation.

IRL interrupts IRL15 to IRL1 have interrupt priority levels of 15 to 1, respectively. IRQ and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 fo module by setting interrupt priority registers A to E (IPRA to IPRE). The ranking of int sources for IPRA to IPRE, however, must be the order listed under Priority within IPR Unit in table 5.4 and cannot be changed. A reset assigns priority level 0 to on-chip peri module interrupts. If the same priority level is assigned to two or more interrupt source interrupts from those sources occur simultaneously, their priority order is the default pr indicated at the right in table 5.4.

H-UDI		13	15	—	—	—	
IRL15*4		71*1	15	—	—	—	
IRL14*4			14	—	—	—	
IRL13*4		70*1	13	—	—	—	
IRL12*4			12	—	—	—	
IRL11*4		69*1	11	—	—	—	
IRL10*4			10	—	—	—	
IRL9*4		68*1	9	—	—	—	
IRL8*4			8	—	—	—	
IRL7*4		67*1	7	—	—	—	
IRL6*4			6	—	—	—	
IRL5*4		66*1	5	—	—	—	
IRL4*4			4	—	—	—	
IRL3*4		65*1	3	—	—	—	
IRL2*4			2	—	—	—	
IRL1*4		64*1	1	—	—	—	
DMAC0	Transfer end	0–127*2	15–0 (0)	IPRA (11–8)	High ↕ Low	VCRDMA0 (6–0)	
DMAC1	Transfer end	0–127*2	15–0 (0)		High ↕ Low	VCRDMA1 (6–0)	

REF*3	CMI	0–127*2	15–0 (0)		High	VCRWDT (6–0)	
					↕		
					Low		
E-DMAC	EINT*6	0–127*2	15–0 (0)	IPRB (15–12)	High	VCRA (14–8	
					↕		
Reserved					Low	VCRB (14–0	
FRT	ICI	0–127*2	15–0 (0)	IPRB (11–8)	High	VCRC (14–8	
	OCI	0–127*2			↕	VCRC (6–0)	
	OVI	0–127*2			Low	VCRD (14–8	
TPU0	TGI0A	0–127*2	15–0 (0)	IPRD (15–12)	High	VCRE (14–8	
	TGI0B	0–127*2			↕	VCRE (6–0)	
	TGI0C	0–127*2				VCRF (14–8	
	TGI0D	0–127*2				VCRF (6–0)	
	TCI0V	0–127*2			Low	VCRG (14–8	
TPU1	TGI1A	0–127*2	15–0 (0)	IPRD (11–8)	High	VCRH (14–8	
	TGI1B	0–127*2			↕	VCRH (6–0)	
	TCI1V	0–127*2				VCRI (14–8)	
	TCI1U	0–127*2			Low	VCRI (6–0)	
TPU2	TGI2A	0–127*2	15–0 (0)	IPRD (7–4)	High	VCRJ (14–8	
	TGI2B	0–127*2			↕	VCRJ (6–0)	
	TCI2V	0–127*2				VCRK (14–8	
	TCI2U	0–127*2			Low	VCRK (6–0)	

SCIF2	ERI2	0–127*2		15–0 (0)	IPRE (15–12)	High	VCRN (14–
	RXI2	0–127*2				↕	VCRN (6–0
	BRI2	0–127*2					VCRO (14–
	TXI2	0–127*2				Low	VCRO (6–0
SIO0	RERI0	0–127*2		15–0 (0)	IPRE (11–8)	High	VCRP (14–
	TERI0	0–127*2				↕	VCRP (6–0
	RDFI0	0–127*2					VCRQ (14–
	TDEI0	0–127*2				Low	VCRQ (6–0
SIO1	RERI1	0–127*2		15–0 (0)	IPRE (7–4)	High	VCRR (14–
	TERI1	0–127*2				↕	VCRR (6–0
	RDFI1	0–127*2					VCRS (14–
	TDEI1	0–127*2				Low	VCRS (6–0
SIO2	RERI2	0–127*2		15–0 (0)	IPRE (3–0)	High	VCRT (14–
	TERI2	0–127*2				↕	VCRT (6–0
	RDFI2	0–127*2					VCRU (14–
	TDEI2	0–127*2				Low	VCRU (6–0
Reserved		128–255	—	—	—	—	—

Notes: 1. An external vector number fetch can be performed without using the auto-v
numbers shown in this table. The external vector numbers are 0 to 127.
2. Vector numbers are set in the on-chip vector number register.
3. REF is the refresh control unit within the bus state controller.
4. Set to IRL1 to IRL15 or IRQ0 to IRQ3 by the EXIMD bit in ICR.
5. In the SH7615, VCRB is a reserved register and must not be accessed.
6. The E-DMAC interrupt (EINT) is the OR of those of the 19 interrupt sources
EtherC/E-DMAC status register (EESR) that are enabled by the EtherC/E-D
interrupt permission register (EESIPR). As the three status bits in the Ether
register (ECSR) can be copied into the ECI bit in EESR as an interrupt sou
input to the INTC as the OR of a maximum of 22 interrupt sources.

H-UDI		13	15	—	—	—	
IRQ0*4		64*1	15–0 (0)	IPRC (15–12)	—	—	
IRQ1*4		65*1	15–0 (0)	IPRC (11–8)	—	—	
IRQ2*4		66*1	15–0 (0)	IPRC (7–4)	—	—	
IRQ3*4		67*1	15–0 (0)	IPRC (3–0)	—	—	
DMAC0	Transfer end	0–127*2	15–0 (0)	IPRA (11–8)	High ↕ Low	VCRDMA0 (6–0)	
DMAC1	Transfer end	0–127*2	15–0 (0)		High ↕ Low	VCRDMA1 (6–0)	
WDT	ITI	0–127*2	15–0 (0)	IPRA (7–4)	High ↕ Low	VCRWDT (14–8)	
REF*3	CMI	0–127*2	15–0 (0)		High ↕ Low	VCRWDT (6–0)	

FRT	ICI	0–127*2	15–0 (0)	IPRB (11–8)	High	VCRC (14	
	OCIA/B	0–127*2			↕	VCRC (6-	
	OVI	0–127*2				VCRD (14	
					Low		
TPU0	TGI0A	0–127*2	15–0 (0)	IPRD (15–12)	High	VCRE (14	
	TGI0B	0–127*2			↕	VCRE (6–	
	TGI0C	0–127*2				VCRF (14	
	TGI0D	0–127*2				VCRF (6–	
	TCI0V	0–127*2			Low	VCRG (14	
TPU1	TGI1A	0–127*2	15–0 (0)	IPRD (11–8)	High	VCRH (14	
	TGI1B	0–127*2			↕	VCRH (6-	
	TCI1V	0–127*2				VCRI (14-	
	TCI1U	0–127*2			Low	VCRI (6–(	
TPU2	TGI2A	0–127*2	15–0 (0)	IPRD (7–4)	High	VCRJ (14	
	TGI2B	0–127*2			↕	VCRJ (6–	
	TCI2V	0–127*2				VCRK (14	
	TCI2U	0–127*2			Low	VCRK (6–	
SCIF1	ERI1	0–127*2	15–0 (0)	IPRD (3–0)	High	VCRL (14	
	RXI1	0–127*2			↕	VCRL (6–	
	BRI1	0–127*2				VCRM (14	
	TXI1	0–127*2			Low	VCRM (6-	
SCIF2	ERI2	0–127*2	15–0 (0)	IPRE (15–12)	High	VCRN (14	
	RXI2	0–127*2			↕	VCRN (6-	
	BRI2	0–127*2				VCRO (14	
	TXI2	0–127*2			Low	VCRO (6-	

SIO1	RERI1	0–127*2	15–0 (0)	IPRE (7–4)	High	VCRR (14–
	TERI1	0–127*2			↑	VCRR (6–0
	RDFI1	0–127*2			↕	VCRS (14–
	TDEI1	0–127*2			Low	VCRS (6–0
SIO2	RERI2	0–127*2	15–0 (0)	IPRE (3–0)	High	VCRT (14–
	TERI2	0–127*2			↑	VCRT (6–0
	RDFI2	0–127*2			↕	VCRU (14–
	TDEI2	0–127*2			Low	VCRU (6–0
Reserved		128–255 —	—	—	—	—

Notes: 1. An external vector number fetch can be performed without using the auto-ve numbers shown in this table. The external vector numbers are 0 to 127.

2. Vector numbers are set in the on-chip vector number register.
3. REF is the refresh control unit within the bus state controller.
4. Set to IRL1 to IRL15 or IRQ0 to IRQ3 by the EXIMD bit in ICR.
5. In the SH7615, VCRB is a reserved register and must not be accessed.
6. The E-DMAC interrupt (EINT) is the OR of those of the 19 interrupt sources EtherC/E-DMAC status register (EESR) that are enabled by the EtherC/E-D interrupt permission register (EESIPR). As the three status bits in the EtherC register (ECSR) can be copied into the ECI bit in EESR as an interrupt sourc input to the INTC as the OR of a maximum of 22 interrupt sources.

Bit:	15	14	13	12	11	10	9
	—	—	—	—	DMAC IP3	DMAC IP2	DMA IP1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	WDT IP3	WDT IP2	WDT IP1	WDT IP0	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R

Bits 15 to 12—Reserved: These bits are always read as 0. The write value should alwa

Bits 11 to 8—Direct Memory Access Controller (DMAC) Interrupt Priority Level 3 to (DMACIP3 to DMACIP0): These bits set the direct memory access controller (DMAC priority level. There are four bits, so levels 0 to 15 can be set. The same level is set for DMAC channels. When interrupts occur simultaneously, channel 0 has priority.

Bits 7 to 4—Watchdog Timer (WDT) Interrupt Priority Level 3 to 0 (WDTIP3 to WD These bits set the watchdog timer (WDT) interrupt priority level and bus state controll interrupt priority level. There are four bits, so levels 0 to 15 can be set. When WDT an interrupts occur simultaneously, the WDT interrupt has priority.

Bits 3 to 0—Reserved: These bits are always read as 0. The write value should always

Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 15 to 12—Ethernet Controller Direct Memory Access Controller (E-DMAC) Inter
Priority Level 3 to 0 (E-DMACIP3 to E-DMACIP0): These bits set the Ethernet contro
memory access controller (E-DMAC) interrupt priority level. There are four bits, so lev
can be set.

Bits 11 to 8—16-Bit Free-Running Timer (FRT) Interrupt Priority Level 3 to 0 (FRTIP
FRTIP0): These bits set the 16-bit free-running timer (FRT) interrupt priority level. The
bits, so levels 0 to 15 can be set.

Bits 7 to 0—Reserved: These bits are always read as 0. The write value should always b

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	IRQ2IP3	IRQ2IP2	IRQ2IP1	IRQ2IP0	IRQ3IP3	IRQ3IP2	IRQ3I
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0—IRQ0 to IRQ3 Priority Level 3 to 0 (IRQnIP3 to IRQnIP0, n = 0 to 3): T
the IRQ0 to IRQ3 priority levels. There are four bits for each interrupt, so the value ca
between 0 and 15.

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	TPU2IP3	TPU2IP2	TPU2IP1	TPU2IP0	SCF1IP3	SCF1IP2	SCF1IP
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 4—16-Bit Timer Pulse Unit 0 to 2 (TPU0 to TPU2) Interrupt Priority Level 3
(TPUnIP3 to TPUnIP0, n = 0 to 2): These bits set the 16-bit timer pulse unit 0 to 2 (TPU
TPU2) interrupt priority levels. There are four bits for each interrupt, so the value can b
between 0 and 15.

Bits 3 to 0—Serial Communication Interface with FIFO 1 (SCIF1) Interrupt Priority Le
(SCF1IP3 to SCF1IP0): These bits set the serial communication interface with FIFO 1
interrupt priority level. There are four bits, so the value can be set between 0 and 15.

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	SIO1IP3	SIO1IP2	SIO1IP1	SIO1IP0	SIO2IP3	SIO2IP2	SIO2I
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 12—Serial Communication Interface with FIFO 2 (SCIF2) Interrupt Priorit
0 (SCF2IP3 to SCF2IP0): These bits set the serial communication interface with FIFO
interrupt priority level. There are four bits, so the value can be set between 0 and 15.

Bits 11 to 0—Serial I/O 0 to 2 (SIO0 to SIO2) Interrupt Priority Level 3 to 0 (SIOnIP3
SIOnIP0, n = 0 to 2): These bits set the serial I/O 0 to 2 (SIO0 to SIO2) interrupt prior
There are four bits for each interrupt, so the value can be set between 0 and 15.

Table 5.5 shows the relationship between on-chip peripheral module interrupts and int
priority level setting registers.

Table 5.5 Interrupt Request Sources and IPRA to IPRE

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits
Interrupt priority level setting register A	Reserved	DMAC0, DMAC1	WDT, REF	Res
Interrupt priority level setting register B	E-DMAC	FRT	Reserved	Res
Interrupt priority level setting register C	IRQ0	IRQ1	IRQ2	IRQ
Interrupt priority level setting register D	TPU0	TPU1	TPU2	SCI
Interrupt priority level setting register E	SCIF2	SIO0	SIO1	SIO

Vector number setting register WDT (VCRWDT) is a 16-bit read/write register that set
interval interrupt and BSC compare match interrupt vector numbers (0 to 127). VCRW
initialized to H'0000 by a reset. It is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9
	—	WITV6	WITV5	WITV4	WITV3	WITV2	WITV1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	BCMV6	BCMV5	BCMV4	BCMV3	BCMV2	BCMV
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—Watchdog Timer (WDT) Interval Interrupt Vector Number 6 to 0 (WITV
WITV0): These bits set the vector number for the interval interrupt (ITI) of the watchd
(WDT). There are seven bits, so the value can be set between 0 and 127.

Bits 6 to 0—Bus State Controller (BSC) Compare Match Interrupt Vector Number 6 to
(BCMV6 to BCMV0): These bits set the vector number for the compare match interrup
the bus state controller (BSC). There are seven bits, so the value can be set between 0 a

R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 15 and 7 to 0—Reserved: These bits are always read as 0. The write value should
0.

Bits 14 to 8—Ethernet Controller Direct Memory Access Controller (E-DMAC) Inter
Number 6 to 0 (EINV6 to EINV0): These bits set the vector number for Ethernet cont
memory access controller (E-DMAC) interrupt (EINT). There are seven bits, so the va
set between 0 and 127.

5.3.8 Vector Number Setting Register B (VCRB)

Vector number setting register B (VCRB) is a 16-bit reserved register. Access to this r
prohibited. VCRB is initialized to H'0000 by a reset. It is not initialized in standby mo

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	FOCV6	FOCV5	FOCV4	FOCV3	FOCV2	FOCV1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—16-Bit Free-Running Timer (FRT) Input-Capture Interrupt Vector Numb
(FICV6 to FICV0): These bits set the vector number for the 16-bit free-running timer (
capture interrupt (ICI). There are seven bits, so the value can be set between 0 and 127.

Bits 6 to 0—16-Bit Free-Running Timer (FRT) Output-Compare Interrupt Vector Num
(FOCV6 to FOCV0): These bits set the vector number for the 16-bit free-running timer
output-compare interrupt (OCI). There are seven bits, so the value can be set between 0

R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 15 and 7 to 0—Reserved: These bits are always read as 0. The write value should
0.

Bits 14 to 8—16-Bit Free-Running Timer (FRT) Overflow Interrupt Vector Number 6
(FOVV6 to FOVV0): These bits set the vector number for the 16-bit free-running time
overflow interrupt (OVI). There are seven bits, so the value can be set between 0 and

	—	TG0AV6	TG0AV5	TG0AV4	TG0AV3	TG0AV2	TG0AV
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TG0BV6	TG0BV5	TG0BV4	TG0BV3	TG0BV2	TG0BV
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—16-Bit Timer pulse unit 0 (TPU0) TGR0A Input Capture/Compare Match
Vector Number 6 to 0 (TG0AV6 to TG0AV0): These bits set the vector number for the
timer pulse unit 0 (TPU0) TGR0A input capture/compare match interrupt. There are se
the value can be set between 0 and 127.

Bits 6 to 0—16-Bit Timer pulse unit 0 (TPU0) TGR0B Input Capture/Compare Match
Vector Number 6 to 0 (TG0BV6 to TG0BV0): These bits set the vector number for the
timer pulse unit 0 (TPU0) TGR0B input capture/compare match interrupt. There are se
the value can be set between 0 and 127.

	—	TG0CV6	TG0CV5	TG0CV4	TG0CV3	TG0CV2	TG0C
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TG0DV6	TG0DV5	TG0DV4	TG0DV3	TG0DV2	TG0D
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alw

Bits 14 to 8—16-Bit Timer pulse unit 0 (TPU0) TGR0C Input Capture/Compare Mat
Vector Number 6 to 0 (TG0CV6 to TG0CV0): These bits set the vector number for th
timer pulse unit 0 (TPU0) TGR0C input capture/compare match interrupt. There are s
the value can be set between 0 and 127.

Bits 6 to 0—16-Bit Timer pulse unit 0 (TPU0) TGR0D Input Capture/Compare Matc
Vector Number 6 to 0 (TG0DV6 to TG0DV0): These bits set the vector number for th
timer pulse unit 0 (TPU0) TGR0D input capture/compare match interrupt. There are s
the value can be set between 0 and 127.

Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 15 and 7 to 0—Reserved: These bits are always read as 0. The write value should a
0.

Bits 14 to 8—16-Bit Timer pulse unit 0 (TPU0) TCNT0 Overflow Interrupt Vector Nu
(TC0VV6 to TV0VV0): These bits set the vector number for the 16-bit timer pulse unit
TCNT0 overflow interrupt. There are seven bits, so the value can be set between 0 and

	—	TG1AV6	TG1AV5	TG1AV4	TG1AV3	TG1AV2	TG1A
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TG1BV6	TG1BV5	TG1BV4	TG1BV3	TG1BV2	TG1B
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alw

Bits 14 to 8—16-Bit Timer pulse unit 1 (TPU1) TGR1A Input Capture/Compare Mat
Vector Number 6 to 0 (TG1AV6 to TG1AV0): These bits set the vector number for th
timer pulse unit 1 (TPU1) TGR1A input capture/compare match interrupt. There are s
the value can be set between 0 and 127.

Bits 6 to 0—16-Bit Timer pulse unit 1 (TPU1) TGR1B Input Capture/Compare Match
Vector Number 6 to 0 (TG1BV6 to TG1BV0): These bits set the vector number for th
timer pulse unit 1 (TPU1) TGR1B input capture/compare match interrupt. There are s
the value can be set between 0 and 127.

Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TC1UV6	TC1UV5	TC1UV4	TC1UV3	TC1UV2	TC1UV
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—16-Bit Timer pulse unit 1 (TPU1) TCNT1 Overflow Interrupt Vector Nu
(TC1VV6 to TC1VV0): These bits set the vector number for the 16-bit timer pulse unit
TCNT1 overflow interrupt. There are seven bits, so the value can be set between 0 and

Bits 6 to 0—16-Bit Timer pulse unit 1 (TPU1) TCNT1 Underflow Interrupt Vector Nu
(TC1UV6 to TC1UV0): These bits set the vector number for the 16-bit timer pulse unit
TCNT1 underflow interrupt. There are seven bits, so the value can be set between 0 and

	—	TG2AV6	TG2AV5	TG2AV4	TG2AV3	TG2AV2	TG2A
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TG2BV6	TG2BV5	TG2BV4	TG2BV3	TG2BV2	TG2B
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alw

Bits 14 to 8—16-Bit Timer pulse unit 2 (TPU2) TGR2A Input Capture/Compare Mat
Vector Number 6 to 0 (TG2AV6 to TG2AV0): These bits set the vector number for th
timer pulse unit 2 (TPU2) TGR2A input capture/compare match interrupt. There are s
the value can be set between 0 and 127.

Bits 6 to 0—16-Bit Timer pulse unit 2 (TPU2) TGR2B Input Capture/Compare Match
Vector Number 6 to 0 (TG2BV6 to TG2BV0): These bits set the vector number for th
timer pulse unit 2 (TPU2) TGR2B input capture/compare match interrupt. There are s
the value can be set between 0 and 127.

Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TC2UV6	TC2UV5	TC2UV4	TC2UV3	TC2UV2	TC2UV
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—16-Bit Timer pulse unit 2 (TPU2) TCNT2 Overflow Interrupt Vector Nu
(TC2VV6 to TC2VV0): These bits set the vector number for the 16-bit timer pulse unit
TCNT2 overflow interrupt. There are seven bits, so the value can be set between 0 and

Bits 6 to 0—16-Bit Timer pulse unit 2 (TPU2) TCNT2 Underflow Interrupt Vector Nu
(TC2UV6 to TC2UV0): These bits set the vector number for the 16-bit timer pulse unit
TCNT2 underflow interrupt. There are seven bits, so the value can be set between 0 and

	—	SER1V6	SER1V5	SER1V4	SER1V3	SER1V2	SER1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	SRX1V6	SRX1V5	SRX1V4	SRX1V3	SRX1V2	SRX1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—Serial Communication Interface with FIFO 1 (SCIF1) Receive-Error Int
Vector Number 6 to 0 (SER1V6 to SER1V0): These bits set the vector number for the
communication interface with FIFO 1 (SCIF1) receive-error interrupt. There are sever
value can be set between 0 and 127.

Bits 6 to 0—Serial Communication Interface with FIFO 1 (SCIF1) Receive-Data-Full
Interrupt Vector Number 6 to 0 (SRX1V6 to SRX1V0): These bits set the vector numb
serial communication interface with FIFO 1 (SCIF1) receive-data-full/data-ready inter
are seven bits, so the value can be set between 0 and 127.

	—	SBR1V6	SBR1V5	SBR1V4	SBR1V3	SBR1V2	SBR1V
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	STX1V6	STX1V5	STX1V4	STX1V3	STX1V2	STX1V
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—Serial Communication Interface with FIFO 1 (SCIF1) Break Interrupt Ve
Number 6 to 0 (SBR1V6 to SBR1V0): These bits set the vector number for the serial
communication interface with FIFO 1 (SCIF1) break interrupt. There are seven bits, so
can be set between 0 and 127.

Bits 6 to 0—Serial Communication Interface with FIFO 1 (SCIF1) Transmit-Data-Emp
Vector Number 6 to 0 (STE1V6 to STE1V0): These bits set the vector number for the s
communication interface with FIFO 1 (SCIF1) transmit-data-empty interrupt. There are
so the value can be set between 0 and 127.

	—	SER2V6	SER2V5	SER2V4	SER2V3	SER2V2	SER2
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	SRX2V6	SRX2V5	SRX2V4	SRX2V3	SRX2V2	SRX2
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—Serial Communication Interface with FIFO 2 (SCIF2) Receive-Error Int
Vector Number 6 to 0 (SER2V6 to SER2V0): These bits set the vector number for the
communication interface with FIFO 2 (SCIF2) receive-error interrupt. There are sever
value can be set between 0 and 127.

Bits 6 to 0—Serial Communication Interface with FIFO 2 (SCIF2) Receive-Data-Full
Interrupt Vector Number 6 to 0 (SRX2V6 to SRX2V0): These bits set the vector numb
serial communication interface with FIFO 2 (SCIF2) receive-data-full/data-ready inter
are seven bits, so the value can be set between 0 and 127.

	—	SBR2V6	SBR2V5	SBR2V4	SBR2V3	SBR2V2	SBR2V
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	STX2V6	STX2V5	STX2V4	STX2V3	STX2V2	STX2V
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—Serial Communication Interface with FIFO 2 (SCIF2) Break Interrupt Ve
Number 6 to 0 (SBR2V6 to SBR2V0): These bits set the vector number for the serial
communication interface with FIFO 2 (SCIF2) break interrupt. There are seven bits, so
can be set between 0 and 127.

Bits 6 to 0—Serial Communication Interface with FIFO 2 (SCIF2) Transmit-Data-Emp
Vector Number 6 to 0 (STE2V6 to STE2V0): These bits set the vector number for the s
communication interface with FIFO 2 (SCIF2) transmit-data-empty interrupt. There are
so the value can be set between 0 and 127.

	—	RER0V6	RER0V5	RER0V4	RER0V3	RER0V2	RER0
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TER0V6	TER0V5	TER0V4	TER0V3	TER0V2	TER0
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—Serial I/O 0 (SIO0) Receive Overrun Error Interrupt Vector Number 6 t
(RER0V6 to RER0V0): These bits set the vector number for the serial I/O 0 (SIO0) re
overrun error interrupt. There are seven bits, so the value can be set between 0 and 12

Bits 6 to 0—Serial I/O 0 (SIO0) Transmit Underrun Error Interrupt Vector Number 6
(TER0V6 to TER0V0): These bits set the vector number for the serial I/O 0 (SIO0) tra
underrun error interrupt. There are seven bits, so the value can be set between 0 and 1

Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TDE0V6	TDE0V5	TDE0V4	TDE0V3	TDE0V2	TDE0V
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—Serial I/O 0 (SIO0) Receive-Data-Full Interrupt Vector Number 6 to 0 (R
RDF0V0): These bits set the vector number for the serial I/O 0 (SIO0) receive-data-full
There are seven bits, so the value can be set between 0 and 127.

Bits 6 to 0—Serial I/O 0 (SIO0) Transmit-Data-Empty Interrupt Vector Number 6 to 0
to TDE0V0): These bits set the vector number for the serial I/O 0 (SIO0) transmit-data-
interrupt. There are seven bits, so the value can be set between 0 and 127.

	—	RER1V6	RER1V5	RER1V4	RER1V3	RER1V2	RER1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TER1V6	TER1V5	TER1V4	TER1V3	TER1V2	TER1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—Serial I/O 1 (SIO1) Receive Overrun Error Interrupt Vector Number 6 t
(RER1V6 to RER1V0): These bits set the vector number for the serial I/O 1 (SIO1) re
overrun error interrupt. There are seven bits, so the value can be set between 0 and 12

Bits 6 to 0—Serial I/O 1 (SIO1) Transmit Underrun Error Interrupt Vector Number 6
(TER1V6 to TER1V0): These bits set the vector number for the serial I/O 1 (SIO1) tra
underrun error interrupt. There are seven bits, so the value can be set between 0 and 1

Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TDE1V6	TDE1V5	TDE1V4	TDE1V3	TDE1V2	TDE1V
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—Serial I/O 1 (SIO1) Receive-Data-Full Interrupt Vector Number 6 to 0 (R
RDF1V0): These bits set the vector number for the serial I/O 1 (SIO1) receive-data-full
There are seven bits, so the value can be set between 0 and 127.

Bits 6 to 0—Serial I/O 1 (SIO1) Transmit-Data-Empty Interrupt Vector Number 6 to 0
to TDE1V0): These bits set the vector number for the serial I/O 1 (SIO1) transmit-data-
interrupt. There are seven bits, so the value can be set between 0 and 127.

	—	RER2V6	RER2V5	RER2V4	RER2V3	RER2V2	RER2
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TER2V6	TER2V5	TER2V4	TER2V3	TER2V2	TER2
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bits 14 to 8—Serial I/O 2 (SIO2) Receive Overrun Error Interrupt Vector Number 6 t
(RER2V6 to RER2V0): These bits set the vector number for the serial I/O 2 (SIO2) re
overrun error interrupt. There are seven bits, so the value can be set between 0 and 12

Bits 6 to 0—Serial I/O 2 (SIO2) Transmit Underrun Error Interrupt Vector Number 6
(TER2V6 to TER2V0): These bits set the vector number for the serial I/O 2 (SIO2) tra
underrun error interrupt. There are seven bits, so the value can be set between 0 and 1

Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	TDE2V6	TDE2V5	TDE2V4	TDE2V3	TDE2V2	TDE2V
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 7—Reserved. These bits are always read as 0. The write value should alwa

Bits 14 to 8—Serial I/O 2 (SIO2) Receive-Data-Full Interrupt Vector Number 6 to 0 (R
RDF2V0): These bits set the vector number for the serial I/O 2 (SIO2) receive-data-full
There are seven bits, so the value can be set between 0 and 127.

Bits 6 to 0—Serial I/O 2 (SIO2) Transmit-Data-Empty Interrupt Vector Number 6 to 0
to TDE2V0): These bits set the vector number for the serial I/O 2 (SIO2) transmit-data-
interrupt. There are seven bits, so the value can be set between 0 and 127.

Tables 5.6 and 5.7 show the relationship between on-chip peripheral module interrupts
interrupt vector number setting registers.

Vector number setting register C	Input capture interrupt (FRT)	Output compare i (FRT)
Vector number setting register D	Overflow interrupt (FRT)	Reserved
Vector number setting register E	Input capture/compare match interrupt (TPU0/TGR0A)	Input capture/com interrupt (TPU0/T
Vector number setting register F	Input capture/compare match interrupt (TPU0/TGR0C)	Input capture/com interrupt (TPU0/T
Vector number setting register G	Overflow interrupt (TPU0/TCNT0)	Reserved
Vector number setting register H	Input capture/compare match interrupt (TPU1/TGR1A)	Input capture/com interrupt (TPU1/T
Vector number setting register I	Overflow interrupt (TPU1/TCNT1)	Underflow interrup (TPU1/TCNT1)
Vector number setting register J	Input capture/compare match interrupt (TPU2/TGR2A)	Input capture/com interrupt (TPU2/T
Vector number setting register K	Overflow interrupt (TPU2/TCNT2)	Underflow interrup (TPU2/TCNT2)
Vector number setting register L	Receive-error interrupt (SCIF1)	Receive-data-full/ interrupt (SCIF1)
Vector number setting register M	Break interrupt (SCIF1)	Transmit-data-em (SCIF1)
Vector number setting register N	Receive-error interrupt (SCIF2)	Receive-data-full/ interrupt (SCIF2)
Vector number setting register O	Break interrupt (SCIF2)	Transmit-data-em (SCIF2)
Vector number setting register P	Receive overrun error interrupt (SIO0)	Transmit underrun interrupt (SIO0)
Vector number setting register Q	Receive-data-full interrupt (SIO0)	Transmit-data-em (SIO0)

Vector number setting register 3	Receive data full interrupt (SIO2)	Transmit data empty (SIO2)

As table 5.6 shows, two on-chip peripheral module interrupts are assigned to each regis
vector numbers by setting the corresponding 7-bit groups (bits 14 to 8 and bits 6 to 0) v
in the range of H'00 (0000000) to H'7F (1111111). H'00 is vector number 0 (the lowest
vector number 127 (the highest). The vector table address is calculated by the following

Vector table address = VBR + (vector number × 4)

A reset initializes a vector number setting register to H'0000. They are not initialized in
mode.

Table 5.7 Interrupt Request Sources and Vector Number Setting Registers (2)

Register	Setting Function
Vector number setting register DMA0 (VCRDMA0)	Channel 0 transfer end interrupt for D
Vector number setting register DMA1 (VCRDMA1)	Channel 1 transfer end interrupt for D

As shown in table 5.7 the vector numbers for direct memory access controller transfer-
interrupts are set in VCRDMA0 and VCRDMA1. See sections 11, Direct Memory Acc
Controller (DMAC), for more details.

Bit:	15	14	13	12	11	10	9
	NMIL	—	—	—	—	—	—
Initial value:	0/1*	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	EXIM
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W

Note: * When NMI input is high: 1; when NMI input is low: 0

Bit 15—NMI Input Level (NMIL): Sets the level of the signal input at the NMI pin. T
be read to determine the NMI pin level. This bit cannot be modified.

Bit 15: NMIL	Description
0	NMI input level is low
1	NMI input level is high

Bits 14 to 9—Reserved: These bits are always read as 0. The write value should alway

Bit 8—NMI Edge Select (NMIE): Selects whether the falling or rising edge of the inte
request signal to the NMI pin is detected.

Bit 8: NMIE	Description
0	Interrupt request is detected on falling edge of NMI input (
1	Interrupt request is detected on rising edge of NMI input

Bits 7 to 2—Reserved: These bits are always read as 0. The write value should always

Bit 0—Interrupt Vector Mode Select (VECMD): This bit selects auto-vector mode or e
vector mode for IRL/IRQ interrupt vector number setting. In auto-vector mode, an inter
determined vector number is set. The IRL15 and IRL14 interrupt vector numbers are se
the IRL1 vector number is set to 64. In external vector mode, a value between 0 and 12
input as the vector number from the external vector number input pins (D7 to D0).

Bit 0: VECMD	Description
0	Auto vector mode, vector number automatically set internall (Ir
1	External vector mode, vector number set by external input

5.3.29 IRQ Control/Status Register (IRQCSR)

The IRQ control/status register (IRQCSR) is a 16-bit register that sets the $\overline{\text{IRL0}}$ to $\overline{\text{IRL3}}$ signal detection mode, indicates the input signal levels at pins $\overline{\text{IRL0}}$ to $\overline{\text{IRL3}}$, and also i IRQ interrupt status. IRQCSR is initialized by a reset. It is not initialized in standby mo

Bit:	15	14	13	12	11	10	9
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	IRL3PS	IRL2PS	IRL1PS	IRL0PS	IRQ3F	IRQ2F	IRQ1F
Initial value:	0/1	0/1	0/1	0/1	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, to clear the flag (in case of edge detection).

Note: n = 0 to 3

Bits 7 to 4—IRL Pin Status Bits (IRL3PS to IRL0PS): These bits indicate the $\overline{\text{IRL3}}$ to
status. The $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pin levels can be ascertained by reading these bits. These bit
modified.

Bits 7 to 4: IRLnPS	Description
0	Low level is being input to pin $\overline{\text{IRLn}}$
1	High level is being input to pin $\overline{\text{IRLn}}$

Note: n = 0 to 3

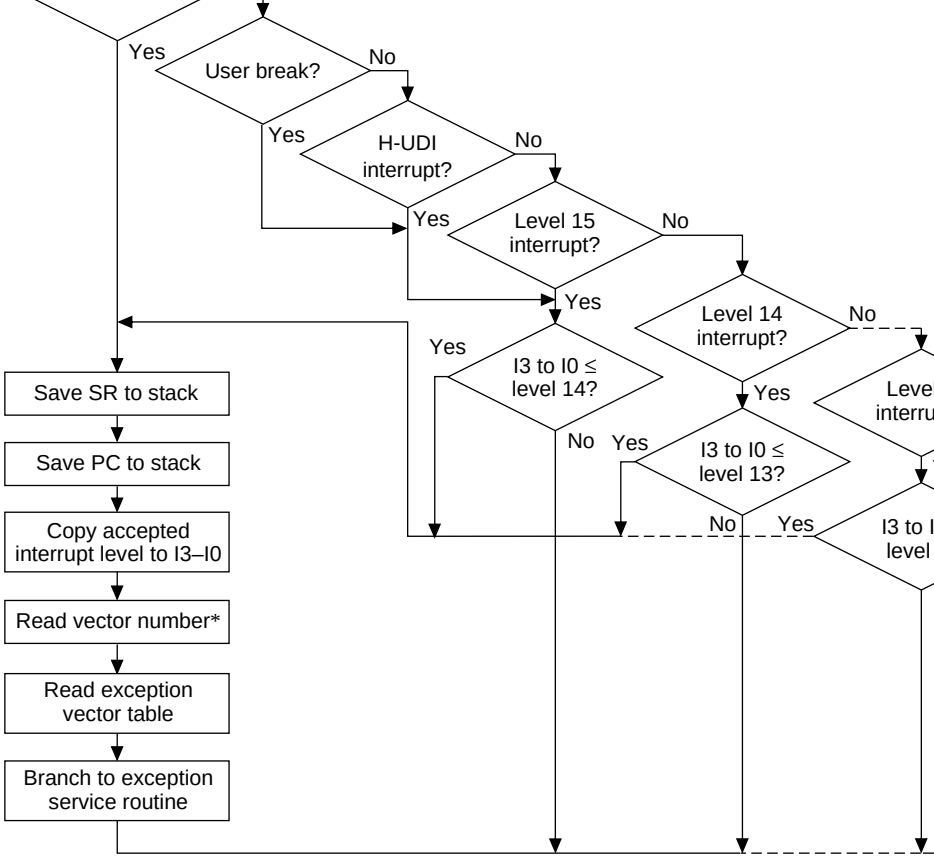
Bits 3 to 0—IRQ3 to IRQ0 Flags (IRQ3F to IRQ0F): These bits indicate the IRQ3 to
interrupt request status.

		• When 0 is written to IRQnF after reading • When an IRQn interrupt is accepted
1	Level detection	There is an IRQn interrupt request [Setting condition] When $\overline{\text{IRLn}}$ input is low
	Edge detection	An IRQn interrupt request has been detected [Setting condition] When an $\overline{\text{IRLn}}$ input edge is detected

Note: n = 0 to 3

according to the priority levels set in interrupt priority level setting registers A to F
IPRE). Lower-priority interrupts are held pending. If two or more of these interrup
same priority level or if multiple interrupts occur within a single module, the inter
highest default priority or the highest priority within its IPR setting unit (as indicat
5.4) is selected.

3. The interrupt controller compares the priority level of the selected interrupt reques
interrupt mask bits (I3 to I0) in the CPU's status register (SR). If the request priori
equal to or less than the level set in I3 to I0, the request is held pending. If the requ
level is higher than the level in bits I3 to I0, the interrupt controller accepts the inte
sends an interrupt request signal to the CPU.
4. The CPU detects the interrupt request sent from the interrupt controller when it de
next instruction to be executed. Instead of executing the decoded instruction, the C
interrupt exception handling.
5. Status register (SR) and program counter (PC) are saved onto the stack.
6. The priority level of the accepted interrupt is copied to the interrupt mask level bits
the status register (SR).
7. When external vector mode is specified for the IRL/IRQ interrupt, the vector numb
from the external vector number input pins (D7 to D0).
8. The CPU reads the start address of the exception service routine from the exceptio
table entry for the accepted interrupt, jumps to that address, and starts executing th
there. This jump is not a delayed branch.



I3–I0: Status register interrupt mask bits.

Note: * The vector number is only read from an external source when an external vect
specified for the IRL/IRQ interrupt vector number.

Figure 5.8 Interrupt Sequence Flowchart

PC: Start address of return destination instruction (instruction after executing instructi

Figure 5.9 Stack State after Interrupt Exception Handling

5.5 Interrupt Response Time

Table 5.8 shows the interrupt response time, which is the time from the occurrence of
request until interrupt exception handling starts and fetching of the first instruction of
service routine begins.

executed by CPU						interrup error ex handlin Icyc + m m3 + m interrup instructi howeve may be during r instructi executio
Time from interrupt exception handling (SR and PC saves and vector address fetch) until fetch of first instruction of exception service routine starts		5.0 × Icyc + m1 + m2 + m3	5.0 × Icyc + m1 + m2 + m3	5.0 × Icyc + m1 + m2 + m3	5.0 × Icyc + m1 + m2 + m3	
Response time	Total:	X + 7.0 × Icyc + m1 + m2 + m3	X + 5.5 × Icyc + 1.0 × Ecyc + 1.5 × Pcyc + m1 + m2 + m3	X + 5.5 × Icyc + 1.0 × Pcyc + m1 + m2 + m3	X + 5.0 × Icyc + 1.0 × Pcyc + m1 + m2 + m3	
	Minimum:	10	11	9.5	9	Iϕ:Eϕ:Pϕ
	Maximum:	11 + 2 (m1 + m2 + m3) + m4	19.5 + 2 (m1 + m2 + m3) + m4	13.5 + 2 (m1 + m2 + m3) + m4	13.0 + 2 (m1 + m2 + m3) + m4	Iϕ:Eϕ:Pϕ

Note: m1 to m4 are the number of states needed for the following memory accesses
m1: SR save (longword write)
m2: PC save (longword write)
m3: Vector address read (longword read)
m4: Fetch of first instruction of interrupt service routine
Icyc: Iϕ cycle time
Ecyc: Eϕ cycle time
Pcyc: Pϕ cycle time
Peripheral modules A: DMAC, REF (BSC)
Peripheral modules B: WDT, FRT, TPU, SCIF, SIO, E-DMAC

When an external vector is fetched, the interrupt source can also be cleared when the e
vector fetch cycle is detected.

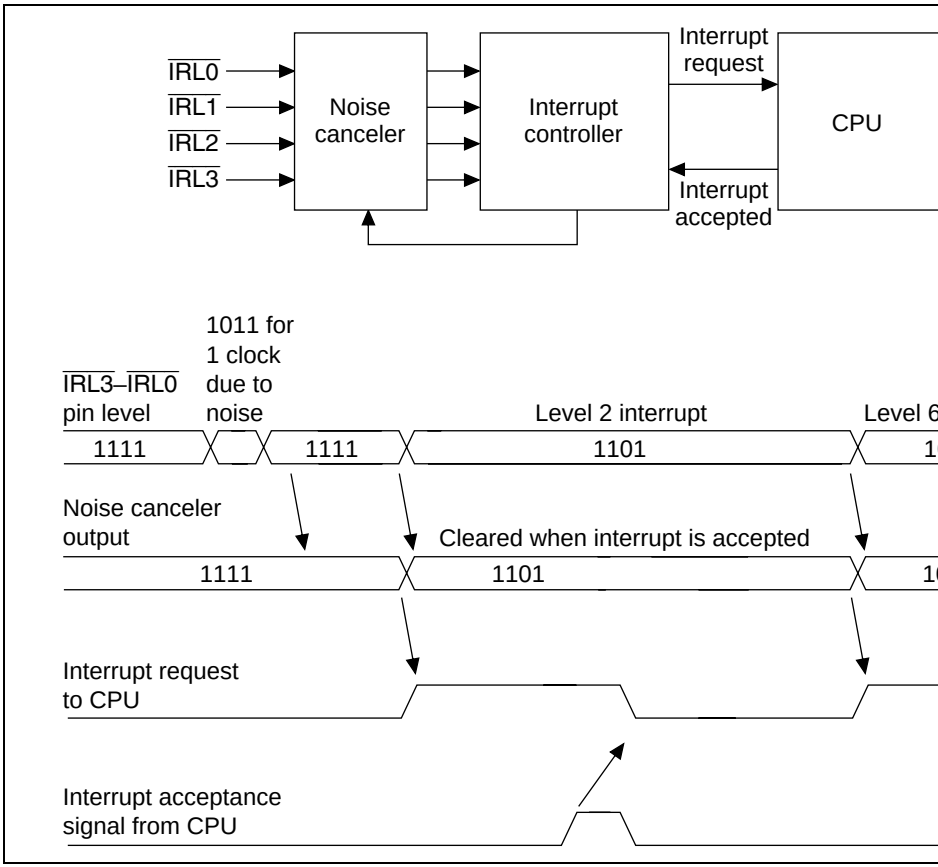


Figure 5.10 $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ Pin Sampling

If interrupt source clearing is performed by writing to an IO address (external), t
instruction will be executed before completion of the write operation because of
buffer. To ensure that the write operation is completed before the next instructio
executed, synchronization is achieved when a read is performed from the same a
following the write.

a. When returning from interrupt handling by means of RTE instruction

   When the RTE instruction is used to return from interrupt handling, as show
   5.11, consider the cycles to be inserted between the read instruction for sync
   and the RTE instruction, according to the set clock ratio (Iϕ:Eϕ:Pϕ) and exte
   cycle.

   IRL3—IRL0 should be negated at least 0.5 Icyc + 1.0 Ecyc + 1.5 Pcyc befor
   interrupt acceptance becomes possible.

   For example, if clock ratio Iϕ:Eϕ:Pϕ is 4:2:2, at least 5.5 Icyc should be inse

b. When changing level during interrupt handling

   When the SR value is changed by means of an LDC instruction and multiple
   implementation of other interrupts is enabled, also, consider the cycles to be
   between the synchronization instruction and the LDC instruction as shown i
   5.12, according to the set clock ratio (Iϕ:Eϕ:Pϕ) and external bus cycle.

   IRL3 to IRL0 should be negated at least 0.5 Icyc + 1.0 Ecyc + 1.5 Pcyc befo
   interrupt acceptance becomes possible.

   For example, if clock ratio Iϕ:Eϕ:Pϕ is 4:2:2, at least 5.5 Icyc should be inse

F: Instruction fetch ..... Instruction is fetched from memory in which program i
D: Instruction decode ..... Fetched instruction is decoded
E: Instruction execution ..... Data operation or address operation is performed in a
with result of decoding
M: Memory access ..... Memory data access is performed
W: Write-back ..... Data read from memory is written to register

Figure 5.11 Pipeline Operation when Returning by Means of RTE Instru

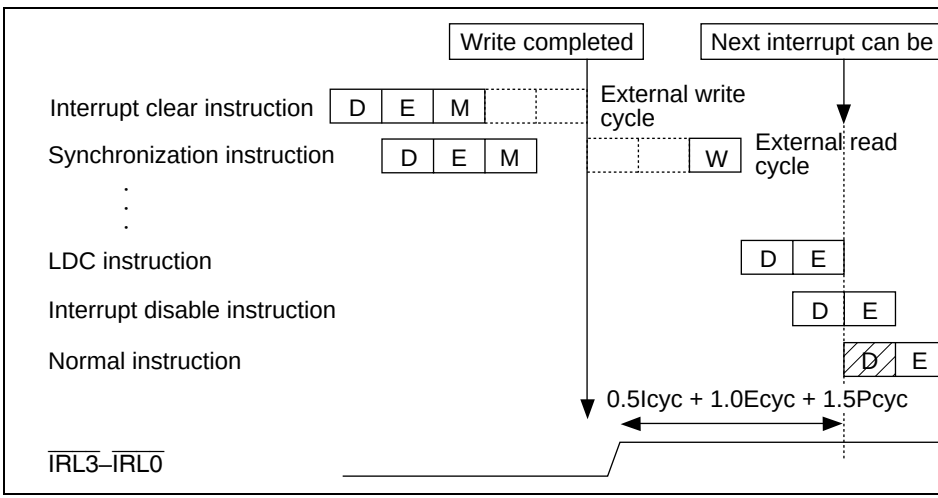


Figure 5.12 Pipeline Operation when Interrupts are Enabled by Means of SR M

When an interrupt source is cleared by the program, pipeline operation must be
to ensure that the same interrupt is not implemented again.

5.13, consider the cycles to be inserted between the read instruction for sync
and the RTE instruction, according to the set clock ratio (Iϕ:Eϕ:Pϕ).

The on-chip peripheral interrupt signal should be negated at least 0.5 Icyc +
before next interrupt acceptance becomes possible.

For example, if clock ratio Iϕ:Eϕ:Pϕ is 4:2:2, at least 2.5 Icyc should be inse

b. When changing level during interrupt handling

When the SR value is changed by means of an LDC instruction and multiple
implementation of other interrupts is enabled, consider the cycles to be inser
between the synchronization instruction and the LDC instruction as shown i
5.14, according to the set clock ratio (Iϕ:Eϕ:Pϕ).

The on-chip peripheral interrupt signal should be negated at least 0.5 Icyc +
before next interrupt acceptance becomes possible.

For example, if clock ratio Iϕ:Eϕ:Pϕ is 4:2:2, at least 2.5 Icyc should be inse

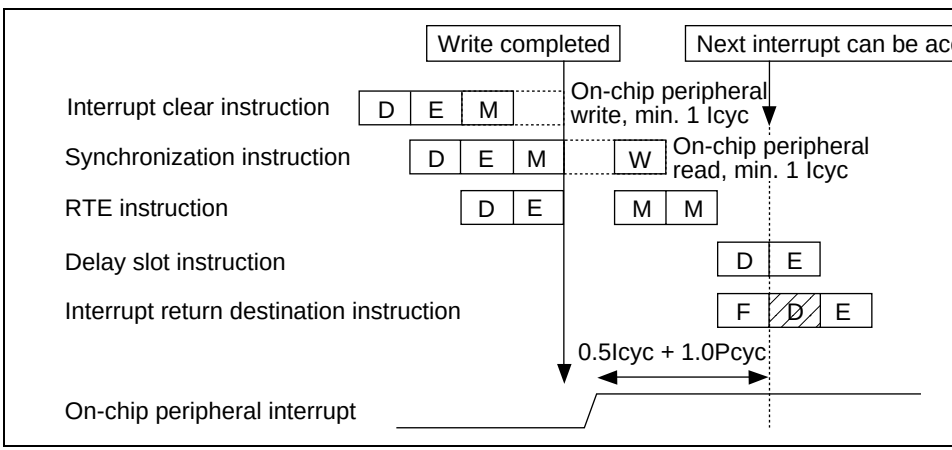


Figure 5.13 Pipeline Operation when Returning by Means of RTE Instruc

Figure 5.14 Pipeline Operation when Interrupts are Enabled by Means of SR M

In the above figure, the stage in which the instruction fetch occurs cannot be sp
because of the mix of DSP instructions in this chip, so instruction fetch F is om
cases during pipeline operation.

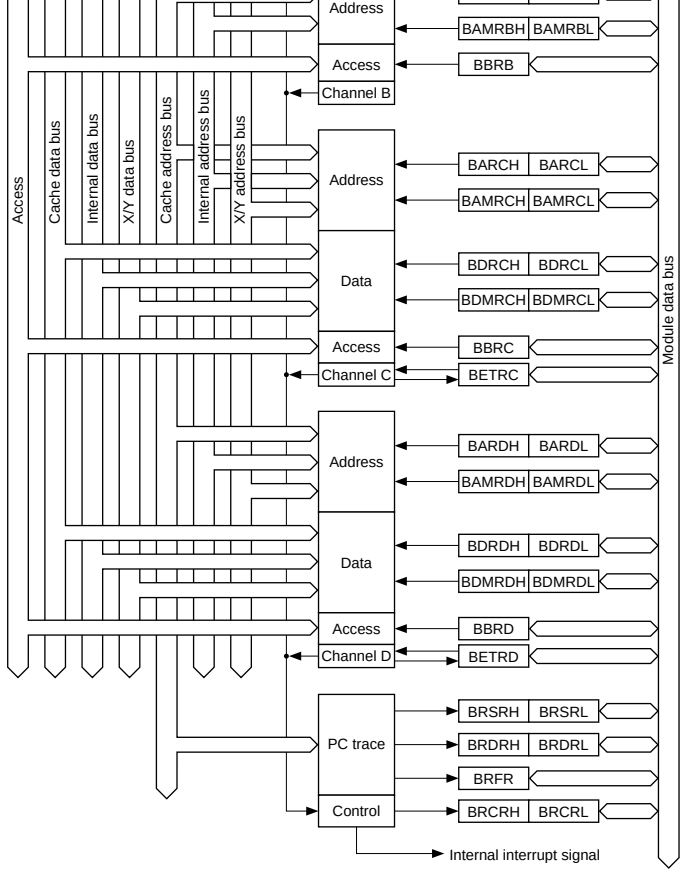
This function makes it easy to design a sophisticated self-monitoring debugger, enabli
to be debugged with the chip alone, without using an in-circuit emulator.

6.1.1 Features

The UBC has the following features:

- The following can be set as break conditions:
  - — Number of break channels: Four (channels A, B, C, and D)
    User break interrupts can be generated on independent or sequential conditions
    channels A, B, C, and D.
  - — Sequential break settings
    - Channel A → channel B → channel C → channel D
    - Channel B → channel C → channel D
    - Channel C → channel D

  1. Address: 32-bit masking capability, individual address setting possible (cache b
     internal bus (DMAC, E-DMAC), X/Y bus)
  2. Data (channels C and D only,): 32-bit masking capability, individual address se
     possible (cache bus (CPU), internal bus (DMAC, E-DMAC), X/Y bus)
  3. Bus master: CPU cycle/on-chip DMAC (DMAC, E-DMAC) cycle
  4. Bus cycle: Instruction fetch/data access
  5. Read/write
  6. Operand cycle: Byte/word/longword
- User break interrupt generation on occurrence of break condition
  A user-written user break interrupt exception routine can be executed.
- Processing can be stopped before or after instruction execution in an instruction fe
- Break with specification of number of executions (channels C and D only)
  Settable number of executions: maximum $2^{12} - 1$ (4095)
- PC trace function
  The branch source/branch destination can be traced when a branch instruction is fe
  (maximum 8 addresses (4 pairs)).



BARAH/L:	Break address register AH/L	BETRC:	Break execution times register C
BAMRAH/L:	Break address mask register AH/L	BARDH/L:	Break address register DH/L
BBRA:	Break bus cycle register A	BAMRDH/L:	Break address mask register DH/L
BARBH/L:	Break address register BH/L	BDRDH/L:	Break data register DH/L
BAMRBH/L:	Break address mask register BH/L	BDMRDH/L:	Break data mask register DH/L
BBRB:	Break bus cycle register B	BBRD:	Break bus cycle register D
BARCH/L:	Break address register CH/L	BETRD:	Break execution times register D
BAMRCH/L:	Break address mask register CH/L	BRCRH/L:	Break control register H/L
BDRCH/L:	Break data register CH/L	BRFR:	Branch flag register
BDMRCH/L:	Break data mask register CH/L	BRSRH/L:	Branch source register H/L
BBRC:	Break bus cycle register C	BRDRH/L:	Branch destination register H/L

Figure 6.1 Block Diagram of User Break Controller

Break address mask register AL	BAMRAL	R/W	H'0000	H'FFFFFF06	16
Break bus cycle register A	BBRA	R/W	H'0000	H'FFFFFF08	16
Break address register BH	BARBH	R/W	H'0000	H'FFFFFF20	16
Break address register BL	BARBL	R/W	H'0000	H'FFFFFF22	16
Break address mask register BH	BAMRBH	R/W	H'0000	H'FFFFFF24	16
Break address mask register BL	BAMRBL	R/W	H'0000	H'FFFFFF26	16
Break bus cycle register B	BBRB	R/W	H'0000	H'FFFFFF28	16
Break address register CH	BARCH	R/W	H'0000	H'FFFFFF40	16
Break address register CL	BARCL	R/W	H'0000	H'FFFFFF42	16
Break address mask register CH	BAMRCH	R/W	H'0000	H'FFFFFF44	16
Break address mask register CL	BAMRCL	R/W	H'0000	H'FFFFFF46	16
Break data register CH	BDRCH	R/W	H'0000	H'FFFFFF50	16
Break data register CL	BDRCL	R/W	H'0000	H'FFFFFF52	16
Break data mask register CH	BDMRCH	R/W	H'0000	H'FFFFFF54	16
Break data mask register CL	BDMRCL	R/W	H'0000	H'FFFFFF56	16
Break bus cycle register C	BBRC	R/W	H'0000	H'FFFFFF48	16
Break execution times register C	BETRC	R/W	H'0000	H'FFFFFF58	16
Break address register DH	BARDH	R/W	H'0000	H'FFFFFF60	16
Break address register DL	BARDL	R/W	H'0000	H'FFFFFF62	16
Break address mask register DH	BAMRDH	R/W	H'0000	H'FFFFFF64	16
Break address mask register DL	BAMRDL	R/W	H'0000	H'FFFFFF66	16
Break data register DH	BDRDH	R/W	H'0000	H'FFFFFF70	16
Break data register DL	BDRDL	R/W	H'0000	H'FFFFFF72	16
Break data mask register DH	BDMRDH	R/W	H'0000	H'FFFFFF74	16
Break data mask register DL	BDMRDL	R/W	H'0000	H'FFFFFF76	16

Branch source register L	BRSRL	R	Undefined	H'FFFFFF16	16
Branch destination register H	BRDRH	R	Undefined	H'FFFFFF18	16
Branch destination register L	BRDRL	R	Undefined	H'FFFFFF1A	16

Notes: 1. Initialized by a power-on reset. Value is retained in standby mode, and is un
after a manual reset.
2. Byte access cannot be used.
3. Bits SVF and DVF in BRFR are initialized by a power-on reset; the other bits
are not initialized.

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BARAL

Bit:	15	14	13	12	11	10	9
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break address register A (BARA) consists of two 16-bit readable/writable registers: b
register AH (BARAH) and break address register AL (BARAL). BARAH specifies th
(bits 31 to 16) of the address used as a channel A break condition, and BARAL specif
half (bits 15 to 0). BARAH and BARAL are initialized to H'0000 by a power-on reset
manual reset, their values are undefined.

BARAH Bits 15 to 0—Break Address A31 to A16 (BAA31 to BAA16): These bits st
half (bits 31 to 16) of the address used as a channel A break condition.

BARAL Bits 15 to 0—Break Address A15 to A0 (BAA15 to BAA0): These bits store
half (bits 15 to 0) of the address used as a channel A break condition.

Bit:	7	6	5	4	3	2	1
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BAMRAL

Bit:	15	14	13	12	11	10	9
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break address mask register A (BAMRA) consists of two 16-bit readable/writable regis
address mask register AH (BAMRAH) and break address mask register AL (BAMRAL
BAMRAH specifies which bits of the break address set in BARAH are to be masked, a
BAMRAL specifies which bits of the break address set in BARAL are to be masked. B
and BAMRAL are initialized to H'0000 by a power-on reset; after a manual reset, their
undefined.

BAMRAH Bits 15 to 0—Break Address Mask A31 to A16 (BAMA31 to BAMA16): T
specify whether or not corresponding channel A break address bits 31 to 16 (BAA31 to
set in BARAH are to be masked.

BAMRAL Bits 15 to 0—Break Address Mask A15 to A0 (BAMA15 to BAMA0): The
specify whether or not corresponding channel A break address bits 15 to 0 (BAA15 to
in BARAL are to be masked.

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	CPA1	CPA0	IDA1	IDA0	RWA1	RWA0	SZA1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break bus cycle register A (BBRA) is a 16-bit readable/writable register that sets four
break conditions: (1) CPU cycle/on-chip DMAC (DMAC, E-DMAC) cycle, (2) instru
fetch/data access, (3) read/write, and (4) operand size. BBRA is initialized to H'0000 b
on reset; after a manual reset, its value is undefined.

Bits 15 to 8—Reserved: These bits are always read as 0. The write value should alway

Bits 7 and 6—CPU/DMAC, E-DMAC Cycle Select A (CPA1, CPA0): These bits spec
a CPU cycle, or a DMAC or E-DMAC cycle, is to be selected as the bus cycle used as
break condition.

Bit 7: CPA1	**Bit 6: CPA0**	**Description**
0	0	Channel A user break interrupt is not generated (
	1	CPU cycle is selected as user break condition
1	0	DMAC or E-DMAC cycle is selected as user break condition
	1	CPU, DMAC, or E-DMAC cycle is selected as user break condit

	1	Instruction fetch cycle or data access cycle is selected as break c

Bits 3 and 2—Read/Write Select A (RWA1, RWA0): These bits specify whether a read write cycle is to be selected as the bus cycle used as a channel A break condition.

Bit 3: RWA1	Bit 2: RWA0	Description
0	0	Channel A user break interrupt is not generated (In
	1	Read cycle is selected as break condition
1	0	Write cycle is selected as break condition
	1	Read cycle or write cycle is selected as break condition

Bits 1 and 0—Operand Size Select A (SZA1, SZA0): These bits select the operand size cycle used as a channel A break condition.

Bit 1: SZA1	Bit 0: SZA0	Description
0	0	Operand size is not included in break conditions (In
	1	Byte access is selected as break condition
1	0	Word access is selected as break condition
	1	Longword access is selected as break condition

Notes: When a break is to be executed on an instruction fetch, clear the SZA0 bit to 0. A instructions are regarded as being accessed using word size (instruction fetches performed as longword).

In the case of an instruction, the operand size is word; in the case of a CPU/DMA DMAC data access, it is determined by the specified operand size. Note that the size is not determined by the bus width of the space accessed.

Bit:	7	6	5	4	3	2	1
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BARBL

Bit:	15	14	13	12	11	10	9
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break address register B (BARB) consists of two 16-bit readable/writable registers: br
register BH (BARBH) and break address register BL (BARBL). BARBH specifies the
(bits 31 to 16) of the address used as a channel B break condition, and BARBL specifi
half (bits 15 to 0). BARBH and BARBL are initialized to H'0000 by a power-on reset;
manual reset, their values are undefined.

BARBH Bits 15 to 0—Break Address B31 to B16 (BAB31 to BAB16): These bits sto
half (bits 31 to 16) of the address used as a channel B break condition.

BARBL Bits 15 to 0—Break Address B15 to B0 (BAB15 to BAB0): These bits store
half (bits 15 to 0) of the address used as a channel B break condition.

Bit:	7	6	5	4	3	2	1
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BAMRBL

Bit:	15	14	13	12	11	10	9
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break address mask register B (BAMRB) consists of two 16-bit readable/writable regis address mask register BH (BAMRBH) and break address mask register BL (BAMRBL BAMRBH specifies which bits of the break address set in BARBH are to be masked, a BAMRBL specifies which bits of the break address set in BARBL are to be masked. B and BAMRBL are initialized to H'0000 by a power-on reset; after a manual reset, their undefined.

BAMRBH Bits 15 to 0—Break Address Mask B31 to B16 (BAMB31 to BAMB16): T specify whether or not corresponding channel B break address bits 31 to 16 (BAB31 to set in BARBH are to be masked.

BAMRBL Bits 15 to 0—Break Address Mask B15 to B0 (BAMB15 to BAMB0): Thes specify whether or not corresponding channel B break address bits 15 to 0 (BAB15 to in BARBL are to be masked.

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	CPB1	CPB0	IDB1	IDB0	RWB1	RWB0	SZB1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break bus cycle register B (BBRB) is a 16-bit readable/writable register that sets four
break conditions: (1) CPU cycle/on-chip DMAC (DMAC, E-DMAC) cycle, (2) instru
fetch/data access, (3) read/write, and (4) operand size. BBRB is initialized to H'0000 b
on reset; after a manual reset, its value is undefined.

Bits 15 to 8—Reserved: These bits are always read as 0. The write value should alway

Bits 7 and 6—CPU/DMAC, E-DMAC Cycle Select B (CPB1, CPB0): These bits spec
a CPU cycle, or a DMAC or E-DMAC cycle, is to be selected as the bus cycle used as
break condition.

Bit 7: CPB1	**Bit 6: CPB0**	**Description**
0	0	Channel B user break interrupt is not generated (
	1	CPU cycle is selected as user break condition
1	0	DMAC or E-DMAC cycle is selected as user break condition
	1	CPU, DMAC, or E-DMAC cycle is selected as user break condit

	1	Instruction fetch cycle or data access cycle is selected as break c

Bits 3 and 2—Read/Write Select B (RWB1, RWB0): These bits specify whether a read
write cycle is to be selected as the bus cycle used as a channel B break condition.

Bit 3: RWB1	Bit 2: RWB0	Description
0	0	Channel B user break interrupt is not generated (In
	1	Read cycle is selected as break condition
1	0	Write cycle is selected as break condition
	1	Read cycle or write cycle is selected as break condition

Bits 1 and 0—Operand Size Select B (SZB1, SZB0): These bits select the operand size
cycle used as a channel B break condition.

Bit 1: SZB1	Bit 0: SZB0	Description
0	0	Operand size is not included in break conditions (In
	1	Byte access is selected as break condition
1	0	Word access is selected as break condition
	1	Longword access is selected as break condition

Notes: When a break is to be executed on an instruction fetch, clear the SZB0 bit to 0. A
instructions are regarded as being accessed using word size (instruction fetches
performed as longword).

In the case of an instruction, the operand size is word; in the case of a CPU/DMA
DMAC data access, it is determined by the specified operand size. Note that the
size is not determined by the bus width of the space accessed.

Bit:	7	6	5	4	3	2	1
	BAC23	BAC22	BAC21	BAC20	BAC19	BAC18	BAC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BARCL

Bit:	15	14	13	12	11	10	9
	BAC15	BAC14	BAC13	BAC12	BAC11	BAC10	BAC9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAC7	BAC6	BAC5	BAC4	BAC3	BAC2	BAC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break address register C (BARC) consists of two 16-bit readable/writable registers: br register CH (BARCH) and break address register CL (BARCL). BARCH specifies the (bits 31 to 16) of the address used as a channel C break condition, and BARCL specifi half (bits 15 to 0). The address bus connected to the X/Y memory can also be specifie condition by making a setting in the XYEC bit/XYSC bit in break bus cycle register C When XYEC = 0, BAC31 to BAC0 specify the address. When XYEC = 1, the upper 1 (BAC31 to BAC16) of BARC specify the X address bus, and the lower 16 bits (BAC1 specify the Y address bus. BARCH and BARCL are initialized to H'0000 by a power- after a manual reset, their values are undefined.

Note: * As an X/Y bus access is always a word access, the values of XAB0 and YAB
included in the break condition.

6.2.8 Break Address Mask Register C (BAMRC)

BAMRCH

Bit:	15	14	13	12	11	10	9
	BAMC31	BAMC30	BAMC29	BAMC28	BAMC27	BAMC26	BAMC2
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAMC23	BAMC22	BAMC21	BAMC20	BAMC19	BAMC18	BAMC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BAMRCL

Bit:	15	14	13	12	11	10	9
	BAMC15	BAMC14	BAMC13	BAMC12	BAMC11	BAMC10	BAMC9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAMC7	BAMC6	BAMC5	BAMC4	BAMC3	BAMC2	BAMC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

XYEC = 0	Address	Upper 16 bits maskable	Lower 16 bits ma
XYEC = 1	X address (when XYSC = 0)	Maskable	—
	Y address (when XYSC = 1	—	Maskable

Bit 31 to 0:

BAMCn	**Description**
0	Channel C break address bit BACn is included in break condition (
1	Channel C break address bit BACn is masked, and not included in

Note: n = 31 to 0

Bit:	7	6	5	4	3	2	1
	BDC23	BDC22	BDC21	BDC20	BDC19	BDC18	BDC17
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BDRCL

Bit:	15	14	13	12	11	10	9
	BDC15	BDC14	BDC13	BDC12	BDC11	BDC10	BDC9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BDC7	BDC6	BDC5	BDC4	BDC3	BDC2	BDC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break data register C (BDRC) consists of two 16-bit readable/writable registers: break
register CH (BDRCH) and break data register CL (BDRCL). BDRCH specifies the upp
(bits 31 to 16) of the data used as a channel C break condition, and BDRCL specifies th
half (bits 15 to 0). The data bus connected to the X/Y memory can also be specified as
condition by making a setting in the XYEC bit/XYSC bit in break bus cycle register C
When XYEC = 1, the upper 16 bits (BDC31 to BDC16) of BDRC specify the X data bu
lower 16 bits (BDC15 to BDC0) specify the Y data bus.

6.2.10 Break Data Mask Register C (BDMRC)

BDMRCH

Bit:	15	14	13	12	11	10	9
	BDMC31	BDMC30	BDMC29	BDMC28	BDMC27	BDMC26	BDMC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BDMC23	BDMC22	BDMC21	BDMC20	BDMC19	BDMC18	BDMC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BDMRCL

Bit:	15	14	13	12	11	10	9
	BDMC15	BDMC14	BDMC13	BDMC12	BDMC11	BDMC10	BDMC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BDMC7	BDMC6	BDMC5	BDMC4	BDMC3	BDMC2	BDMC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break data mask register C (BDMRC) consists of two 16-bit readable/writable registe
data mask register CH (BDMRCH) and break data mask register CL (BDMRCL). BD
specifies which bits of the break data set in BDRCH are to be masked, and BDMRCL
which bits of the break data set in BDRCL are to be masked. Operation also depends o
XYEC and XYSC in BBRC as shown below. BDMRCH and BDMRCL are initialized
by a power-on reset; after a manual reset, their values are undefined.

Bits 31 to 0:

BDMCn	Description
0	Channel C break data bit BDCn is included in break condition (In
1	Channel C break data bit BDCn is masked, and not included in cond

Notes: 1. n = 31 to 0

2. When including the data bus value in the break condition, specify the operan

3. When specifying byte size, and using odd-address data as a break condition value in bits 7 to 0 of BDRC and BDMRC. When using even-address data as condition, set the value in bits 15 to 8. The unused 8 bits of these registers h effect on the break condition.

	CPC1	CPC0	IDC1	IDC0	RWC1	RWC0	SZC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break bus cycle register C (BBRC) is a 16-bit readable/writable register that sets five
break conditions: (1) internal bus (C-bus, I-bus)/X memory bus/Y memory bus), (2) C
cycle/on-chip DMAC (DMAC, E-DMAC) cycle, (3) instruction fetch/data access, (4)
and (5) operand size. BBRC is initialized to H'0000 by a power-on reset; after a manu
value is undefined.

Bits 15 to 10—Reserved: These bits are always read as 0. The write value should alwa

Bit 9—X/Y Memory Bus Enable C (XYEC): Selects whether the X/Y bus is used as a
break condition.

Bit 9: XYEC	Description
0	Cache bus or internal bus is selected as condition for channel C ad (
1	X/Y bus is selected as condition for channel C address/data

Bit 8—X Bus/Y Bus Select C (XYSC): Selects whether the X bus or the Y bus is used
channel C break condition. This bit is valid only when bit XYEC = 1.

Bit 8: XYSC	Description
0	X bus is selected as channel C break condition (
1	Y bus is selected as channel C break condition

The configuration of bits 7 to 0 is the same as for BBRA.

	ETRC7	ETRC6	ETRC5	ETRC4	ETRC3	ETRC2	ETRC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a channel C execution-times break condition is enabled (by setting the ETBEC b
BRCR), this 16-bit register specifies the number of times a channel C break condition c
before a user break interrupt is requested. The maximum value is $2^{12} – 1$ times. Each tim
channel C break condition occurs, the value in BETRC is decremented by 1. After the I
value reaches H'0001, an interrupt is requested when a break condition next occurs.

As exceptions and interrupts cannot be accepted for instructions in a repeat loop compr
more than three instructions, BETRC is not decremented by the occurrence of a break c
for an instruction in such a repeat loop (see 4.6, When Exception Sources Are Not Acc

Bits 15 to 12 are always read as 0, and should only be written with 0.

BETRC is initialized to H'0000 by a power-on reset.

Bit	7	6	5	4	3	2	1
	BAD23	BAD22	BAD21	BAD20	BAD19	BAD18	BAD1
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BARDL

Bit	15	14	13	12	11	10	9
	BAD15	BAD14	BAD13	BAD12	BAD11	BAD10	BAD9
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
	BAD7	BAD6	BAD5	BAD4	BAD3	BAD2	BAD1
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break address register D (BARD) consists of two 16-bit readable/writable registers: b
register DH (BARDH) and break address register DL (BARDL). BARDH specifies th
(bits 31 to 16) of the address used as a channel D break condition, and BARDL specif
half (bits 15 to 0). The address bus connected to the X/Y memory can also be specifie
condition by making a setting in the XYED bit/XYSD bit in break bus cycle register D
When XYED = 0, BAD31 to BAD0 specify the address. When XYED = 1, the upper
(BAD31 to BAD16) of BARD specify the X address bus, and the lower 16 bits (BAD
specify the Y address bus. BARDH and BARDL are initialized to H'0000 by a power-
after a manual reset, their values are undefined.

Note: * As an X/Y bus access is always a word access, the values of XAB0 and YAB
included in the break condition.

6.2.14 Break Address Mask Register D (BAMRD)

BAMRDH

Bit:	15	14	13	12	11	10	9
	BAMD31	BAMD30	BAMD29	BAMD28	BAMD27	BAMD26	BAMD2
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAMD23	BAMD22	BAMD21	BAMD20	BAMD19	BAMD18	BAMD1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BAMRDL

Bit:	15	14	13	12	11	10	9
	BAMD15	BAMD14	BAMD13	BAMD12	BAMD11	BAMD10	BAMD9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BAMD7	BAMD6	BAMD5	BAMD4	BAMD3	BAMD2	BAMD1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

XYED = 0	Address	Upper 16 bits maskable	Lower 16 bits masi
XYED = 1	X address (when XYSD = 0)	Maskable	—
	Y address (when XYSD = 1)	—	Maskable

Bits 31 to 0:

BAMDn	**Description**
0	Channel D break address bit BADn is included in break condition
1	Channel D break address bit BADn is masked, and not included in

Note: n = 31 to 0

Bit:	7	6	5	4	3	2	1
	BDD23	BDD22	BDD21	BDD20	BDD19	BDD18	BDD17
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BDRDL

Bit:	15	14	13	12	11	10	9
	BDD15	BDD14	BDD13	BDD12	BDD11	BDD10	BDD9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BDD7	BDD6	BDD5	BDD4	BDD3	BDD2	BDD1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break data register D (BDRD) consists of two 16-bit readable/writable registers: break
register DH (BDRDH) and break data register DL (BDRDL). BDRDH specifies the up
(bits 31 to 16) of the data used as a channel D break condition, and BDRDL specifies th
half (bits 15 to 0). The data bus connected to the X/Y memory can also be specified as
condition by making a setting in the XYED bit/XYSD bit in break bus cycle register D
When XYED = 1, the upper 16 bits (BDD31 to BDD16) of BDRD specify the X data b
lower 16 bits (BDD15 to BDD0) specify the Y data bus.

6.2.16 Break Data Mask Register D (BDMRD)

BDMRDH

Bit:	15	14	13	12	11	10	9
	BDMD31	BDMD30	BDMD29	BDMD28	BDMD27	BDMD26	BDMD
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BDMD23	BDMD22	BDMD21	BDMD20	BDMD19	BDMD18	BDMD
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BDMRDL

Bit:	15	14	13	12	11	10	9
	BDMD15	BDMD14	BDMD13	BDMD12	BDMD11	BDMD10	BDMD
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BDMD7	BDMD6	BDMD5	BDMD4	BDMD3	BDMD2	BDMD
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break data mask register D (BDMRD) consists of two 16-bit readable/writable registe
data mask register DH (BDMRDH) and break data mask register DL (BDMRDL). BD
specifies which bits of the break data set in BDRDH are to be masked, and BDMRDL
which bits of the break data set in BDRDL are to be masked. Operation also depends

	(when XYSD = 0)		
	Y data (when XYSD = 1)	—	Maskable

Bits 31 to 0:

BDMDn	Description
0	Channel D break data bit BDDn is included in break condition (In
1	Channel D break data bit BDDn is masked, and not included in cond

Notes: 1. n = 31 to 0

2. When including the data bus value in the break condition, specify the operan

3. When specifying byte size, and using odd-address data as a break condition value in bits 7 to 0 of BDRD and BDMRD. When using even-address data as condition, set the value in bits 15 to 8. The unused 8 bits of these registers h effect on the break condition.

	CPD1	CPD0	IDD1	IDD0	RWD1	RWD0	SZD1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break bus cycle register D (BBRD) is a 16-bit readable/writable register that sets five
break conditions: (1) internal bus (C-bus, I-bus)/X memory bus/Y memory bus), (2) C
cycle/on-chip DMAC (DMAC, E-DMAC) cycle, (3) instruction fetch/data access, (4)
and (5) operand size. BBRD is initialized to H'0000 by a power-on reset; after a manu
value is undefined.

Bits 15 to 10—Reserved: These bits are always read as 0. The write value should alwa

Bit 9—X/Y Memory Bus Enable D (XYED): Selects whether the X/Y bus is used as a
break condition.

Bit 9: XYED	Description
0	Cache bus or internal bus is selected as condition for channel D ad ()
1	X/Y bus is selected as condition for channel D address/data

Bit 8—X Bus/Y Bus Select D (XYSD): Selects whether the X bus or the Y bus is used
channel D break condition. This bit is valid only when bit XYED = 1.

Bit 8: XYSD	Description
0	X bus is selected as channel D break condition (
1	Y bus is selected as channel D break condition

The configuration of bits 7 to 0 is the same as for BBRA.

	ETRD7	ETRD6	ETRD5	ETRD4	ETRD3	ETRD2	ETRD1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a channel D execution-times break condition is enabled (by setting the ETBED b
BRCR), this 16-bit register specifies the number of times a channel D break condition c
before a user break interrupt is requested. The maximum value is $2^{12} – 1$ times. Each tim
channel D break condition occurs, the value in BETRD is decremented by 1. After the
value reaches H'0001, an interrupt is requested when a break condition next occurs.

As exceptions and interrupts cannot be accepted for instructions in a repeat loop compr
more than three instructions, BETRD is not decremented by the occurrence of a break c
for an instruction in such a repeat loop (see section 4.6, When Exception Sources Are N
Accepted).

Bits 15 to 12 are always read as 0, and should only be written with 0.

BETRD is initialized to H'0000 by a power-on reset.

Bit:	7	6	5	4	3	2	1
	CMFCB	CMFPB	—	SEQ1	SEQ0	PCBB	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BRCRL

Bit:	15	14	13	12	11	10	9
	CMFCC	CMFPC	ETBEC	—	DBEC	PCBC	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	CMFCD	CMFPD	ETBED	—	DBED	PCBD	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The break control register (BRCR) is used to make the following settings:

1. Setting of independent channel mode or sequential condition mode for channels A
2. Selection of pre- or post-instruction-execution break in case of an instruction fetch
3. Selection of whether the data bus is to be included in the comparison conditions fo and D
4. Selection of whether an execution-times break is to be set for channels C and D
5. Selection of whether a PC trace is to be executed

BRCR also contains flags that are set when a condition is satisfied. BRCR is initialize H'00000000 by a power-on reset; after a manual reset, its value is undefined.

Bit 30—DMAC Condition Match Flag A (CMFPA): This flag is set to 1 when an on-ch
bus cycle condition, among the break conditions set for channel A, is satisfied. This fla
cleared to 0 (if the flag setting is to be checked again after it has once been set, the flag
cleared by a write).

Bit 30: CMFPA	Description
0	User break interrupt has not been generated by a channel A on-chip cycle condition (Ir
1	User break interrupt has been generated by a channel A on-chip DM condition

Bits 29 and 28—Reserved: These bits are always read as 0. The write value should alw

Bit 27—PC Trace Enable (PCTE): Selects whether a PC trace is to be executed.

Bit 27: PCTE	Description
0	PC trace is not executed (Ir
1	PC trace is executed

Bit 26—PC Break Select A (PCBA): Selects whether a channel A instruction fetch cyc
effected before or after execution of the instruction.

Bit 26: PCBA	Description
0	Channel A instruction fetch cycle break is effected before instruction (Ir
1	Channel A instruction fetch cycle break is effected after instruction e

Bits 25 and 24—Reserved: These bits are always read as 0. The write value should alw

Bit 22—DMAC Condition Match Flag B (CMFPB): This flag is set to 1 when an on-c
bus cycle condition, among the break conditions set for channel B, is satisfied. This fl
cleared to 0 (if the flag setting is to be checked again after it has once been set, the flag
cleared by a write).

Bit 22: CMFPB	Description
0	User break interrupt has not been generated by a channel B on-ch cycle condition (
1	User break interrupt has been generated by a channel B on-chip D condition

Bit 21—Reserved: This bit is always read as 0. The write value should always be 0.

Bits 20 and 19—Sequence Condition Select (SEQ1, SEQ0): These bits select indepen
sequential conditions for channels A, B, C, and D.

Bit 20: SEQ1	Bit 19: SEQ0	Description
0	0	Comparison based on independent conditions for channels A, B (
	1	Channel C → D sequential condition; channels A and B indepen
1	0	Channel B → C → D sequential condition; channel A independe
	1	Channel A → B → C → D sequential condition

Bit 18—PC Break Select B (PCBB): Selects whether a channel B instruction fetch cyc
effected before or after execution of the instruction.

Bit 18: PCBB	Description
0	Channel B instruction fetch cycle break is effected before instructio (
1	Channel B instruction fetch cycle break is effected after instruction

	condition (In
1	User break interrupt has been generated by a channel C CPU cycle

Bit 14—DMAC Condition Match Flag C (CMFPC): This flag is set to 1 when an on-ch bus cycle condition, among the break conditions set for channel C, is satisfied. This flag cleared to 0 (if the flag setting is to be checked again after it has once been set, the flag cleared by a write).

Bit 14: CMFPC	Description
0	User break interrupt has not been generated by a channel C on-chip cycle condition (In
1	User break interrupt has been generated by a channel C on-chip DM condition

Bit 13—Execution-Times Break Enable C (ETBEC): Enables a channel C execution-ti condition. When this bit is 1, a user break interrupt is generated when the number of br conditions that have occurred equals the number of executions specified by the break ex times register (BETRC).

Bit 13: ETBEC	Description
0	Channel C execution-times break condition is disabled (In
1	Channel C execution-times break condition is enabled

Bit 12—Reserved: This bit is always read as 0. The write value should always be 0.

Bit 11—Data Break Enable C (DBEC): Selects whether a data bus condition is to be in the channel C break conditions.

Bit 11: DBEC	Description
0	Data bus condition is not included in channel C conditions (In
1	Data bus condition is included in channel C conditions

Bit 7—CPU Condition Match Flag D (CMFCD): This flag is set to 1 when a CPU bus condition, among the break conditions set for channel D, is satisfied. This flag is not c (if the flag setting is to be checked again after it has once been set, the flag must be cle write).

Bit 7: CMFCD	Description
0	User break interrupt has not been generated by a channel D CPU condition (
1	User break interrupt has been generated by a channel D CPU cycl

Bit 6—DMAC Condition Match Flag D (CMFPD): This flag is set to 1 when a DMAC condition, among the break conditions set for channel D, is satisfied. This flag is not c (if the flag setting is to be checked again after it has once been set, the flag must be cle write).

Bit 6: CMFPD	Description
0	User break interrupt has not been generated by a channel D on-ch cycle condition (
1	User break interrupt has been generated by a channel D on-chip D condition

Bit 5—Execution-Times Break Enable D (ETBED): Enables a channel D execution-ti condition. When this bit is 1, a user break interrupt is generated when the number of b conditions that have occurred equals the number of executions specified by the break times register (BETRD).

Bit 5: ETBED	Description
0	Channel D execution-times break condition is disabled (
1	Channel D execution-times break condition is enabled

Bit 4—Reserved: This bit is always read as 0. The write value should always be 0.

effected before or after execution of the instruction.

Bit 2: PCBD	Description
0	Channel D instruction fetch cycle break is effected before instruction (In
1	Channel D instruction fetch cycle break is effected after instruction e

Bits 1 and 0—Reserved: These bits are always read as 0. The write value should always

6.2.20 Branch Flag Registers (BRFR)

Bit:	15	14	13	12	11	10	9
	SVF	PID2	PID1	PID0	—	—	—
Initial value:	0	Undefined	Undefined	Undefined	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	DVF	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

The branch flag registers (BRFR) comprise a set of four 16-bit read-only registers. The registers contain flags indicating whether the actual branch addresses (in a branch instru repeat, interrupt, etc.) have been saved in BRSR and BRDR, and a 3-bit pointer indicati number of cycles from fetch to execution of the last instruction executed. The BRFR re form a FIFO (first-in first-out) queue for PC trace use. The queue is shifted at each bran

Bits SVF and DVF are initialized by a power-on reset, but bits PID2 to PID0 are not.

Bit 15—Source Verify Flag (SVF): Indicates whether the address and pointer that enab branch source address to be calculated have been stored in BRSR. This flag is set when instruction at the branch destination address is fetched, and reset when BRSR is read.

Odd	PID indicates instruction buffer number	(
Even	PID+2 indicates instruction buffer number	

Bits 11 to 8, 6 to 0—Reserved: These bits are always read as 0. The write value shoul
0.

Bit 7—Destination Verify Flag (DVF): Indicates whether the branch source address h
stored in BRDR. This flag is set when the instruction at the branch destination address
and reset when BRDR is read.

Bit 7: DVF	Description	
0	BRDR value is invalid	(
1	BRDR value is valid	

See the PC trace description for the method of executing a PC trace using the branch s
registers (BRSR), branch destination registers (BRDR), and branch flag registers (BR

6.2.21 Branch Source Registers (BRSR)

BRSRH

Bit:	15	14	13	12	11	10	9
	BSA31	BSA30	BSA29	BSA28	BSA27	BSA26	BSA25
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W:	R	R	R	R	R	R	R

Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R	R	R	R	R

The branch source registers (BRSR) comprise a set of four 32-bit read-only registers. T
in these registers are used to calculate the address of the last instruction executed befor
when performing a PC trace. The BRSR registers form a FIFO (first-in first-out) queue
trace use. The queue is shifted at each branch.

The BRSR registers are not initialized by a reset.

6.2.22 Branch Destination Registers (BRDR)

BRDRH

Bit:	15	14	13	12	11	10	9
	BDA31	BDA30	BDA29	BDA28	BDA27	BDA26	BDA25
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R	R	R	R	R

Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W:	R	R	R	R	R	R	R

The branch destination registers (BRDR) comprise a set of four 32-bit read-only regis
registers store the branch destination fetch addresses used when performing a PC trace
BRDR registers form a FIFO (first-in first-out) queue for PC trace use. The queue is s
each branch.

The BRDR registers are not initialized by a reset.

6.3 Operation

6.3.1 User Break Operation Sequence

The sequence of operations from setting of break conditions to user break interrupt ex
handling is described below.

1. Set the break address in the break address register (BARA/BARB/BARC/BARD),
   be masked in the break address mask register (BAMRA/BAMRB/BAMRC/BAM
   break bits in the break data register (BDRC/BDRD), and the data to be masked in
   data mask register (BDMRC/BDMRD).

   Set the break bus conditions in the break bus cycle register (BBRA/BBRB/BBRC/
   Make three settings—CPU cycle/on-chip DMAC cycle select, instruction fetch/da
   select, and read/write select—for each of BBRA, BBRB, BBRC, and BBRD. A us
   interrupt will not be generated for a channel for which any one of these settings is

   Set the respective conditions in the corresponding BRCR register bits.

2. When a set condition is satisfied, the UBC sends a user break interrupt request to t
   controller (INTC). The CPU condition match flag (CMFCA/CMFCB/CMFCC/CM
   DMAC condition match flag (CMFPA/CMFPB/CMFPC/CMFPD) is also set for t
   condition for the respective channel.

3. Whether a set condition is matched or not can be ascertained from the respective co
match flag (CMFCA, CMFPA, CMFCB, CMFPB, CMFCC, CMFPC, CMFCD, or
These flags are set by a match with the set condition, but are not reset. Therefore, if
of a particular flag is to be checked again, the flag must be cleared by writing 0.

When an execution-times break is specified for channel C or D, the CMFCC, CMF
CMFCD, or CMFPD flag is set when the number of executions matches the numbe
executions specified by BETRC or BETRD.

6.3.2 Instruction Fetch Cycle Break

1. If a CPU/instruction fetch/read/word setting is made in the break bus cycle register
BBRB, BBRC, or BBRD), a CPU instruction fetch cycle can be selected as a break
In this case, it is possible to specify whether the break is to be effected before or aft
execution of the relevant instruction by means of the PCBA/PCBB/PCBC/PCBD bi
break control register (BRCR).

2. In the case of an instruction for which pre-execution is set as the break condition, th
performed when it has been confirmed that the instruction has been fetched and is t
executed. Consequently, a break cannot be set for an overrun-fetched instruction (a
instruction fetched but not executed in the event of a branch or interrupt transition).
is set for the delay slot of a delayed branch instruction, or for the instruction followi
instruction for which interrupts are prohibited, such as LCD, an interrupt is generate
execution of the next instruction at which interrupts are accepted.

3. With the post-execution condition, an interrupt is generated after execution of the i
set as the break condition, and before execution of the following instruction. As in 2
break cannot be set for an overrun-fetched instruction. If a break is set for a delayed
instruction, or for an instruction for which interrupts are prohibited, such as LCD, a
is generated before execution of the next instruction at which interrupts are accepte

4. When an instruction fetch cycle is set for channel C or D, break data register C (BD
break data register D (BDRD) is ignored. Therefore, break data need not be set for a
instruction fetch cycle break.

burst write of a synchronous DRAM, or for a dummy access cycle of a single read

2. Table 6.2 shows the bits of the break address register and the address bus that are c
each operand size to determine whether a break condition has been matched.

Table 6.2 Data Access Cycle Address and Operand Size Comparison Conditio

Access Size	Compared Address Bits
Longword	Bits 31 to 2 of break address register compared with bits 31 to 2 of
Word	Bits 31 to 1 of break address register compared with bits 31 to 1 of
Byte	Bits 31 to 0 of break address register compared with bits 31 to 0 of

This means, for example, that if address H'00001003 is set without specifying a siz
bus cycles that satisfy the break conditions are as follows (assuming that all other c
are satisfied):

Longword access at address H'00001000

Word access at address H'00001002

Byte access at address H'00001003

3. When data value is included in break condition in channel C

When the data value is included in the break conditions, specify longword, word, c
operand size in break bus cycle register C (BBRC). When the data value is include
break conditions, a break interrupt is generated on a match of the address condition
data condition.

When byte data is specified, set the same data in the two bytes comprising bits 15 t
7 to 0 in break data register C (BDRC) and break data mask register C (BDMRC).
byte is designated, bits 31 to 16 of BDRC and BDMRC are ignored.

Similar conditions apply when the data value is included in the break conditions fo

so that the saved PC value is the address at which the break occurs.

2. When instruction fetch (post-instruction-execution) is set as break condition

   The program counter (PC) value saved to the stack in user break interrupt exception is the address of the next instruction to be executed after the instruction for which th condition matched. In this case, the fetched instruction is executed, and a user break is generated before execution of the next instruction. However, if a setting is made f instruction for which interrupts are prohibited, the break is effected before executio next instruction at which interrupts are accepted, so that the saved PC value is the a which the break occurs.

3. When data access (CPU/on-chip DMAC) is set as break condition

   The value saved is the start address of the next instruction after the instruction for w execution has been completed when user break exception handling is initiated.

   When data access (CPU/on-chip DMAC) is set as a break condition, the point at wh break is to be made cannot be specified. A break is effected before execution of the about to be fetched around the time of the break data access.

6.3.5 X Memory Bus or Y Memory Bus Cycle Break

A break condition for an X bus cycle or Y bus cycle can only be specified for channel C When XYEC in BBRC or XYED in BBRD is set to 1, break addresses and break data c memory bus or Y memory bus are selected. Either the X memory bus or the Y memory be selected with the XYSC bit in BBRC or the XYSD bit in BBRD; the X and Y memo cannot both be included in the break conditions at the same time. The break conditions to X memory bus cycles or Y memory bus cycles by setting the CPU bus master, data a cycle, read or write access, and word operand size or no operand size specification.

When an X memory address is selected as a break condition, specify the X memory add upper 16 bits of BARC and BAMRC or BARD and BAMRD; when a Y memory addre selected, specify the Y memory address in the lower 16 bits of BARC and BAMRC or BAMRD. The same method is used to specify X memory data or Y memory data for B BDMRC or BDRD and BMRD.

C have already been met when the break conditions for channels C and D are met at th
time, the conditions for channel D are considered to be met and a break occurs.

Channel B to Channel C to Channel D: When SEQ1 in BRCR is set to 1 and SEQ0
sequential break occurs when the conditions are met for channel B, channel C, and the
D, in that order. This causes the BRCR condition match flag for each channel to be se

If the break conditions for channels B and C are met at the same time, and the conditio
already been met for channel B, the conditions are considered to be met for channel B
conditions for channel B have already been met when the break conditions for channe
are met at the same time, the conditions for channel C are considered to be met.

If the break conditions for channels C and D are met at the same time, and the conditio
already been met for channel C, the conditions are considered to be met for channel C
conditions for channel C have already been met when the break conditions for channe
are met at the same time, the conditions for channel D are considered to be met and a
occurs.

Channel A to Channel B to Channel C to Channel D: When SEQ1 in BRCR is set
SEQ0 is set to 1, a sequential break occurs when the conditions are met for channel A,
channel C, and then channel D, in that order. This causes the BRCR condition match f
channel to be set to 1.

If the break conditions for channels A and B are met at the same time, and the conditio
already been met for channel A, the conditions are considered to be met for channel A
conditions for channel A have already been met when the break conditions for channe
are met at the same time, the conditions for channel B are considered to be met.

If the break conditions for channels B and C are met at the same time, and the conditio
already been met for channel B, the conditions are considered to be met for channel B
conditions for channel B have already been met when the break conditions for channe
are met at the same time, the conditions for channel C are considered to be met.

If the break conditions for channels C and D are met at the same time, and the conditio
already been met for channel C, the conditions are considered to be met for channel C
conditions for channel C have already been met when the break conditions for channe

Bus X or bus Y may be selected in the sequential break setting, and it is also possible to
number of executions as a brake condition. For example, if an execution-times break is
channels C and D, a user break interrupt will be issued if, after the execution-times set
set in BETRC has occurred, the execution-times condition set in BETRD for channel D

6.3.7 PC Traces

1. A PC trace is started by setting the PC trace enable bit (PCTE) to 1 in BRCR. When
(branch instruction, repeat, or interrupt) occurs the address that enables the branch s
address to be calculated and the branch destination address are stored in the branch
register (BRSR) and branch destination register (BRDR). The address stored in BRI
branch destination instruction fetch address. The address stored in BRSR is the last
fetch address prior to the branch. A pointer indicating the relationship to the instruc
executed immediately before the branch is stored in the branch flag register (BRFR)

2. The address of the instruction executed immediately before the branch occurred can
calculated from the address stored in BRSR and the pointer stored in BRFR. Design
address stored in BRSR as BSA, the pointer stored in BRFR as PID, and the addres
the branch as IA, then IA is found from the following equation:

$$IA = BSA - 2 \times PID$$

Caution is necessary if an interrupt (branch) occurs before the instruction at the bran
destination is executed. In the case illustrated in figure 6.2., the address of instructio
executed immediately before the branch, is calculated from the equation IA = BSA
However, if branch “branch” is a delayed branch instruction with a delay slot and th
destination is a 4n+2 address, branch destination address “Dest” specified by the bra
instruction is stored directly in BRSR. In this case, therefore, equation IA = BSA –
not applied, and PID is invalid. BSA is at a 4n+2 boundary in this case only, catego
shown in table 6.3.

Figure 6.2 When Interrupt Occurs before Branch Instruction Is Execu

Table 6.3 BSA Values Stored in Exception Handling before Execution of Bran Destination Instruction

Branch	Branch Destination (Dest)	BSA	Branch Source Address Calculable of BRSR and BRFR
Delay	4n	4n	Exec = IA = BSA – 2 × PID
	4n + 2	4n + 2	Dest = BSA
No delay	4n or 4n + 2	4n	Exec = IA = BSA – 2 × PID

If PID is an odd number, the value incremented by 2 indicates the instruction buffe equations in the table do not take this into account. Therefore, the calculation can b using the values of BSA stored in BRSR and PID stored in BRFR.

3. The location indicated by the address before branch occurrence, IA, differs accord kind of branch.
   a. Branch instruction: Branch instruction address
   b. Repeat loop: 2nd instruction from last in repeat loop

```
Repeat_Start: inst (1); → BRDR
              inst(2);
              :
              inst (n-1); → Address calculated from BRSR and
Repeat End:   inst (n);
```

   c. Interrupt: Instruction executed immediately before interrupt
      The address of the first instruction in the interrupt routine is stored in BRDR.

In a repeat loop consisting of no more than three instructions, an instruction fetch generated. As the branch destination address is unknown, a PC trace cannot be per

A. Register settings: BARA = H'00000404 / BAMRA = H'00000000 / BBRA = H'005
BARB = H'00003080 / BAMRB = H'0000007F / BBRB = H'005
BARC = H'00008010 / BAMRC = H'00000006 / BBRC = H'005
BDRC = H'00000000 / BDMRC = H'00000000
BARD =H'0000FF04 / BAMRD = H'00000000 / BBRD = H'005
BDRD = H'00000000 / BDMRD = H'00000000
BRCR = H'04000400

Set conditions: All channels independent

Channel A: Address: H'00000404; address mask: H'0000000
Bus cycle: CPU, instruction fetch (post-execution),
read (operand size not included in cond

Channel B: Address: H'00003080; address mask: H'0000007
Bus cycle: CPU, instruction fetch (pre-execution),
read (operand size not included in cond

Channel C: Address: H'00008010; address mask: H'0000000
Data: H'00000000; data mask: H'00000000
Bus cycle: CPU, instruction fetch (post-execution),
read (operand size not included in cond

Channel D: Address: H'0000FF04; address mask: H'0000000
Data: H'00000000; data mask: H'00000000
Bus cycle: CPU, instruction fetch (pre-execution),
read (operand size not included in cond

A user break interrupt is generated after execution of the instruction at address H'00
before execution of instructions at addresses H'00003080 to H'000030FF, after exec
instructions at addresses H'00008010 to H'00008016, or before execution of the inst
address H'0000FF04.

Channel A:	Address:	H'00027128; address mask: H'000000
	Bus cycle:	CPU, instruction fetch (pre-execution)
Channel B:	Address:	H'00031415; address mask: H'000000
	Bus cycle:	CPU, instruction fetch (pre-execution)
		read (operand size not included in con
Channel C:	Address:	H'00037226; address mask: H'000000
	Data:	H'00000000; data mask: H'00000000
	Bus cycle:	CPU, instruction fetch (pre-execution)
Channel D:	Address:	H'0003722E; address mask: H'000000
	Data:	H'00000000; data mask: H'00000000
	Bus cycle:	CPU, instruction fetch (pre-execution)

On channel A, a user break interrupt is not generated as an instruction fetch is not
cycle.

On channel B, a user break interrupt is not generated as an instruction fetch is perf
even address.

A user break interrupt is generated by a channel C and D sequential condition mat
execution of the instruction at address H'0003722E following execution of the inst
address H'00037226.

Channel A: Not used

Channel B: Not used

Channel C: Address: H'00037226; address mask: H'0000000
Data: H'00000000; data mask: H'00000000
Bus cycle: CPU, instruction fetch (pre-execution),

Channel D: Address: H'0003722E; address mask: H'0000000
Data: H'00000000; data mask: H'00000000
Bus cycle: CPU, instruction fetch (pre-execution),

As the channel C break condition is a write cycle, the condition is not matched, and sequential conditions are not satisfied, a user break interrupt is not generated.

D. Register settings: BBRA = H'0000
BARB = H'00000500 / BAMRB = H'00000000 / BBRB = H'005
BARC = H'00000A00 / BAMRC = H'00000000 / BBRC = H'005
BDRC = H'00000000 / BDMRC = H'00000000
BARD = H'00001000 / BAMRD = H'00000000 / BBRD = H'005
BDRD = H'00000000 / BDMRD = H'00000000
BRCR = H'00102020 / BETRC = H'0005 / BETRD = H'000A

Channel A: Not used

Channel B: Address: H'00000500; address mask: H'0000000
Data: H'00000000; data mask: H'00000000
Bus cycle: CPU, instruction fetch (pre-execution),

Channel C: Address: H'00000A00; address mask: H'0000000
Data: H'00000000; data mask: H'00000000
Bus cycle: CPU, instruction fetch (pre-execution),
Execution-times break enabled (5 times)

CPU Data Access Cycle Break Condition Settings

Register settings: BARA = H'00123456 / BAMRA = H'00000000 / BBRA = H'00
BARB = H'01000000 / BAMRB = H'00000000 / BBRB = H'00
BARC = H'000ABCDE / BAMRC = H'000000FF / BBRC = H'
BDRC = H'0000A512 / BDMRC = H'00000000
BARD = H'1001E000 / BAMRD = H'FFFF0000 / BBRD = H'0
BDRD = H'00004567 / BDMRD = H'00000000
BRCR = H'00000808

Set conditions: All channels independent

Channel A: Address: H'00123456; address mask: H'000000
Bus cycle: CPU, data access, read (operand size n
in conditions)

Channel B: Address: H'01000000; address mask: H'000000
Bus cycle: CPU, data access, read, word

Channel C: Address: H'000ABCDE; address mask: H'000000
Data: H'0000A512; data mask: H'00000000
Bus cycle: CPU, data access, write, word

Channel D: Y address: H'1001E000; address mask: H'FFFF00
Data: H'00004567; data mask: H'00000000
Bus cycle: CPU, data access, write, word

On channel A, a user break interrupt is generated by a longword read at address H'
word read at address H'00123456, or a byte read at address H'00123456.

On channel B, a user break interrupt is generated by a word read at address H'0100

On channel C, a user break interrupt is generated when H'A512 is written by word
address from H'000ABC00 to H'000ABCFE.

On channel D, a user break interrupt is generated when H'4567 is written by word
address H'1001E000 in Y memory space.

Set conditions: All channels independent

Channel A: Address: H'00314156; address mask: H'0000000
Bus cycle: DMAC, instruction fetch, read (operanc
included in conditions)

Channel B: Not used

Channel C: Not used

Channel D: Address: H'00055555; address mask: H'0000000
Data: H'00007878; data mask: H'00000F0F
Bus cycle: DMAC, data access, write, byte

On channel A, a user break interrupt is not generated as an instruction fetch is not p
a DMAC cycle.

On channel D, a user break interrupt is generated when the DMAC writes H'7* (*: I
is written by byte access to address H'00055555.

b. If, of the channels included in a sequential condition, the channel bus cycle co
constituting the first break conditions of adjacent channels are specified as a pr
break (PCB bit cleared to 0 in BRCR) and an instruction fetch (designated by t
cycle register), note that when the bus cycle conditions for the two channels ar
simultaneously, a break is effected and the BRCR condition match flags are se

3. When changing a register setting, the written value normally becomes effective in
In an on-chip memory fetch, two instructions are fetched simultaneously. If the fet
second instruction has been set as a break condition, even if the break condition is
modifying the relevant UBC registers immediately after the fetch of the first instru
break interrupt will still be generated prior to the second instruction. To fix a timin
the setting is definitely changed, the last register value written should be read with
access. The changed setting will be valid from this point on.

4. If a user break interrupt is generated by an instruction fetch condition match, and t
is matched again in the UBC during execution of the exception service routine, ex
handling for that break will be executed when the interrupt request mask value in S
14 or below. Therefore, when masking addresses and setting an instruction fetch/p
execution condition to perform step-execution, ensure that an address match does
during execution of the UBC's exception service routine.

5. Note the following when specifying an instruction in a repeat loop that includes a
instruction as a break condition.

   When an instruction in a repeat loop is specified as a break condition:

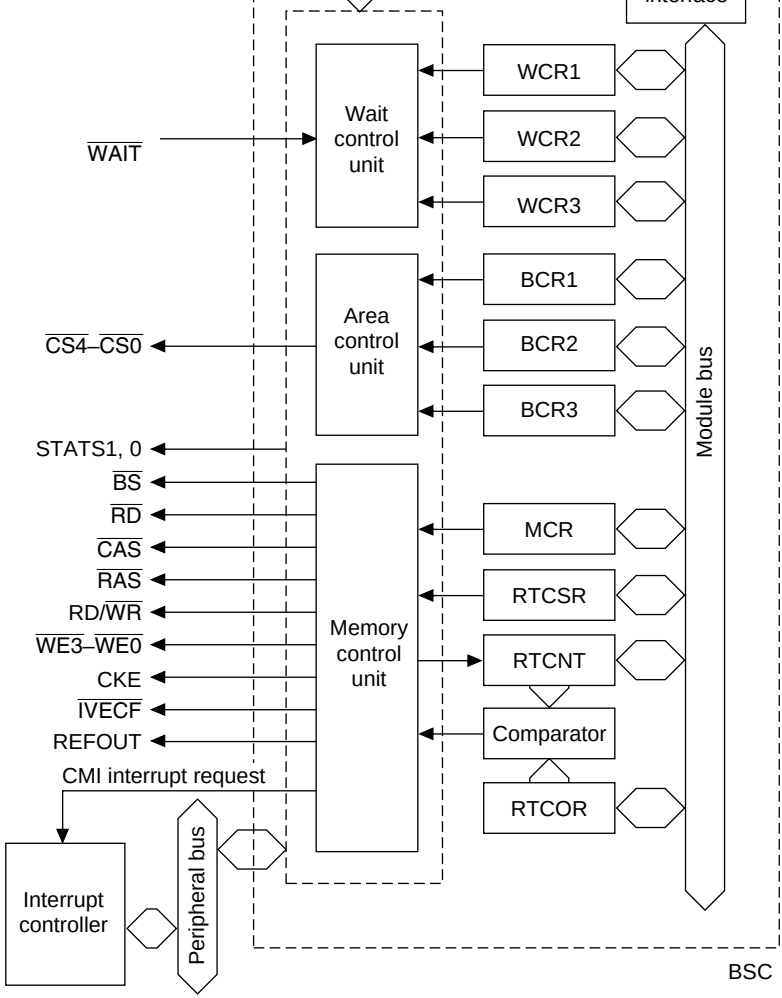
   a. A break will not occur during execution of a repeat loop comprising no more t
   instructions.
   b. When an execution-times break is set, an instruction fetch from memory will n
   during execution of a repeat loop comprising no more than three instructions.
   Consequently, the value in the break execution times register (BETRC or BET
   be decremented.

6. Do not execute a branch instruction immediately after reading a PC trace register (
BRSR, or BRDR).

7.1.1 Features

The BSC has the following features:

- Address space is managed as five spaces
  - Maximum linear 32 Mbytes for each of the address spaces CS0 to CS4
  - Memory type (DRAM, synchronous DRAM, burst ROM, etc.) can be specifie space.
  - Bus width (8, 16, or 32 bits) can be selected for each space.
  - Wait state insertion can be controlled for each space.
  - Control signals are output for each space.
- Cache
  - Cache area and cache-through area can be selected by access address.
  - In cache access, in the event of a cache access miss 16 bytes are read consecuti byte units to fill the cache. Write-through mode/write-back mode can be select writes.
  - In cache-through access, access is performed according to access size.
- Refresh
  - Supports $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh (auto-refresh) and self-refresh.
  - Refresh interval can be set by the refresh counter and clock selection.
  - Intensive refreshing by means of refresh count setting (1, 2, 4, 6, or 8)
- Direct interface to DRAM
  - Row/column address multiplex output.
  - Burst transfer during reads, fast page mode for consecutive accesses.
  - TP cycle generation to secure $\overline{\text{RAS}}$ precharge time.
  - EDO mode
- Direct interface to synchronous DRAM
  - Row/column address multiplex output.
  - Selection of burst read, single write mode or burst read, burst write mode
  - Bank active mode



WCR: Wait control register
BCR: Bus control register
MCR: Individual memory control register
RTCSR: Refresh timer control/status
RTCNT: Refresh timer counter
RTCOR: Refresh time constant regis

Figure 7.1 BSC Block Diagram

D31 to D0	I/O	Hi-Z	32-bit data bus. When reading or writing a 16-bit wi use D15 to D0; when reading or writing a 8-bit width D7 to D0. With 8-bit accesses that read or write a 3 area, input and output the data via the byte position by the lower address bits of the 32-bit bus
$\overline{BS}$	Output	Hi-Z	Indicates start of bus cycle or monitor. With the basi (device interfaces except for DRAM, synchronous D signal is asserted for a single clock cycle simultane address output. The start of the bus cycle can be de by this signal
$\overline{CS0}$ to $\overline{CS4}$	Output	Hi-Z	Chip select. $\overline{CS3}$ is not asserted when the CS3 spa space
RD/$\overline{WR}$	Output	Hi-Z	Read/write signal. Signal that indicates access cycle (read/write). Connected to $\overline{WE}$ pin when DRAM/syn DRAM is connected
$\overline{RAS}$	Output	Hi-Z	$\overline{RAS}$ pin for DRAM/synchronous DRAM
$\overline{CAS}$/$\overline{OE}$	Output	Hi-Z	Open when using DRAM Connected to $\overline{OE}$ pin when using EDO RAM Connected to $\overline{CAS}$ pin when using synchronous DR
$\overline{RD}$	Output	Hi-Z	Read pulse signal (read data output enable signal). connected to the device's $\overline{OE}$ pin; when there is an data buffer, the read cycle data can only be output signal is low
$\overline{WAIT}$	Input	Don't care	Hardware wait input
$\overline{BRLS}$	Input	Input	Bus release request input
$\overline{BGR}$	Output	Output	Bus grant output
CKE	Output	Output	Synchronous DRAM clock enable control. Signal fo synchronous DRAM self-refresh
$\overline{IVECF}$	Output	Output	Interrupt vector fetch
DREQ0	Input	Input	DMA request 0
DACK0	Output	Output	DMA acknowledge 0

DQMLU/ $\overline{WE2}$	Output	Hi-Z	When synchronous DRAM is used, connected to D the second byte (D23 to D16). For ordinary space, writing to the second byte
DQMLU/ $\overline{WE1}$	Output	Hi-Z	When synchronous DRAM is used, connected to D the third byte (D15 to D8). For ordinary space, indi to the third byte
DQMLL/ $\overline{WE0}$	Output	Hi-Z	When synchronous DRAM is used, connected to D the least significant byte (D7 to D0). For ordinary s indicates writing to the least significant byte
$\overline{CAS3}$	Output	Hi-Z	When DRAM is used, connected to $\overline{CAS}$ pin for the significant byte (D31 to D24)
$\overline{CAS2}$	Output	Hi-Z	When DRAM is used, connected to $\overline{CAS}$ pin for the byte (D23 to D16)
$\overline{CAS1}$	Output	Hi-Z	When DRAM is used, connected to $\overline{CAS}$ pin for the (D15 to D8)
$\overline{CAS0}$	Output	Hi-Z	When DRAM is used, connected to $\overline{CAS}$ pin for the significant byte (D7 to D0)
STATS0, STATS1	Output	Output	Bus master identification 00: CPU 01: DMAC 10: E-DMAC 11: Other
$\overline{BUSHIZ}$	Input	Input	Signal used in combination with $\overline{WAIT}$ signal to pla strobe signals in the high-impedance state without bus cycle.

Note: Hi-Z: High impedance

Initialization Procedure: Do not access a space other than CS0 until the settings for th
to memory are completed.

Table 7.2 Register Configuration

Name	Abbre-viation	R/W	Initial Value	Address*1	A
Bus control register 1	BCR1	R/W	H'03F0	H'FFFFFFE0	1
Bus control register 2	BCR2	R/W	H'00FC	H'FFFFFFE4	1
Bus control register 3	BCR3	R/W	H'0F00	H'FFFFFFFC	1
Wait control register 1	WCR1	R/W	H'AAFF	H'FFFFFFE8	1
Wait control register 2	WCR2	R/W	H'000B	H'FFFFFFC0	1
Wait control register 3	WCR3	R/W	H'0000	H'FFFFFFC4	1
Individual memory control register	MCR	R/W	H'0000	H'FFFFFFEC	1
Refresh timer control/status register	RTCSR	R/W	H'0000	H'FFFFFFF0	1
Refresh timer counter	RTCNT	R/W	H'0000	H'FFFFFFF4	1
Refresh time constant register	RTCOR	R/W	H'0000	H'FFFFFFF8	1

Notes: 1. This address is for 32-bit accesses; for 16-bit accesses add 2.
2. 16-bit access is for read only.

The chip has 16-kbyte RAM as on-chip memory. The on-chip RAM is divided into an
a Y area, which can be accessed in parallel with the DSP instruction. See the SH-1/SH
Programming Manual for more information.

There are several spaces for cache control. These include the associative purge space f
purges, address array read/write space for reading and writing addresses (address tags)
array read/write space for forced reads and writes of data arrays.

Table 7.3 Address Map

Address	Space	Memory
H'00000000–H'01FFFFFF	CS0 space, cache area	Ordinary space or burst ROM
H'02000000–H'03FFFFFF	CS1 space, cache area	Ordinary space
H'04000000–H'05FFFFFF	CS2 space, cache area	Ordinary space or synchronous DRAM*2
H'06000000–H'07FFFFFF	CS3 space, cache area	Ordinary space, synchronous DRAM*2, or DRAM
H'08000000–H'09FFFFFF	CS4 space, cache area	Ordinary space (I/O device)
H'0A000000–H'0FFFFFFF	Reserved*1	
H'10000000–H'1000DFFF	Reserved*1	
H'1000E000–H'1000EFFF	On-chip X RAM area	
H'1000F000–H'1001DFFF	Reserved*1	
H'1001E000–H'1001EFFF	On-chip Y RAM area	
H'1001F000–H'1FFFFFFF	Reserved*1	
H'20000000–H'21FFFFFF	CS0 space, cache-through area	Ordinary space or burst ROM
H'22000000–H'23FFFFFF	CS1 space, cache-through area	Ordinary space

H'40000000–H'49FFFFFF	Associative purge space	1
H'4A000000–H'5FFFFFFF	Reserved*1	
H'60000000–H'7FFFFFFF	Address array, read/write space	5
H'80000000–H'BFFFFFFF	Reserved*1	
H'C0000000–H'C0000FFF	Data array, read/write space	4
H'C0001000–H'DFFFFFFF	Reserved*1	
H'E0000000–H'FFFEFFFF	Reserved*1	
H'FFFF0000–H'FFFF0FFF	For setting synchronous DRAM mode	4
H'FFFF1000–H'FFFF7FFF	Reserved*1	
H'FFFF8000–H'FFFF8FFF	For setting synchronous DRAM mode	4
H'FFFFC000–H'FFFFBFFF	Reserved*1	
H'FFFFFC00–H'FFFFFFFF	On-chip peripheral modules	

Notes: 1. Do not access reserved spaces, as operation cannot be guaranteed.

2. Bank-active mode is not supported for CS2 space synchronous DRAM acce
precharge mode is always used.

Bank-active mode is supported for CS3 space synchronous DRAM access.

R/W:	R	R/W	R/W	R/W	R/W	R	R/W

Bit:	7	6	5	4	3	2	1
	A1LW1	A1LW0	A0LW1	A0LW0	A4EN DIAN	DRAM2	DRAM
Initial value:	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initialize the ENDIAN, BSTROM, PSHR, and DRAM2 to DRAM0 bits after a power
and do not change their values thereafter. To change other bits by writing to them, wri
value as they are initialized to. Do not access any space other than CS0 until the regist
initialization ends.

Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.

Bits 14 and 13—Long Wait Specification for Area 4 (A4LW1, A4LW0): From 3 to 14
are inserted in CS4 space accesses when the wait control bits (W41, W40) in wait con
2 (WCR2) are set as long wait (i.e., are set to 11) (see table 7.4).

Bit 12—Endian Specification for Area 2 (A2ENDIAN): In big-endian format, the MS
data is the lowest byte address and byte data goes in order toward the LSB. For little-e
format, the LSB of byte data is the lowest byte address and byte data goes in order tov
MSB. When this bit is 1, the data is rearranged into little-endian format before transfe
CS2 space is read or written to. It is used when handling data with little-endian proces
running programs written with conscious use of little-endian format.

Note: Data rearrangement into little-endian format requires no extra processing time

Bit 12: A2ENDIAN	**Description**
0	Big-endian
1	Little-endian

memory interface setting is made for CS2 and CS3, from 3 to 14 wait cycles are inserte
CS3 accesses when the bits specifying the respective area waits in the wait control bits
W20 or W31, W30) in wait control register 1 (WCR1) are set as long waits (i.e., are set
table 7.4).

Bits 7 and 6—Long Wait Specification for Area 1 (A1LW1, A1LW0): From 3 to 14 wa
are inserted in area 1 accesses when the wait control bits (W11, W10) in wait control re
(WCR1) are set as long wait (i.e., are set to 11) (see table 7.4).

Bits 5 and 4—Long Wait Specification for Area 0 (A0LW1, A0LW0): When the basic
interface setting is made for CS0, from 3 to 14 wait cycles are inserted in CS0 accesses
wait control bits (W01, W00) in wait control register 1 (WCR1) are set as long wait (i.e
11) (see table 7.4).

Bit 3—Endian Specification for Area 4 (A4ENDIAN): In big-endian mode, the most si
byte (MSB) is the lowest byte address, and byte data is aligned in order toward the leas
byte (LSB). In little-endian mode, the LSB is the lowest byte address, and byte data is a
order toward the MSB. When this bit is set to 1, data in read/write accesses to the CS4 s
rearranged into little endian order before being transferred. This is used for data exchan
little-endian processor or when executing a program written with awareness of little-en

Note: Data rearrangement into little-endian format requires no extra processing time.

Bit 3: A4ENDIAN	Description
0	Big endian
1	Little endian

		1	CS2 and CS3 are synchronous DRAM spaces
	1	0	Reserved (do not set)
		1	Reserved (do not set)

Table 7.4 Wait Values Corresponding to BCR1 and BCR3 Register Settings (A

BCR3	BCR1		
AnLW2	**AnLW1**	**AnLW0**	**Wait Value**
0	0	0	3 cycles inserted
		1	4 cycles inserted
	1	0	5 cycles inserted
		1	6 cycles inserted
1	0	0	8 cycles inserted
		1	10 cycles inserted
	1	0	12 cycles inserted
		1	14 cycles inserted (

Note: n = 0 to 4
AHLW2, AHLW1, and AHLW0 are common to CS2 and CS3.

	A3SZ1	A3SZ0	A2SZ1	A2SZ0	A1SZ1	A1SZ0	—
Initial value:	1	1	1	1	1	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R

Initialize BCR2 after a power-on reset and do not write to it thereafter. When writing to
the same values as those the bits are initialized to. Do not access any space other than C
the register initialization ends.

The CS0 space bus size specification is set with pins MD4 and MD3. See section 3.3, C
Bus Width of the CS0 Area, for details.

Bits 15 to 10—Reserved: These bits are always read as 0. The write value should alway

Bits 9 and 8—Bus Size Specification for Area 4 (CS4) (A4SZ1, A4SZ0)

Bit 9: A4SZ1	Bit 8: A4SZ0	Description
0	0	Longword (32-bit) size (Ir
	1	Byte (8-bit) size
1	0	Word (16-bit) size
	1	Longword (32-bit) size

Bits 7 and 6—Bus Size Specification for Area 3 (CS3) (A3SZ1, A3SZ0). Effective onl
ordinary space is set.

Bit 7: A3SZ1	Bit 6: A3SZ0	Description
0	0	Reserved (do not set)
	1	Byte (8-bit) size
1	0	Word (16-bit) size
	1	Longword (32-bit) size (Ir

Bits 3 and 2—Bus Size Specification for Area 1 (CS1) (A1SZ1, A1SZ0)

Bit 3: A1SZ1	Bit 2: A1SZ0	Description
0	0	Reserved (do not set)
	1	Byte (8-bit) size
1	0	Word (16-bit) size
	1	Longword (32-bit) size (

Bits 1 and 0—Reserved: These bits are always read as 0. The write value should alwa

7.2.3 Bus Control Register 3 (BCR3)

Bit:	15	14	13	12	11	10	9
	—	—	—	—	A4LW2	AHLW2	A1LW
Initial value:	0	0	0	0	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	DSWW1	DSWW0	—	—	—	BASEL	EDO
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W

Initialize the BASEL, EDO, and BWE bits after a power-on reset and do not write to t
thereafter. To change other bits by writing to them, write the same value as they are in
Do not access any space other than CS0 until the register initialization ends.

Bits 15 to 12—Reserved bits: These bits are always read as 0. The write value should

Bits 11 to 8—Long Wait Specification for Areas 0 to 4 (AnLW2): When the basic me
interface setting is made for CS n, from 3 to 14 wait cycles are inserted in CS n access
according to the combination with the long wait specification bits (AnLW1 and AnLW

	1	1 wait
1	0	2 waits
	1	Reserved (do not set)

Bits 5 to 3—Reserved bits: These bits are always read as 0. The write value should alw

Bit 2—Number of Banks Specification when Using 64M Synchronous DRAM (BASEI
64M synchronous DRAM is specified by AMX2 to AMX0 in MCR, the number of ban
specified.

Bit 2: BASEL	Description	
0	4 banks	(Ir
1	2 banks	

Bit 1—EDO Mode Specification (EDO): Enables EDO mode to be specified when DRA
specified for CS3 space.

Bit 1: EDO	Description	
0	High-speed page mode	(Ir
1	EDO mode	

Bit 0—Synchronous DRAM Burst Write Specification (BWE): Enables burst write mo
specified when synchronous DRAM is specified for CS2 or CS3 space.

Bit 0: BWE	Description	
0	Single write mode	(Ir
1	Burst write mode	

	W31	W30	W21	W20	W11	W10	W01
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Do not access a space other than CS0 until the settings for register initialization are co

Bits 15 to 8—Idles between Cycles for Areas 3 to 0 (IW31 to IW00): These bits speci
cycles inserted between consecutive accesses to different CS spaces. Idles are used to
conflict between ROM or the like, which is slow to turn the read buffer off, and fast m
I/O interfaces. Even when access is to the same space, idle cycles must be inserted wh
access is followed immediately by a write access. The idle cycles to be inserted compl
specification for the previously accessed space. The set values below show the minim
of idle cycles; more cycles than indicated by the Idles between Cycles setting may act
inserted.

IW31, IW21, IW11, IW01	IW30, IW20, IW10, IW00	Description
0	0	No idle cycle
	1	One idle cycle inserted
1	0	Two idle cycles inserted (
	1	Four idle cycles inserted

	1	Complies with the long wait speci bus control register 1, 3 (BCR1, E External wait input is enabled (Ir

- When CS3 is DRAM, the number of $\overline{CAS}$ assert cycles is specified by wait control and W30

Bit 7: W31	Bit 6: W30	Description
0	0	1 cycle
	1	2 cycles
1	0	3 cycles
	1	Reserved (do not set)

When external wait mask bit A3WM in WCR2 is 0 and the number of $\overline{CAS}$ assert c to 2 or more, external wait input is enabled.

- When CS2 or CS3 is synchronous DRAM, CAS latency is specified by wait control and W30, and W21 and W20, respectively

W31, W21	W30, W20	Description
0	0	1 cycle
	1	2 cycles
1	0	3 cycles
	1	4 cycles (Ir

With synchronous DRAM, external wait input is ignored regardless of any setting.

	—	—	—	—	IW41	IW40	W41
Initial value:	0	0	0	0	1	0	1
R/W:	R	R	R	R	R/W	R/W	R/W

Bits 15 and 14—Number of External Waits Specification for Area 4 (A4WD1, A4WD
bits specify the number of cycles between acceptance of CS4 space external wait nega
or $\overline{\text{WEn}}$ negation.

Bit 15: A4WD1	Bit 14: A4WD0	Description
0	0	1 cycle (
	1	2 cycles
1	0	4 cycles
	1	Reserved (do not set)

Bit 13—Reserved bit. This bit is always read as 0. The write value should always be 0

Bits 12 to 8—External Wait Mask Specification for Areas 0 to 4 (A4WM to A0WM):
enable waits to be masked for CS spaces 0 to 4. When a value other than 00 is set in th
control bits for CS spaces 0 to 4 (W41 to W00), external wait input can be enabled, bu
input can be masked by setting these bits to 1. With synchronous DRAM, external wai
ignored regardless of the settings.

1	Don’t care	Don’t care	External wait input ignored

Bits 7 to 4—Reserved bits: These bits are always read as 0. The write value should alw

Bits 3 and 2—Idles between Cycles for Area 4 (IW41, IW40): These bits specify idle c
inserted between cycles in CS4 in the same way as for CS 0 to 3. The set values below
minimum number of idle cycles; more cycles than indicated by the Idles between Cycle
may actually be inserted.

Bit 3: IW41	Bit 2: IW40	Description
0	0	No idle cycle
	1	One idle cycle inserted
1	0	Two idle cycles inserted (Ir
	1	Four idle cycles inserted

Bits 1 and 0—Wait Control for Area 4 (W41, W40): These bits specify waits for CS4 i
way as for areas 0 to 3.

Bit 1: W41	Bit 0: W40	Description
0	0	No wait. External wait input disabled without wa
	1	One wait. External wait input enabled with one
1	0	Two waits. External wait input enabled with two
	1	Complies with the long wait specification of bus registers 1 and 3 (BCR1, BCR3). External wait enabled (Ir

	A3SHW1	A3SHW0	A2SHW1	A2SHW0	A1SHW1	A1SHW0	A0SHW
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 14—Reserved bits: These bits are always read as 0. The write value shoul
0.

Bits 13 to 11—CS4 Address/$\overline{\text{CS4}}$ to $\overline{\text{RD}}$/$\overline{\text{WEn}}$ Assertion (A4SW2 to A4SW0): These
the number of cycles from address/$\overline{\text{CS4}}$ output to $\overline{\text{RD}}$/$\overline{\text{WEn}}$ assertion for the CS4 spac

Bit 13: A4SW2	Bit 12: A4SW1	Bit 11: A4SW0	Description
0	0	0	0.5 cycles
		1	1.5 cycle
	1	0	3.5 cycles
		1	5.5 cycles
1	0	0	7.5 cycles
		1	Reserved (do not set)
	1	0	Reserved (do not set)
		1	Reserved (do not set)

Bit 10—Reserved bit: This bit is always read as 0. The write value should always be 0

Bits 9 and 8—Area 4 $\overline{\text{RD}}$/$\overline{\text{WEn}}$ Negation to Address/$\overline{\text{CS4}}$ Hold (A4HW1, A4HW0): T
specify the number of cycles from $\overline{\text{RD}}$/$\overline{\text{WEn}}$ negation to address/$\overline{\text{CS4}}$ hold for the CS4

Bit 9: A4HW1	Bit 8: A4HW0	Description
0	0	0.5 cycle, $\overline{\text{CS4}}$ hold cycle = 0 cycles
	1	1.5 cycle, $\overline{\text{CS4}}$ hold cycle = 1 cycle
1	0	3.5 cycle, $\overline{\text{CS4}}$ hold cycle = 3 cycles
	1	5.5 cycle, $\overline{\text{CS4}}$ hold cycle = 5 cycles

	1	1.5 cycle, $\overline{CSn}$* hold cycle = 1 cycle
1	0	2.5 cycle, $\overline{CSn}$* hold cycle = 2 cycles
	1	Reserved (do not set)

Note: * n = 0 to 3

7.2.7 Individual Memory Control Register (MCR)

Bit:	15	14	13	12	11	10	9
	TRP0	RCD0	TRWL0	TRAS1	TRAS0	BE	RASD
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	AMX2	SZ	AMX1	AMX0	RFSH	RMODE	TRP1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The TRP1, TRP0, RCD1, RCD0, TRWL1, TRWL0, TRAS1, TRAS0, BE, RASD, AM
AMX0 and SZ bits are initialized after a power-on reset. Do not write to them thereafte
writing to them, write the same values as they are initialized to. Do not access CS2 or C
register initialization is completed.

Bits 1 and 15—RAS Precharge Time (TRP1, TRP0): When DRAM is connected, speci
minimum number of cycles after $\overline{RAS}$ is negated before the next assert. When synchron
DRAM is connected, specifies the minimum number of cycles after precharge until a ba
command is output. See section 7.5, Synchronous DRAM Interface, for details.

- For synchronous DRAM interface

Bit 1: TRP1	Bit 15: TRP0	Description
0	0	1 cycle
	1	2 cycles
1	0	3 cycles
	1	4 cycles

Bits 0 and 14—$\overline{\text{RAS}}$-$\overline{\text{CAS}}$ Delay (RCD1, RCD0): When DRAM is connected, specifi
number of cycles after $\overline{\text{RAS}}$ is asserted before $\overline{\text{CAS}}$ is asserted. When synchronous DR
connected, specifies the number of cycles after a bank active (ACTV) command is iss
read or write command (READ, READA, WRIT, WRITA) is issued.

Bit 0: RCD1	Bit 14: RCD0	Description
0	0	1 cycle
	1	2 cycles
1	0	3 cycles
	1	Reserved (do not set)

Bits 8 and 13—Write-Precharge Delay (TRWL1, TRWL0): When the synchronous D
in the bank active mode, this bit specifies the number of cycles after the write cycle be
start-up of the auto-precharge. Based on this number of cycles, the timing at which the
command can be issued is calculated within the bus controller. In bank active mode, th
specifies the number of cycles before the precharge command is issued after the write
issued. This bit is ignored when memory other than synchronous DRAM is connected

Bit 8: TRWL1	Bit 13: TRWL0	Description
0	0	1 cycle
	1	2 cycles
1	0	3 cycles
	1	Reserved (do not set)

	1	5 cycles

After an auto-refresh command is issued, a bank active command is not issued for TRA
regardless of the TRP bit setting. For synchronous DRAM, there is no $\overline{\text{RAS}}$ assertion p
there is a limit for the time from the issue of a refresh command until the next access. T
set to observe this limit. Commands are not issued for TRAS cycles when self-refresh i

- For synchronous DRAM interface

Bit 12: TRAS1	Bit 11: TRAS0	Description
0	0	3 cycles (Ir
	1	4 cycles
1	0	6 cycles
	1	9 cycles

Bit 10—Burst Enable (BE)

Bit 10: BE	Description
0	Burst disabled (Ir
1	High-speed page mode during DRAM and ED0 interfacing is enable Burst access conditions are as follows: • Longword access, cache fill access, or DMAC 16-byte transfer, bus width • Cache fill access or DMAC 16-byte transfer, with 32-bit bus widt During synchronous DRAM access, burst operation is always enabl regardless of this bit

	performs writes other than to DRAM, see section 7.6.5, Burst Acce
	For synchronous DRAM, access ends in the bank active state. Thi valid for area 3. When area 2 is synchronous DRAM, the mode is a precharge

Bits 7, 5, and 4—Address Multiplex (AMX2 to AMX0)

- For DRAM interface

Bit 7: AMX2	Bit 5: AMX1	Bit 4: AMX0	Description
0	0	0	8-bit column address DRAM
		1	9-bit column address DRAM
	1	0	10-bit column address DRAM
		1	11-bit column address DRAM
1	0	0	Reserved (do not set)
		1	Reserved (do not set)
	1	0	Reserved (do not set)
		1	Reserved (do not set)

1	0	0	64-Mbit SDRAM (4M × 16 bits), 128-M SDRAM (4M × 32 bits)*3
		1	64-Mbit SDRAM (8M × 8 bits)*1, 128-M SDRAM (8M × 16 bits)*1*4, 256-Mbit S (8M × 32 bits)*1*5
	1	0	Reserved (do not set)
		1	2-Mbit SDRAM (128k × 16 bits)

Notes: 1. When SZ bit in MCR is 0 (16-bit bus width), these settings are reserved and be made.
2. See figure 7.34, 64-Mbit Synchronous DRAM (2 Mwords × 32 Bits) Connecti Example, for the method of connection to a 64-Mbit SDRAM (2M × 32 bits).
3. See figure 7.35 for the method of connection to a 128-Mbit SDRAM (4M × 32
4. Connect a 128-Mbit SDRAM with (8M × 16 bits) through a 32-bit bus as show 7.36.
5. See figure 7.37 for the method of connection to a 256-Mbit SDRAM (8M × 32

Bit 6—Memory Data Size (SZ): For synchronous DRAM and DRAM space, the data b BCR2 is ignored in favor of the specification of this bit.

Bit 6: SZ	Description
0	Word (16 bits) (In
1	Longword (32 bits)

refresh. When the RFSH bit is 0, do not set this bit to 1. When the RFSH bit is 1, self-
mode is entered immediately after the RMODE bit is set to 1. When the RFSH bit is 1
is 0, a $\overline{CAS}$-before-$\overline{RAS}$ refresh or auto-refresh is performed at the interval set in the 8
timer. When a refresh request occurs during an external area access, the refresh is perf
the access cycle is completed. When set for self-refresh, self-refresh mode is entered i
unless the chip is in the middle of a synchronous DRAM area access, in which case se
mode is entered when the access ends. Refresh requests from the interval timer are ign
self-refresh.

Bit 2: RMODE	Description
0	Normal refresh
1	Self-refresh

	CMF	CMIE	CKS2	CKS1	CKS0	RRC2	RRC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always

Bit 7—Compare Match Flag (CMF): This status flag, which indicates that the values of and RTCOR match, is set/cleared under the following conditions:

Bit 7: CMF	**Description**
0	[Clearing condition] After RTCSR is read when CMF is 1, 0 is written in CMF
1	[Setting condition] RTCNT = RTCOR

Bit 6—Compare Match Interrupt Enable (CMIE): Enables or disables an interrupt requ by the CMF bit of RTCSR when CMF is set to 1.

Bit 6: CMIE	**Description**
0	Interrupt request caused by CMF is disabled (Ir
1	Interrupt request caused by CMF is enabled

		1	Pϕ/1024
	1	0	Pϕ/2048
		1	Pϕ/4096

Bits 2 to 0—Refresh Count (RRC2 to RRC0): These bits specify the number of conse
refreshes to be performed when the refresh timer counter (RTCNT) and refresh time c
register (RTCOR) values match and a refresh request is issued.

Bit 2: RRC2	Bit 1: RRC1	Bit 0: RRC0	Description
0	0	0	1 refresh
		1	2 refreshes
	1	0	4 refreshes
		1	6 refreshes
1	0	0	8 refreshes
		1	Reserved (do not set)
	1	0	Reserved (do not set)
		1	Reserved (do not set)

7.2.9 Refresh Timer Counter (RTCNT)

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RTCOR is an 8-bit read/write register. The values of RTCOR and RTCNT are constant
compared. When the values correspond, the compare match flag (CMF) in RTCSR is s
RTCNT is cleared to 0. When the refresh bit (RFSH) in the individual memory control
(MCR) is set to 1, a refresh request signal occurs. The refresh request signal is held unt
operation is actually performed. If the refresh request is not processed before the next n
previous request becomes ineffective.

When the CMIE bit in RTCSR is set to 1, an interrupt request is sent to the controller b
match signal. The interrupt request is output continuously until the CMF bit in RTCSR
When the CMF bit clears, it only affects the interrupt; the refresh request is not cleared
operation. When a refresh is performed and refresh requests are counted using interrupt
can be set simultaneously with the interval timer interrupt.

Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always

mode pins for the CS0 space, or by setting BCR2 for the CS1 to CS4 spaces. However
width of devices connected to the respective spaces is specified statically, and the data
cannot be changed for each access cycle. Figures 7.2 to 7.4 show the relationship betw
data widths and access units.

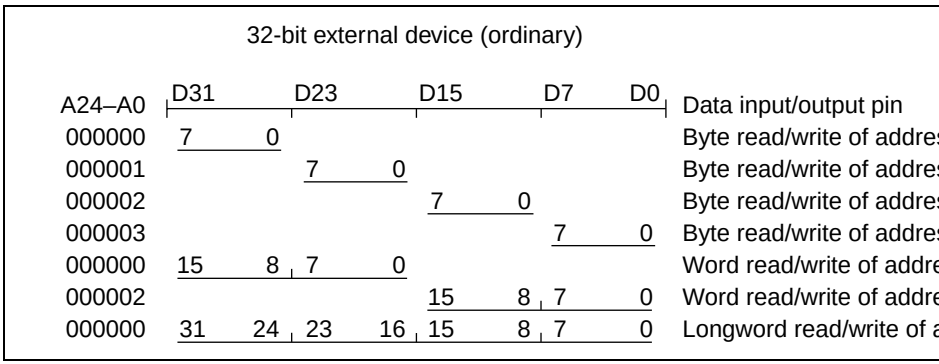


Figure 7.2 32-Bit External Devices and Their Access Units

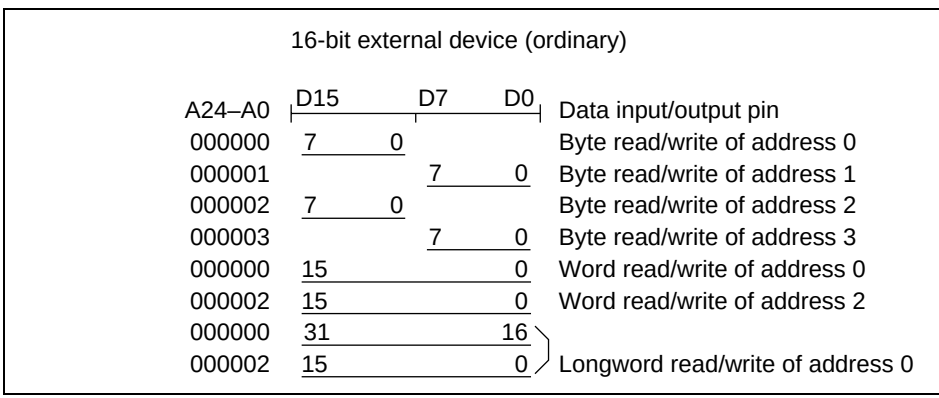


Figure 7.3 16-Bit External Devices and Their Access Units

000002	15	8	Word read/write of address 2
000003	7	0	
000000	31	24	Longword read/write of address 0
000001	23	16	
000002	15	8	
000003	7	0	

Figure 7.4 8-Bit External Devices and Their Access Units

7.3.2 Connection to Little-Endian Devices

The chip provides a conversion function in CS2, CS4 space for connection to and to ma compatibility with devices that use little-endian format (in which the LSB is the 0 posit byte data lineup). When the endian specification bit of BCR1 is set to 1, CS2, CS4 spac endian. The relationship between device data width and access unit for little-endian for shown in figures 7.5, 7.6, and 7.7. When sharing memory or the like with a little-endian master, the SH7615 connects D31 to D24 to the least significant byte (LSB) of the othe master and D7 to D0 to the most significant byte (MSB), when the bus width is 32 bits. width is 16 bits, the SH7615 connects D15 to D8 to the least significant byte of the othe master and D7 to D0 to the most significant byte.

Only data conversion is supported by this function. For this reason, be careful not to pla program code or constants in the CS2, CS4 space. When this function is used, make sur access unit is the same for writing and reading. For example, data written by longword should be read by longword access. If the read access unit is different from the write ac an incorrect value will be read.

000000	7 0	15 8	23 16	31 24	Longword read/write of a

Figure 7.5 32-Bit External Devices and Their Access Units

16-bit external device (little-endian)

A24–A0	D15		D7	D0	Data input/output pin
000000	7	0			Byte read/write of address 0
000001			7	0	Byte read/write of address 1
000002	7	0			Byte read/write of address 2
000003			7	0	Byte read/write of address 3
000000	7	0	15	8	Word read/write of address 0
000002	7	0	15	8	Word read/write of address 2
000000	7	0	15	8	Longword read/write of address 0
000002	23	16	31	24	

Figure 7.6 16-Bit External Devices and Their Access Units

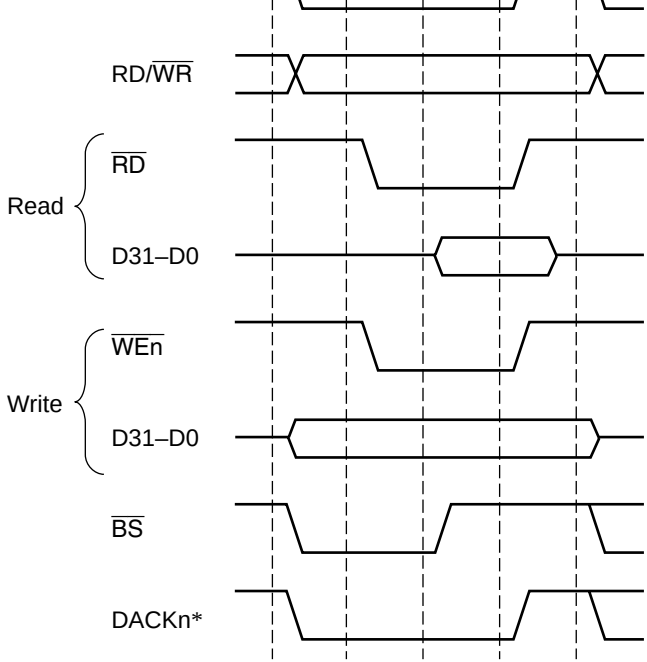
8-bit external device (little-endian)

A24–A0	D7	D0	Data input/output pin
000000	7	0	Byte read/write of address 0
000001	7	0	Byte read/write of address 1
000002	7	0	Byte read/write of address 2
000003	7	0	Byte read/write of address 3
000000	7	0	Word read/write of address 0
000001	15	8	
000002	7	0	Word read/write of address 2
000003	15	8	
000000	7	0	Longword read/write of address 0
000001	15	8	
000002	23	16	
000003	31	24	

Figure 7.7 8-Bit External Devices and Their Access Units

The access size is not specified during a read. The correct access start address will be o
LSB of the address, but since no access size is specified, the read will always be 32 bits
devices and 16 bits for 16-bit devices. For writes, only the $\overline{WE}$ signal of the byte that w
written is asserted. For 32-bit devices, $\overline{WE3}$ specifies writing to a 4n address and $\overline{WE0}$
writing to a 4n+3 address. For 16-bit devices, $\overline{WE1}$ specifies writing to a 2n address an
specifies writing to a 2n+1 address. For 8-bit devices, only $\overline{WE0}$ is used.

When data buses are provided with buffers, the $\overline{RD}$ signal must be used for data output
direction. When RD/$\overline{WR}$ signals do not perform accesses, the chip stays in read status,
danger of conflicts occurring with output when this is used to control the external data b

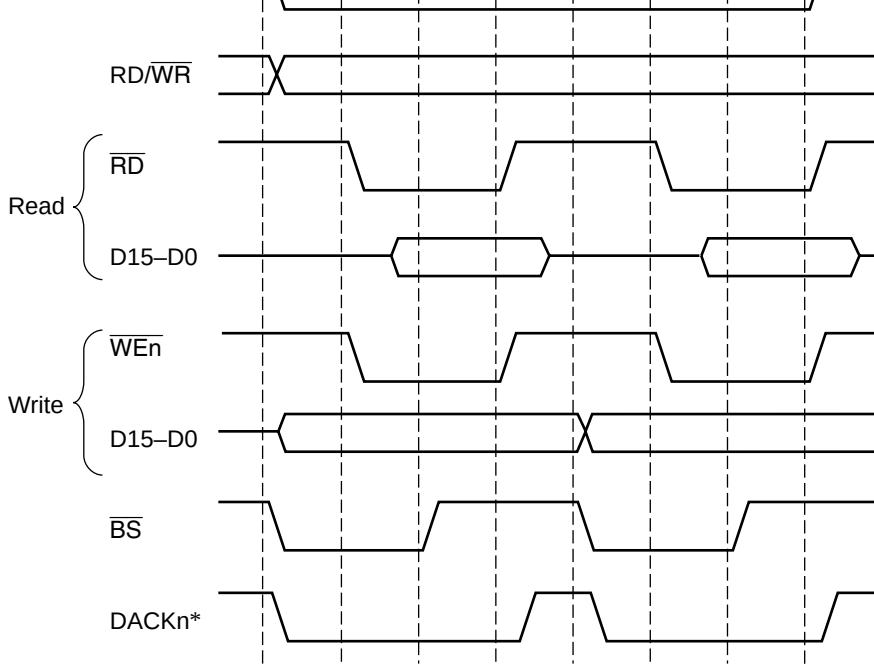


Note: * DACKn waveform when active-low is specified.

Figure 7.8 Basic Timing of Ordinary Space Access

When making a word or longword access with an 8-bit bus width, or a longword acces
bit bus width, the bus state controller performs multiple accesses.

When clock ratio except Iϕ:Eϕ = 1:1, the basic timing shown in figure 7.8 is repeated,
clock ratio Iϕ:Eϕ is 1:1, burst access with no $\overline{\text{CSn}}$ negate period is performed as shown
7.9.



Note: * DACKn waveform when active-low is specified.

Figure 7.9 Timing of Longword Access in Ordinary Space Using 16-Bit Bus (Clock Ratio Iϕ:Eϕ = 1:1)

Figure 7.10 shows an example of 32-bit data width SRAM connection, figure 7.11 an e
16-bit data width SRAM connection, and figure 7.12 an example of 8-bit data width SR
connection.

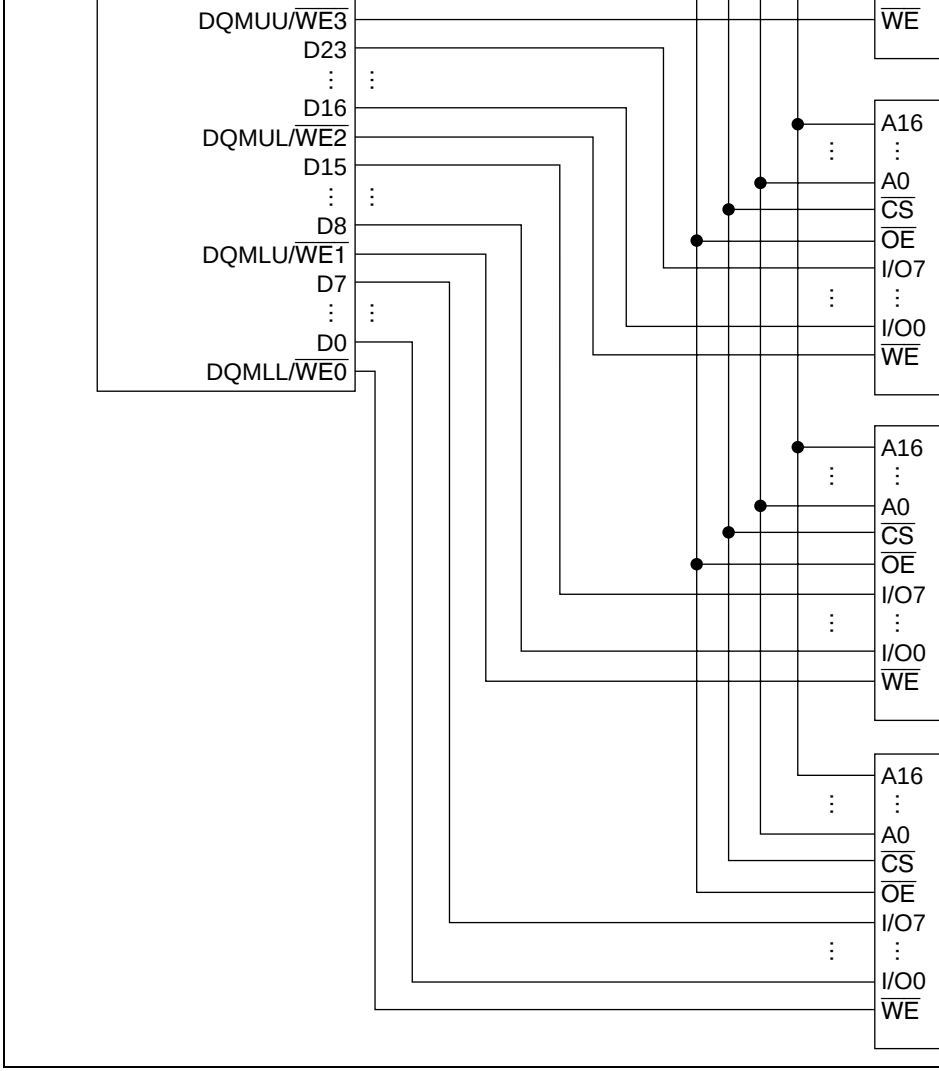


Figure 7.10 Example of 32-Bit Data Width SRAM Connection

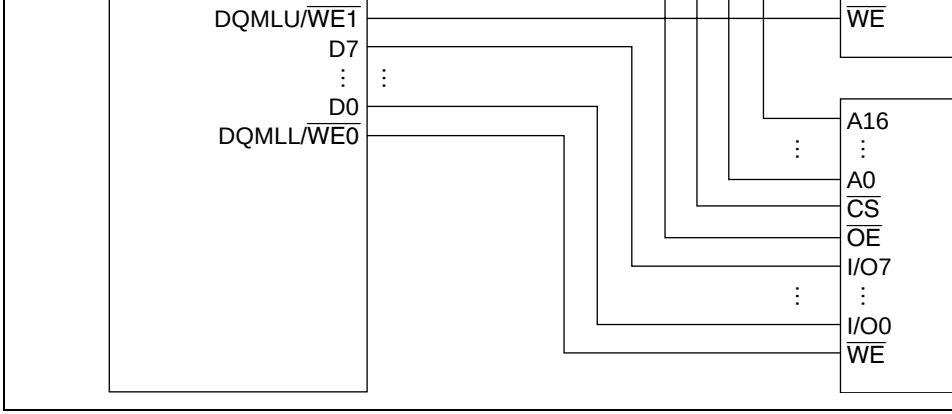


Figure 7.11 Example of 16-Bit Data Width SRAM Connection

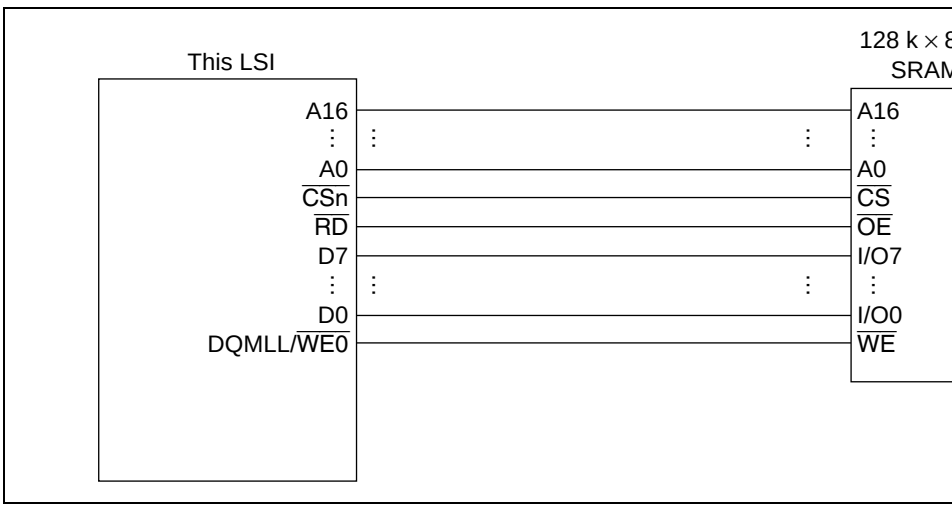


Figure 7.12 Example of 8-Bit Data Width SRAM Connection

and BCR3, a Tw cycle is inserted as a wait cycle as long as the number of specified cy
wait timing for ordinary access space shown in figure 7.13. The names of the control l
specify Tw for each CS space are shown in table 7.5.

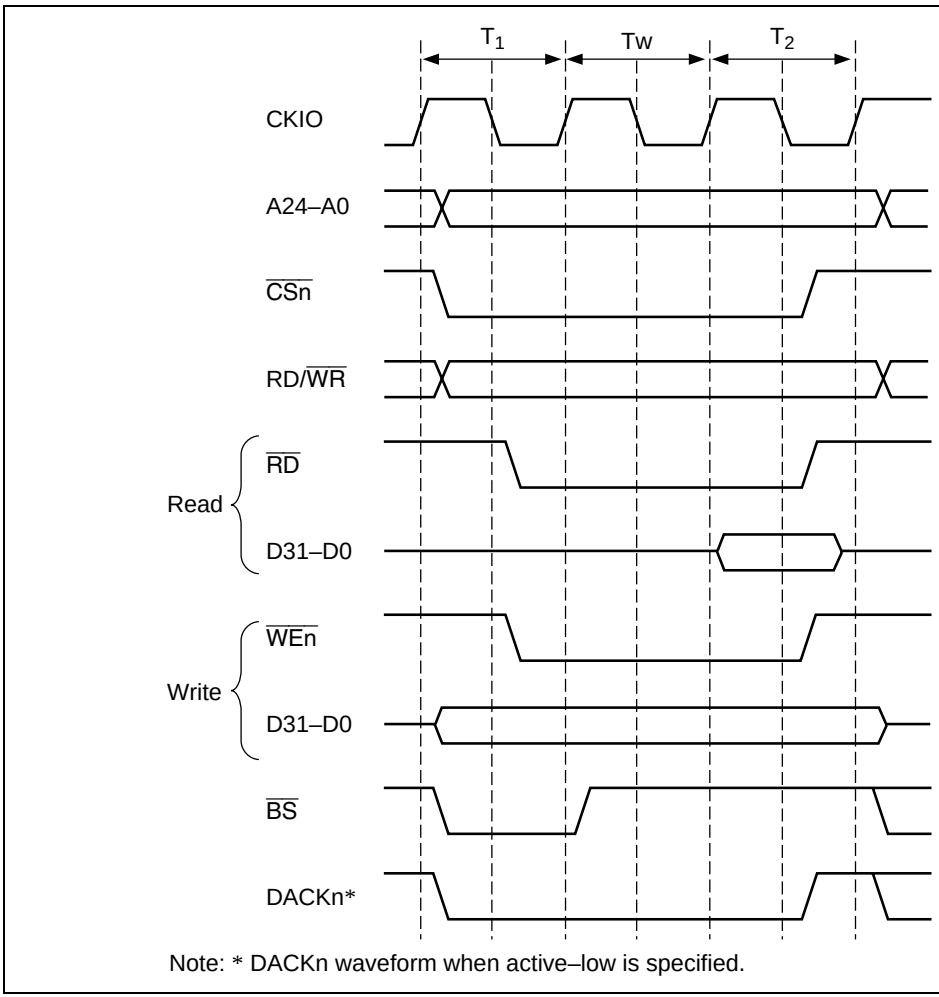
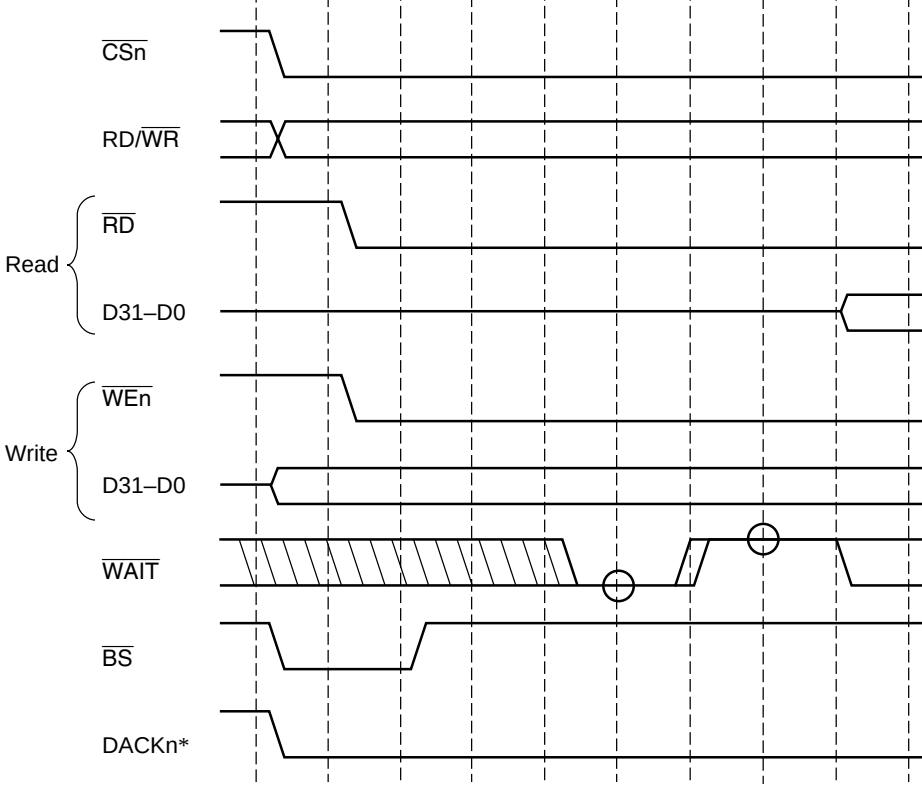


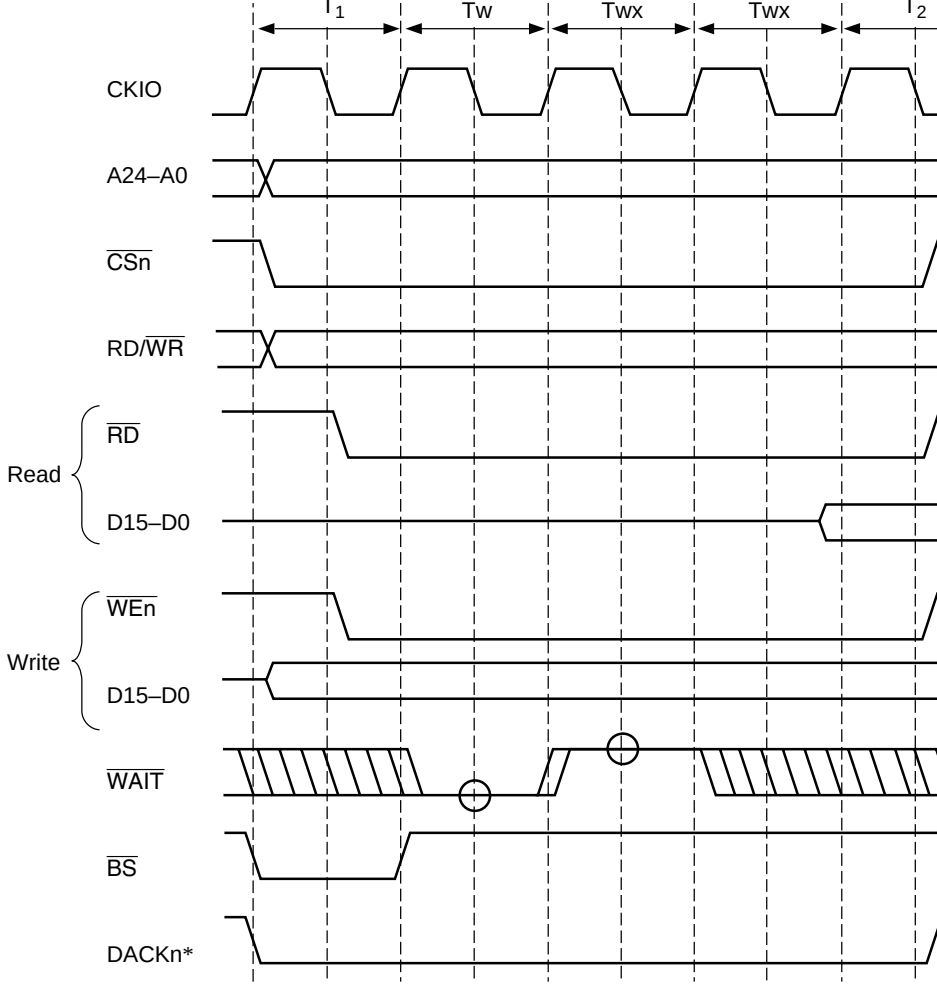
Figure 7.13 Wait Timing of Ordinary Space Access (Software Wait Or

When a wait is specified by software using WCR1 and WCR2 (Wn1, Wn0), and the ex
mask bit (AnWM) is cleared to 0 in WCR2, the wait input $\overline{\text{WAIT}}$ signal from outside is
Figure 7.14 shows $\overline{\text{WAIT}}$ signal sampling. A 2-cycle wait is specified as a software wa
sampling is performed when the Tw state shifts to the T_2 state, so there is no effect ever
$\overline{\text{WAIT}}$ signal is asserted in the T_1 cycle or the first Tw cycle. The $\overline{\text{WAIT}}$ signal is samp
clock fall.



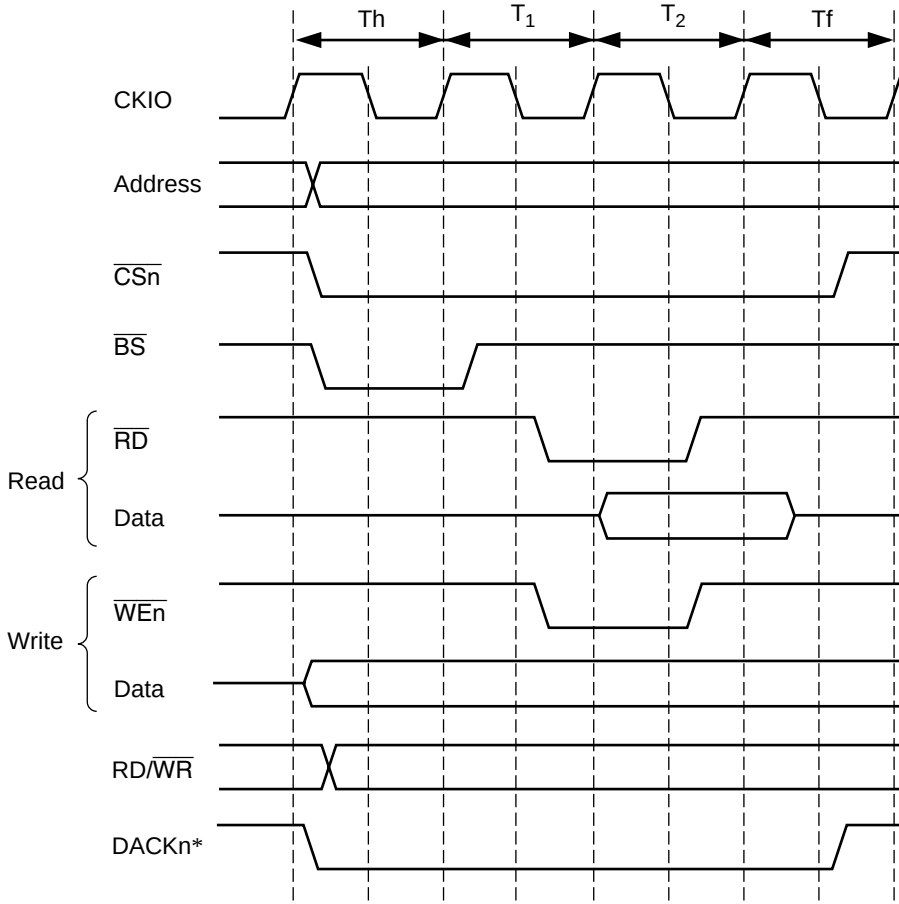
Note: * DACKn waveform when active-low is specified.

Figure 7.14 Wait State Timing of Ordinary Space Access (Wait States from WAIT Signal)



Note: * DACKn waveform when active-low is specified.

Figure 7.15 Wait State Timing of Ordinary Space in CS4 Space



Note: * DACKn waveform when active-low is specified.

Figure 7.16 $\overline{CS}$ Assertion Period Extension Function

7.5.1 Synchronous DRAM Direct Connection

Seven kinds of synchronous DRAM can be connected: 2-Mbit (128k × 16), 4-Mbit (25
16-Mbit (1M × 16, 2M × 8, and 4M × 4), and 64-Mbit (4M × 16 and 8M × 8). This chi
64-Mbit synchronous DRAMs internally divided into two or four banks, and other sync
DRAMs internally divided into two banks. Since synchronous DRAM can be selected b
signal, CS2 and CS3 spaces can be connected using a common $\overline{\text{RAS}}$ or other control si
the memory enable bits for DRAM and other memory (DRAM2 to DRAM0) in BCR1
001, CS2 is ordinary space and CS3 is synchronous DRAM space. When the DRAM2 t
bits are set to 100, CS2 is synchronous DRAM space and CS3 is ordinary space. When
set to 101, both CS2 and CS3 are synchronous DRAM spaces.

Supported synchronous DRAM operating modes are burst read/single write mode (initi
and burst read/burst write mode. The burst length depends on the data bus width, comp
bursts for a 16-bit width, and 4 bursts for a 32-bit width. The data bus width is specifie
bit in MCR. Burst operation is always performed, so the burst enable (BE) bit in MCR
Switching to burst write mode is performed by means of the BWE bit in BCR3.

Control signals for directly connecting synchronous DRAM are the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$/$\overline{\text{OE}}$, RD
or $\overline{\text{CS3}}$, DQMUU, DQMUL, DQMLU, DQMLL, and CKE signals. Signals other than
$\overline{\text{CS3}}$ are common to every area, and signals other than CKE are valid and fetched only
or $\overline{\text{CS3}}$ is true. Therefore, synchronous DRAM can be connected in parallel in multiple
is negated (to the low level) only when a self-refresh is performed; otherwise it is alway
(to the high level).

Commands can be specified for synchronous DRAM using the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$/$\overline{\text{OE}}$, RD/$\overline{\text{WR}}$
certain address signals. These commands are NOP, auto-refresh (REF), self-refresh (SE
bank precharge (PALL), specific bank precharge (PRE), row address strobe/bank active
read (READ), read with precharge (READA), write (WRIT), write with precharge (WR
mode register write (MRS).

Bytes are specified using DQMUU, DQMUL, DQMLU, and DQMLL. The read/write i
performed on the byte whose DQM is low. For 32-bit data, DQMUU specifies 4n addre
and DQMLL specifies 4n + 3 address access. For 16-bit data, only DQMLU and DQM

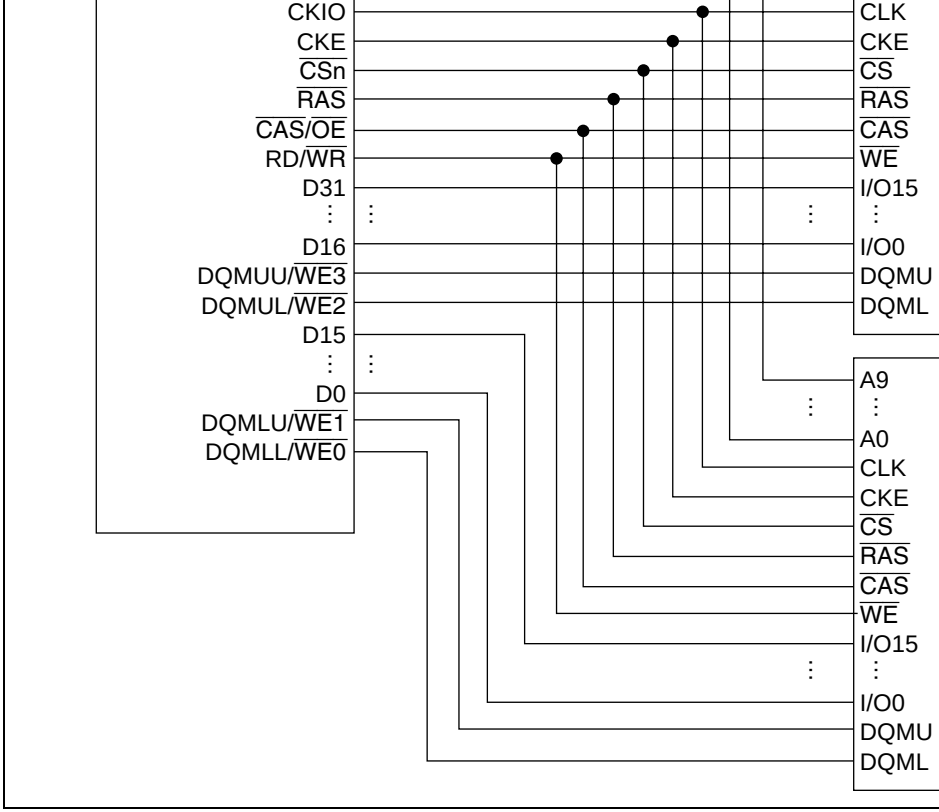


Figure 7.17 Synchronous DRAM 32-bit Device Connection

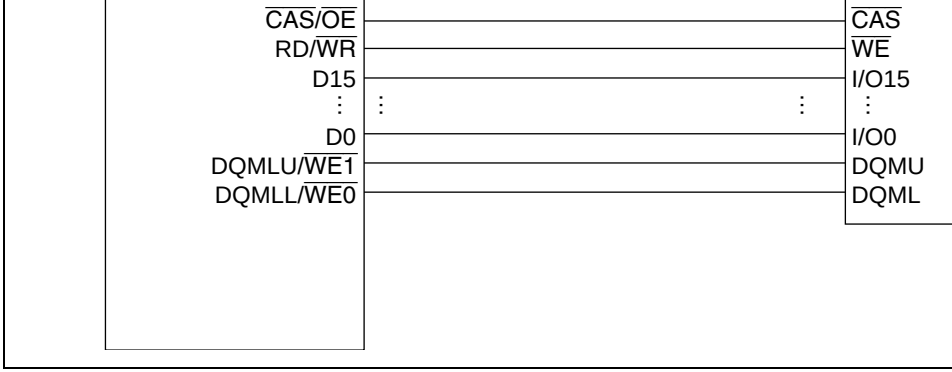


Figure 7.18 Synchronous DRAM 16-bit Device Connection

7.5.2 Address Multiplexing

Addresses are multiplexed according to the MCR's address multiplex specification bits AMX0 and size specification bit SZ so that synchronous DRAMs can be connected to t directly without an external multiplex circuit. Table 7.6 shows the relationship between multiplex specification bits and bit output to the address pins.

A24 to A16 always output the original value regardless of multiplexing.

When SZ = 0, the data width on the synchronous DRAM side is 16 bits and the LSB of device's address pins (A0) specifies word address. The A0 pin of the synchronous DRA connected to the A1 pin of the SH7615, the rest of the connection proceeding in the sar beginning with the A1 pin to the A2 pin.

When SZ = 1, the data width on the synchronous DRAM side is 32 bits and the LSB of device's address pins (A0) specifies longword address. The A0 pin of the synchronous thus connected to the A2 pin of the SH7615, the rest of the connection proceeding in th order, beginning with the A1 pin to the A3 pin.

				address							
				Row address	A10–A17	A18	A19	A20	A21	A22*2	
1	0	1	0	Column address	A1–A8	A9	A10	A11	L/H*1	A23*2	
				Row address	A11–A18	A19	A20	A21	A22	A23*2	
1	0	1	1	Column address	A1–A8	A9	L/H*1	A19*2	A12	A13	
				Row address	A9–A16	A17	A18	A19*2	A20	A21	
1	1	0	0	Column address	A1–A8	A9	A10	A11	L/H*1	A13	
				Row address	A9–A16	A17	A18	A19	A20	A21	
1	1	0	1	Column address	A1–A8	A9	A10	A11	L/H*1	A13	
				Row address	A10–A17	A18	A19	A20	A21	A22	
1	1	1	1	Column address	A1–A8	A9	L/H*1	A18*2	A12	A13	
				Row address	A9–A16	A17	A17	A18*2	A20	A21	
0	0	0	0	Column address	A1–A8	A9	A10	L/H*1	A20*2	A13	
				Row address	A9–A16	A17	A18	A19	A20*2	A21	

				Row address	A9–A16	A17	A18	A19	A20	A21	A
0	1	1	1	Column address	A1–A8	L/H[*1]	A17[*2]	A11	A12	A13	A
				Row address	A9–A16	A16	A17[*2]	A19	A20	A21	A

Notes: AMX2 to AMX0 setting 110 is reserved and must not be used. When SZ = 0, AM
AMX0 settings 001, 010, and 101 are also reserved and must not be used.

1. L/H is a bit used to specify commands. It is fixed at L or H according to the a mode.
2. Bank address specification.
3. Bank address specification when using four banks.

7.5.3 Burst Reads

Figure 7.19 (a) and (b) show the timing charts for burst reads. In the following example synchronous DRAMs of 256k × 16 bits are connected, the data width is 32 bits and the length is 4. After a Tr cycle that performs ACTV command output, a READA comman in the Tc cycle, read data is accepted in cycles Td1 to Td4, and the end of the read sequ waited for in the Tde cycle. One Tde cycle is issued when Iϕ:Eϕ ≠ 1:1, and two cycles = 1:1. Tap is a cycle for waiting for the completion of the auto-precharge based on the command within the synchronous DRAM. During this period, no new access command issued to the same bank. Accesses of the other bank of the synchronous DRAM by ano space are possible. Depending on the TRP1, TRP0 specification in MCR, the chip deter number of Tap cycles and does not issue a command to the same bank during that perio

Figure 7.19 (a) and (b) show examples of the basic cycle. Because a slower synchronou connected, setting WCR1 and MCR bits can extend the cycle. The number of cycles fro ACTV command output cycle Tr to the READA command output cycle Tc can be spec bits RCD1 and RCD0 in MCR. 00 specifies 1 cycle, 01 specifies 2 cycles, and 10 speci cycles. For 2 or 3 cycles, a NOP command issue cycle Trw for the synchronous DRAM between the Tr cycle and the Tc cycle. The number of cycles between the READA com output cycle Tc and the initial read data fetch cycle Td1 can be specified between 1 cyc cycles using the W21/W20 and W31/W30 bits in WCR1. The number of cycles at this

When the data width is 16 bits, 8 burst cycles are required for a 16-byte data transfer. T
fetch cycle goes from Td1 to Td8.

Synchronous DRAM CAS latency is up to 3 cycles, but the CAS latency of the bus sta
can be specified up to 4. This is so that circuits containing latches can be installed betw
synchronous DRAMs and the chip.

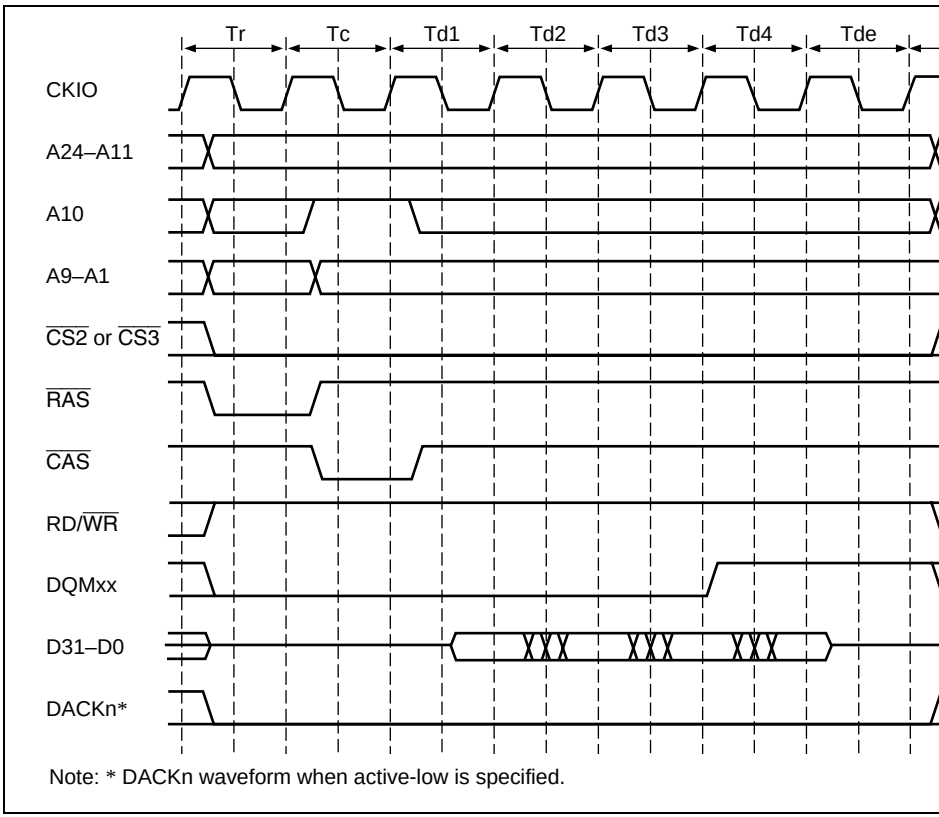
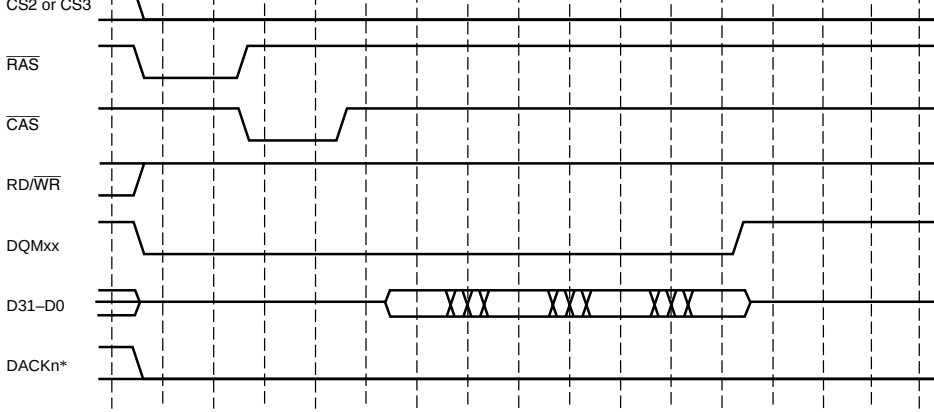
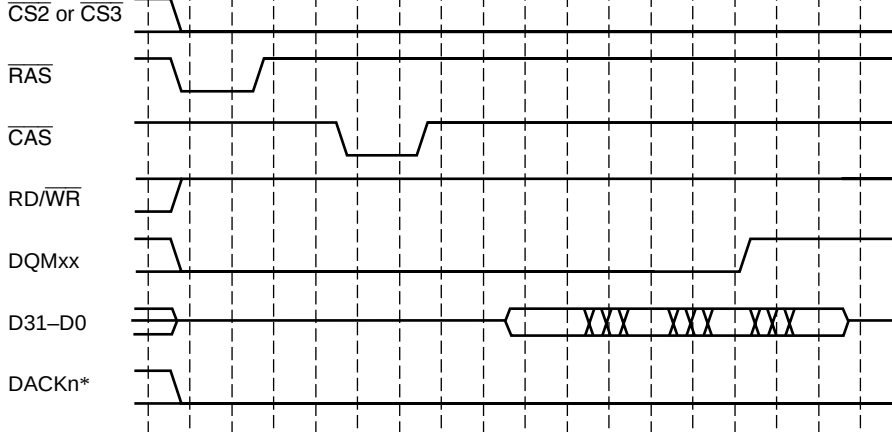


Figure 7.19 (a) Basic Burst Read Timing (Auto-Precharge) Except t_{Ecyc}:t_P



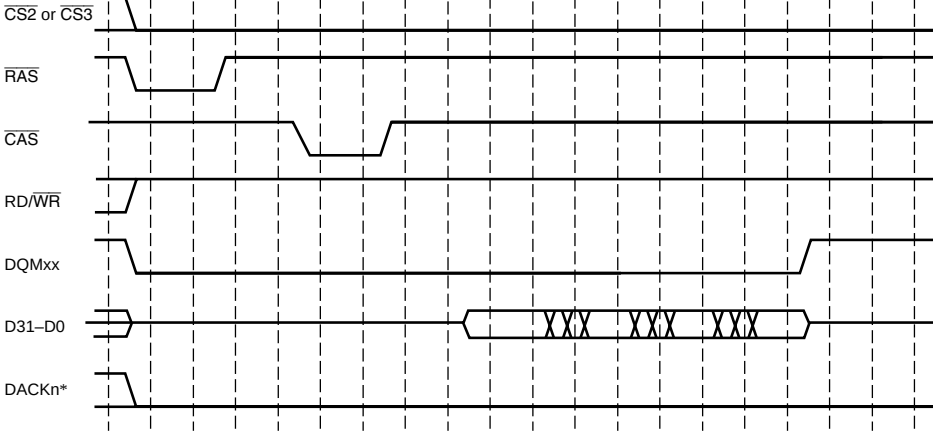
Note: * DACKn waveform when active-low is specified.

Figure 7.19 (b) Basic Burst Read Timing (Auto-Precharge) Iϕ:Eϕ = 1:1



Note: * DACKn waveform when active-low is specified.

Figure 7.20 (a) Burst Read Wait Specification Timing (Auto-Precharg
Except t_{Ecyc}:t_{Pcyc} 1:1



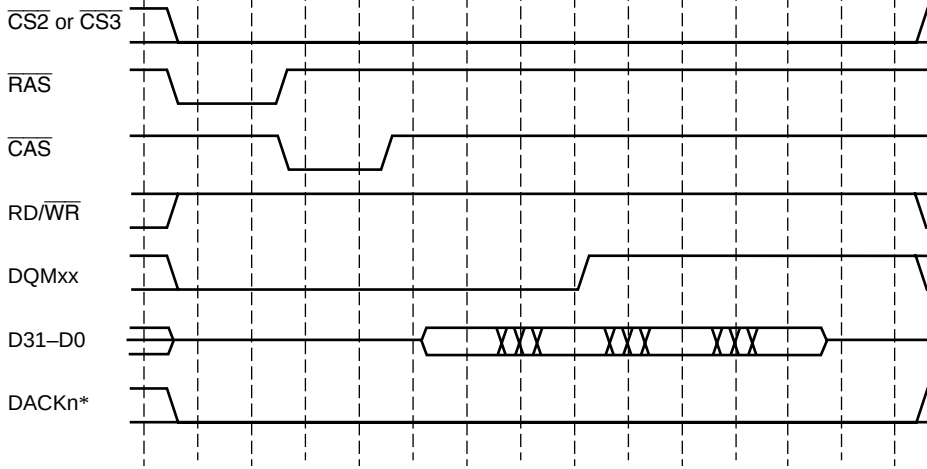
Note: * DACKn waveform when active-low is specified.

Figure 7.20 (b) Burst Read Wait Specification Timing (Auto-Precharge) Iϕ:E

the required data is received. To avoid data conflict, an empty read cycle is performed
Td4 after the required data is read in Td1 and the device waits for the end of synchron
operation.

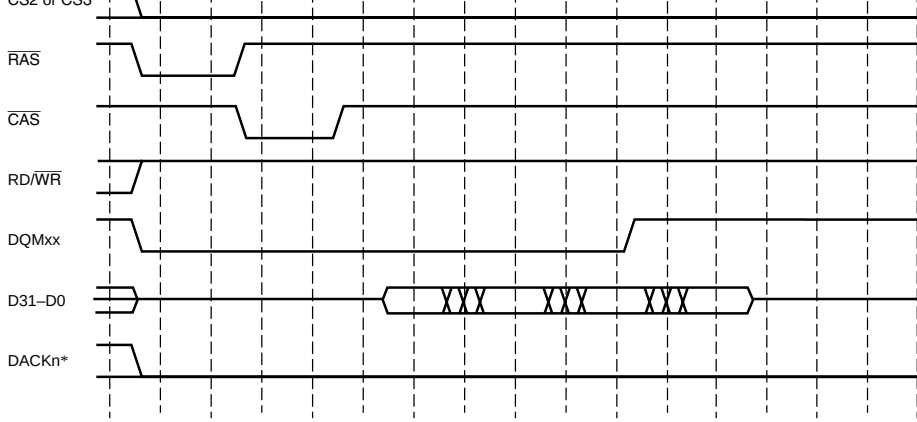
When the data width is 16 bits, the number of burst transfers during a read is 8. Data i
cache-through and other DMA read cycles only in the Td1 and Td2 cycles (of the 8 cy
Td1 to Td8) for longword accesses, and only in the Td1 cycle for word or byte accesse

Empty cycles tend to increase the memory access time, lower the program execution s
lower the DMA transfer speed, so it is important to avoid accessing unnecessary cache
areas and to use data structures that enable 16-byte unit transfers by placing data on 16
boundaries when performing DMA transfers that specify synchronous DRAM as the s



Note: * DACKn waveform when active-low is specified.

Figure 7.21 (a) Single Read Timing (Auto-Precharge) Except t_{Ecyc}:t_{Pcyc} 1

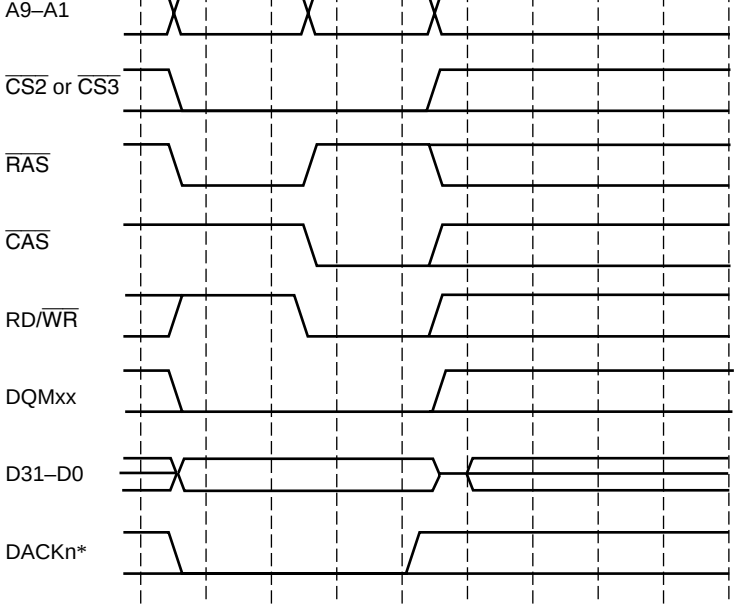


Note: * DACKn waveform when active-low is specified.

Figure 7.21 (b) Single Read Timing (Auto-Precharge) Iϕ:Eϕ = 1:1

7.5.5 Single Writes

Synchronous DRAM writes are executed as single writes or burst writes according to
specification by the BWE bit in BCR3. Figure 7.22 shows the basic timing chart for si
accesses. After the ACTV command Tr, a WRITA command is issued in Tc to perforn
precharge. In the write cycle, the write data is output simultaneously with the write co
When writing with an auto-precharge, the bank is precharged after the completion of t
command within the synchronous DRAM, so no command can be issued to that bank
precharge is completed. For that reason, besides a Tap cycle to wait for the precharge
accesses, a Trw1 cycle is added to wait until the precharge is started, and the issuing o
commands to the same bank is delayed during this period. The number of cycles in the
can be specified using the TRWL1 and TRWL0 bits in MCR.

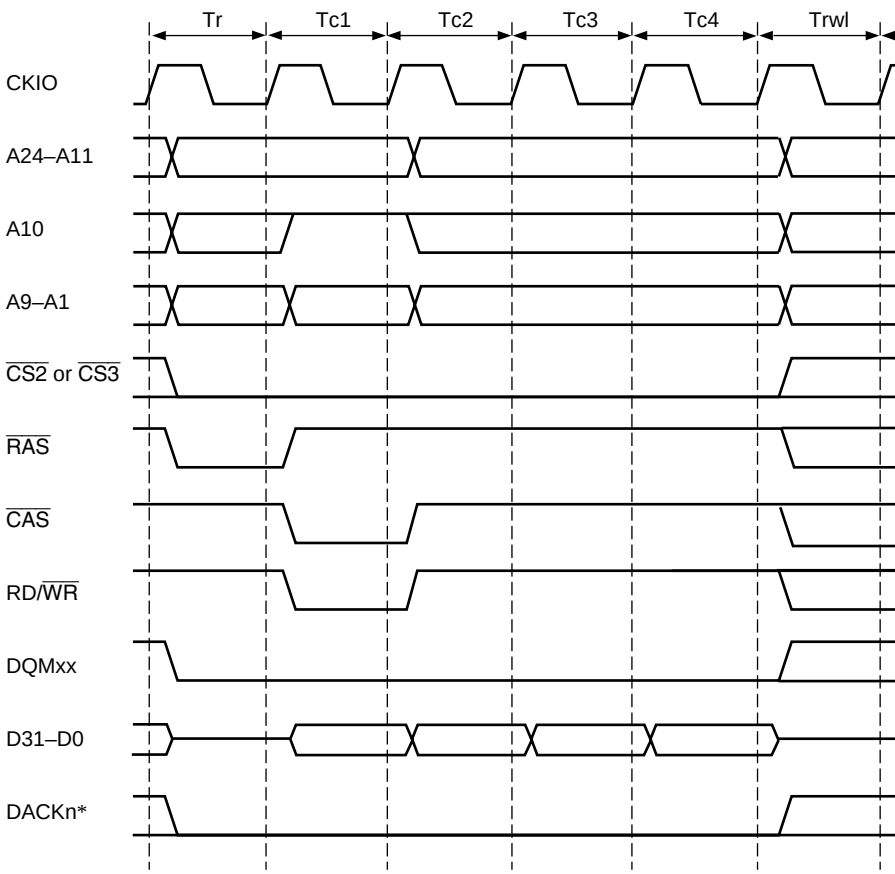


Note: * DACKn waveform when active-low is specified.

Figure 7.22 Basic Single Write Cycle Timing (Auto-Precharge)

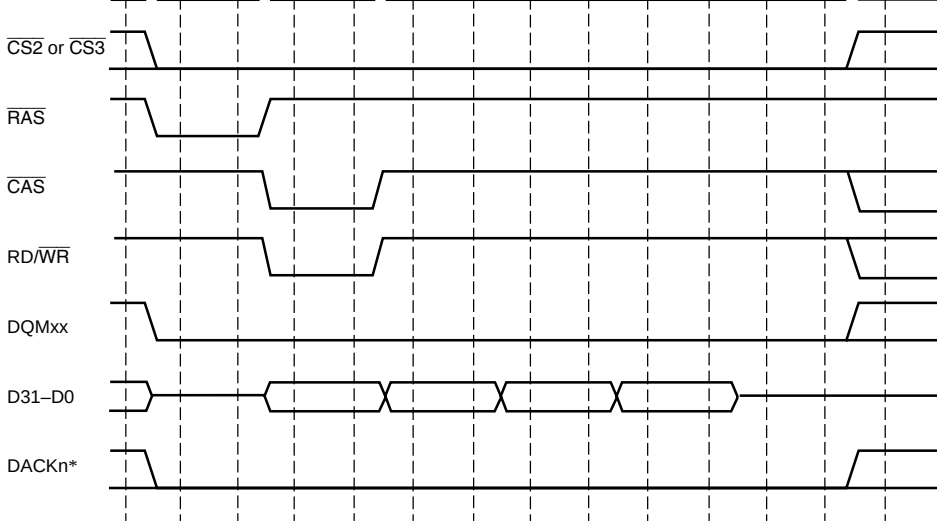
7.5.6 Burst Write Mode

Burst write mode can be selected by setting the BWE bit to 1 in BCR3. The basic timin
burst write access is shown in figure 7.23 (a) and (b). This example assumes a 32-bit bu
and a burst length of 4. In the burst write cycle, the WRITA command that performs au
precharge is issued in Tc1 following the ACTV command Tr cycle. The first 4 bytes of
are output simultaneously with the WRITA command in Tc1, and the remaining 12 byt
are output consecutively in Tc2, Tc3, and Tc4. In a write with auto-precharge, as with a
write, a Trw1 cycle that provides the waiting time until precharge is started is inserted a
of the write data, followed by a Tap cycle for the precharge wait in a write access. The
Tap cycles can be set respectively in MCR by bits TRWL1 and TRWL0, and bits TRP1
TRP0.



Note: * DACKn waveform when active-low is specified.

Figure 7.23 (a) Basic Burst Write Timing (Auto-Precharge) Except t_{Ecyc}:t_{E}



Note: * DACKn waveform when active-low is specified.

Figure 7.23 (b) Basic Burst Write Timing (Auto-Precharge) Iϕ:Eϕ = 1:

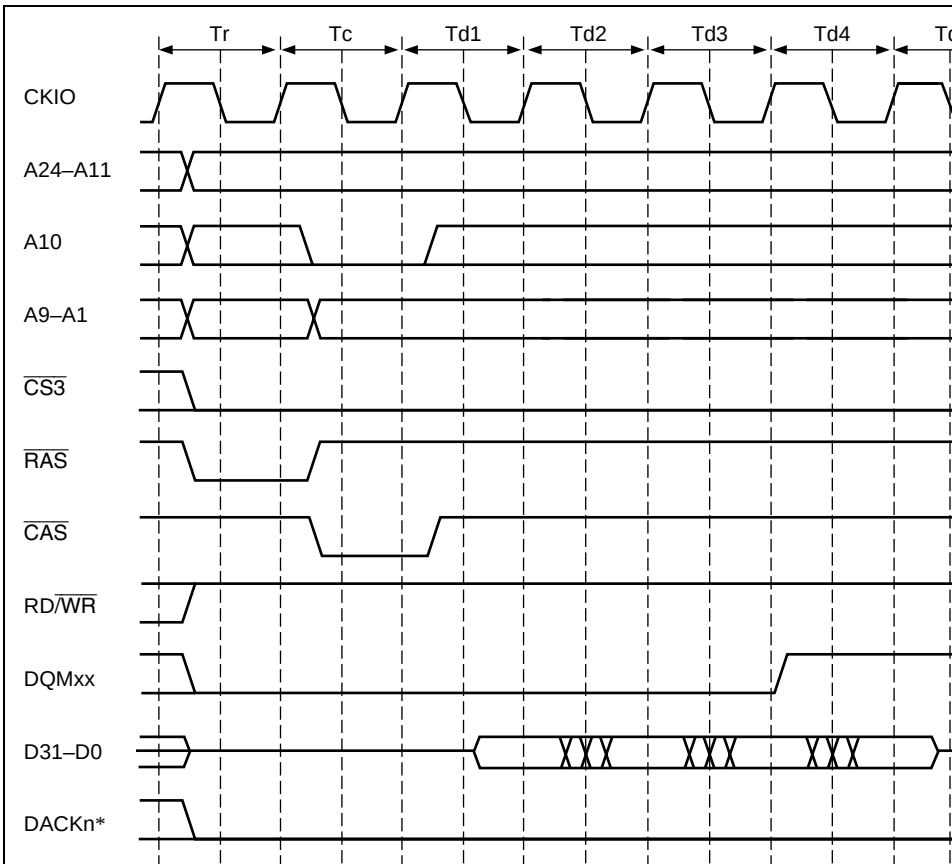
high-speed page mode. Synchronous DRAM is divided into two banks, so one row ad
can stay active. When the next access is to a different row address, a PRE command is
to precharge the bank, and access is performed by an ACTV command and READ or
command in order, after the precharge is completed. With successive accesses to diffe
addresses, the precharge is performed after the access request occurs, so the access tim
When writing, performing an auto-precharge means that no command can be called fo
tAP cycles after a WRITA command is called. When the bank active mode is used, RE
WRIT commands can be issued consecutively if the row address is the same. This sho
number of cycles by tRWL + tAP for each write. The number of cycles between the iss
precharge command and the row address strobe command is determined by the TRP1,
MCR.

Whether execution is faster when the bank active mode is used or when basic access is
determined by the proportion of accesses to the same row address (P1) and the averag
cycles from the end of one access to the next access (tA). When tA is longer than tAP,
waiting for the precharge during a read becomes invisible. If tA is longer than tRWL +
delay waiting for the precharge also becomes invisible during writes. The difference b
bank active mode and basic access speeds in these cases is the number of cycles betwe
of access and the issue of the read/write command: (tRP + tRCD) × (1 – P1) and tRCD,
respectively.

The time that a bank can be kept active, tRAS, is limited. When the period will be prov
program execution, and it is not assured that another row address will be accessed wit
the cache, the synchronous DRAM must be set to auto-refresh and the refresh cycle m
the maximum value tRAS or less. This enables the limit on the maximum active perio
bank to be ensured. When auto-refresh is not being used, some measure must be taken
program to ensure that the bank does not stay active for longer than the prescribed per

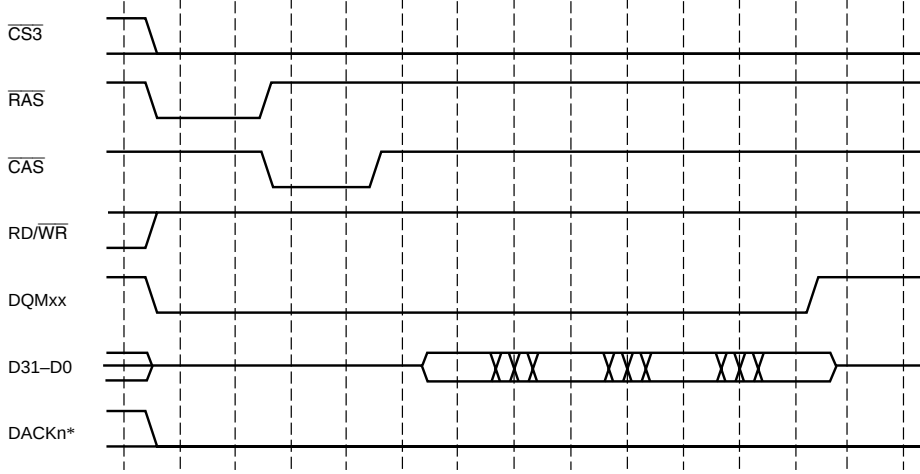
Figure 7.24 (a) and (b) show burst read cycles that is not an auto-precharge cycle, figu
and (b) show burst read cycles to a same row address, figure 7.26 (a) and (b) show bur
cycles to different row addresses, figure 7.27 shows a write cycle without auto-precha
7.28 shows a write cycle to a same row address, and figure 7.29 shows a write cycle to
row addresses.

accesses to the respective banks of the CS3 space are considered. Accesses to other CS during this period do not affect this operation. When an access occurs to a different row while the bank is active, figure 7.26 or figure 7.29 will be substituted for figures 7.25 a after this is detected. Both banks will become inactive even in the bank active mode aft refresh cycle ends or after the bus is released by bus arbitration.



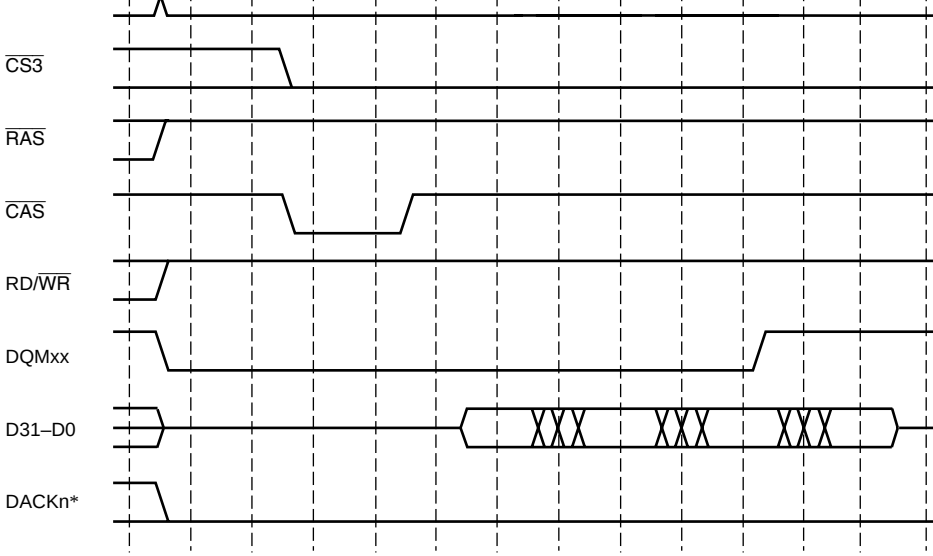
Note: * DACKn waveform when active-low is specified.

Figure 7.24 (a) Burst Read Timing (No Precharge) Except t_{Ecyc}:t_{Pcyc} 1:1



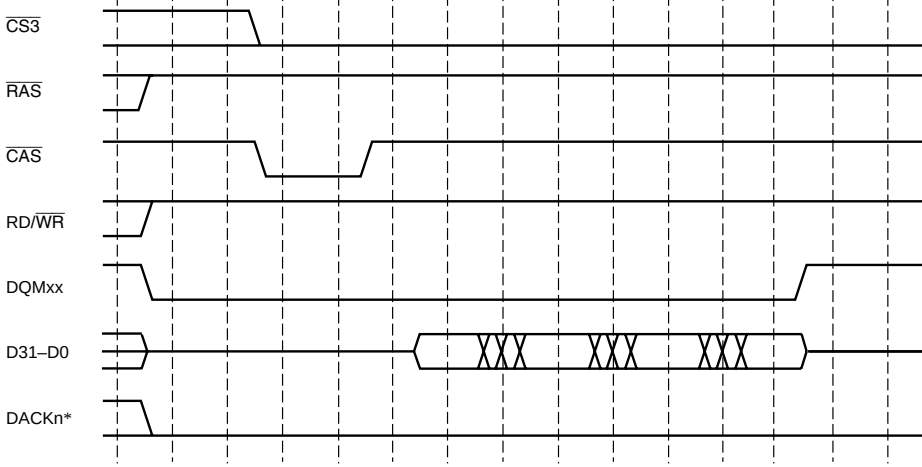
Note: * DACKn waveform when active-low is specified.

Figure 7.24 (b) Burst Read Timing (No Precharge) Iϕ:Eϕ = 1:1



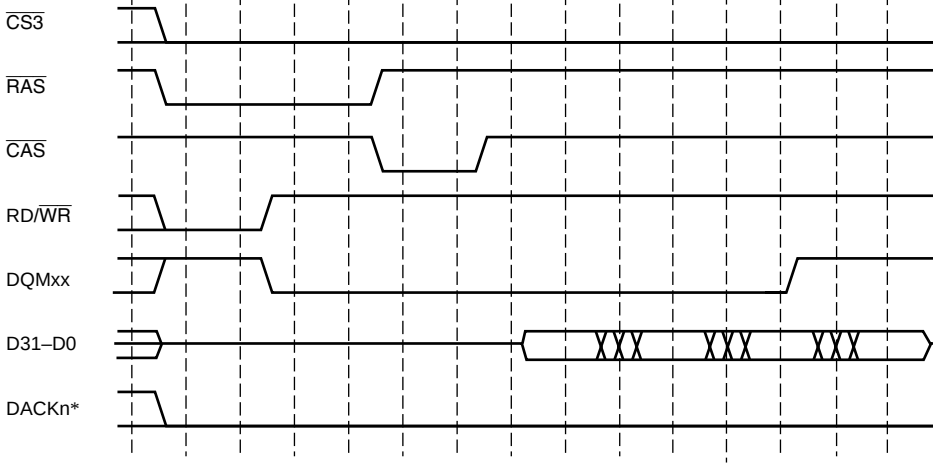
Note: * DACKn waveform when active-low is specified.

Figure 7.25 (a) Burst Read Timing (Bank Active, Same Row Address) Except t_{Ecyc}:t_{Pcyc} 1:1



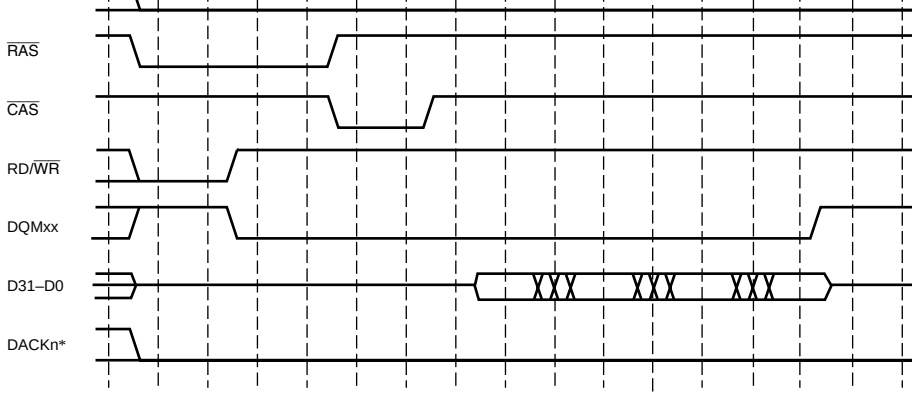
Note: * DACKn waveform when active-low is specified.

Figure 7.25 (b) Burst Read Timing (Bank Active, Same Row Address) Iϕ:E



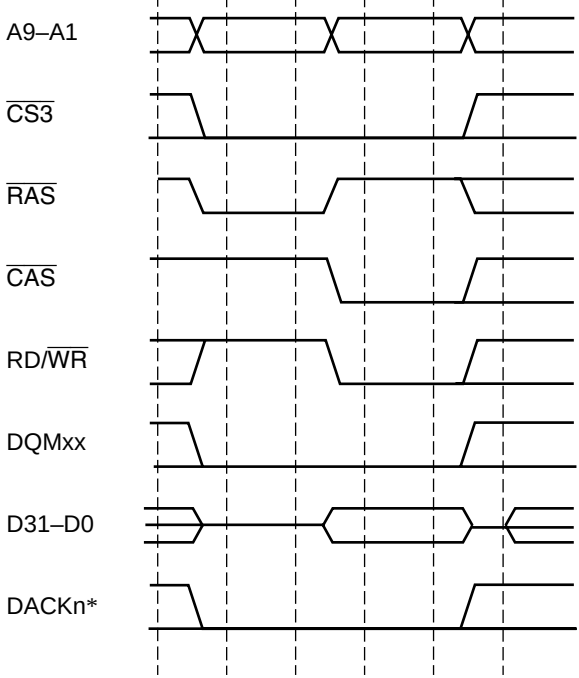
Note: * DACKn waveform when active-low is specified.

Figure 7.26 (a) Burst Read Timing (Bank Active, Different Row Address Except t_{Ecyc}:t_{Pcyc} 1:1



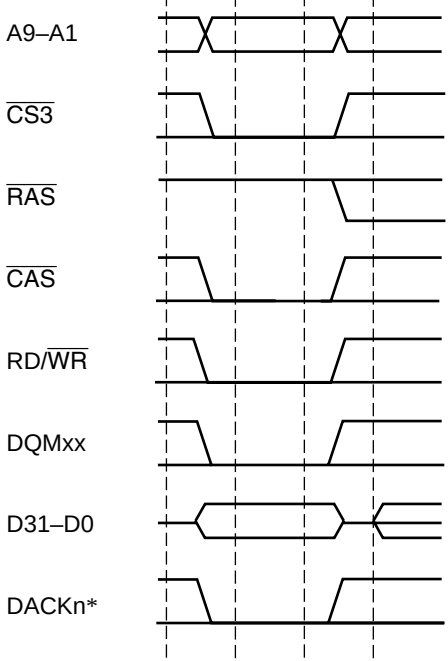
Note: * DACKn waveform when active-low is specified.

Figure 7.26 (b) Burst Read Timing (Bank Active, Different Row Addresses) Ic



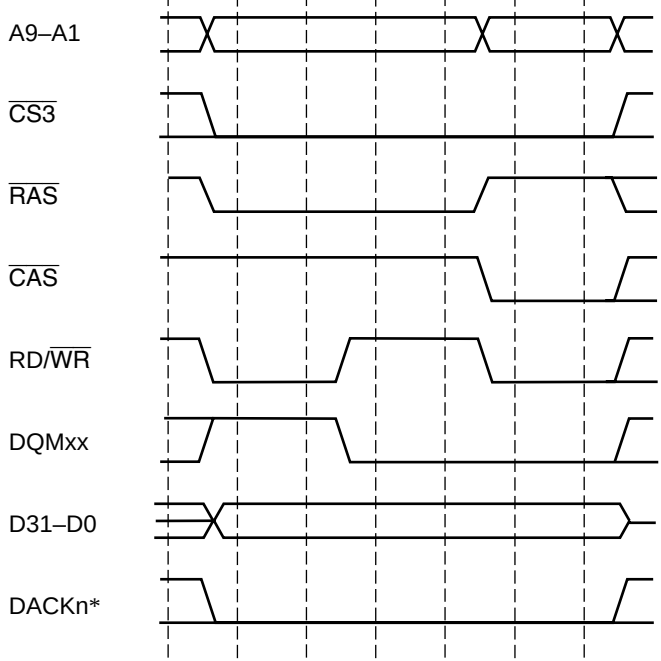
Note: * DACKn waveform when active-low is specified.

Figure 7.27 Single Write Mode Timing (No Precharge)



Note: * DACKn waveform when active-low is specified.

Figure 7.28 Single Write Mode Timing (Bank Active, Same Row Addre



Note: * DACKn waveform when active-low is specified.

Figure 7.29 Single Write Mode Timing (Bank Active, Different Row Addre

the interval determined by the input clock selected by the CKS2 to CKS0 bits in RTCS
value set in RTCOR. Set the CKS2 to CKS0 bits and RTCOR so that the refresh interv
specifications of the synchronous DRAM being used are satisfied. First , set RTCOR,
and the RMODE and RFSH bits in MCR, then set the CKS2 to CKS0 and RRC2 to R
RTCSR. When a clock is selected with the CKS2 to CKS0 bits, RTCNT starts countin
the value at that time. The RTCNT value is constantly compared to the RTCOR value
the two values match, a refresh request is made, and the number of auto-refreshes set i
RRC0 are performed. RTCNT is cleared to 0 at that time and the count up starts again
shows the timing for the auto-refresh cycle.

First, a PALL command is issued during the Tp cycle to change all the banks from act
precharge states. Then number of idle cycles equal to one less than the value set in TR
TRP0 are inserted, and a REF command is issued in the Trr cycle. After the Trr cycle,
commands are output for the number of cycles specified in the TRAS bit in MCR. The
must be set to satisfy the refresh cycle time specifications (active/active command del
the synchronous DRAM. When the set value of the TRP1 and TRP0 bits in MCR is 2
NOP cycle is inserted between the Tp cycle and Trr cycle.

During a manual reset, no refresh request is issued, since there is no RTCNT count-up
a refresh properly, make the manual reset period shorter than the refresh cycle interval
RTCNT to (RTCOR – 1) so that the refresh is performed immediately after the manua
cleared.

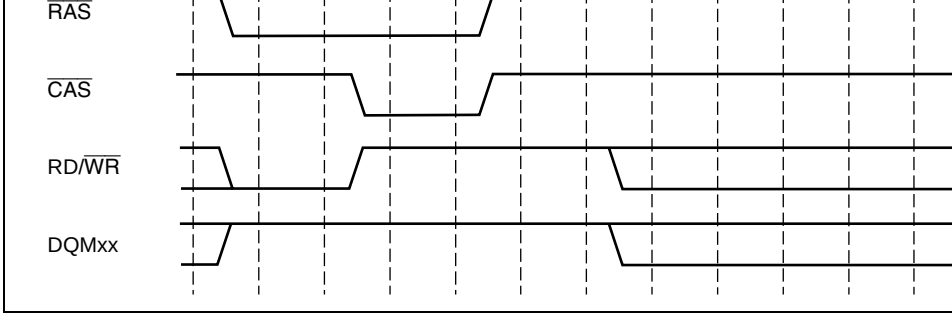


Figure 7.30 Auto-Refresh Timing

Self-Refreshes: The self-refresh mode is a type of standby mode that produces refresh
refresh addresses within the synchronous DRAM. It is started up by setting the RMOD
RFSH bits to 1. The synchronous DRAM is in self-refresh mode when the CKE signal
low. During the self-refresh, the synchronous DRAM cannot be accessed. To clear the
set the RMODE bit to 0. After self-refresh mode is cleared, issuing of commands is pro
the number of cycles specified in the TRAS1 and TRAS0 bits in MCR. Figure 7.31 sho
refresh timing. Settings must be made so that self-refresh clearing and data retention ar
performed correctly, and auto-refreshing is performed without delay at the correct inter
self-refresh mode is entered while the synchronous DRAM is set for auto-refresh or wh
the standby mode with a manual reset or NMI, auto-refresh can be re-started if RFSH is
RMODE is 0 when the self-refresh mode is cleared. When time is required between cle
self-refresh mode and starting the auto-refresh mode, this time must be reflected in the
RTCNT setting. When the RTCNT value is set to RTCOR – 1, the refresh can be starte
immediately.

If the standby function of the chip is used to enter the standby mode after the self-refres
set, the self-refresh state continues; the self-refresh state will also be maintained after re
from a standby using an NMI. A manual reset cannot be used to exit the self-refresh sta

During a power-on reset, the bus state controller register is initialized, so the self-refres
ended.

If a bus arbitration request occurs during a self-refresh, the bus is not released until the
is cleared.

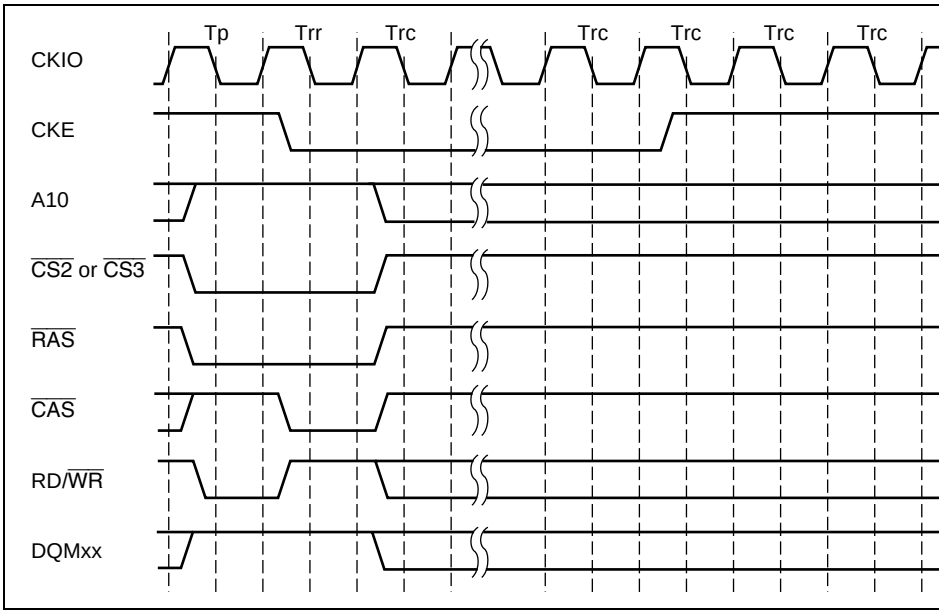


Figure 7.31 Self-Refresh Timing

overlap that occurs when memory spaces CS2 and CS3 are connected to SDRAM (tabl
3).

Table 7.7 Cases of Overlap Between Tap Cycle and Next Access

No.	First Access	Second Access
1	Space CS3, auto precharge mode	Access to different space among CS0, CS1, CS2,
2		Access to different bank in CS3
3	Space CS2, auto precharge mode	Access to different space among CS0, CS1, CS2,
4		Access to different bank in CS2

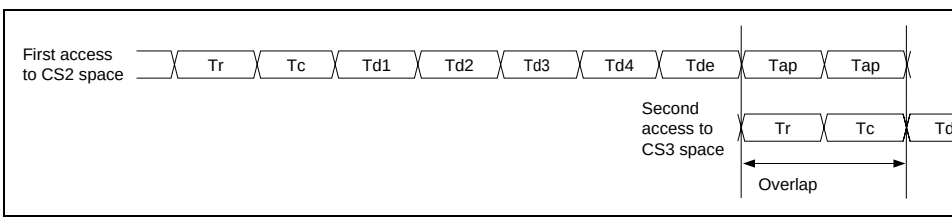


Figure 7.32 Conceptual Diagram of Overlap (Conditions: SDRAM Connected to
($\overline{RAS}$ Precharge Time Set to 2 Cycles) and SDRAM Connected to CS3 Spa

H'FFFF0000 or X + H'FFFF8000 is used depends on the specifications of the synchro
DRAM. Use a value in the range H'000 to H'FFF for X. Data is ignored at this time, b
is written using word as the size.

Write any data in word size to the following addresses to select the burst read single w
supported by the chip, a CAS latency of 1 to 3, a sequential wrap type, and a burst len
(depending on whether the width is 16 bits or 32 bits).

- Burst Read/Single Write

For 16 bits:	CAS latency 1	H'FFFF0426	(H'FFFF8426)
	CAS latency 2	H'FFFF0446	(H'FFFF8446)
	CAS latency 3	H'FFFF0466	(H'FFFF8466)
For 32 bits:	CAS latency 1	H'FFFF0848	(H'FFFF8848)
	CAS latency 2	H'FFFF0888	(H'FFFF8888)
	CAS latency 3	H'FFFF08C8	(H'FFFF88C8)

  To set burst read, burst write, CAS latency 1 to 3, wrap-type sequential, and burst
  (depending on whether the width is 16 bits or 32 bits), arbitrary data is written to t
  addresses, using the word size.

- Burst Read/Burst Write

16-bit width:	CAS latency 1	H'FFFF0026	(H'FFFF8026)
	CAS latency 2	H'FFFF0046	(H'FFFF8046)
	CAS latency 3	H'FFFF0066	(H'FFFF8066)
32-bit width:	CAS latency 1	H'FFFF0048	(H'FFFF8048)
	CAS latency 2	H'FFFF0088	(H'FFFF8088)
	CAS latency 3	H'FFFF00C8	(H'FFFF80C8)

Figure 7.33 shows the mode register setting timing.

Writing to address X + H'FFFF0000 or X + H'FFFF8000 first issues an all-bank prech
command (PALL), then issues eight dummy auto-refresh commands (REF) required f
synchronous DRAM power-on sequence. Lastly, a mode register write command (MR

command is issued. Refer to the synchronous DRAM manual for the necessary idle tim
the pulse width of the reset signal is longer than the idle time, the mode register may be
immediately without problem. However, care is required if the pulse width of the reset
shorter than the idle time.

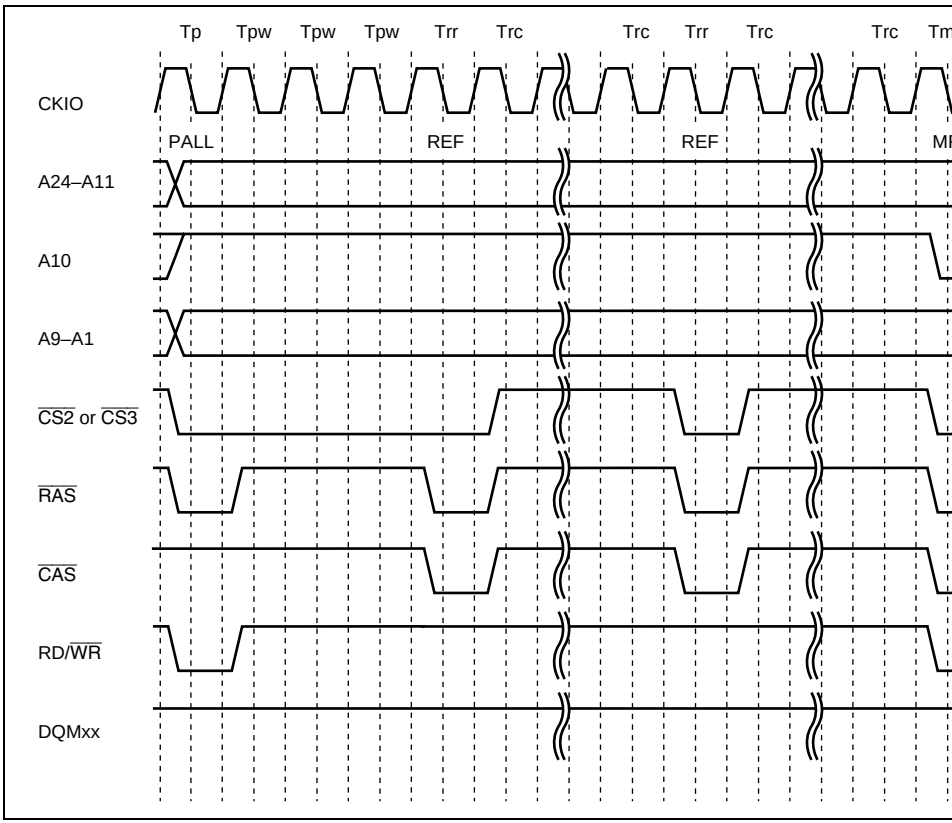


Figure 7.33 Synchronous DRAM Mode Write Timing

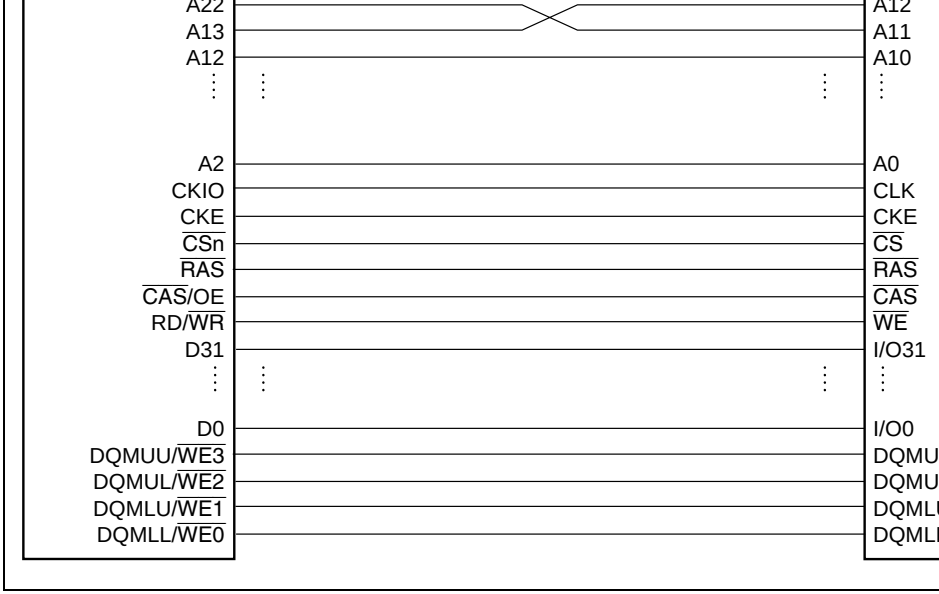


Figure 7.34 64-Mbit Synchronous DRAM (2 Mwords × 32 Bits) Connection

Bus Status Controller (BSC) Register Settings: Set the individual bits in the memor
register (MCR) as follows.

MCR (bit 6) SZ = 1
MCR (bit 7) AMX2 = 0
MCR (bit 5) AMX1 = 0
MCR (bit 4) AMX0 = 0

Synchronous DRAM Mode Settings: To make mode settings for the synchronous D
to address X + H'FFFF0000 or X + H'FFFF8000 from the CPU. (X represents the setti
Whether to use X + H'FFFF0000 or X + H'FFFF8000 determines on the synchronous
used.

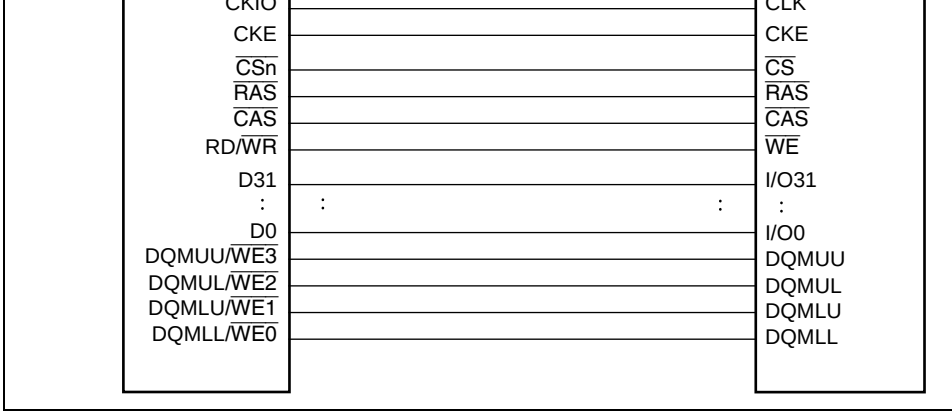


Figure 7.35 128-Mbit Synchronous DRAM (4 Mwords × 32 Bits) Connection E

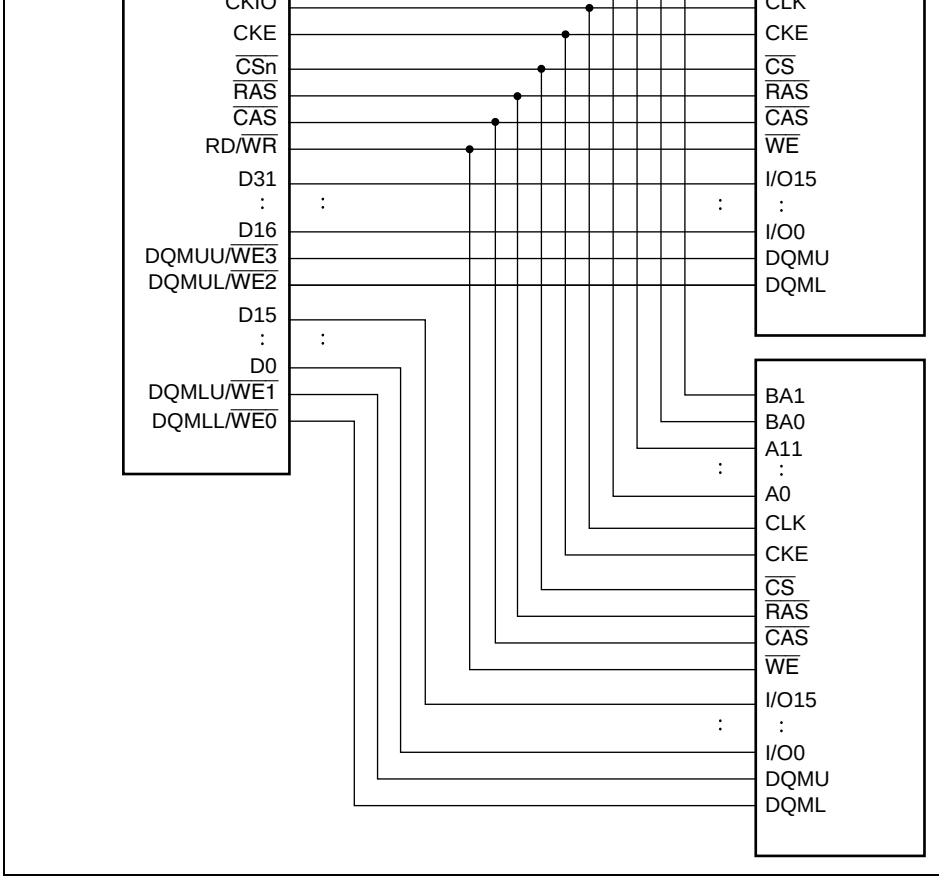


Figure 7.36 128-Mbit Synchronous DRAM (8 Mwords × 16 Bits) Connection

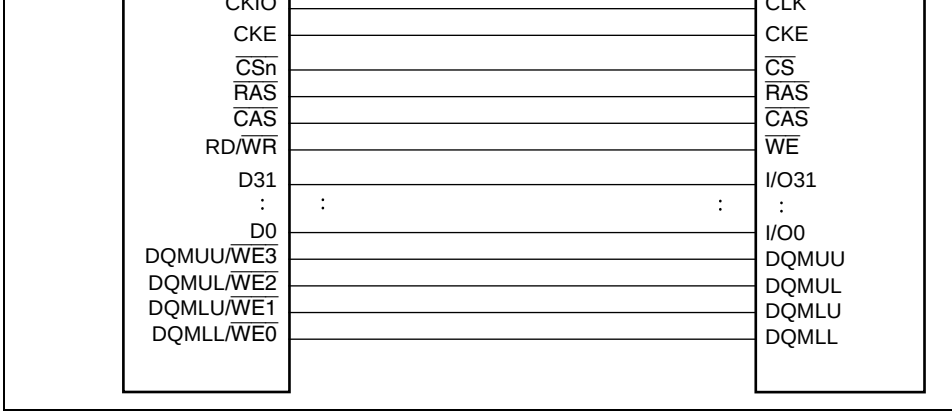


Figure 7.37 256-Mbit Synchronous DRAM (8 Mwords × 32 Bits) Connection E

DRAMs can be connected, since $\overline{CAS}$ is used to control byte access. The $\overline{RAS}$, $\overline{CAS3}$
and RD/$\overline{WR}$ signals are used to connect the DRAM. When the data width is 16 bits, $\overline{C}$
$\overline{CAS2}$ are not used. In addition to ordinary read and write access, burst access using h
page mode is also supported.

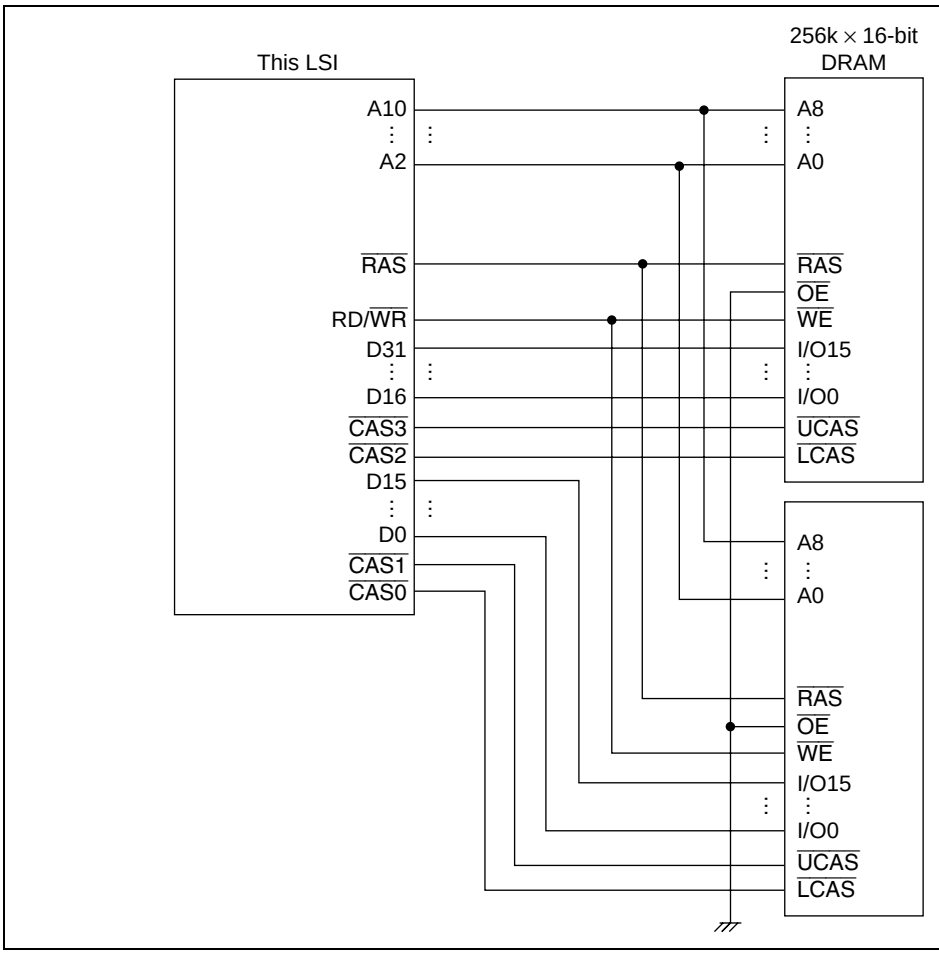


Figure 7.38 Example of DRAM Connection (32-Bit Data Width)

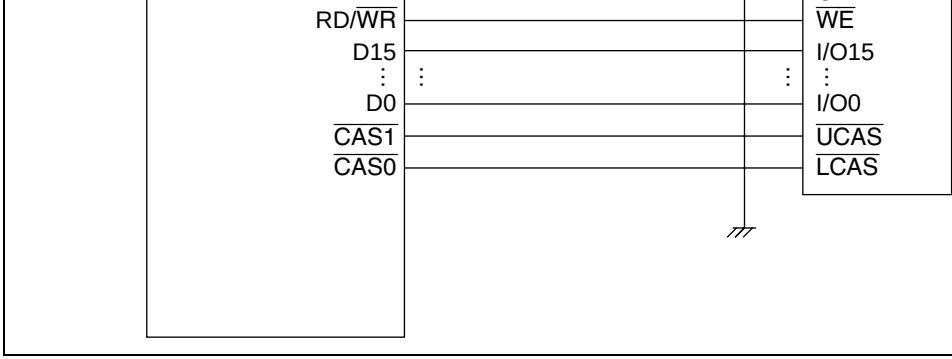


Figure 7.39 Example of DRAM Connection (16-Bit Data Width)

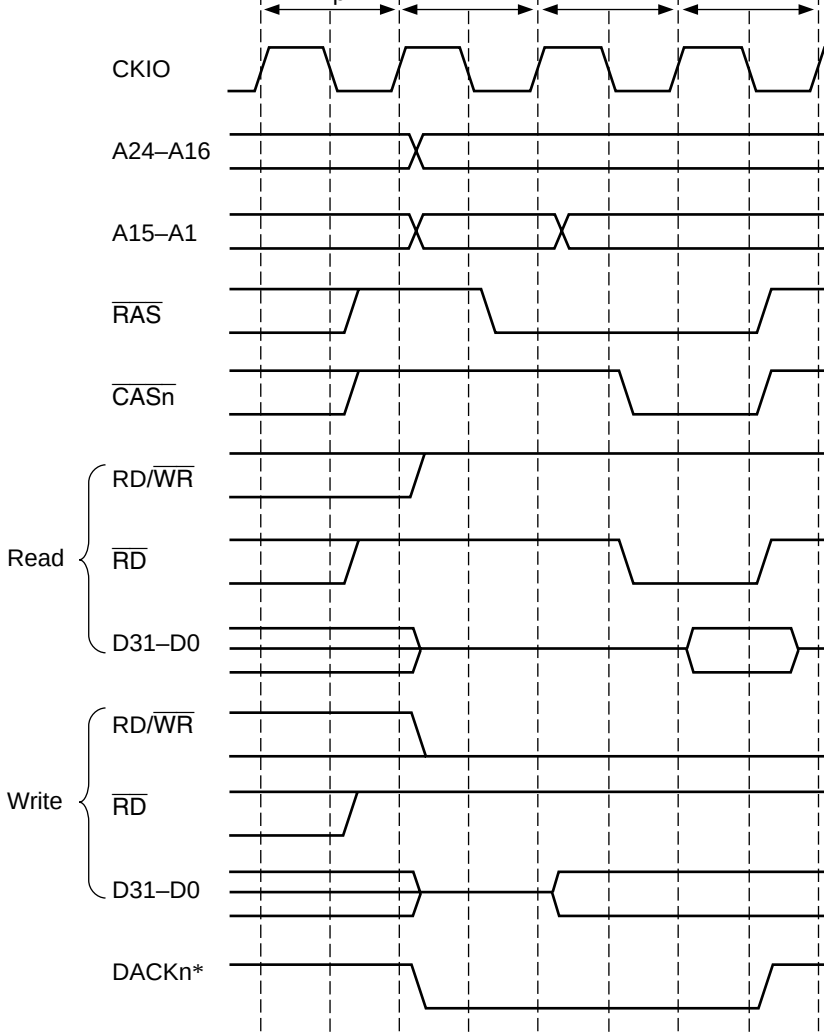
7.6.2 Address Multiplexing

When the CS3 space is set to DRAM, addresses are always multiplexed. This allows D
require multiplexing of row and column addresses to be connected directly without add
address multiplexing circuits. There are four ways of multiplexing, which can be select
the AMX1 and AMX0 bits in MCR. Table 7.8 illustrates the relationship between the A
AMX0 bits and address multiplexing. Address multiplexing is performed on address ou
A15 to A1. The original addresses are output to pins A24 to A16. During DRAM acces
is reserved, so set it to 0.

Table 7.8 Relationship between AMX1 and AMX0 and Address Multiplexing

AMX1	AMX0	No. of Column Address Bits	Row Address Output	Column Address Out
0	0	8 bits	A23 to A9	A15 to A1
	1	9 bits	A24 to A10	A15 to A1
1	0	10 bits	A24 to A11*1	A15 to A1
	1	11 bits	A24 to A12*2	A15 to A1

Notes: 1. Address output pin A15 is high.
2. Address output pins A15 and A14 are high.

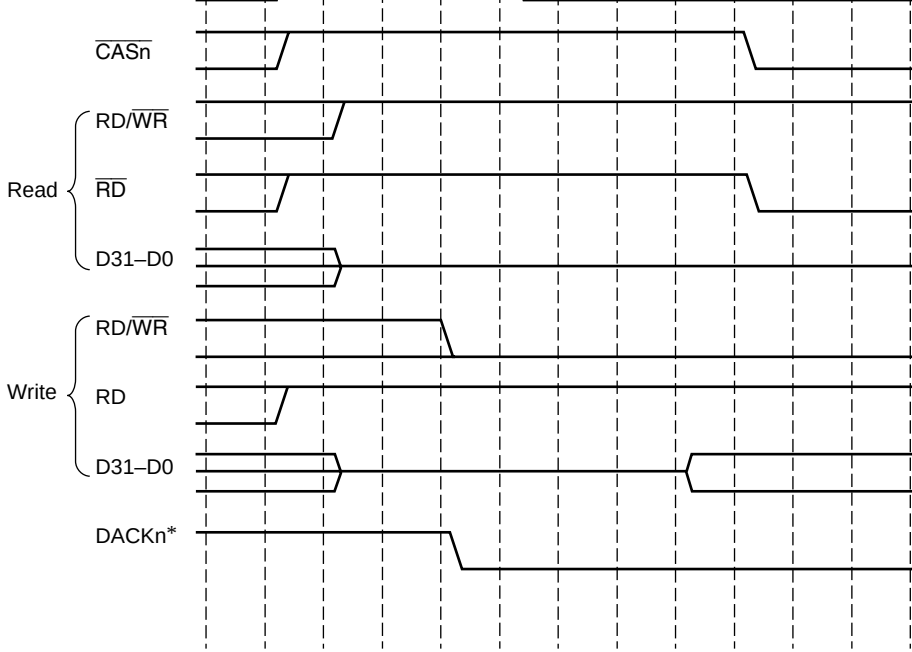


Note: * DACKn waveform when active-low is specified.

Figure 7.40 Basic Access Timing

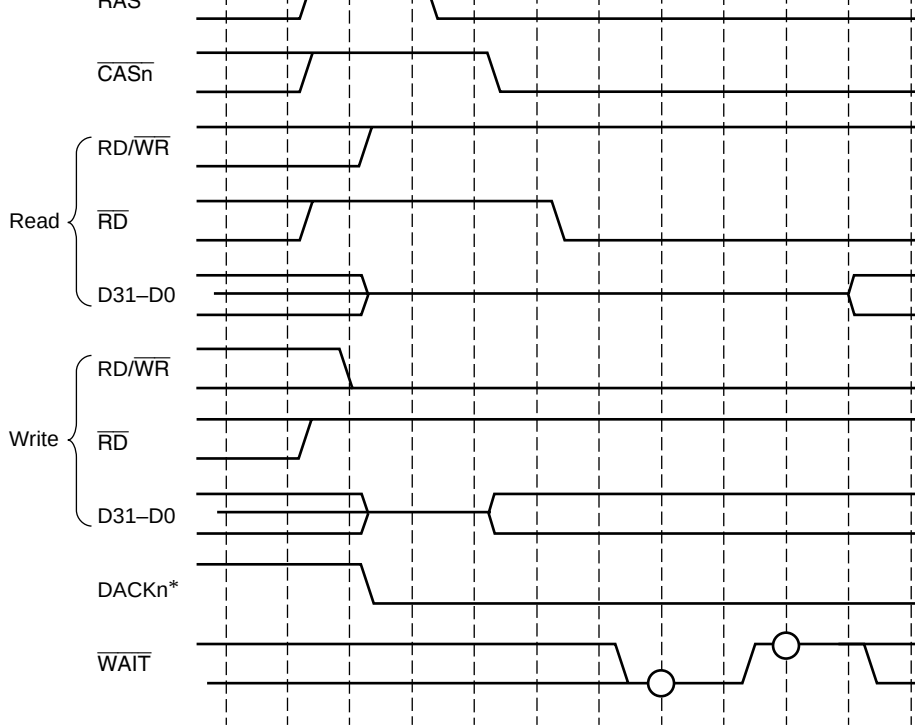
a Trw cycle by means of the RCD1, RCD0 bit in MCR. The number of cycles from CA
to the end of access can be extended from 1 cycle to 3 cycles by setting the W31/W30 b
WCR1. When external wait mask bit A3WM in WCR2 is cleared to 0 and bits W31 and
WCR1 are set to a value other than 00, the external wait pin is also sampled, so the num
cycles can be further increased. When bit A3WM in WCR2 is set to 1, external wait inp
ignored regardless of the setting of W31 and W30 in WCR1. Figure 7.42 shows the tim
state control using the $\overline{\text{WAIT}}$ pin.

In either case, when consecutive accesses occur, the Tp cycle access overlaps the Tc2 c
previous access. In DRAM access, $\overline{\text{BS}}$ is not asserted, and so $\overline{\text{RAS}}$, $\overline{\text{CAS}}$n, $\overline{\text{RD}}$, etc., sho
used for $\overline{\text{WAIT}}$ pin control.



Note: * DACKn waveform when active-low is specified

Figure 7.41 Wait State Timing

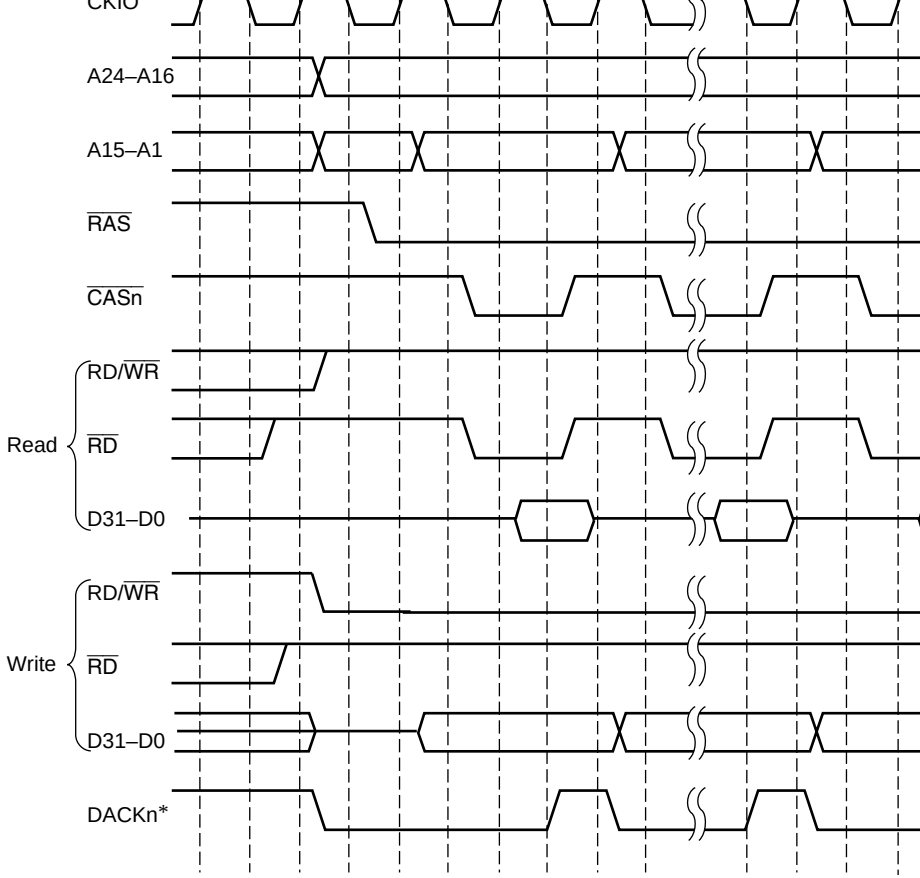


Note: * DACKn waveform when active-low is specified

Figure 7.42 External Wait State Timing

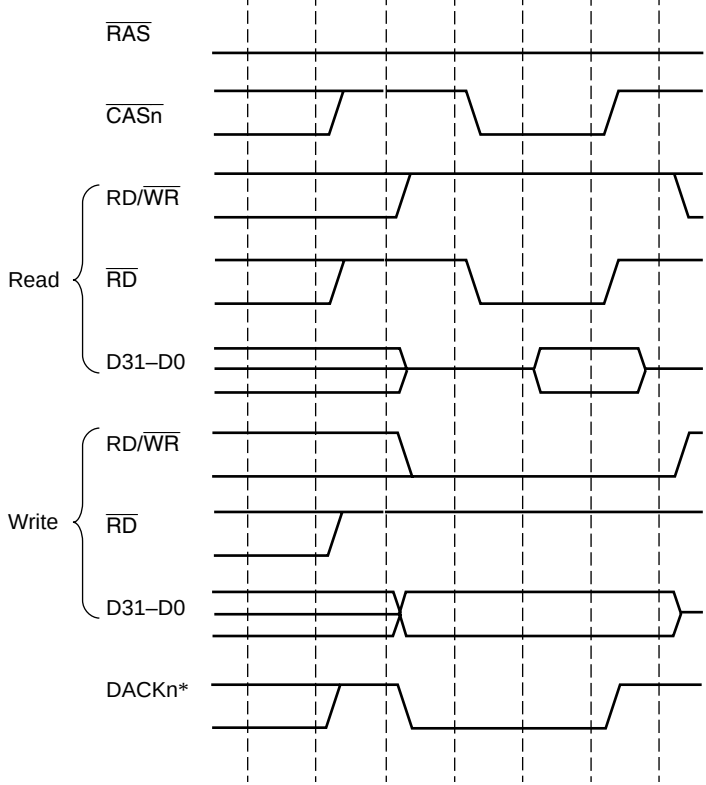
7.6.5 Burst Access

In addition to the ordinary mode of DRAM access, in which row addresses are output a access and data is then accessed, DRAM also has a high-speed page mode for use when continuously accessing the same row that enables fast access of data by changing only t address after the row address is output. Select ordinary access or high-speed page mode the burst enable bit (BE) in MCR. Figure 7.43 shows the timing of burst access in high- mode. When performing burst access, cycles can be inserted using the wait state contro



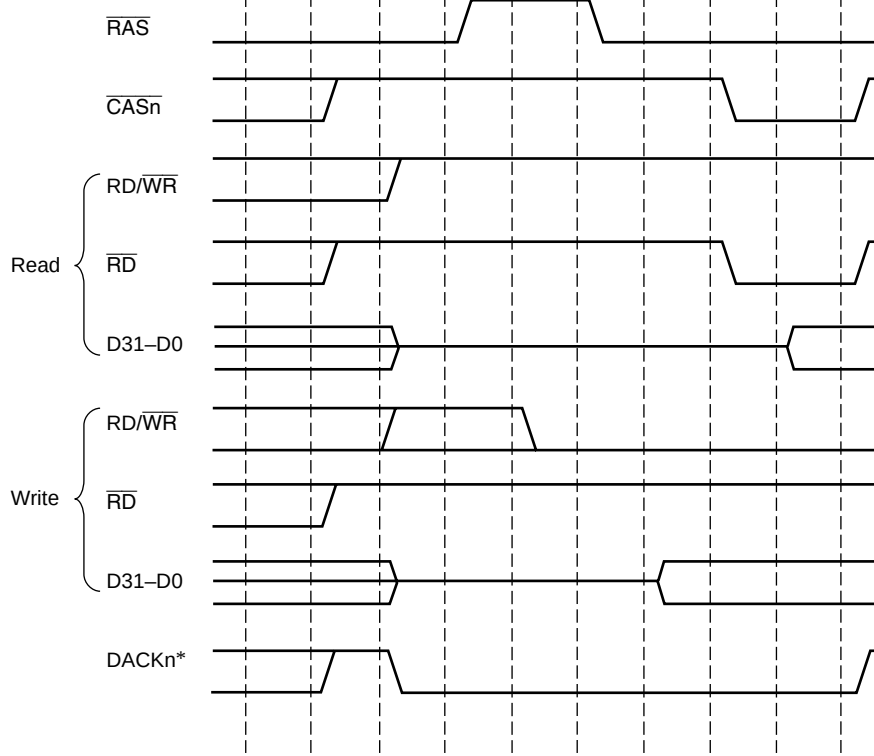
Note: * DACKn waveform when active-low is specified

Figure 7.43 Burst Access Timing



Note: * DACKn waveform when active-low is specified

Figure 7.44 $\overline{\text{RAS}}$ Down Mode Same Row Access Timing



Note: * DACKn waveform when active-low is specified

Figure 7.45 $\overline{\text{RAS}}$ Down Mode Different Row Access Timing

7.6.6 EDO Mode

In addition to the kind of DRAM in which data is output to the data bus only while the
signal is asserted in a data read cycle, there is another kind provided with an EDO mo
while both $\overline{\text{RAS}}$ and $\overline{\text{OE}}$ are asserted, once the $\overline{\text{CASn}}$ signal is asserted data is output t
bus until $\overline{\text{CASn}}$ is next asserted, even though $\overline{\text{CASn}}$ is negated during this time.

by 1/2 cycle, making it at the rise of the CRIO clock.

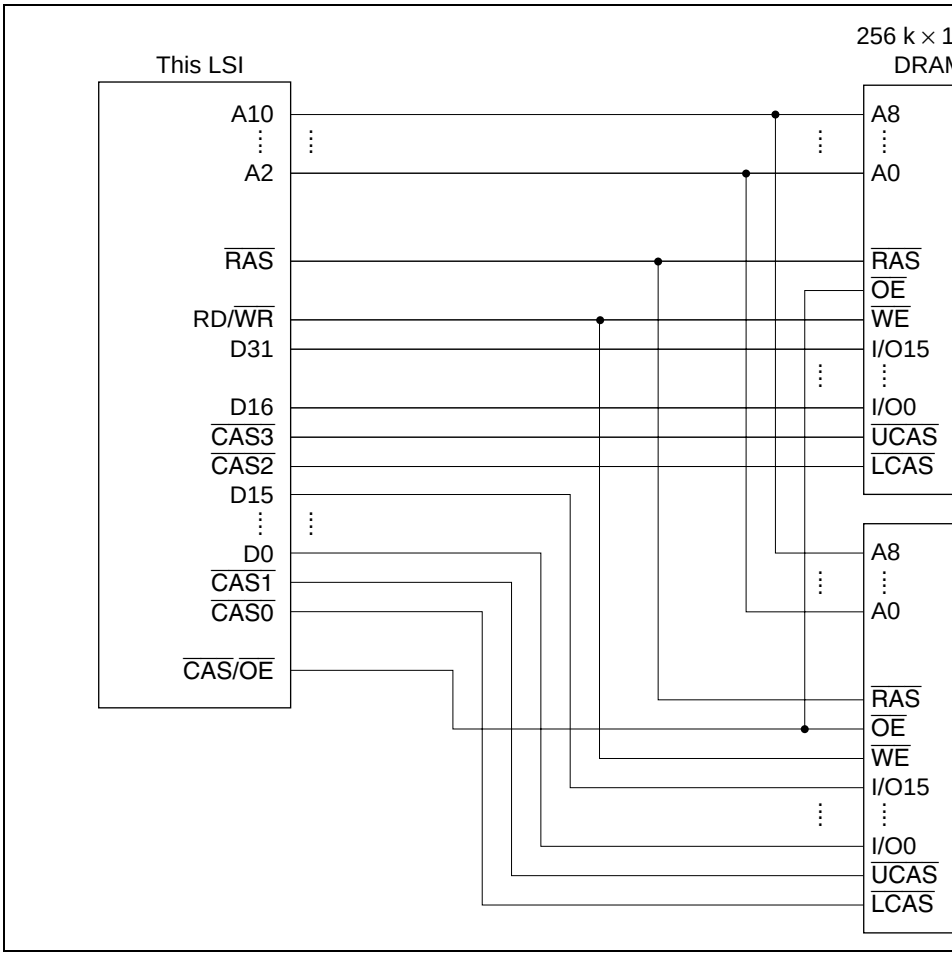


Figure 7.46 Example of EDO DRAM Connection (32-Bit Data Width)

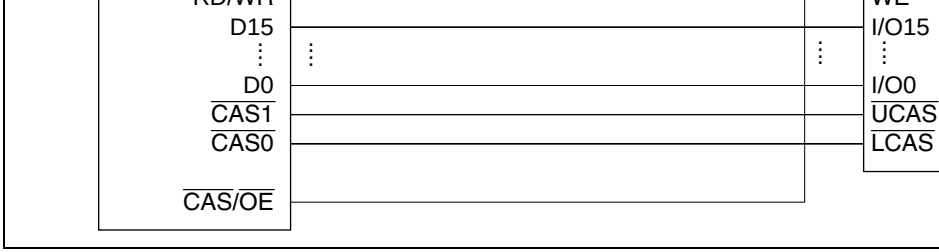
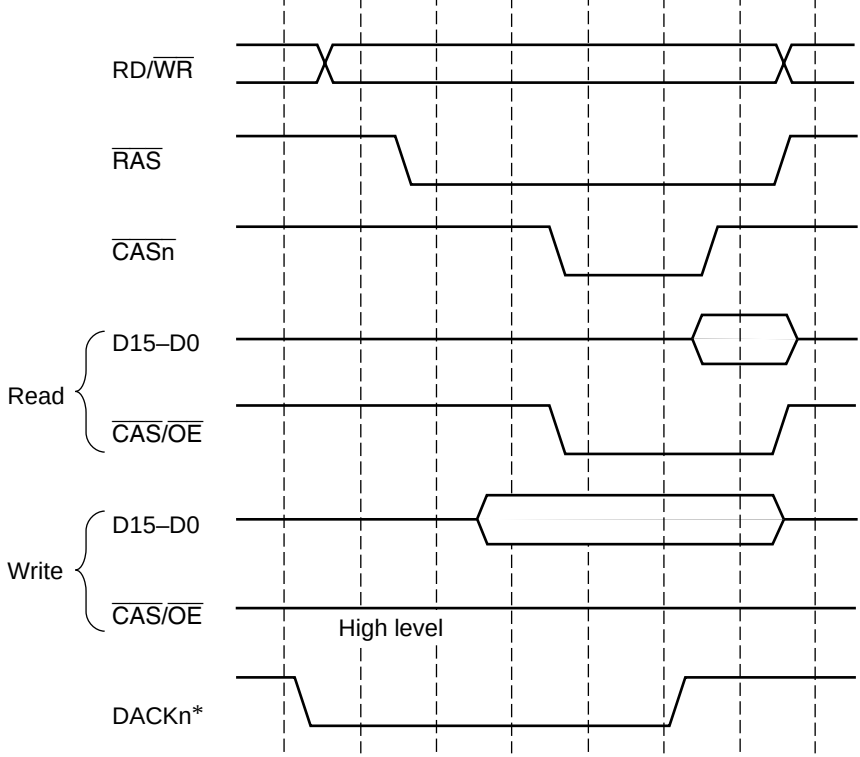
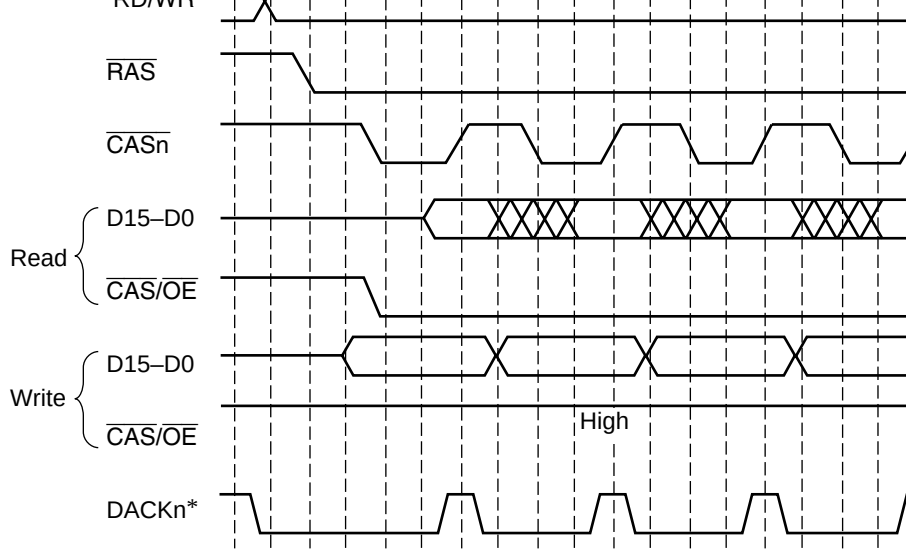


Figure 7.47 Example of EDO DRAM Connection (16-Bit Data Width



Note: * DACKn waveform when active-low is specified

Figure 7.48 DRAM EDO Mode Ordinary Access Timing

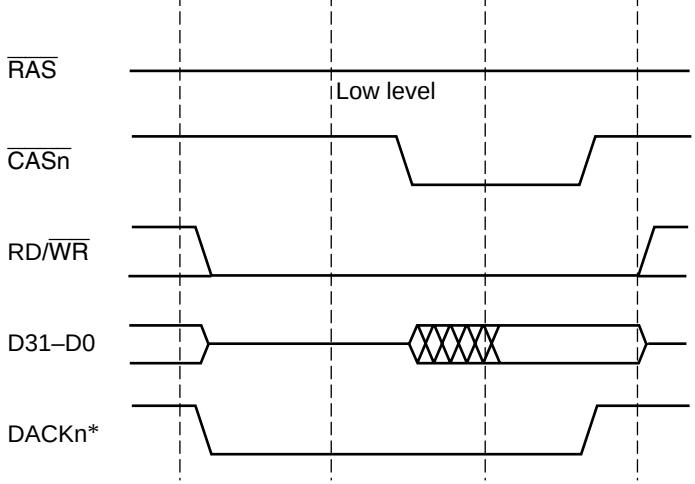


Note: * DACKn waveform when active-low is specified

Figure 7.49 DRAM EDO Mode Burst Access Timing

7.6.7 DRAM Single Transfer

Wait states equivalent to the value set in bits DSWW1 and DSWW0 in BCR3 can be i
between DACKn assertion and $\overline{\text{CASn}}$ assertion in a write in DMA single address trans
Inserting wait states allows the data setup time for external device memory. Figure 7.5
write cycle timing in DMA single transfer mode when DSWW1/DSWW0 = 01 and R
The DMA single transfer mode read cycle is the same as a CPU or DMA dual transfer
cycle.



Note: * DACKn waveform when active-low is specified

Figure 7.50 DMA Single Transfer Mode Write Cycle Timing (RAS Down Mode, Same Row Address)

7.6.8 Refreshing

The bus state controller includes a function for controlling DRAM refreshing. Distribut
refreshing using a CAS-before-RAS refresh cycle can be performed by clearing the RM
0 and setting the RFSH bit to 1 in MCR. Consecutive refreshes can be generated by set
RRC2 to RRC0 in RTCSR. If DRAM is not accessed for a long period, self-refresh mo
uses little power consumption for data retention, can be activated by setting both the RM
RFSH bits to 1.

CAS-Before-RAS Refreshing: Refreshing is performed at intervals determined by the
selected by bits CKS2 to CKS0 in RTCSR, and the value set in RTCOR. The RTCOR
the value of bits CKS2 to CKS0 in RTCSR should be set so as to satisfy the refresh inte
specification for the DRAM used. First make the settings for RTCOR, RTCNT, and the
and RFSH bits in MCR, then make the CKS2 to CKS0 and RRC2 to RRC0 settings in

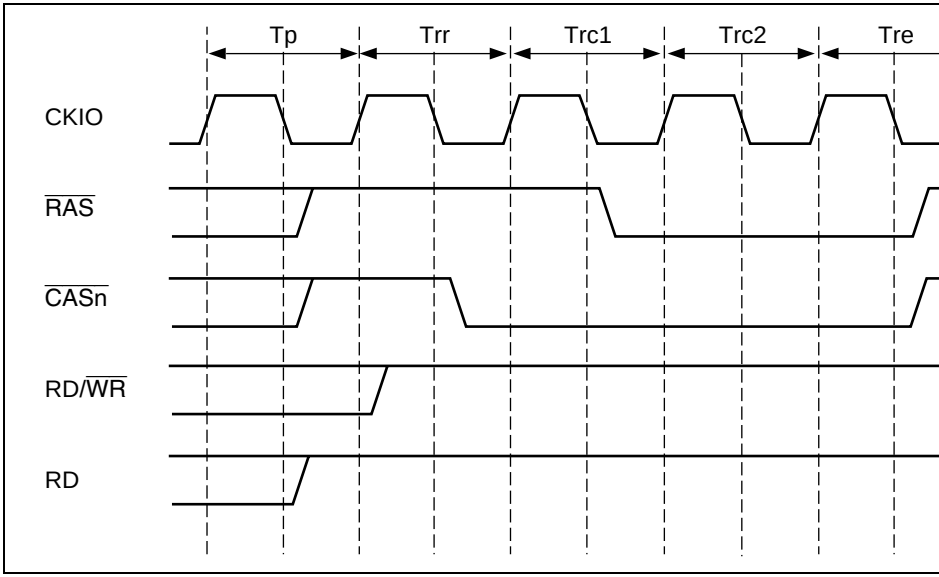


Figure 7.51 DRAM CAS-before-RAS Refresh Cycle Timing

Self-Refreshing: A self-refresh is started by setting both the RMODE bit and the RFS
During the self-refresh, DRAM cannot be accessed. Self-refreshing is cleared by clear
RMODE bit to 0. Self-refresh timing is shown in figure 7.52. Settings must be made s
refresh clearing and data retention are performed correctly, and CAS-before-RAS refr
immediately performed at the correct intervals. When self-refreshing is started from th
which CAS-before-RAS refreshing is set, or when exiting standby mode by means of
reset or NMI, auto-refreshing is restarted if RFSH is set to 1 and RMODE is cleared to
self-refresh mode is cleared. If the transition from clearing of self-refresh mode to star
refresh takes time, this time should be taken into consideration when setting the initial
RTCNT. When the RTCNT value is set to RTCOR-1, the refresh can be started imme

After self-refreshing has been set, the self-refresh state continues even if the chip stand
entered using the chip's standby function. The self-refresh state is also maintained eve
recovery from standby mode by means of NMI input.

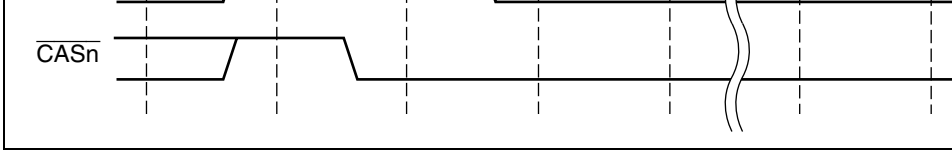


Figure 7.52 DRAM Self-Refresh Cycle Timing

7.6.9 Power-On Sequence

When DRAM is used after the power is turned on, there is a requirement for a waiting p
during which accesses cannot be performed (100 µs or 200 µs minimum) followed by a
prescribed number of dummy $\overline{CAS}$-before-$\overline{RAS}$ refresh cycles (usually 8). The bus stat
(BSC) does not perform any special operations for the power-on reset, so the required p
sequence must be implemented by the initialization program executed after a power-on

7.7 Burst ROM Interface

Set the BSTROM bit in BCR1 to set the CS0 space for connection to burst ROM. The b
interface is used to permit fast access to ROMs that have the nibble access function. Fig
shows the timing of nibble accesses to burst ROM. Set for two wait cycles. The access
the same as an ordinary access, but when the first cycle ends, only the address is change
CS0 signal is not negated, enabling the next access to be conducted without the T1 cycl
for ordinary space access. From the second time on, the T1 cycle is omitted, so access i
faster than ordinary accesses. Currently, the nibble access can only be used on 4-addres
This function can only be utilized for word or longword reads to 8-bit ROM and longw
16-bit ROM. Mask ROMs have slow access speeds and require 4 instruction fetches for
widths and 16 accesses for cache filling. Limited support of nibble access was thus add
alleviate this problem. When connecting to an 8-bit width ROM, a maximum of 4 conse
accesses are performed; when connecting to a 16-bit width ROM, a maximum of 2 cons
accesses are performed. Figure 7.53 shows the relationship between data width and acc
For cache filling and DMAC 16-byte transfers, longword accesses are repeated 4 times

8-bit bus-width word access

T1	Tw	T2

8-bit bus-width byte access

T1	Tw	T2	Tw	T2

16-bit bus-width longword access

T1	Tw	T2

16-bit bus-width word access

T1	Tw	T2

16-bit bus-width byte access

T1	Tw	T2

32-bit bus-width longword access

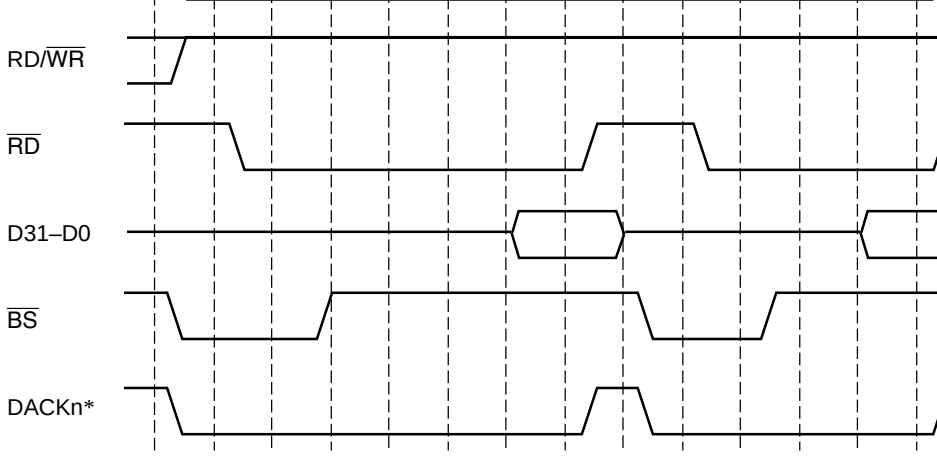
T1	Tw	T2

32-bit bus-width word access

T1	Tw	T2

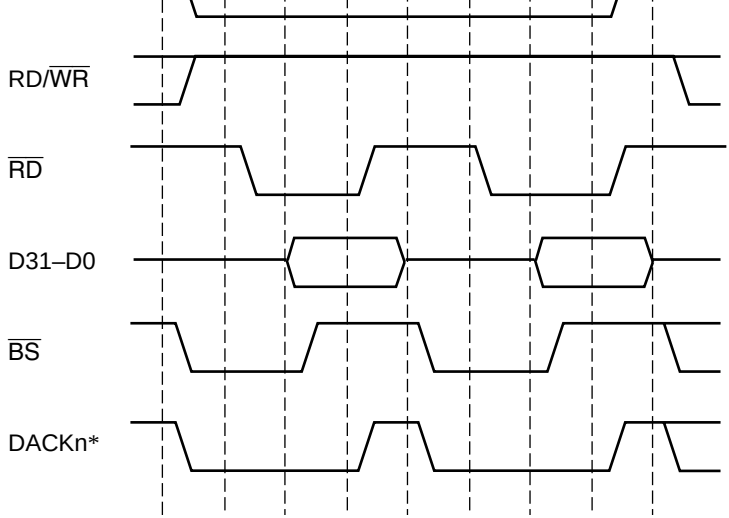
32-bit bus-width byte access

Figure 7.53 Data Width and Burst ROM Access (1 Wait State)



Note: * DACKn waveform when active-low is specified.

Figure 7.54 Burst ROM Nibble Access (2 Wait States)



Note: * DACKn waveform when active-low is specified.

Figure 7.55 Burst ROM Nibble Access (No Wait States)

immediately by a write from the chip. When the chip is writing continuously, the data c
always from the chip to other memory, and there are no particular problems. Neither is
particular problem if the following read access is to the same CS space, since data is ou
the same data buffer. The number of idle cycles to be inserted into the access cycle whe
from another CS space, or performing a write, after a read from the CS3 space, is speci
IW31 and IW30 bits in WCR1. Likewise, IW21 and IW20 specify the number of idle c
CS2 reads, IW11 and IW10 specify the number after CS1 reads, and IW01 and IW00 sp
number after CS0 reads. The number of idle cycles after a CS4 read is specified by the
IW40 bits in WCR2. From 0, 1, 2, or 4 cycles can be specified. When there is already a
between accesses, the number of empty cycles is subtracted from the number of idle cy
insertion. When a write cycle is performed immediately after a read access, 1 idle cycle
even when 0 is specified for waits between access cycles.

When the chip shifts to a read cycle immediately after a write, the write data becomes h
impedance when the clock rises, but the $\overline{RD}$ signal, which indicates read cycle data outp
is not asserted until the clock falls. The result is that no idles are inserted into the cycle.

When bus arbitration is being performed, an empty cycle is inserted for arbitration, so n
inserted between cycles.

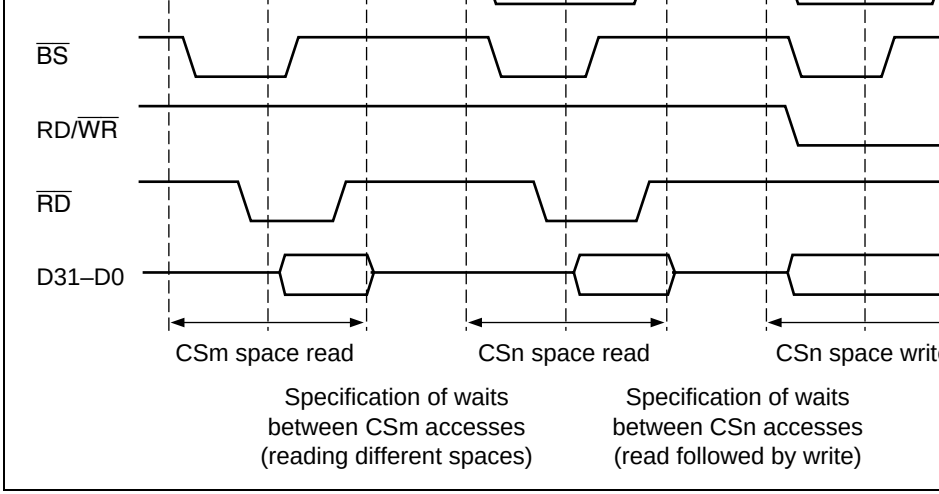


Figure 7.56 Idles between Cycles

The chip has three internal bus masters, the CPU, the DMAC and the E-DMAC. When synchronous DRAM or DRAM is connected and refresh control is performed, the refre becomes a fourth master. In addition to these, there are also bus requests from external The priority for bus requests when they occur simultaneously is as follows.

Refresh request → External device → E-DMAC → DMAC → CPU

However, only one E-DMAC channel can hold the bus during one bus-mastership cycl

The E-DMAC has two channels to handle both transmission and reception. Arbitration the channels is performed automatically within the E-DMAC module, with bus masters alternating between the transmit channel and the receive channel. For arbitration betwe DMAC channels, either fixed priority mode or round robin mode can be selected by me priority mode bit (PR) in the DMA operation register (DMAOR).

When the bus is being passed between slave and master, all bus control signals are nega the bus is released to prevent erroneous operation of the connected devices. When the b transferred, also, the bus control signals begin bus driving from the negated state. The r slave passing the bus between them drive the same signal values, so output buffer confl avoided. A pull-up resistance is required for the bus control signals to prevent malfunct by external noise when they are at high impedance.

Bus permission is granted at the end of the bus cycle. When the bus is requested, the bu released immediately if there is no ongoing bus cycle. If there is a current bus cycle, th released until the bus cycle ends. Even when a bus cycle does not appear to be in progr viewed from off-chip, it is not possible to determine immediately whether the bus has b released by looking at $\overline{CS}$n or other control signals, since a bus cycle (such as wait inse between access cycles) may have been started internally. The bus cannot be released du transfers for cache filling, DMAC 16-byte block transfers (16 + 16 = 32-byte transfers address mode), or E-DMAC 16-byte block transfers. Likewise, the bus cannot be releas the read and write cycles of a TAS instruction. Arbitration is also not performed betwe bus cycles produced by a data width smaller than the access size, such as a longword ac

Figures 7.57 (a) and 7.57 (b) show the timing charts in the cases that bus requests occu
simultaneously from the E-DMAC, DMAC, and CPU. These cases are based on the fo
settings:

- The CS2 and CS3 spaces are set for synchronous DRAM.
- The CAS latency is one cycle.
- The E-DMAC is enabled at both the transmitter and receiver (the buffer and descri
  CS3 space).
- The DMAC is enabled in only one channel that is set to auto-request mode, cycle-s
  and 16-byte dual-address transmission (CS2 space).
- Burst read and single write are set to synchronous DRAM.

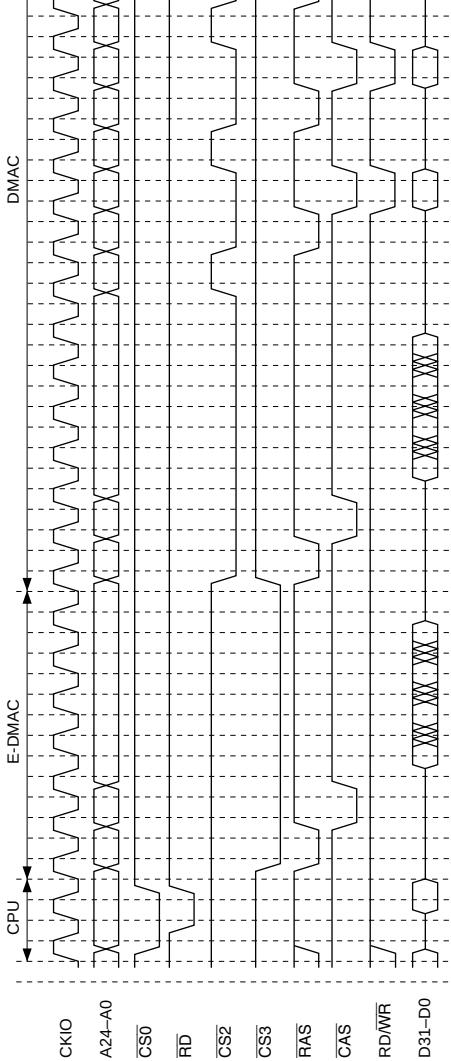


Figure 7.57 (a) Bus Arbitration Timing
(E-DMAC Read → DMAC 16-Byte Transmission → CPU Read)

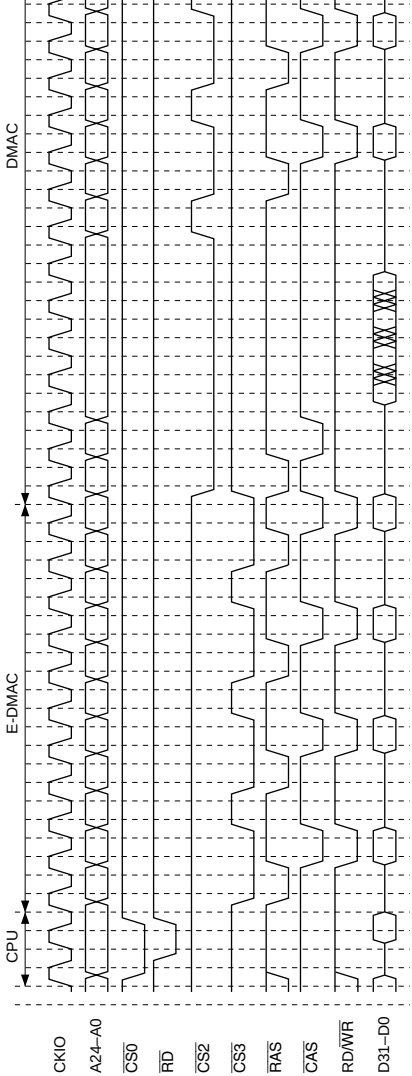


Figure 7.57 (b) Bus Arbitration Timing
(E-DMAC Write → DMAC 16-Byte Transmission → CPU Read)

signals DACK0 and DACK1.

When the DRAM has finished precharging, the bus is released. The synchronous DRA issues a precharge command to the active bank. After this is completed, the bus is relea

The specific bus release sequence is as follows. First, the address bus and data bus beco impedance synchronously with a rise of the clock. Half a cycle later, the bus use enable asserted synchronously with a fall of the clock. Thereafter the bus control signals ($\overline{BS}$, $\overline{CASn}$, $\overline{WEn}$, $\overline{RD}$, RD/$\overline{WR}$) become high impedance at a rise of the clock. These bus co signals are driven high at least 2 cycles before they become high impedance. Sampling request signals occurs at the clock fall.

The sequence when the bus is taken back from the slave is as follows. When the negati is detected at a clock fall, high-level driving of the bus control signals starts half a cycle bus use enable signal is then negated at the next clock fall. The address bus and data bu starting at the next clock rise. The bus control signals are asserted and the bus cycle act from the same clock rise at which the address and data signals are driven, at the earliest 7.58 shows the timing of bus arbitration.

To reduce the overhead due to arbitration with a user-designed slave, a number of cons accesses may be attempted. In this case, to insure dependable refreshing, the design mu for the slave to release the bus before it has held it for a period exceeding the refresh cy SH7615 is provided with the REFOUT pin to send a signal requesting the bus while ref execution is being kept waiting. REFOUT is asserted while refresh execution is being k until the bus is acquired. When the external slave device receives this signal and release the bus is returned to the chip and refreshing can be executed.

control signals

Figure 7.58 Bus Arbitration

7.10 Additional Items

7.10.1 Resets

The bus state controller is completely initialized only in a power-on reset. All signals a
immediately negated, regardless of whether or not the chip is in the middle of a bus cy
negation is simultaneous with turning the output buffer off. All control registers are in
standby mode, sleep mode, and a manual reset, no bus state controller control register
initialized. When a manual reset is performed, the currently executing bus cycle only i
and then the chip waits for an access. When a cache-filling or DMAC/E-DMAC 16-by
executing, the CPU, DMAC, or E-DMAC that is the bus master ends the access in a lo
unit, since the access request is canceled by the manual reset. This means that when a
is executed during a cache filling, the cache contents can no longer be guaranteed. Du
manual reset, the RTCNT does not count up, so no refresh request is generated, and a
is not initiated. To preserve the data of the DRAM and synchronous DRAM, the pulse
manual reset must be shorter than the refresh interval. Master mode chips accept arbitr
requests even when a manual reset signal is asserted. When a reset is executed only fo
master mode while the bus is released, the $\overline{\text{BGR}}$ signal is negated to indicate this. If th
signal is continuously asserted, the bus release state is maintained.

7.10.2 Access as Viewed from CPU, DMAC or E-DMAC

The chip is internally divided into three buses: cache, internal, and peripheral. The CP
memory are connected to the cache bus, the DMAC, E-DMAC and bus state controlle
connected to the internal bus, and the low-speed peripheral devices and mode register
connected to the peripheral bus. On-chip memory other than cache memory and the us
controller are connected to both the cache bus and the internal bus. The internal bus ca
accessed from the cache bus, but not the other way around. The peripheral bus can be
from the internal bus, but not the other way around. This results in the following.

consecutive longword reads occur. For misses that occur when byte or word operands a
or branches occur to odd word boundaries (4n + 2 addresses), the filling is always perfo
longword accesses on the chip-external interface. In the cache-through area, the access
actual access address. When the access is an instruction fetch, the access size is always

For cache-through areas and on-chip peripheral module read cycles, after an extra cycle
to determine the cycle, the read cycle is started through the internal bus. Read data is se
CPU through the cache bus.

When write cycles access the cache area, the cache is searched. When the data of the re
address is found, it is written here. The actual write occurs in parallel to this via the inte
write-through mode. In write-back mode, the actual write is not performed until a repla
operation occurs for the relevant address. When the right to use the internal bus is held,
notified that the write is completed without waiting for the end of the actual off-chip wr
the right to use the internal bus is not held, as when it is being used by the DMAC or th
is a wait until the bus is acquired before the CPU is notified of completion.

Accesses to cache-through areas and on-chip peripheral modules work the same as in th
area, except for the cache search and write.

Because the bus state controller has one level of write buffer, the internal bus can be us
another access even when the chip-external bus cycle has not ended. After a write has b
performed to low-speed memory off the chip, performing a read or write with an on-chi
peripheral module enables an access to the on-chip peripheral module without having to
the completion of the write to low-speed memory.

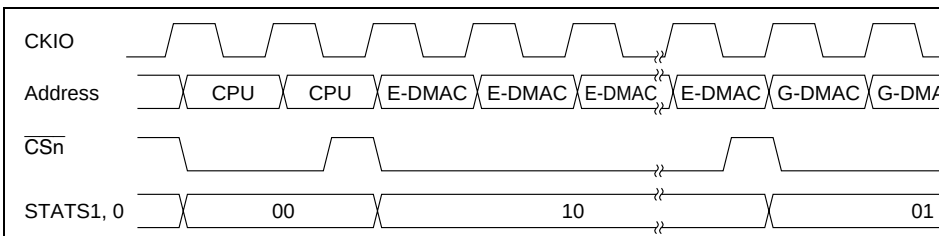
During reads, the CPU always has to wait for the end of the operation. To immediately
processing after checking that the write to the device of actual data has ended, perform
read access to the same address consecutively to check that the write has ended.

The bus state controller's write buffer functions in the same way during accesses from t
A dual-address DMA transfer thus starts in the next read cycle without waiting for the e
write cycle. When both the source address and destination address of the DMA are exte
to the chip, however, it must wait until the completion of the previous write cycle befor
the next read cycle.

encoding patterns are shown in table 7.9, and the output timing in figure 7.59.

Table 7.9 Encoding Patterns

Identification	STATS1	STATS0
CPU	0	0
DMAC		1
E-DMAC	1	0
Others		1



Note: In on-chip I/O → on-chip RAM or on-chip I/O → memory transfers using the DMAC, accesses to on-chip I/O and on-chip RAM are included in the “Others” category.

Figure 7.59 STATS Output Timing

7.10.4 $\overline{\text{BUSHiZ}}$ Specification

The $\overline{\text{BUSHiZ}}$ pin is needed when the SH7615 is connected to a PCI controller via a PC
and the PCI master and SH7615 share local memory on the SH7615 bus. By using this
combination with the $\overline{\text{WAIT}}$ pin, it is possible to place the bus and specific control sig
high-impedance state while keeping the SH7615's internal state halted. The conditions
establishing the high-impedance state, the applicable pins, and the bus timing (figure 7
shown below. See the Application Note for an example of PCI bridge connection.

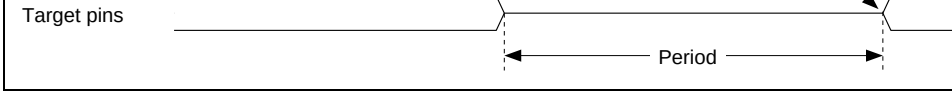


Figure 7.60 $\overline{\text{BUSHiZ}}$ Bus Timing

1. Can be used when memory is shared by the CPU and an external device.
2. When $\overline{\text{BUSHiZ}}$ is asserted after asserting $\overline{\text{WAIT}}$, the CPU appears to release the bus
3. When it becomes possible to access the shared memory, $\overline{\text{BUSHiZ}}$ is negated.
4. When the data is ready, $\overline{\text{WAIT}}$ is negated.

This procedure allows the CPU and an external device to share memory.

7.11 Usage Notes

7.11.1 Normal Space Access after Synchronous DRAM Write when Using DMA

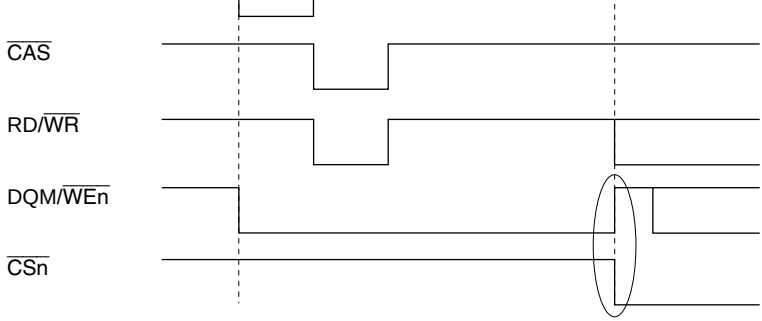
Negation of the DQMn/$\overline{\text{WEn}}$ signal in a synchronous DRAM write and $\overline{\text{CSn}}$ assertion i
immediately following normal space access both occur at the same rising edge of CKIO
7.61). As there is a risk of an erroneous write to normal space in this case, when synchr
DRAM or a high-speed device is connected to normal space, it is recommended that $\overline{\text{CS}}$
delayed on the system side.

Cases in which a synchronous DRAM write and normal space access occur consecutive
shown in table 7.10.

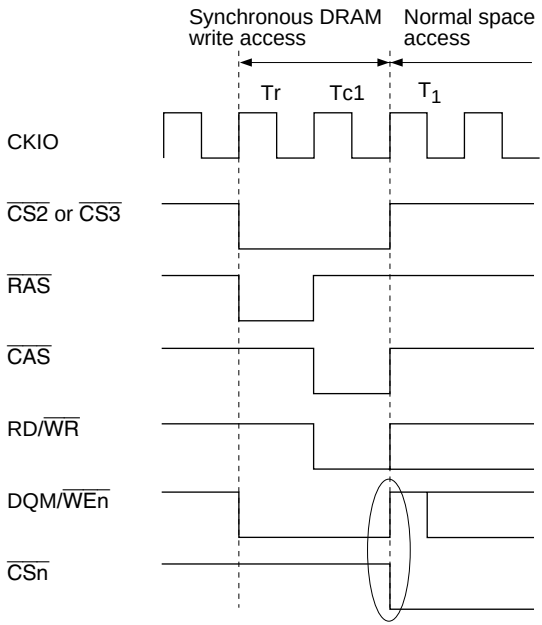
Table 7.10 Access Sequence

Write to Synchronous DRAM	Normal Space Access
CPU	DMA
DMA	CPU
DMA	DMA

Note: When an access by the CPU is performed immediately after a write by the CPU,
the accesses are not consecutive.



(a) Burst write mode



(b) Single write mode

Figure 7.61 Normal Space Access Immediately after Synchronous DRAM

In SDRAM burst write mode and bank active mode, wrong data may be output to SDR
the Ethernet controller direct memory access controller (E-DMAC) performs DMA rec
using SDRAM as the receive buffer, when the direct memory access controller (DMAC
16-bit transmission to SDRAM (destination address), or when the cache controller perf
back to SDRAM.

Conditions: When all of the following conditions are satisfied, the previous data writte
SDRAM is erroneously output to the SDRAM as the first four bytes of the 16-byte SDI
data.

- The clock ratio of external clock (Eϕ):internal clock (Iϕ) is not set to 1:1.
- SDRAM burst write mode is used.
- SDRAM bank active mode is used.
- The E-DMAC performs DMA reception by using SDRAM as the receive buffer, the
  performs 16-byte transfer (source address = on-chip memory or on-chip peripheral
  space, and destination address = SDRAM), or the cache controller performs write-b
  SDRAM.

Countermeasures: This problem in SDRAM burst write mode is avoided by any of the
countermeasures.

- Set the clock ratio of external clock (Eϕ): internal clock (Iϕ) to 1:1.
- Specify SDRAM auto-precharge mode.

supported for cache operation.

Each line of cache memory consists of 16 bytes. Cache memory is always updated in l
Four 32-bit accesses are required to update a line. Since the number of entries is 64, th
(A9 to A4) in each address determine the entry. A four-way set associative configurati
so up to four different instructions/data can be stored in the cache even when entry ad
match. To efficiently use four ways having the same entry address, replacement is pro
on a pseudo-LRU (least-recently used) replacement algorithm.

The cache configuration is shown in figure 8.1, and addresses in figure 8.2.

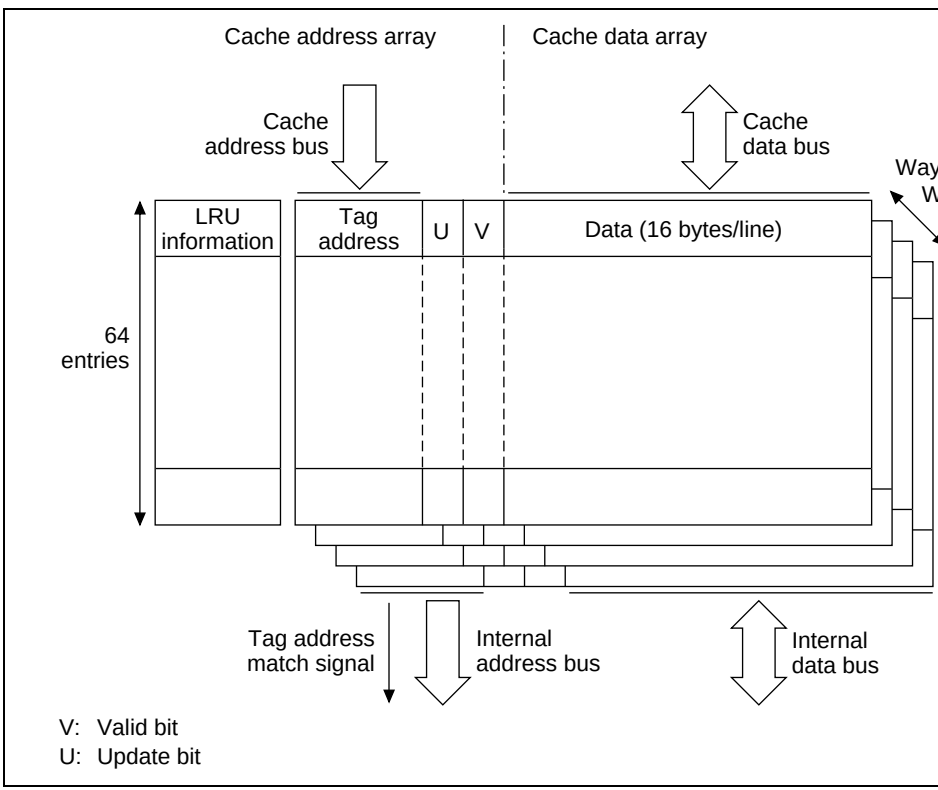


Figure 8.1 Cache Configuration

8.1.1 Register Configuration

Table 8.1 shows the cache register configuration.

Table 8.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Addres
Cache control register	CCR	R/W	H'00	H'FFFF

8.2 Register Description

8.2.1 Cache Control Register (CCR)

The cache control register (CCR) is used for cache control. CCR must be set and the ca
be initialized before use. CCR is initialized to H'00 by a power-on reset or manual reset

Bit:	7	6	5	4	3	2	1
	W1	W0	WB	CP	TW	OD	ID
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Way Specification Bit (W1, W0): W1 and W0 specify the way when an a
array is directly accessed by address specification.

Bit 7: W1	Bit 6: W0	Description
0	0	Way 0 (In
	1	Way 1
1	0	Way 2
	1	Way 3

and LRU information of all ways are initialized to 0. After initialization is completed,
reverts to 0. The CP bit always reads 0. Read the cache to check if initialization is com

Bit 4: CP	Description
0	Normal operation (
1	Cache purge

Note: Always read 0.

Bit 3—Two-Way Mode (TW): TW is the two-way mode bit. The cache operates as a
associative cache when TW is 0 and as a two-way set associative cache and 2-kbyte R
TW is 1. In the two-way mode, ways 2 and 3 are cache and ways 0 and 1 are RAM. W
are read or written by direct access of the data array according to address space specif

Bit 3: TW	Description
0	Four-way mode (
1	Two-way mode

Bit 2—Data Replacement Disable Bit (OD): OD is the bit for disabling data replacem
this bit is 1, data fetched from external memory is not written to the cache even if ther
miss. Cache data is, however, read or updated during cache hits. OD is valid only whe

Bit 2: OD	Description
0	Normal operation (
1	Data not replaced even when cache miss occurs in data access

Bit 0—Cache Enable Bit (CE): CE is the cache enable bit. Cache can be used when CE

Bit 0: CE	Description
0	Cache disabled (In
1	Cache enabled

8.3 Address Space and the Cache

The address space is divided into six partitions. The cache access operation is specified addresses. Table 8.2 lists the partitions and their cache operations. For more informatio address spaces, see section 7, Bus State Controller. Note that the spaces of the cache ar cache-through area are the same.

Table 8.2 Address Space and Cache Operation

Addresses A31 to A29	Partition	Cache Operation
000	Cache area	Cache is used when the CE bit in C
001	Cache-through area	Cache is not used
010	Associative purge area	Cache line of the specified address (disabled)
011	Address array read/write area	Cache address array is accessed d
110	Data array read/write area	Cache data array is accessed direc
111	I/O area	Cache is not used

output from the CPU match). In proper use, the tag addresses of each way differ from
and the tag address of only one way will match. When none of the way tag addresses
called a cache miss. Tag addresses of entries with valid bits of 0 will not match in any

When a cache hit occurs, data is read from the data array of the way that was matched
the entry address, the byte address within the line, and the access data size, and is sent
The address output on the cache address bus is calculated in the CPU's instruction exe
and the results of the read are written during the CPU's write-back stage. The cache ac
and cache data bus both operate as pipelines in concert with the CPU's pipeline structu
address comparison to data read requires 1 cycle; since the address and data operate as
consecutive reads can be performed at each cycle with no waits (figure 8.3).

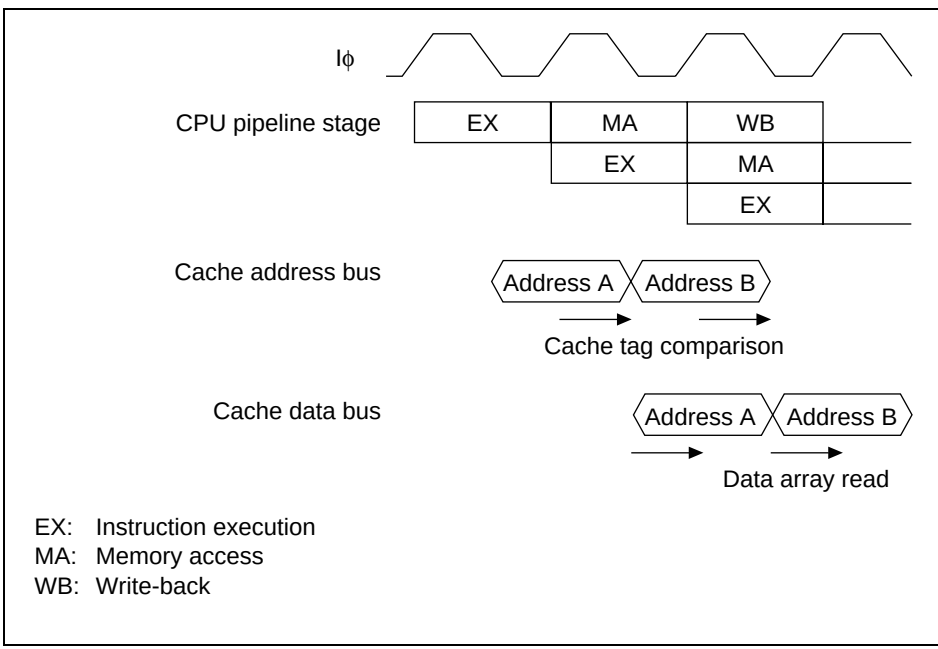


Figure 8.3 Read Access in Case of a Cache Hit

The internal address bus and internal data bus also function as pipelines, just like the ca
(figure 8.4).

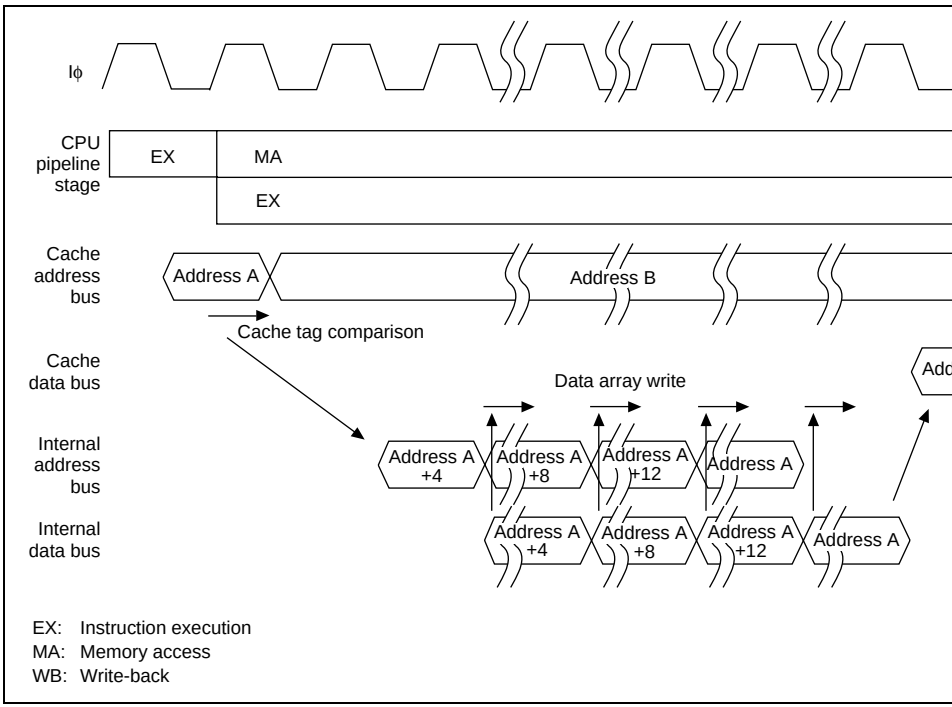


Figure 8.4 Read Access in Case of a Cache Miss

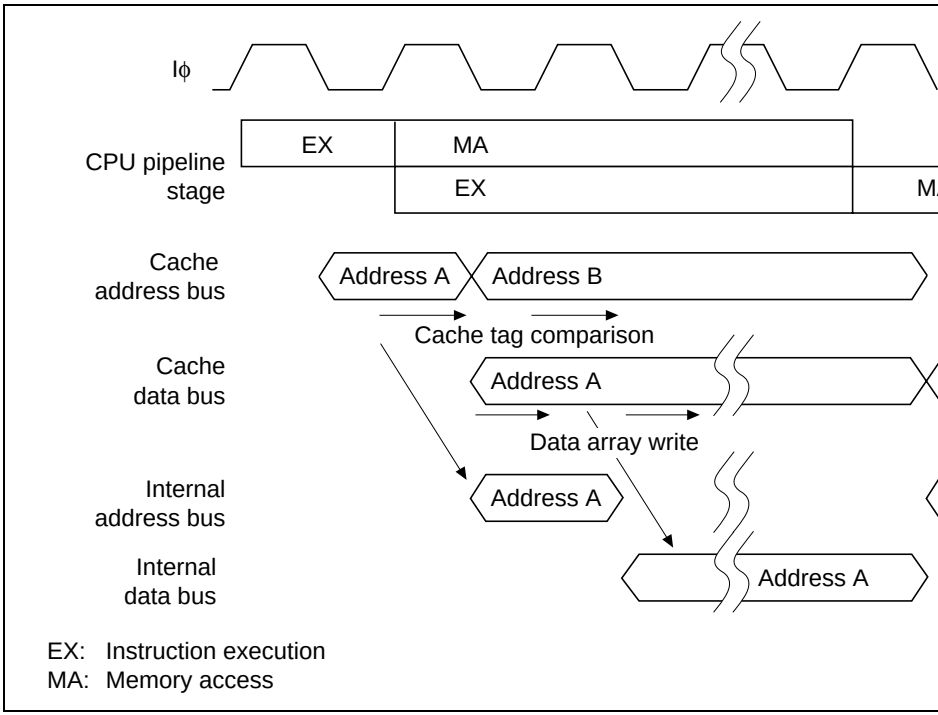


Figure 8.5 Write Access (Write-Through)

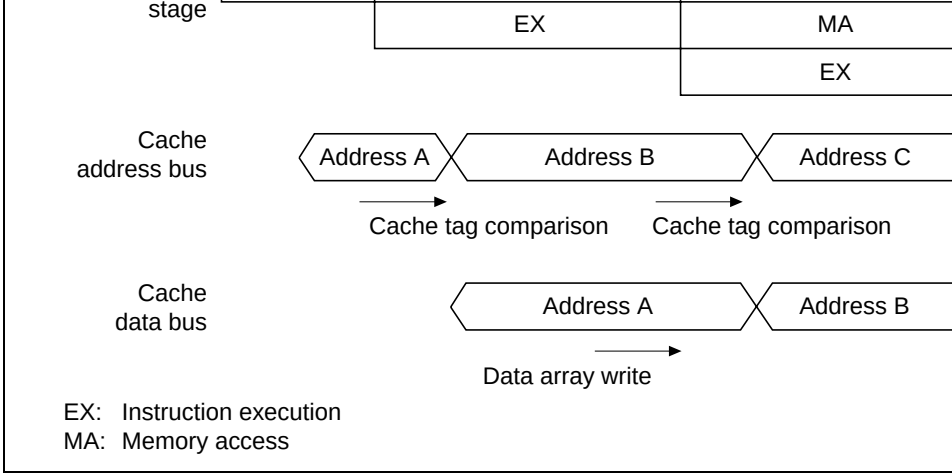


Figure 8.6 Write Access in Case of a Cache Hit (Write-Back)

When a cache miss occurs, the way for replacement is determined using the LRU infor
the write address from the CPU is written in the address array for that way. Simultaneo
valid bit and update bit are set to 1. Since the 16 bytes of data for replacing the data arr
simultaneously read when the data on the cache bus is written to the cache, the address
cache address bus is output to the internal address bus and 4 longwords are read consec
The access order is such that, for the address output to the internal address, the byte add
the line is sequentially incremented by 4, so that the longword that contains the address
from the cache comes last. The read data on the internal data bus is written sequentially
cache data array.

The internal address bus and internal data bus also function as pipelines, just like the ca
(figure 8.7).

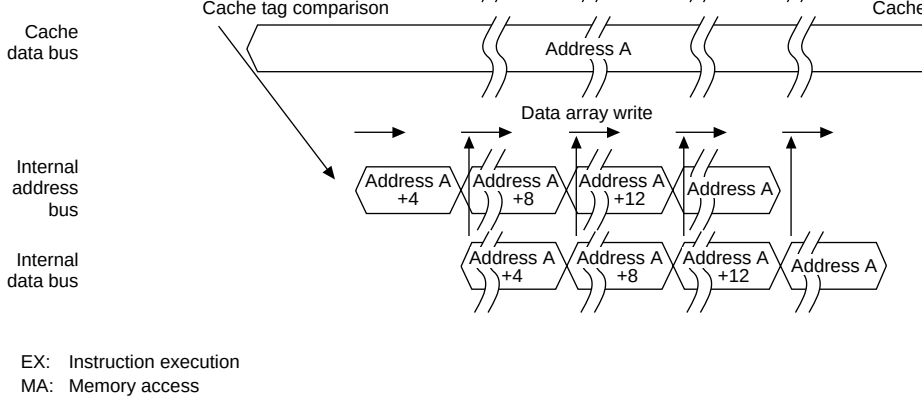


Figure 8.7 Write Access in Case of a Cache Miss (Write-Back)

When the update bit of an entry to be replaced in write-back mode is 1, write-back to
memory is necessary. To improve performance, the entry to be replaced is first transfe
write-back buffer, and fetching of the new entry into the cache is given priority over t
back. When the new entry has been fetched into the cache, the write-back buffer conte
written back to external memory. The cache can be accessed during this write-back.

The write-back buffer can hold one cache line (16 bytes) of data and its address. The c
of the write-back buffer is shown in figure 8.8.

A (31–4)	Longword 0	Longword 1	Longword 2	Longwor

A (31–4): Address for write-back to external memory
Longwords 0–3: One cache line of data for write-back to external memory

Figure 8.8 Write-Back Buffer Configuration

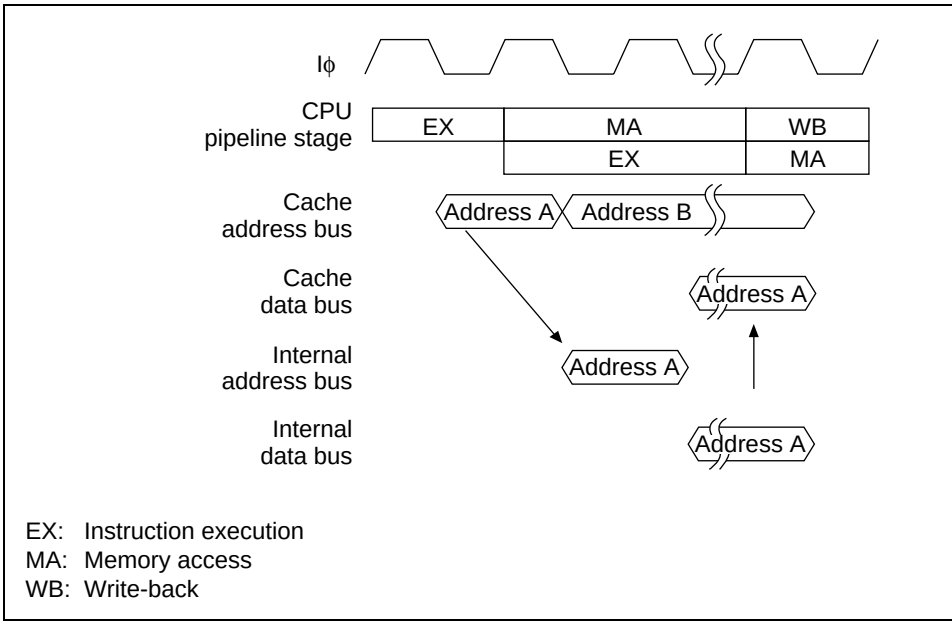


Figure 8.9 Reading Cache-Through Areas

8.4.4 The TAS Instruction

The TAS instruction reads data from memory, compares it to 0, reflects the result in the
the status register (SR), and sets the most significant bit to 1. It is an instruction that wr
same address. Accesses to the cache area are handled in the same way as ordinary data

8.4.5 Pseudo-LRU and Cache Replacement

When a cache miss occurs during a read, the data of the missed address is read from 1 l
bytes) of memory and replaced. It is therefore necessary to decide which of the four wa
replaced. It can generally be expected that a way that has been infrequently used recent
unlikely to be used next. This algorithm for replacing ways is called the least recently u
replacement algorithm, or LRU. The hardware to implement it, however, is complex. F

replacement occurs after a cache miss. Table 8.3 shows the rewrite values; table 8.4 sh
the way to be replaced is selected.

After a cache purge by means of the CP bit in CCR, all the LRU information is zeroiz
initial order of use is way 3 → way 2 → way 1 → way 0. Thereafter, the way is select
to the order of access in the program. Since the replacement will not be correct if the I
inappropriate value, the address array write function can be used to rewrite. When this
sure not to write a value other than 0 as the LRU information.

When the OD bit or ID bit in CCR is 1, cache replacement is not performed even if a c
occurs during data read or instruction fetch. Instead of replacing, the missed address d
and directly transferred to the CPU.

The two-way mode of the cache set by CCR's TW bit can only be implemented by rep
2 and 3. Comparisons of address array tag addresses are carried out on all four ways e
way mode, so the valid bits of ways 1 and 0 must be cleared to 0 before beginning ope
two-way mode.

Writing for the tag address and valid bit for cache replacement does not wait for the re
memory to be completed. If a memory access is aborted due to a reset, etc., during rep
there will be a discrepancy between the cache contents and memory contents, so a pur
performed.

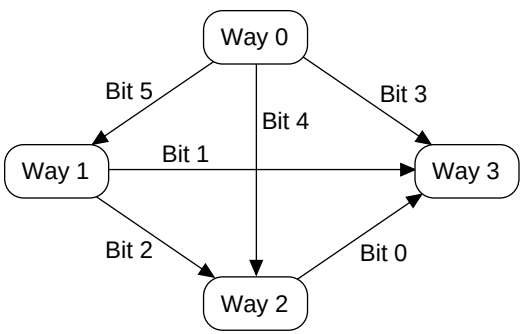


Figure 8.10 LRU Information and Access Sequence

Table 8.4 Selection Conditions for Replaced Way

	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	B
Way 0	1	1	1	—	—	—
Way 1	0	—	—	1	1	—
Way 2	—	0	—	0	—	1
Way 3	—	—	0	—	0	0

Note: —: Don’t care.

8.4.6 Cache Initialization

Purges of the entire cache area can only be carried out by writing 1 to the CP bit in CCI
1 to the CP bit initializes the valid bit of the address array, and all bits of the LRU infor
0. Cache purges are completed in 1 cycle, but additional time is required for writing to
Always initialize the valid bit and LRU before enabling the cache.

When the cache is enabled, instructions are read from the cache even during writing to
means that the prefetched instructions are read from the cache. To do a proper purge, w
CCR’s CE bit, then disable the cache and purge. Since CCR’s CE bit is cleared to 0 by
reset or manual reset, the cache can be purged immediately by a reset.

8.4.7 Associative Purges

Associative purges invalidate 1 line (16 bytes) corresponding to specific address conter
the contents are in the cache.

When the contents of a shared address are rewritten by one CPU in a multiprocessor co
or a configuration in which the chip's internal E-DMAC (or DMAC) and CPU share me
address must be invalidated in the cache of the other CPU if it is present there.

When writing to or reading the address obtained by adding H'40000000 to the address t
purged, the valid bit of the entry storing the address prior to addition are initialized to 0

Address	010	Tag address	Entry address	–
Number of bits	3	19	6	4

Figure 8.11 Associative Purge Access

8.4.8 Cache Flushing

When the CPU rewrites the contents of a specific shared address in the cache by write
multiprocessor configuration or a configuration in which the chip's internal E-DMAC
and CPU share memory, the rewritten data must be written back to the main memory,
cache contents invalidated, before the bus is granted by the CPU in the chip to another
(external master, E-DMAC, or DMAC). The chip does not support an instruction or p
flushing the contents of specific addresses, so in order to execute a cache flush it is ne
perform reads in a 4-kbyte space (cache area) other than the address space to be flushe
cache, and intentionally create cache misses. For this purpose, cache accesses should b
every 16 bytes. By this means, write-backs are generated and the contents written to th
the CPU in the chip are written back to the main memory, enabling flushing to be exe
However, this method incurs an overhead consisting of the cache fill time due to read
the time for rereading data to be left in the cache. Therefore, if the overhead due to us
write-back method is of concern when constructing a system in which a number of ma
memory, the shared area should be made a cache-through area in order to maintain co

8.4.9 Data Array Access

The cache data array can be read or written directly via the data array read/write area.
or longword access can be used on the data array. Data array accesses are completed i
a read and 2 cycles for a write. Since only the cache bus is used, the operation can pro
parallel even when another master, such as the DMAC, is using the bus. The data arra
mapped on H'C0000000 to H'C00003FF, way 1 on H'C0000400 to H'C00007FF, way
H'C0000800 to H'C0000BFF and way 3 on H'C0000C00 to H'C0000FFF. When the tw
mode is being used, the area H'C0000000 to H'C00007FF is accessed as 2 kbytes of o
RAM. When the cache is disabled, the area H'C0000000 to H'C0000FFF can be used
of on-chip RAM.

Bit	31		
Data		Data	
Number of bits		32	

BA: Byte address within line W: Way specification

Figure 8.12 Data Array Access

8.4.10 Address Array Access

The address array of the cache can be accessed so that the contents fetched to the cache
checked for purposes of program debugging or the like. The address array is mapped o
H'60000000 to H'600003FF. Since all of the ways are mapped to the same addresses, w
selected by rewriting the W1 and W0 bits in CCR. The address array can only be acces
longwords.

When the address array is read, the tag address, LRU information, and valid bit are out
When the address array is written to, the tag address, and valid bit are written from the
address bus. The write address must therefore be calculated before the write is perform
information is written from the cache data bus, but 0 must always be written to prevent
malfunctions.

Address array write:

Bit	31 29	28 10	9 4	3	2	1	0
Address	011	Tag address	Entry address	—	V	—	
Number of bits	3	19	6	1	1	2	

Bit	31 10	9
Data	—	L infor
Number of bits	22	6

V: Valid bit

Figure 8.13 Address Array Access

8.5 Cache Use

8.5.1 Initialization

Cache memory is not initialized in a reset. Therefore, the cache must be initialized by
before use. The cache is initialized by zeroizing all address array valid bits and LRU i
The address array write function can be used to initialize each line, but it is simpler to
once by writing 1 to the CP bit in CCR. Figure 8.14 shows how to initialize the cache.

```
MOV.W  #H'FE92, R1
MOV.B  @R1, R0      ;
AND    #H'FE, R0    ;
MOV.B  #R0, @R1     ; Cache disable
OR     #H'10, R0
MOV.B  R0, @R1      ; Cache purge
OR     #H'01, R0
MOV.B  R0, R1       ; Cache enable
```

Figure 8.14 Cache Initialization

An associative purge is used to purge specific lines. Since cache lines are 16 bytes long
performed in a 16-byte units. The four ways are checked simultaneously, and only lines
data corresponding to specified addresses are purged. When addresses do not match, th
the specified address is not fetched to the cache, so no purge occurs.

```
; Purging 32 bytes from address R3
MOV.L      #H'40000000, R0
XOR        R1, R1
MOV.L      R1, @(R0, R3)
ADD        #16, R3
MOV.L      R1, @(R0, R3)
```

Figure 8.15 Purging Specific Addresses

When it is troublesome to purge the cache after every DMA transfer, it is recommended
OD bit in CCR be set to 1 in advance. When the OD bit is 1, the cache operates as cach
only for instructions. However, when data is already fetched into cache memory, specif
cache memory must be purged for DMA transfers.

8.5.3 Cache Data Coherency

The cache memory does not have a snoop function. This means that when data is share
bus master other than the CPU, software must be used to ensure the coherency of data.
purpose, the cache-through area can be used, or a cache purge can be performed with p
logic using write-through.

When the cache-through area is to be used, the data shared by the bus masters is placed
cache-through area. This makes it easy to maintain data coherency, since access of the
through area does not fetch data into the cache. When the shared data is accessed repea
the frequency of data rewrites is low, a lower access speed can adversely affect perform

To purge the cache using program logic, the data updates are detected by the program f
cache is then purged. For example, if the program inputs data from a disk, whenever rea
unit (such as a sector) is completed, the buffer address used for reading or the entire ca

purged. When the update unit is larger, it is faster to purge the entire cache rather than
the addresses in order, and then read the data that previously existed in the cache again
external memory.

When write-back is used, coherency can be maintained by executing write-backs (flus
memory by means of intentional cache miss reads, but since executing flushing incurs
overhead, use of write-through or accessing the cache-through area is recommended in
which a number of masters share memory.

8.5.4 Two-Way Cache Mode

The 4-kbyte cache can be used as 2-kbyte RAM and 2-kbyte mixed instruction/data ca
by setting the TW bit in CCR to 1. Ways 2 and 3 become cache, and ways 0 and 1 bec

Initialization is performed by writing 1 to the CP bit in CCR, in the same way as with
valid bit, and LRU bits are cleared to 0.

When LRU information is initialized to zero, the initial order of use is way 3 → way 2
way 3 or way 2 is selected for replacement in accordance with the LRU information. T
conditions for updating the LRU information are the same as for four-way mode, exce
number of ways is two.

When designated as 2-kbyte RAM, ways 0 and 1 are accessed by data array access. Fi
shows the address mapping.

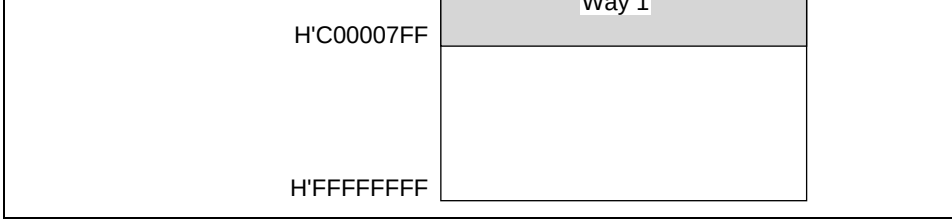


Figure 8.16 Address Mapping of 2-kbyte RAM in the Two-Way Mode

8.6 Usage Notes

8.6.1 Standby

Disable the cache before entering the standby mode for power-down operation. After re
from standby, initialize the cache before use.

8.6.2 Cache Control Register

Changing the contents of CCR also changes cache operation. The chip makes full use o
operations, so it is difficult to synchronize access. For this reason, change the contents
control register simultaneously when disabling the cache or after the cache is disabled.
changing the CCR contents, perform a CCR read.

receive Ethernet DMACs (E-DMACs) in the SH7615, and carries out high-speed data
and from memory.

9.1.1 Features

The EtherC has the following features:

- Transmission and reception of Ethernet/IEEE802.3 frames
- Supports 10/100 Mbps 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

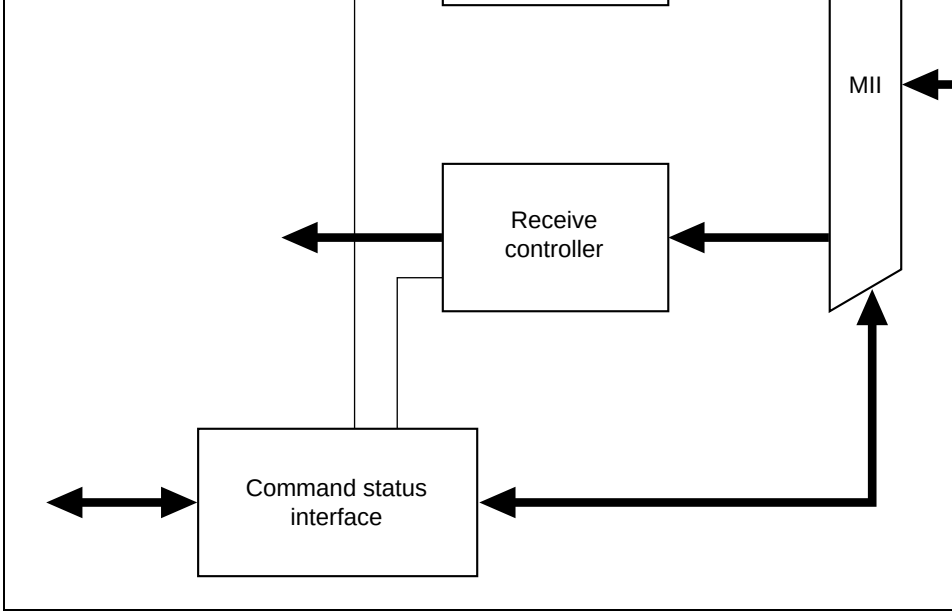


Figure 9.1 Configuration of Ethernet Controller (EtherC)

Transmit Controller: Transmit data is stored in the transmit FIFO from memory via th
E-DMAC. The transmit controller assembles this data into an Ethernet/IEEE802.3 fram
outputs to the MII. After passing through the MII, the transmit data is sent onto the line
LSI. The main functions of the transmit controller are as follows:

- Frame assembly and transmission
- CRC calculation and provision to frames
- Data retransmission in case of a collision (up to 15 times)
- Compliant with MII in IEEE802.3u standard
- Byte-nibble conversion supporting PHY-LSI speed

- Nibble-byte conversion supporting PHY-LSI speed
- Magic Packet monitoring

Command/Status Interface: This interface provides various command/status registe
the EtherC, and performs access to PHY-LSI internal registers via the MII.

MII	TX-CLK	Transmit clock	Input	TX-EN, ETXD0 to ETXD3, TX-ER reference signal
	RX-CLK	Receive clock	Input	RX-DV, ERXD0 to ERXD3, RX-EI reference signal
	TX-EN	Transmit enable	Output	Indicates that transmit data is rea ETXD0 to ETXD3
	ETXD0 to ETXD3	Transmit data (4-bit)	Output	4-bit transmit data
	TX-ER	Transmit error	Output	Notifies PHY-LSI of error during tr
	RX-DV	Receive data valid	Input	Indicates that there is valid receiv ERXD0 to ERXD3
	ERXD0 to ERXD3	Receive data (4-bit)	Input	4-bit receive data
	RX-ER	Receive error	Input	Identifies error state occurring dur reception
	CRS	Carrier detect	Input	Carrier detection signal
	COL	Collision detect	Input	Collision detection signal
	MDC	Management data clock	Output	Reference clock signal for informa transfer via MDIO
	MDIO	Management data input/output	Input/ output	Bidirectional signal for exchange o management information between PHY
Other	LNKSTA	Link status	Input	Inputs link status from PHY-LSI
	EXOUT	General-purpose external output	Output	External output pin
	WOL	Wake-On-LAN	Output	Magic packet reception

EtherC status register	ECSR	R/W	H'00000000	H'F
EtherC interrupt permission register	ECSIPR	R/W	H'00000000	H'F
PHY interface register	PIR	R/W	H'0000000X	H'F
MAC address high register	MAHR	R/W	H'00000000	H'F
MAC address low register	MALR	R/W	H'00000000	H'F
Receive flame length register	RFLR	R/W	H'00000000	H'F
PHY status register	PSR	R	H'00000000	H'F
Transmit retry over counter register	TROCR	R/W*2	H'00000000	H'F
Collision detect counter register	CDCR	R/W*2	H'00000000	H'F
Lost carrier counter register	LCCR	R/W*2	H'00000000	H'F
Carrier not detect counter register	CNDCR	R/W*2	H'00000000	H'F
Illegal frame length counter register	IFLCR	R/W*2	H'00000000	H'F
CRC error frame receive counter register	CEFCR	R/W*2	H'00000000	H'F
Frame receive error counter register	FRECR	R/W*2	H'00000000	H'F
Too-short frame receive counter register	TSFRCR	R/W*2	H'00000000	H'F
Too-long frame receive counter register	TLFRCR	R/W*2	H'00000000	H'F
Residual-bit frame counter register	RFCR	R/W*2	H'00000000	H'F
Multicast address frame counter register	MAFCR	R/W*2	H'00000000	H'F

Notes: All registers must be accessed as 32-bit units.
Reserved bits in a register should only be written with 0.
The value read from a reserved bit is not guaranteed.
1. Individual bits are cleared by writing 1.
2. Cleared by a write to the register.

Bit:	15	14	13	12	11	10	9
	—	—	—	PRCEF	—	—	MPDE
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W

Bit:	7	6	5	4	3	2	1
	—	RE	TE	—	ILB	ELB	DM
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R/W	R/W	R/W

The EtherC mode register specifies the operating mode of the Ethernet controller. The s
this register are normally made in the initialization process following a reset.

Note: Operating mode settings must not be changed while the transmitter and receive
enabled. To modify the operating mode settings or change the operating mode
EtherC is running, first return the EtherC and E-DMAC modules to their initial
means of the software reset bit (SWR) in the E-DMAC mode register (EDMR),
new settings.

Bits 31 to 13—Reserved: These bits are always read as 0. The write value should alway

Bit 12—Permit Receive CRC Error Frame (PRCEF): Specifies the treatment of a recei
containing a CRC error.

Bit 12: PRCEF	Description
0	Reception of a frame with a CRC error is treated as an error (In
1	Reception of a frame with a CRC error is not treated as an error

Note: When this bit is set to 1, the CRC error frame counter register (CEFCR: see sect
is not incremented when a CRC error is detected.

1	Magic Packet detection is enabled

Bits 8 and 7—Reserved: These bits are always read as 0. The write value should alwa

Bit 6—Receiver Enable (RE): Enables or disables the receiver.

Bit 6: RE	Description
0	Receiver is disabled (
1	Receiver is enabled

Note: If a switch is made from the receiver-enabled state (RE = 1) to the receiver-disa (RE = 0) while a frame is being received, the receiver will not be disabled until the frame is completed.

Bit 5—Transmitter Enable (TE): Enables or disables the transmitter.

Bit 5: TE	Description
0	Transmitter is disabled (
1	Transmitter is enabled

Note: If a switch is made from the transmitter-enabled state (TE = 1) to the transmitte state (TE = 0) while a frame is being transmitted, the transmitter will not be disa transmission of the frame is completed.

Bit 4—Reserved: This bit is always read as 0. The write value should always be 0.

Bit 3—Internal Loop Back Mode (ILB): Specifies loopback mode in the EtherC.

Bit 3: ILB	Description
0	Normal data transmission/reception is performed (
1	Data loopback is performed inside the EtherC

Note: A loopback mode specification can only be made with full-duplex transfer (DM = register).

Bit 1—Duplex Mode (DM): Specifies the EtherC transfer method.

Bit 1: DM	Description
0	Half-duplex transfer is specified (In
1	Full-duplex transfer is specified

Note: When internal loopback mode is specified (ILB = 1), full-duplex transfer (DM = 1) used.
The duplex mode information (half-duplex or full-duplex) detected by the PHY-LS set to the DM bit. If this setting does not match the duplex mode in the PHY-LSI, transfer rate may be degraded or a data collision may occur.

Bit 0—Promiscuous Mode (PRM): Setting this bit enables all Ethernet frames to be rec

Bit 0: PRM	Description
0	EtherC performs normal operation (In
1	EtherC performs promiscuous mode operation

Note: “All Ethernet frames” means all receivable frames, irrespective of differences or enabled/disabled status (destination address, broadcast address, multicast bit, e

	—	—	—	—	—	LCHNG	MPD
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W*	R/W*

Note: * The flag is cleared by writing 1. Writing 0 does not affect the flag.

The EtherC status register shows the internal status of the EtherC. This status informa
reported to the CPU by means of interrupts. Individual bits are cleared by writing 1 to
bits that generate an interrupt, the interrupt can be enabled or disabled by means of the
corresponding bit in the EtherC status interrupt permission register (ECSIPR).

Bits 31 to 3—Reserved: These bits are always read as 0. The write value should alway

Bit 2—LINK Signal Changed (LCHNG): Indicates that the LNKSTA signal input fro
LSI has changed from high to low, or from low to high. This bit is cleared by writing
Writing 0 to this bit has no effect.

Bit 2: LCHNG	Description
0	LNKSTA signal change has not been detected (
1	LNKSTA signal change (high-to-low or low-to-high) has been dete

Notes: 1. The current link status can be checked by referencing the LMON bit in the
interface status register (PSR).
2. Signal variation may be detected when the LNKSTA function is selected by
control register (PACR) of the pin function controller (PFC).

Bit 1—Magic Packet Detection (MPD): Indicates that a Magic Packet has been detect
line. This bit is cleared by writing 1 to it. Writing 0 to this bit has no effect.

Bit 1: MPD	Description
0	Magic Packet has not been detected (
1	Magic Packet has been detected

Bit 0—Illegal Carrier Detection (ICD): Indicates that PHY-LSI has detected an illegal
the line. This bit is cleared by writing 1 to it. Writing 0 to this bit has no effect.

Bit:	31	30	29	. . .	11	10	9
	—	—	—	. . .	—	—	—
Initial value:	0	0	0	. . .	0	0	0
R/W:	R	R	R	. . .	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	LCHNGIP	MPDIP
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W

This register enables or disables the interrupt sources indicated by the EtherC status reg
bit in this register enables or disables the interrupt indicated by the corresponding bit in
status register.

Bits 31 to 3—Reserved: These bits are always read as 0. The write value should always

Bit 2— LINK Signal Changed Interrupt Permission (LCHNGIP): Controls interrupt no
by the LINK Signal Changed bit.

Bit 2: LCHNGIP	Description
0	Interrupt notification by LCHNG bit in ECSR is disabled (Ir
1	Interrupt notification by LCHNG bit in ECSR is enabled

Bit 1—Magic Packet Detection Interrupt Permission (MPDIP): Controls interrupt notif
the Magic Packet Detection bit.

Bit 1: MPDIP	Description
0	Interrupt notification by MPD bit in ECSR is disabled (Ir
1	Interrupt notification by MPD bit in ECSR is enabled

Bit:	31	30	29	. . .	11	10	9
	—	—	—	. . .	—	—	—
Initial value:	0	0	0	. . .	0	0	0
R/W:	R	R	R	. . .	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	—	—	MDI	MDO	MMD
Initial value:	0	0	0	0	*	0	0
R/W:	R	R	R	R	R	R/W	R/W

Note: * Undefined

PIR provides a means of accessing PHY-LSI internal registers via the MII.

Bits 31 to 4—Reserved: These bits are always read as 0. The write value should alway

Bit 3— MII Management Data-In (MDI): Indicates the level of the MDIO pin.

Bit 2— MII Management Data-Out (MDO): Outputs the value set to this bit by the M
when the MMD bit is 1.

Bit 1— MII Management Mode (MMD): Specifies the data read/write direction with
MII. Read direction is indicated by 0, and write direction by 1.

Bit 0— MII Management Data Clock (MDC): Outputs the value set to this bit by the
and supplies the MII with the management data clock.

For the method of accessing MII registers, see section 9.3.4, Accessing MII Registers.

	MA39	MA38	MA37	MA36	MA35	MA34	MA33
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	MA31	MA30	MA29	MA28	MA27	MA26	MA25
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	MA23	MA22	MA21	MA20	MA19	MA18	MA17
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The upper 32 bits of the 48-bit MAC address are set in MARH. The setting in this regis
normally made in the initialization process after a reset.

Note: The MAC address setting must not be changed while the transmitter and receiv
enabled. First return the EtherC and E-DMAC modules to their initial state by r
the SWR bit in the E-DMAC mode register (EDMR), then make the new settin

Bits 31 to 0—MAC Address Bits 47 to 16 (MA47 to MA16): Used to set the upper 32 b
MAC address.

Note: If the MAC address to be set in the SH7615 is 01-23-45-67-89-AB (hexadecim
value set in this register is H'01234567.

	MA15	MA14	MA13	MA12	MA11	MA10	MA9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	MA7	MA6	MA5	MA4	MA3	MA2	MA1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The lower 16 bits of the 48-bit MAC address are set in MARL. The setting in this regi
normally made in the initialization process after a reset.

Note: The MAC address setting must not be changed while the transmitter and recei
enabled. First return the EtherC and E-DMAC modules to their initial state by
the SWR bit in the E-DMAC mode register (EDMR), then make the new setti

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alwa

Bits 15 to 0—MAC Address Bits 15 to 0 (MA15 to MA0): Used to set the lower 16 bi
MAC address.

Note: If the MAC address to be set in the SH7615 is 01-23-45-67-89-AB (hexadecim
value set in this register is H'000089AB.

	—	—	—	—	RFL11	RFL10	RFL9
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	RFL7	RFL6	RFL5	RFL4	RFL3	RFL2	RFL1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register specifies the maximum frame length (in bytes) that can be received by the

Bits 31 to 12—Reserved: These bits are always read as 0. The write value should alway

Bits 11 to 0—Receive Frame Length (RFL)

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

Notes: 1. The frame length refers to all fields from the destination address up to and in CRC data.

2. When data that exceeds the specified value is received, the part of the data higher than the specified value is discarded.

Frame contents from the destination address up to and including the data ar transferred to memory. CRC data is not included in the transfer.

	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

PSR enables interface signals from the PHY-LSI to be read.

Bits 31 to 1—Reserved: These bits are always read as 0. The write value should alway

Bit 0— Link Monitor (LMON): The link status can be read by connecting the LINK s
from the PHY-LSI. For information on the polarity, refer to the specifications for the
be connected.

Note: The LMON bit is cleared to 0 when the LNKSTA pin is at a high level, and is
when the pin is at a low level.

	TROC15	TROC14	TROC13	TROC12	TROC11	TROC10	TROC9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	TROC7	TROC6	TROC5	TROC4	TROC3	TROC2	TROC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TROCR is a 16-bit counter that indicates the number of frames that were unable to be t in 16 retransmission attempts. When 16 transmission attempts have failed, TROCR is i by 1. When the value in this register reaches H'FFFF (65,535), the count is halted. The value is cleared to 0 by a write to this register (the write value is immaterial).

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alway

Bits 15 to 0—Transmit Retry Over Count 15 to 0 (TROC15 to TROC0): These bits ind number of frames that were unable to be transmitted in 16 retransmission attempts.

	COLDC15	COLDC14	COLDC13	COLDC12	COLDC11	COLDC10	COLDC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	COLDC7	COLDC6	COLDC5	COLDC4	COLDC3	COLDC2	COLDC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CDCR is a 16-bit counter that indicates the number of collisions that occurred on the l
counting from a point 512 bits after the start of data transmission. When the value in t
reaches H'FFFF (65,535), the count is halted. The counter value is cleared to 0 by a wr
register (the write value is immaterial).

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alwa

Bits 15 to 0—Collision Detect Count 15 to 0 (COLDC15 to COLDC0): These bits ind
count of collisions from a point 512 bits after the start of data transmission.

	LCC15	LCC14	LCC13	LCC12	LCC11	LCC10	LCC9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	LCC7	LCC6	LCC5	LCC4	LCC3	LCC2	LCC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LCCR is a 16-bit counter that indicates the number of times the carrier was lost during transmission. When the value in this register reaches H'FFFF (65,535), the count is halt counter value is cleared to 0 by a write to this register (the write value is immaterial).

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alway

Bits 15 to 0—Lost Carrier Count 15 to 0 (LCC15 to LCC0): These bits indicate the nur times the carrier was lost during data transmission.

	CNDC15	CNDC14	CNDC13	CNDC12	CNDC11	CNDC10	CNDC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	CNDC7	CNDC6	CNDC5	CNDC4	CNDC3	CNDC2	CNDC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CNDCR is a 16-bit counter that indicates the number of times the carrier could not be
while the preamble was being sent. When the value in this register reaches H'FFFF (6
count is halted. The counter value is cleared to 0 by a write to this register (the write v
immaterial).

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alwa

Bits 15 to 0—Carrier Not Detect Count 15 to 0 (CNDC15 to CNDC0): These bits indi
number of times the carrier was not detected.

	IFLC15	IFLC14	IFLC13	IFLC12	IFLC11	IFLC10	IFLC9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	IFLC7	IFLC6	IFLC5	IFLC4	IFLC3	IFLC2	IFLC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IFLCR is a 16-bit counter that indicates the number of times transmission of a packet w
length of less than four bytes was attempted during data transmission. When the value i
register reaches H'FFFF (65,535), the count is halted. The counter value is cleared to 0
to this register (the write value is immaterial).

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alway

Bits 15 to 0—Illegal Frame Length Count 15 to 0 (IFLC15 to IFLC0): These bits indica
count of illegal frame length transmission attempts.

	CEFC15	CEFC14	CEFC13	CEFC12	CEFC11	CEFC10	CEFC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	CEFC7	CEFC6	CEFC5	CEFC4	CEFC3	CEFC2	CEFC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CEFCR is a 16-bit counter that indicates the number of times a frame with a CRC erro
received. When the value in this register reaches H'FFFF (65,535), the count is halted.
value is cleared to 0 by a write to this register (the write value is immaterial).

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alwa

Bits 15 to 0—CRC Error Frame Count 15 to 0 (CEFC15 to CEFC0): These bits indica
of CRC error frames received.

Note: When the Permit Receive CRC Error Frame bit (PRCEF) is set to 1 in the Eth
Register (ECMR), CEFCR is not incremented by reception of a frame with a C

	FREC15	FREC14	FREC13	FREC12	FREC11	FREC10	FREC9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	FREC7	FREC6	FREC5	FREC4	FREC3	FREC2	FREC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

FRECR is a 16-bit counter that indicates the number of frames input from the PHY-LSI a receive error was indicated by the RX-ER pin. FRECR is incremented each time this becomes active. When the value in this register reaches H'FFFF (65,535), the count is h counter value is cleared to 0 by a write to this register (the write value is immaterial).

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alway

Bits 15 to 0—Frame Receive Error Count 15 to 0 (FREC15 to FREC0): These bits indi count of errors during frame reception.

	TSFC15	TSFC14	TSFC13	TSFC12	TSFC11	TSFC10	TSFC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	TSFC7	TSFC6	TSFC5	TSFC4	TSFC3	TSFC2	TSFC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TSFRCR is a 16-bit counter that indicates the number of frames of fewer than 64 byte been received. When the value in this register reaches H'FFFF (65,535), the count is h counter value is cleared to 0 by a write to this register (the write value is immaterial).

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alwa

Bits 15 to 0—Too-Short Frame Receive Count 15 to 0 (TSFC15 to TSFC0): These bit the count of frames received with a length of less than 64 bytes.

	TLFC15	TLFC14	TLFC13	TLFC12	TLFC11	TLFC10	TLFC9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	TLFC7	TLFC6	TLFC5	TLFC4	TLFC3	TLFC2	TLFC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TLFRCR is a 16-bit counter that indicates the number of frames received with a length
the value specified by the receive frame length register (RFLR). When the value in this
reaches H'FFFF (65,535), the count is halted. The counter value is cleared to 0 by a wri
register (the write value is immaterial).

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alway

Bits 15 to 0—Too-Long Frame Receive Count 15 to 0 (TLFC15 to TLFC0): These bits
the count of frames received with a length exceeding the value in RFLR.

Notes: If the value specified by RFLR is 1518 bytes, TLFRCR is incremented by recep
frame with a length of 1519 bytes or more.

TLFRCR is not incremented when a frame containing residual bits is received.
case, the reception of the frame is indicated in the residual-bit frame counter reg
(RFCR).

	RFC15	RFC14	RFC13	RFC12	RFC11	RFC10	RFC9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	RFC7	RFC6	RFC5	RFC4	RFC3	RFC2	RFC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RFCR is a 16-bit counter that indicates the number of frames received containing resi
(less than an 8-bit unit). When the value in this register reaches H'FFFF (65,535), the
halted. The counter value is cleared to 0 by a write to this register (the write value is i

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alwa

Bits 15 to 0—Residual-Bit Frame Count 15 to 0 (RFC15 to RFC0): These bits indicat
of frames received containing residual bits.

	MAFC15	MAFC14	MAFC13	MAFC12	MAFC11	MAFC10	MAFC9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	MAFC7	MAFC6	MAFC5	MAFC4	MAFC3	MAFC2	MAFC1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MAFCR is a 16-bit counter that indicates the number of frames received with a multica
specified. When the value in this register reaches H'FFFF (65,535), the count is halted.
counter value is cleared to 0 by a write to this register (the write value is immaterial).

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alway

Bits 15 to 0—Multicast Address Frame Count 15 to 0 (MAFC15 to MAFC0): These bi
the count of multicast frames received.

Notes: 1. In actual EtherC operation, frame transmission and reception is performed continuously in combination with the E-DMACs. For details of continuous see the description of E-DMAC operation.

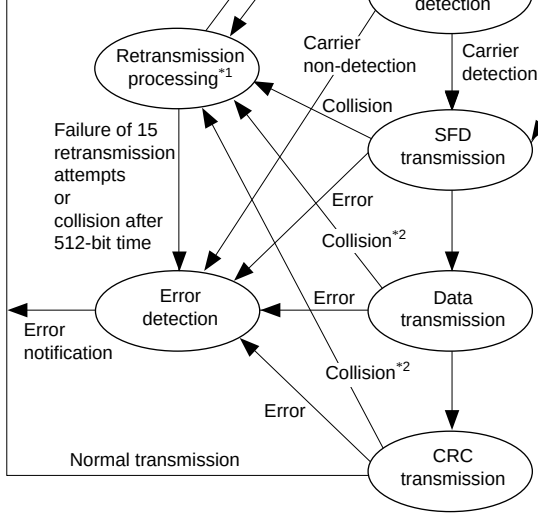
2. The receive data transferred to memory by the receive data E-DMAC does CRC data.

9.3.1 Transmission

The main transmit functions of the EtherC are as follows:

- Frame generation and transmission: Monitors the line status, then adds the preamb CRC to the data to be transmitted, and sends it to the MII
- CRC generation: Generates the CRC for the data field, and adds it to the transmit f
- Transmission retry: when a collision is detected in the collision window (during th transmission of the 512-bit data that includes the preamble and SFD from the start transmission), transmission is retried up to 15 times based on the back-off algorith

The state transitions of the EtherC transmitter are shown in figure 9.2.



Notes: 1. Transmission retry processing includes both jam transmission that depends on collision detection and
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 transm
When a collision is detected during the transmission of data greater than 512 bits, only jam is transmi
transmission based on the back-off algorithm is not retried.

Figure 9.2 EtherC Transmitter State Transitions

1. When the transmit enable (TE) bit is set, the transmitter enters the transmit idle stat
2. When a transmit request is issued by the transmit E-DMAC, the EtherC sends the p
after a transmission delay equivalent to the frame interval time.

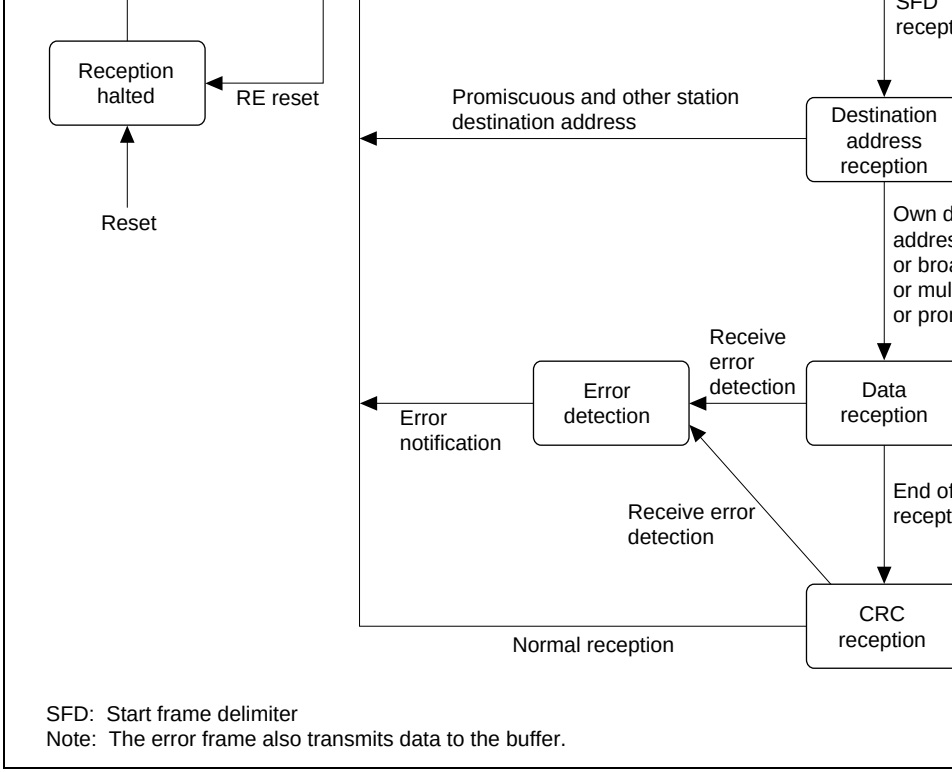
Note: If full-duplex transfer is selected, which does not require carrier detection, the p
sent as soon as a transmit request is issued by the transmit E-DMAC.

3. The transmitter sends the SFD, data, and CRC sequentially. At the end of transmiss
transmit E-DMAC generates a transmission complete interrupt (TC).

Note: If a collision or the carrier-not-detected state occurs during data transmission, t
reported as interrupt sources.

- Receive frame header check: Checks the preamble and SFD, and discards a frame
invalid pattern
- Receive frame data check: Checks the data length in the header, and reports an err
the data length is less than 64 bytes or greater than the specified number of bytes
- Receive CRC check: Performs a CRC check on the frame data field, and reports a
in the case of an abnormality
- Line status monitoring: Reports an error status if an illegal carrier is detected by m
fault detection signal from the PHY-LSI
- Magic Packet monitoring: Detects a Magic Packet from all receive frames

The state transitions of the EtherC receiver are shown in figure 9.3.



SFD: Start frame delimiter
Note: The error frame also transmits data to the buffer.

Figure 9.3 EtherC Receiver State Transitions

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 starts receive processing.
3. If the destination address matches the receiver's own address, or if broadcast or mul transmission or promiscuous mode is specified, the receiver starts data reception.
4. Following data reception, the receiver carries out a CRC check. The result is indicat status bit in the descriptor after the frame data has been written to memory.
5. After one frame has been received, if the receive enable bit is set (RE = 1) in the Et register, the receiver prepares to receive the next frame.

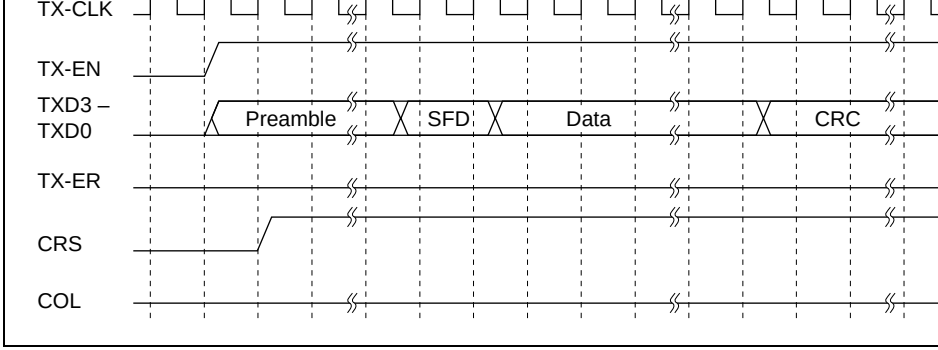


Figure 9.4 (a) MII Frame Transmit Timing (Normal Transmission)

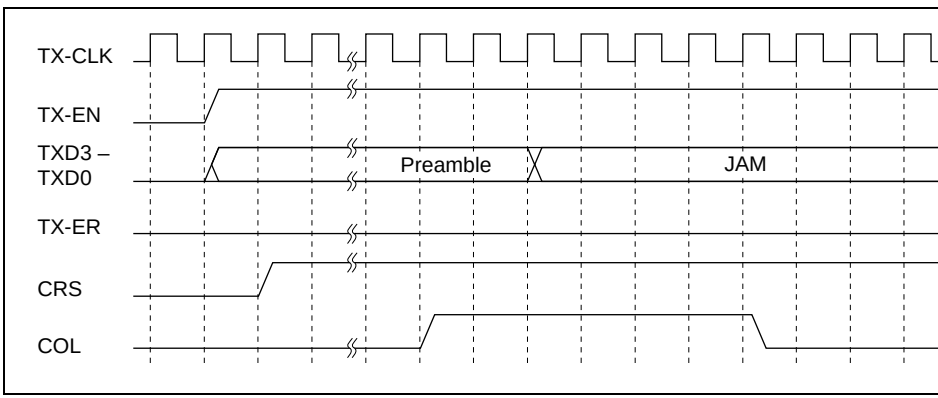


Figure 9.4 (b) MII Frame Transmit Timing (Collision)

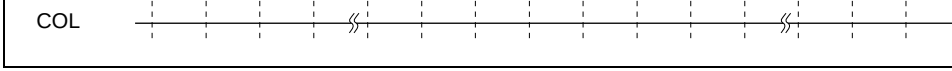


Figure 9.4 (c) MII Frame Transmit Timing (Transmit Error)

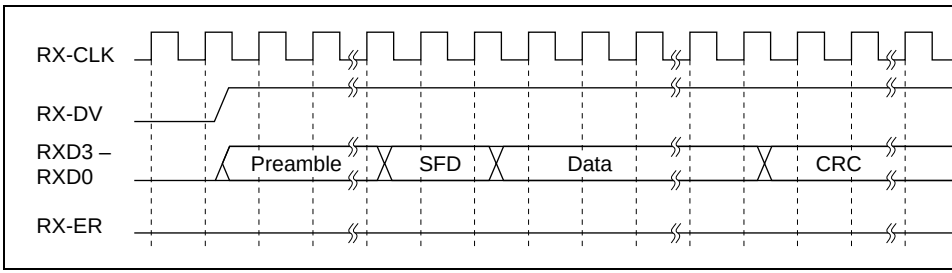


Figure 9.4 (d) MII Frame Receive Timing (Normal Reception)

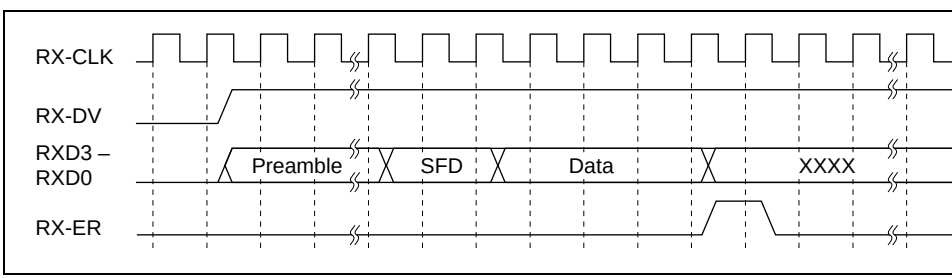


Figure 9.4 (e) MII Frame Receive Timing (Receive Error (1))

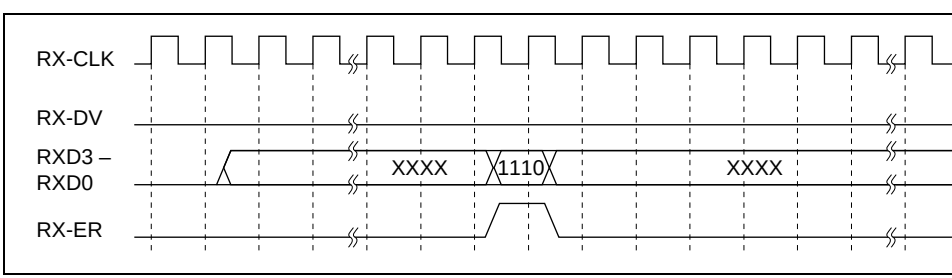


Figure 9.4 (f) MII Frame Receive Timing (Receive Error (2))

Access Type	MII Management Frame						
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

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 MS
This bit changes depending on the PHY-LSI address.

REGAD: Write of 0001 if the register address is 1 (sequential write starting with the MSI
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 9.5 MII Management Frame Format

MII Register Access Procedure: The program accesses MII registers via the PHY in
register (PIR). Access is implemented by a combination of 1-bit-unit data write, 1-bit-
read, bus release, and independent bus release. Examples 1 through 4 below show the
access timing. The timing will differ depending on the PHY-LSI type.

1. The MII register write procedure is shown in figure 9.6 (a).
2. The bus release procedure is shown in figure 9.6 (b).
3. The MII register read procedure is shown in figure 9.6 (c).

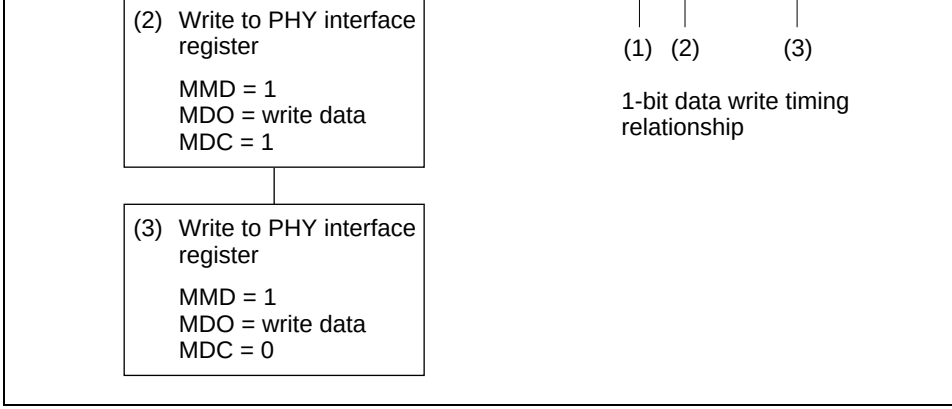


Figure 9.6 (a) 1-Bit Data Write Flowchart

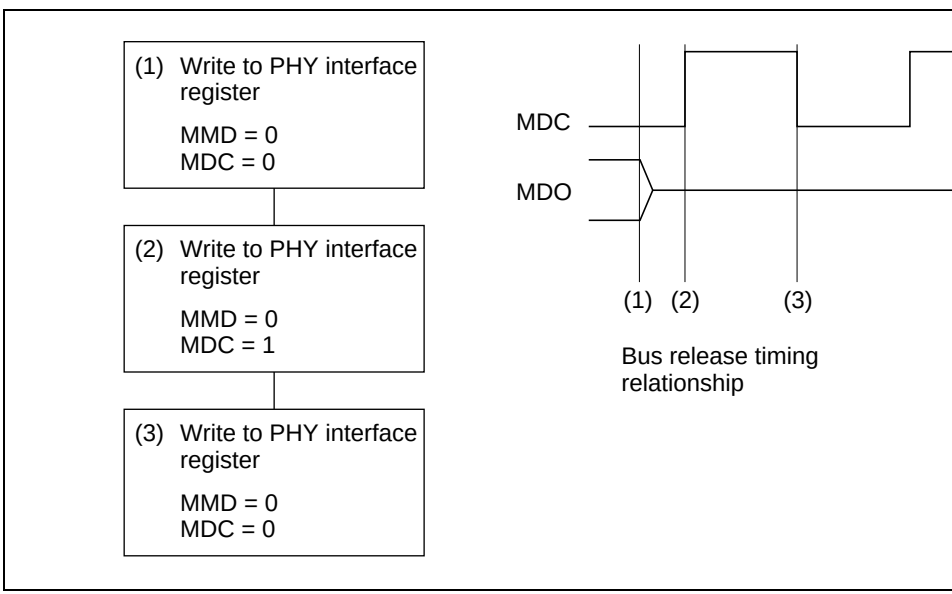


Figure 9.6 (b) Bus Release Flowchart (TA in Read in Figure 9.5)

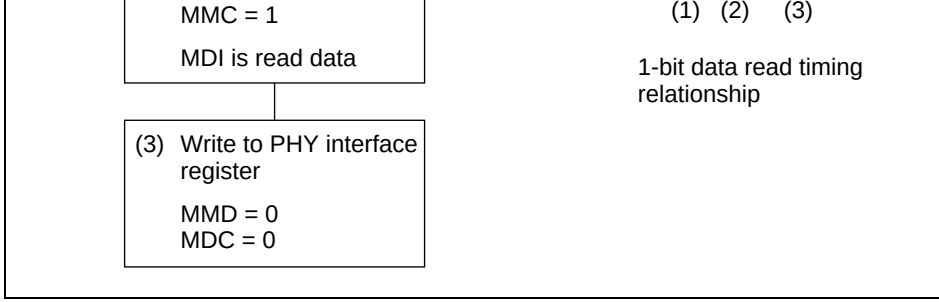


Figure 9.6 (c) 1-Bit Data Read Flowchart

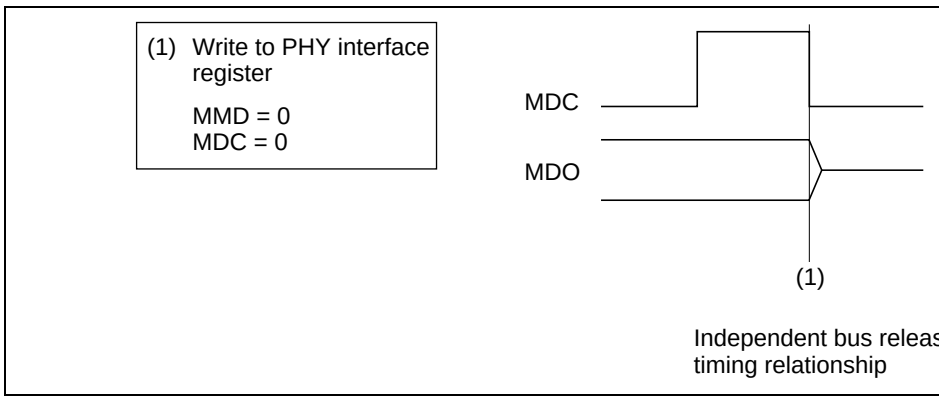


Figure 9.6 (d) Independent Bus Release Flowchart (IDLE in Write in Figu

1. Disable interrupt source output by means of the various interrupt enable/mask regis
2. Set the Magic Packet detection enable bit (MPDE) in the EtherC mode register (EC
3. Set the Magic Packet detection interrupt enable bit (MPDIP) in the EtherC interrupt register (ECSIPR) to the enable setting.
4. If necessary, set the CPU operating mode to sleep mode or set supporting functions standby mode.
5. When a Magic Packet is detected, an interrupt is sent to the CPU. The WOL pin not peripheral LSIs that the Magic Packet has been detected.

Notes: 1. When the Magic Packet is detected, data is stored in the receive FIFO by the packet that has received data previously and the EtherC is notified of the rec status. To return to normal operation from the interrupt processing, initialize EtherC and E-DMAC by using the software reset bit (SWR) in the E-DMAC register (EDMR).

2. With a Magic Packet, reception is performed regardless of the destination ad result, this function is valid, and the WOL pin enabled, only in the case of a the destination address specified by the format in the Magic Packet.

Sleep Mode: In sleep mode, the operation of the CPU and DSP is halted. The EtherC, supporting functions, and external pins continue to operate. Recovery from sleep mod carried out by means of an interrupt from the EtherC or a supporting module, or a rese control external pins and the WOL pin by means of Magic Packet reception, the releva be set beforehand.

Note: In order to specify recovery by means of a magic packet, supporting function sources should be masked before sleep mode is entered. See section 9.3.5, Ma Detection, for the setting procedure.

Standby Mode: In standby mode, the on-chip oscillation circuit is halted. Consequen is not supplied to the EtherC, and interrupts from the EtherC and other supporting mo be reported. It is therefore not possible to restore normal operation by these means, an WOL function cannot be used.

Notes: This mode can be selected to halt all functions including the EtherC. However interrupt, power-on reset, or manual reset is necessary in order to restore norm

When the SH7615 has been placed in standby mode, the CPU, DSP, and bus s controller are among the functions halted. When DRAM is connected, refresh halted, and therefore initialization of memory, etc., is necessary after recovery way as in a reset.

Module Standby Mode: Module standby mode allows individual supporting modules halted. However, due to the nature of its function, the operation of the EtherC cannot b During normal operation, module standby mode can be used to halt unnecessary suppo functions. The CPU and DSP continue to operate in this mode.

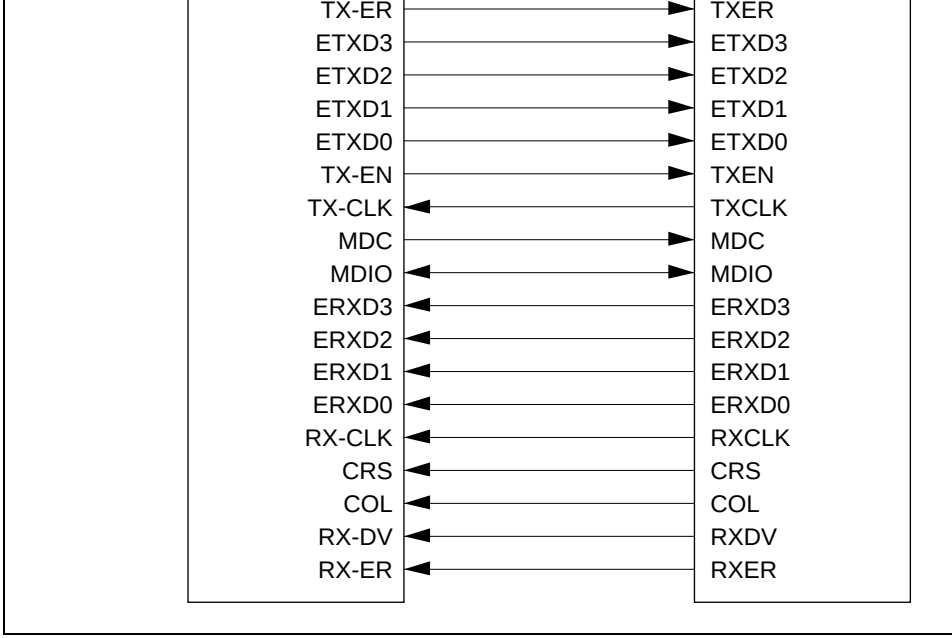


Figure 9.7 Example of Connection to AM79C873

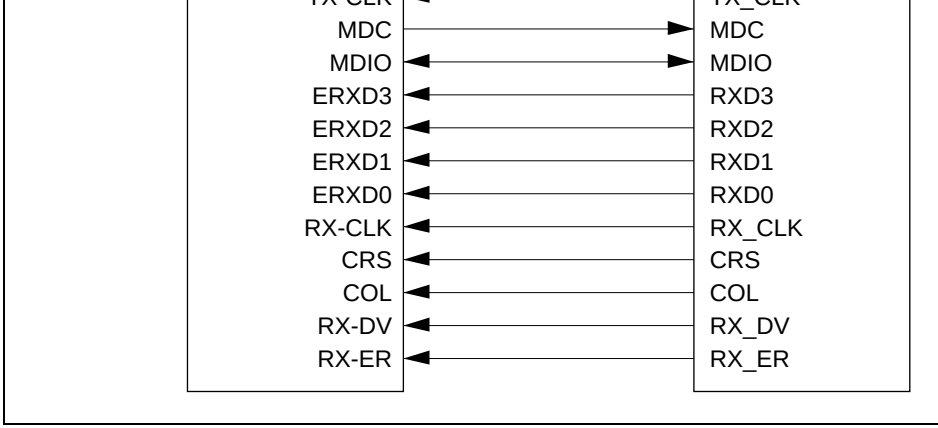
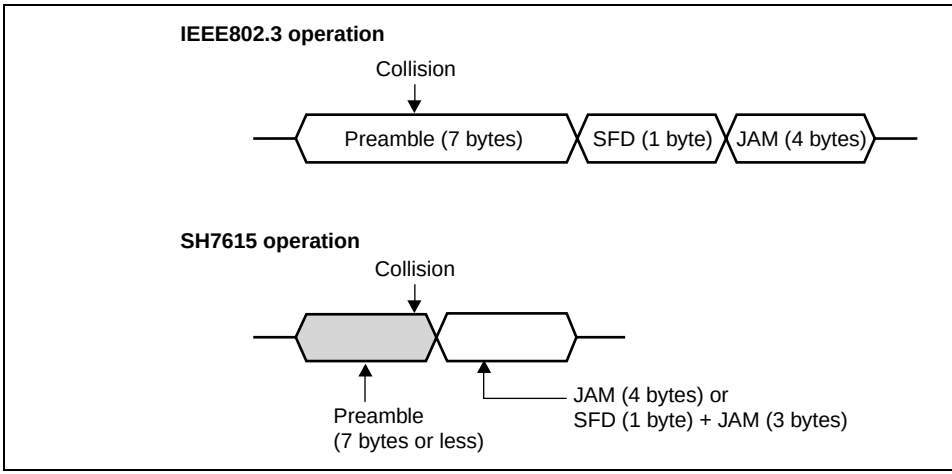


Figure 9.8 Example of Connection to DP83843



Influence on System: The other transmitting station in the collision cannot detect the c
and may mistake its data to be transmitted successfully on the MAC layer. In this case,
layer (such as TCP/IP) generally recovers the error by re-transmission to complete the c
transmission, though the efficiency of transmission may be degraded.

efficient data transfer control to be achieved.

10.1.1 Features

The E-DMAC has the following features:

- The load on the CPU is reduced by means of a descriptor management system
- Transmit/receive frame status information is indicated in descriptors
- Achieves efficient system bus utilization through the use of block transfer (16-byte
- Supports single-frame/multi-buffer operation

Note: The E-DMAC cannot handle transfers to on-chip RAM and supporting modul

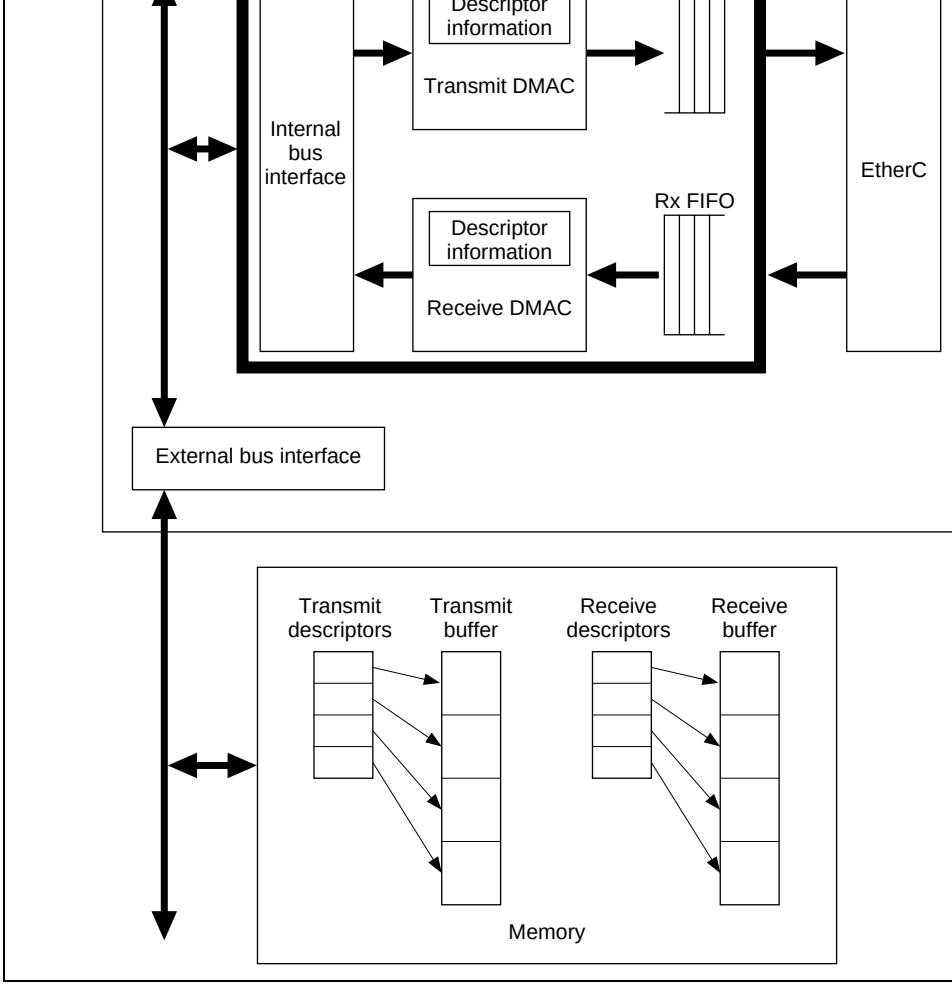


Figure 10.1 Configuration of E-DMAC, and Descriptors and Buffers

transmission can be carried out.

Reception: For each start of a receive DMA transfer, the receive E-DMAC fetches a r
buffer address from the top of the receive descriptor list. When receive data is stored i
FIFO, the E-DMAC transfers this data to the receive buffer. When reception of one fra
finished, the E-DMAC performs a receive status write and fetches the receive buffer a
the next descriptor. By repeating this sequence, consecutive frames can be received.

10.1.4 Register Configuration

The E-DMAC has the seventeen 32-bit registers shown in table 10.1.

Notes: 1. All registers must be accessed as 32-bit units.
2. Reserved bits in a register should only be written with 0.
3. The value read from a reserved bit is not guaranteed.

EtherC/E-DMAC status register	EESR	R/W*1	H'00000000	H'F
EtherC/E-DMAC status interrupt permission register	EESIPR	R/W	H'00000000	H'F
Transmit/receive status copy enable register	TRSCER	R/W	H'00000000	H'F
Receive missed-frame counter register	RMFCR	R/W*2	H'00000000	H'F
Transmit FIFO threshold register	TFTR	R/W	H'00000000	H'F
FIFO depth register	FDR	R/W	H'00000000	H'F
Receiver control register	RCR	R/W	H'00000000	H'F
E-DMAC operation control register	EDOCR	R/W	H'00000000	H'F
Receive buffer write address register	RBWAR	R	H'00000000	H'F
Receive descriptor fetch address register	RDFAR	R	H'00000000	H'F
Transmit buffer read address register	TBRAR	R	H'00000000	H'F
Transmit descriptor fetch address register	TDFAR	R	H'00000000	H'F

Notes: 1. Individual bits are cleared by writing 1.
2. Cleared by reading the register.

their initial state by means of the software reset bit (SWR) in this register, ther
settings.

Bit:	31	30	29	. . .	11	10	9
	—	—	—	. . .	—	—	—
Initial value:	0	0	0	. . .	0	0	0
R/W:	R	R	R	. . .	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	DL1	DL0	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R

Bits 31 to 6—Reserved: These bits are always read as 0. The write value should alway

Bits 5 and 4—Descriptor Length 1, 0 (DL1, DL0): These bits specify the descriptor le

Bit 5: DL1	**Bit 4: DL0**	**Description**
0	0	16 bytes (
	1	32 bytes
1	0	64 bytes
	1	Reserved (setting prohibited)

Bits 3 to 1—Reserved: These bits are always read as 0. The write value should always

Bit 0—Software Reset (SWR): The EtherC and E-DMAC can be initialized by softwa
bits should only be written with 0.

10.2.2 E-DMAC Transmit Request Register (EDTRR)

The E-DMAC transmit request register issues transmit directives to the E-DMAC.

Bit:	31	30	29	. . .	11	10	9
	—	—	—	. . .	—	—	—
Initial value:	0	0	0	. . .	0	0	0
R/W:	R	R	R	. . .	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 31 to 1—Reserved: These bits are always read as 0. The write value should always

Bit 0—Transmit Request (TR): When 1 is written to this bit, the E-DMAC reads a desc
in the case of an active descriptor, transfers the data in the transmit buffer to the EtherC

Bit 0: TR	Description
0	Transmission-halted state. Writing 0 does not stop transmission. Te transmission is controlled by the active bit in the transmit descriptor
1	Start of transmission. The relevant descriptor is read and a frame is the transmit active bit set to 1

Note: When transmission of one frame is completed, the next descriptor is read. If the descriptor active bit in this descriptor has the “active” setting, transmission is cor the transmit descriptor active bit has the “inactive” setting, the TR bit is cleared a operation of the transmit DMAC is halted.

For details on writing to the register, see section 10.4, Usage Notes.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 31 to 1—Reserved: These bits are always read as 0. The write value should alway

Bit 0—Receive Request (RR): When 1 is written to this bit, the E-DMAC reads a desc
then transfers receive data to the buffer in response to receive requests from the Ether

Bit 0: RR	**Description**
0	After frame reception is completed, the receiver is disabled
1	A receive descriptor is read, and transfer is enabled

Notes: In order to receive a frame in response to a receive request, the receive descri
bit in the receive descriptor must be set to “active” beforehand.

1. When the receive request bit is set, the E-DMAC reads the relevant receive
2. If the receive descriptor active bit in the descriptor has the “active” setting, t
prepares for a receive request from the EtherC.
3. When one receive buffer of data has been received, the E-DMAC reads the
descriptor and prepares to receive the next frame. If the receive descriptor
the descriptor has the “inactive” setting, the RR bit is cleared and operation
receive DMAC is halted.

For details on writing to the register, see section 10.4, Usage Notes.

Bit:	23	22	21	20	19	18	17
	TDLA23	TDLA22	TDLA21	TDLA20	TDLA19	TDLA18	TDLA17
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	TDLA15	TDLA14	TDLA13	TDLA12	TDLA11	TDLA10	TDLA9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	TDLA7	TDLA6	TDLA5	TDLA4	TDLA3	TDLA2	TDLA1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 31 to 0—Transmit Descriptor Start Address 31 to 0 (TDLA31 to TDLA0): These b
only be written with 0.

Notes: The lower bits are set as follows according to the specified descriptor length.
16-byte boundary: TDLA[3:0] = 0000
32-byte boundary: TDLA[4:0] = 00000
64-byte boundary: TDLA[5:0] = 000000

This register must not be written to during transmission. Modifications to this r
should only be made while transmission is disabled.

Bit:	23	22	21	20	19	18	17
	RDLA23	RDLA22	RDLA21	RDLA20	RDLA19	RDLA18	RDLA1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	RDLA15	RDLA14	RDLA13	RDLA12	RDLA11	RDLA10	RDLA
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	RDLA7	RDLA6	RDLA5	RDLA4	RDLA3	RDLA2	RDLA
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 31 to 0—Receive Descriptor Start Address 31 to 0 (RDLA31 to RDLA0)

Notes: The lower bits are set as follows according to the specified descriptor length.
16-byte boundary: RDLA[3:0] = 0000
32-byte boundary: RDLA[4:0] = 00000
64-byte boundary: RDLA[5:0] = 000000

Modifications made to this register during reception are invalid. This register
be modified while reception is disabled.

Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W

Bit:	23	22	21	20	19	18	17
	—	ECI	TC	TDE	TFUF	FR	RDE
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	—	—	—	ITF	CND	DLC	CD
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	RMAF	—	—	RRF	RTLF	RTSF	PRE
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W

Bits 31 to 27—Reserved: These bits are always read as 0. The write value should alway

EESR bit 11: Carrier Not Detected (CND)
EESR bit 10: Detect Loss of Carrier (DLC)
EESR bit 9: Delayed Collision Detect (CD)

Bit 25— Receive abort detected (RABT): Indicates whether or not a receive abort was

Bit 25: RABT	Description
0	Receive abort not detected (
1	Receive abort detected

This bit will be set when any one or more of the following bits are set.

EESR bit 4: Receive Residual-Bit Frame (RRF)
EESR bit 3: Receive Too-Long Frame (RTLF)
EESR bit 2: Receive Too-Short Frame (RTSF)
EESR bit 1: PHY-LSI Receive Error (PRE)
EESR bit 0: CRC Error on Received Frame (CERF)

frame counter register. The eight frames in the receive FIFO are retained, and a
transferred to memory when DMA transfer becomes possible. When the frame c
value falls below 8, another frame is received.

Bit 23—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 22—EtherC States Register Interrupt (ECI): Indicates that an interrupt due to an Eth
register (ECSR) source has been detected.

Bit 22: ECI	Description	
0	EtherC status interrupt source not detected	(Ir
1	EtherC status interrupt source detected (interrupt source)	

Note: EESR is a read-only register. When this register is cleared by a source in ECSR
EtherC, this bit is also cleared.

Bit 21—Frame Transmit Complete (TC): Indicates that all the data specified by the tran
descriptor has been transmitted to the EtherC. The transfer status is written back to the
descriptor. When 1-frame transmission is completed for 1-frame/1-buffer processing, o
last data in the frame is transmitted and the transmission descriptor valid bit (TACT) in
descriptor is not set for multiple-frame buffer processing, transmission is completed an
set to 1. After frame transmission, the E-DMAC writes the transmission status back to t
descriptor.

Bit 21: TC	Description	
0	Transfer not complete, or no transfer directive	(Ir
1	Transfer complete (interrupt source)	

Note: As data is sent onto the line by the PHY-LSI from the EtherC via the MII, the actu
transmission completion time is longer.

transmission. In this case, the address that is stored in the transmit descriptor li
register (TDLAR) is transmitted first.

Bit 19—Transmit FIFO Underflow (TFUF): Indicates that underflow has occurred in
FIFO during frame transmission. Incomplete data is sent onto the line.

Bit 19: TFUF	Description	
0	Underflow has not occurred	(
1	Underflow has occurred (interrupt source)	

Note: Whether E-DMAC operation continues or halts after underflow is controlled by t
operation control register (EDOCR).

Bit 18—Frame Received (FR): Indicates that a frame has been received and the receiv
has been updated. This bit is set to 1 each time a frame is received.

Note: The actual receive frame status is indicated in the receive status field in the de

Bit 18: FR	Description	
0	Frame not received	(
1	Frame received (interrupt source)	

Bit 17—Receive Descriptor Exhausted (RDE): This bit is set if the receive descriptor
(RACT) setting is “inactive” (RACT = 0) when the E-DMAC reads a receive descript

Bit 17: RDE	Description	
0	“1” receive descriptor active bit (RACT) detected	(
1	“0” receive descriptor active bit (RACT) detected (interrupt source)	

Note: When receive descriptor empty (RDE = 1) occurs, receiving can be restarted by
RACT = 1 in the receive descriptor and initiating receiving.

2. Whether E-DMAC operation continues or halts after overflow is controlled by DMAC operation control register (EDOCR).

Bits 15 to 13—Reserved: These bits are always read as 0. The write value should alway

Bit 12—Illegal Transmit Frame (ITF): Indicates that the transmit frame length specifica than four bytes.

Bit 12: ITF	Description	
0	Normal transmit frame length	(Ir
1	Illegal transmit frame length (interrupt source)	

Bit 11—Carrier Not Detect (CND): Indicates the carrier detection status.

Bit 11: CND	Description	
0	A carrier is detected when transmission starts	(Ir
1	Carrier not detected (interrupt source)	

Bit 10—Detect Loss of Carrier (DLC): Indicates that loss of the carrier has been detect frame transmission.

Bit 10: DLC	Description	
0	Loss of carrier not detected	(Ir
1	Loss of carrier detected (interrupt source)	

Bit 9—Delayed Collision Detect (CD): Indicates that a delayed collision has been detec frame transmission.

Bit 9: CD	Description	
0	Delayed Collision not detected	(Ir
1	Delayed Collision detected (interrupt source)	

Bit 7—Receive Multicast Address Frame (RMAF): Indicates that a multicast address
been received.

Bit 7: RMAF	Description
0	Multicast address frame has not been received
1	Multicast address frame has been received (interrupt source)

Bits 6 and 5—Reserved: These bits are always read as 0. The write value should alwa

Bit 4—Receive Residual-Bit Frame (RRF): Indicates that a residual-bit frame has bee

Bit 4: RRF	Description
0	Residual-bit frame has not been received
1	Residual-bit frame has been received (interrupt source)

Bit 3—Receive Too-Long Frame (RTLF): Indicates that a frame of 1519 bytes or lon
received.

Bit 3: RTLF	Description
0	Too-long frame has not been received
1	Too-long frame has been received (interrupt source)

Bit 2—Receive Too-Short Frame (RTSF): Indicates that a frame of fewer than 64 byt
received.

Bit 2: RTSF	Description
0	Too-short frame has not been received
1	Too-short frame has been received (interrupt source)

Bit 0: CERF	Description
0	CRC error not detected (Ir
1	CRC error detected (interrupt source)

10.2.7 EtherC/E-DMAC Status Interrupt Permission Register (EESIPR)

EESIPR enables interrupts corresponding to individual bits in the EtherC/E-DMAC sta
An interrupt is enabled by writing 1 to the corresponding bit. In the initial state, interru
enabled.

Bit:	31	30	29	28	27	26	25
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17
	—	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9
	—	—	—	ITFIP	CNDIP	DLCIP	CDIP
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W

Bit 23—Reserved: This bit is always read as 0. The write value should always be 0.

Bit 22—EtherC Status Register Interrupt Permission (ECIP): Enables interrupts due to
status register sources.

Bit 22: ECIP	Description	
0	EtherC status interrupts are disabled	(
1	EtherC status interrupts are enabled	

Bit 21—Frame Transmit Complete Interrupt Permission (TCIP): Enables the frame tra
complete interrupt.

Bit 21: TCIP	Description	
0	Frame transmit complete interrupt is disabled	(
1	Frame transmit complete interrupt is enabled	

Bit 20—Transmit Descriptor Exhausted Interrupt Permission (TDEIP): Enables the tra
descriptor exhausted interrupt.

Bit 20: TDEIP	Description	
0	Transmit descriptor exhausted interrupt is disabled	(
1	Transmit descriptor exhausted interrupt is enabled	

Bit 19—Transmit FIFO Underflow Interrupt Permission (TFUFIP): Enables the transm
underflow interrupt.

Bit 19: TFUFIP	Description	
0	Transmit FIFO underflow interrupt is disabled	(
1	Transmit FIFO underflow interrupt is enabled	

Bit 17: RDEIP	Description	
0	Receive descriptor exhausted interrupt is disabled	(Ir
1	Receive descriptor exhausted interrupt is enabled	

Bit 16—Receive FIFO Overflow Interrupt Permission (RFOFIP): Enables the receive F
overflow interrupt.

Bit 16: RFOFIP	Description	
0	Receive FIFO overflow interrupt is disabled	(Ir
1	Receive FIFO overflow interrupt is enabled	

Bits 15 to 13—Reserved: These bits are always read as 0. The write value should alway

Bit 12—Illegal Transmit Frame Interrupt Permission (ITFIP): Enables the illegal transr
interrupt.

Bit 12: ITFIP	Description	
0	Illegal transmit frame interrupt is disabled	(Ir
1	Illegal transmit frame interrupt is enabled	

Bit 11—Carrier Not Detect Interrupt Permission (CNDIP): Enables the carrier not detec

Bit 11: CNDIP	Description	
0	Carrier not detect interrupt is disabled	(Ir
1	Carrier not detect interrupt is enabled	

Bit 9: CDIP	Description	
0	Collision detect interrupt is disabled	(
1	Collision detect interrupt is enabled	

Bit 8—Transmit Retry Over Interrupt Permission (TROIP): Enables the transmit retry
interrupt.

Bit 8: TROIP	Description	
0	Transmit retry over interrupt is disabled	(
1	Transmit retry over interrupt is enabled	

Bit 7—Receive Multicast Address Frame Interrupt Permission (RMAFIP): Enables th
multicast address frame interrupt.

Bit 7: RMAFIP	Description	
0	Receive multicast address frame interrupt is disabled	(
1	Receive multicast address frame interrupt is enabled	

Bits 6 and 5—Reserved: These bits are always read as 0. The write value should alwa

Bit 4—Receive Residual-Bit Frame Interrupt Permission (RRFIP): Enables the receiv
frame interrupt.

Bit 4: RRFIP	Description	
0	Receive residual-bit frame interrupt is disabled	(
1	Receive residual-bit frame interrupt is enabled	

frame interrupt.

Bit 2: RTSFIP	Description
0	Receive too-short frame interrupt is disabled (Ir
1	Receive too-short frame interrupt is enabled

Bit 1—PHY-LSI Receive Error Interrupt Permission (PREIP): Enables the PHY-LSI re
interrupt.

Bit 1: PREIP	Description
0	PHY-LSI receive error interrupt is disabled (Ir
1	PHY-LSI receive error interrupt is enabled

Bit 0—CRC Error on Received Frame Interrupt Permission (PREIP): Enables the CRC
received frame interrupt.

Bit 0: CERFIP	Description
0	CRC error on received frame interrupt is disabled (Ir
1	CRC error on received frame interrupt is enabled

Bit:	31	30	29	. . .	19	18	17
	—	—	—	. . .	—	—	—
Initial value:	0	0	0	. . .	0	0	0
R/W:	R	R	R	. . .	R	R	R

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	RMAFCE	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R

Bits 31 to 8—Reserved: These bits are always read as 0. The write value should alway

Bit 7—Multicast Address Frame Receive (RMAF) Bit Copy Enable (RMAFCE)

Bit 7: RMAFCE	**Description**
0	Enables the RMAF bit status to be indicated in the RFS7 bit in the descriptor.
1	Disables occurrence of corresponding source to be indicated in the the receive descriptor.

Bits 6 to 0—Reserved: These bits are always read as 0. The write value should always

	—	—	—	. . .	—	—	—
Initial value:	0	0	0	. . .	0	0	0
R/W:	R	R	R	. . .	R	R	R

Bit:	15	14	13	12	11	10	9
	MFC15	MFC14	MFC13	MFC12	MFC11	MFC10	MFC9
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	MFC7	MFC6	MFC5	MFC4	MFC3	MFC2	MFC1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	31	30	29	. . .	23	22	21
	—	—	—	. . .	—	—	—
Initial value:	0	0	0	. . .	0	0	0
R/W:	R	R	R	. . .	R	R	R

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	TFT10	TFT9
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1
	TFT7	TFT6	TFT5	TFT4	TFT3	TFT2	TFT1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 31 to 11—Reserved: These bits are always read as 0. The write value should alwa

H'1F	124 bytes
H'20	128 bytes
:	:
H'3F	252 bytes
H'40	256 bytes
:	:
H'7F	508 bytes
H'80	512 bytes

Note: When setting a transmit FIFO threshold of 256 bytes or more, a FIFO depth of 5
must be selected.

Restriction: When the transfer rate is 10 Mbps (TX-CLK clock input frequency = 2.5 M
the value of transmit FIFO threshold bits 10 to 0 (TFT10 to TFT0) of TFTR is set to a v
H’001 (4 bytes) to H’00C (48 bytes), the Ethernet controller (EtherC) may not transmit

Therefore, when the transfer rate is 10 Mbps, set TFT10 to TFT0 of TFTR to H’000 (st
forward mode), or H’00D (52 bytes) or a larger value. (Refer to the following table.)

When the transfer rate is 100 Mbps (TX-CLK clock input frequency = 25 MHz), this re
not applicable.

H'006	24 bytes
H'007	28 bytes
H'008	32 bytes
H'009	36 bytes
H'00A	40 bytes
H'00B	44 bytes
H'00C	48 bytes
H'00D	52 bytes
H'00E	56 bytes
H'00F	60 bytes
:	:
H'01F	124 bytes
H'020	128 bytes
:	:
H'03F	252 bytes
H'040	256 bytes
:	:
H'07F	508 bytes
H'080	512 bytes

malfunction when the
rate is 10 Mbps.

Bit:	15	14	13	12	11	10	9
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 31 to 9—Reserved: These bits are always read as 0. The write value should always

Bit 8—Transmit FIFO Depth (TFD): Specifies either 256 or 512 bytes as the depth (siz
transmit FIFO (which has a maximum capacity of 512 bytes). The setting cannot be cha
transmission/reception has started.

Bit 8: TFD	Description
0	256 bytes (Ir
1	512 bytes

Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always b

Bit 0—Receive FIFO Depth (RFD): Specifies either 256 or 512 bytes as the depth (size
receive FIFO (which has a maximum capacity of 512 bytes). The actual FIFO depth is
the set value. The setting cannot be changed after transmission/reception has started.

Bit 0: RFD	Description
0	256 bytes (Ir
1	512 bytes

R/W:	R	R	R	. . .	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 31 to 1—Reserved: These bits are always read as 0. The write value should alway

Bit 0—Receive Enable Control (RNC)

Bit 0: RNC	**Description**
0	When reception of one frame is completed, the E-DMAC writes the status into the descriptor and clears the RR bit in EDRRR (
1	When reception of one frame is completed, the E-DMAC writes the status into the descriptor, reads the next descriptor, and prepares the next frame*

Note: * This setting is normally used for continuous frame reception.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	FEC	AEC	EDH
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W

Bits 31 to 4—Reserved: These bits are always read as 0. The write value should always

Bit 3—FIFO Error Control (FEC): Specifies E-DMAC operation when transmit FIFO u
or receive FIFO overflow occurs.

Bit 3: FEC	Description
0	E-DMAC operation continues when underflow or overflow occurs (In
1	E-DMAC operation halts when underflow or overflow occurs

Bit 2—Address Error Control (AEC): Indicates detection of an illegal memory address
attempted E-DMAC transfer.

Bit 2: AEC	Description
0	Illegal memory address not detected (normal operation) (In
1	Illegal memory address detected. Can be cleared by writing 0.

Note: This error occurs if the memory address setting in the descriptor used by the E-D
illegal.

Bit 1—E-DMAC Halted (EDH): When the SH7615's NMI input pin is asserted, E-DM
operation is halted.

Bit 1: EDH	Description
0	The E-DMAC is operating normally (In
1	The E-DMAC has been halted by NMI pin assertion. E-DMAC opera restarted by writing 0.

	RBWA31	RBWA30	RBWA29	RBWA28	RBWA27	RBWA26	RBWA
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17
	RBWA23	RBWA22	RBWA21	RBWA20	RBWA19	RBWA18	RBWA
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9
	RBWA15	RBWA14	RBWA13	RBWA12	RBWA11	RBWA10	RBWA
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	RBWA7	RBWA6	RBWA5	RBWA4	RBWA3	RBWA2	RBWA
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 31 to 0—Receiving-buffer write address (RBWA): This bit can only be read. Wri
disabled.

Note: The buffer write processing result from the E-DMAC and the value read by th
may not be the same.

Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17
	RDFA23	RDFA22	RDFA21	RDFA20	RDFA19	RDFA18	RDFA17
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9
	RDFA15	RDFA14	RDFA13	RDFA12	RDFA11	RDFA10	RDFA9
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	RDFA7	RDFA6	RDFA5	RDFA4	RDFA3	RDFA2	RDFA1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 31 to 0—Receiving-descriptor fetch address (RDFA): This bit can only be read. W
disabled.

Note: The descriptor fetch processing result from the E-DMAC and the value read by
register may not be the same.

R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17
	TBRA23	TBRA22	TBRA21	TBRA20	TBRA19	TBRA18	TBRA1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9
	TBRA15	TBRA14	TBRA13	TBRA12	TBRA11	TBRA10	TBRA
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	TBRA7	TBRA6	TBRA5	TBRA4	TBRA3	TBRA2	TBRA
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 31 to 0—Transmission-buffer read address (TBRD): This bit can only be read. W
disabled.

Note: The buffer read processing result from the E-DMAC and the value read by the
may not be the same.

Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17
	TDFA23	TDFA22	TDFA21	TDFA20	TDFA19	TDFA18	TDFA17
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9
	TDFA15	TDFA14	TDFA13	TDFA12	TDFA11	TDFA10	TDFA9
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	TDFA7	TDFA6	TDFA5	TDFA4	TDFA3	TDFA2	TDFA1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 31 to 0—Transmission-descriptor fetch address (TDFA): This bit can only be read
disabled.

Note: The descriptor fetch processing result from the E-DMAC and the value read by
register may not be the same.

10.3.1 Descriptor List and Data Buffers

Before starting transmission/reception, the communication program creates transmit a
descriptor lists in memory. The start addresses of these lists are then set in the transmi
descriptor list start address registers.

Transmit Descriptor

Figure 10.2 shows the relationship between a transmit descriptor and the transmit buff
According to the specification in this descriptor, the relationship between the transmit
transmit buffer can be defined as one frame/one buffer or one frame/multi-buffer.

Notes: 1. The descriptor start address must be specified to align with an address bou
corresponding to the descriptor length specified in the E-DMAC mode reg
(EDMR).

2. The transmit buffer start address must be specified to align with a longwor
Note, however, that it must be aligned with a 16-byte boundary when SDR
connected.

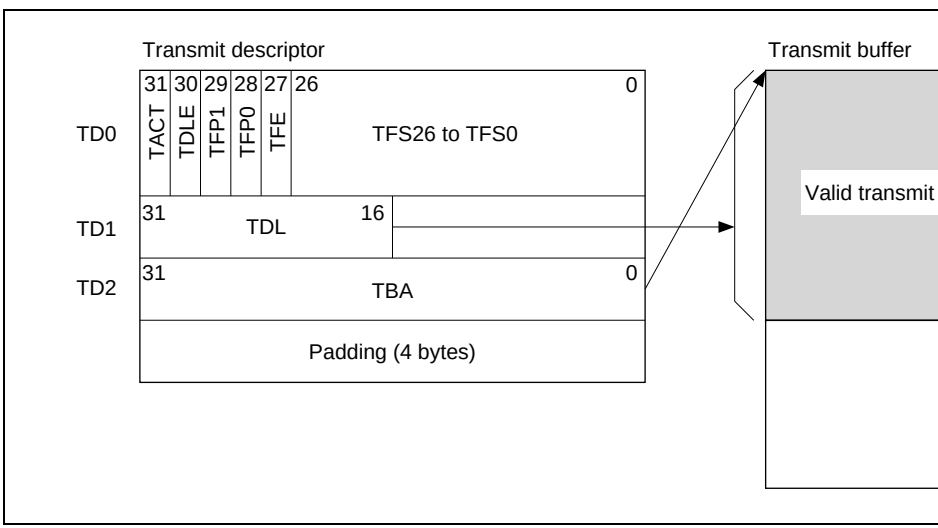


Figure 10.2 Relationship between Transmit Descriptor and Transmit Bu

	Indicates that valid data has not been written to the transmit buffer b or this bit has been reset by a write-back operation on termination o frame transfer processing (completion or suspension of transmissio If this state is recognized in an E-DMAC descriptor read, the E-DMA terminates transmit processing and transmit operations cannot be c restart is necessary)
1	The transmit descriptor is valid Indicates that valid data has been written to the transmit buffer by th frame transfer processing has not yet been executed, or that frame in progress When this state is recognized in an E-DMAC descriptor read, the E- continues with the transmit operation

Bit 30—Transmit Descriptor List Last (TDLE): Indicates that this descriptor is the last transmit descriptor list. After completion of the corresponding buffer transfer, the E-DM references the first descriptor. This specification is used to set a ring configuration for t descriptors.

Bit 30: TDLE	Description
0	This is not the last transmit descriptor list
1	This is the last transmit descriptor list

1	0	Transmit buffer indicated by this descriptor is start of frame (fran concluded)
	1	Contents of transmit buffer indicated by this descriptor are equiv frame (one frame/one buffer)

Note: In the preceding and following descriptors, a logically positive relationship must maintained between the settings of this bit and the TDLE bit.

Bit 27—Transmit Frame Error (TFE): Indicates that one or other bit of the transmit fra indicated by bits 26 to 0 is set.

Bit 27: TFE	Description
0	No error during transmission
1	An error of some kind occurred during transmission (see bits 26 to

Bits 26 to 0—Transmit Frame Status 26 to 0 (TFS26 to TFS0): These bits indicate the during frame transmission.

- TFS26 to TFS9—Reserved
- TFS8—Transmit Abort Error Detect

  Note: This bit is set to 1 when any of transmit frame status bits 4 to 0 (TFS4 to TF When this bit is set, the transmit frame error bit (bit 27: TFE) is set to 1.
- TFS7 to TFS5—Reserved
- TFS4—Illegal Transmit Frame (corresponds to ITF bit in EESR)
- TFS3—Carrier Not Detect (corresponds to CND bit in EESR)
- TFS2—Detect Loss of Carrier (corresponds to DLC bit in EESR)
- TFS1—Delayed Collision Detect in Transmission (corresponds to CD bit in EESR
- TFS0—Transmit Retry Over (corresponds to TRO bit in EESR)

Transmit Descriptor 2 (TD2): Specifies the 32-bit transmit buffer start address.

Note: The transmit buffer start address must be specified to align with a longword bo
However, it must be aligned with a 16-byte boundary when SDRAM is connec

Bits 31 to 0—Transmit Buffer Address (TBA)

Receive Descriptor

Figure 10.3 shows the relationship between a receive descriptor and the receive buffer.
reception, the E-DMAC performs data rewriting up to a receive buffer 16-byte boundar
regardless of the receive frame length. Finally, the actual receive frame length is report
lower 16 bits of RD1 in the descriptor. Data transfer to the receive buffer is performed
automatically by the E-DMAC to give a one frame/one buffer or one frame/multi-buffe
configuration according to the size of one received frame.

Notes: 1. The descriptor start address must be specified to align with an address boun
corresponding to the descriptor length specified in the E-DMAC mode regis
(EDMR).
2. The receive buffer start address must be specified to align with a longword b
Note, however, that it must be aligned with a 16-byte boundary when SDRA
connected.
Specify an appropriate size of receive buffer so that it aligns with a 16-byte
Example: H'0500 (= 1536 bytes)

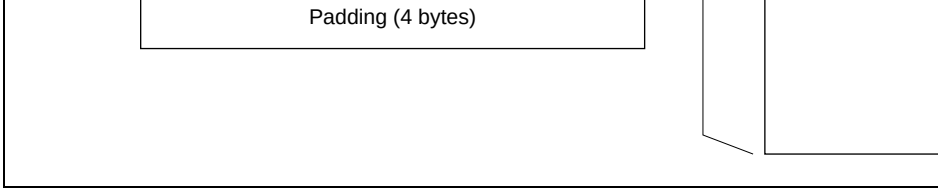


Figure 10.3 Relationship between Receive Descriptor and Receive Buf

Receive Descriptor 0 (RD0): RD0 indicates the receive frame status. The CPU and E
RD0 to report the frame transmission status.

Bit 31—Receive Descriptor Active (RACT): Indicates that this descriptor is active. Th
resets this bit after receive data has been transferred to the receive buffer. On completi
receive frame processing, the CPU sets this bit to prepare for reception.

Bit 31: RACT	Description
0	The receive descriptor is invalid Indicates that the receive buffer is not ready (access disabled by E this bit has been reset by a write-back operation on termination of frame transfer processing (completion or suspension of reception) If this state is recognized in an E-DMAC descriptor read, the E-DM terminates receive processing and receive operations cannot be co Reception can be restarted by setting RACT to 1 and executing re initiation.
1	The receive descriptor is valid Indicates that the receive buffer is ready (access enabled) and pro frame transfer from the FIFO has not been executed, or that frame progress When this state is recognized in an E-DMAC descriptor read, the E continues with the receive operation

Bit 30—Receive Descriptor List Last (RDLE): Indicates that this descriptor is the last
receive descriptor list. After completion of the corresponding buffer transfer, the E-DM

relationship between the receive buffer and receive frame.

Bit 29: RFP	Bit 28: RFP	Description
0	0	Frame reception for receive buffer indicated by this descriptor co (frame is not concluded)
	1	Receive buffer indicated by this descriptor contains end of frame concluded)
1	0	Receive buffer indicated by this descriptor is start of frame (frame concluded)
	1	Contents of receive buffer indicated by this descriptor are equival frame (one frame/one buffer)

Bit 27—Receive Frame Error (RFE): Indicates that one or other bit of the receive fram indicated by bits 26 to 0 is set. Whether or not the multicast address frame receive infor which is part of the receive frame status, is copied into this bit is specified by the transr status copy enable register (TRSCER).

Bit 27: RFE	Description
0	No error during reception (Ir
1	An error of some kind occurred during reception (see bits 26 to 0)

Bits 26 to 0—Receive Frame Status 26 to 0 (RFS26 to RFS0): These bits indicate the e during frame reception.

- RFS26 to RFS10—Reserved
- RFS9—Receive FIFO Overflow (corresponds to RMAF bit in EESR)
- RFS8—Receive Abort Error Detect

  Note: This bit is set to 1 when any of receive frame status bits 9 (RFS9), 7 (RFS7), (RFS4 to RFS0) is set. When this bit is set, the receive frame error (RFE) bit
- RFS7—Receive Multicast Address Frame (corresponds to RMAF bit in EESR)
- RFS6, RFS5—Reserved
- RFS4—Receive Residual-Bit Frame (corresponds to RRF bit in EESR)

Notes: The transfer byte length must align with a 16-byte boundary (bits 19 to 16 clea
The maximum receive frame length with one frame per buffer is 1,514 bytes,
the CRC data. Therefore, for the receive buffer length specification, a value of
(H'05F0) that takes account of a 16-byte boundary is set as the maximum rece
length.

Bits 15 to 0—Receive Data Length (RDL): These bits specify the data length of a rece
stored in the receive buffer.

Note: The receive data transferred to the receive buffer does not include the 4-byte C
the end of the frame. The receive frame length is reported as the number of wo
data bytes) not including this CRC data.

Receive Descriptor 2 (RD2): Specifies the 32-bit receive buffer start address.

Note: The receive buffer start address must be specified to align with a longword bo
However, it must be aligned with a 16-byte boundary when SDRAM is conne

Bits 31 to 0—Receive Buffer Address (RBA)

10.3.2 Transmission

When the transmitter is enabled and the transmit request bit (TR) is set in the E-DMA
request register (EDTRR), the E-DMAC reads the descriptor used last time from the t
descriptor list (in the initial state, the descriptor indicated by the transmission descript
address register (TDLAR)). If the setting of the TACT bit in the read descriptor is “ac
DMAC reads transmit frame data sequentially from the transmit buffer start address sp
TD2, and transfers it to the EtherC. The EtherC creates a transmit frame and starts tran
the MII. After DMA transfer of data equivalent to the buffer length specified in the de
following processing is carried out according to the TFP value.

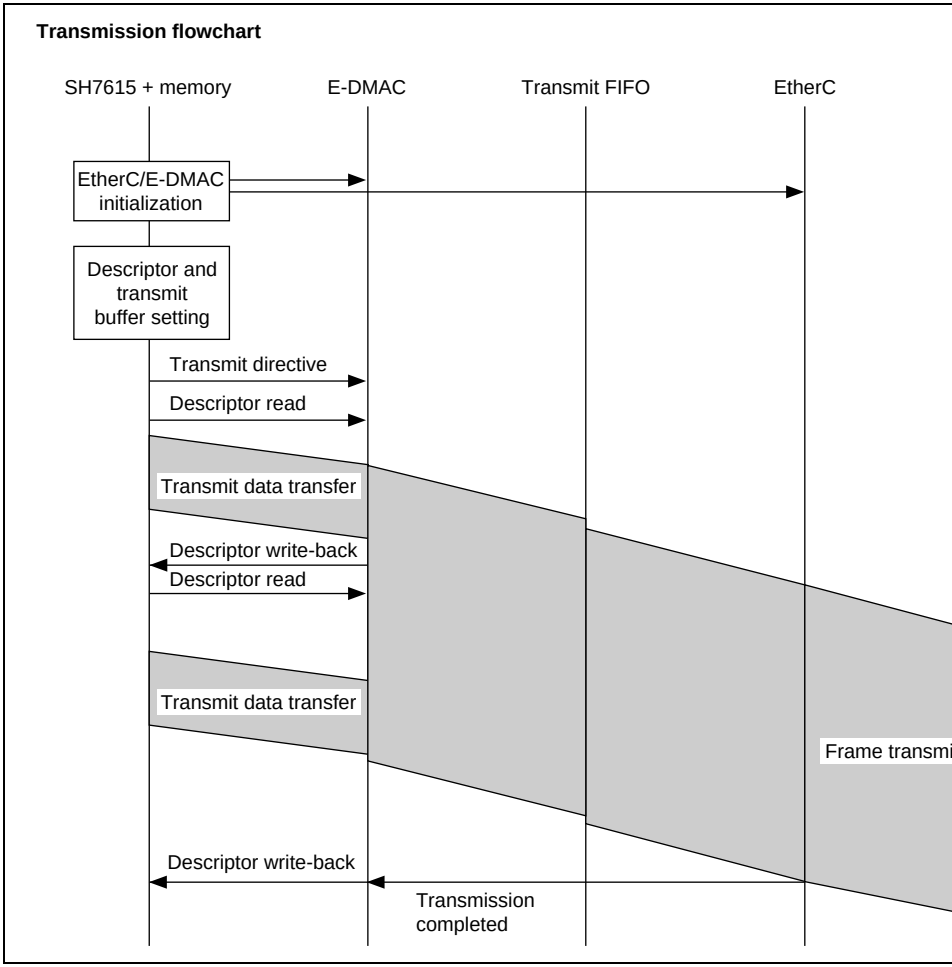


Figure 10.4 Sample Transmission Flowchart

when the buffer is full (RFP = 10 or 00), then reads the next descriptor. The E-DMAC continues to transfer data to the receive buffer specified by the new RD2. When frame completed, or if frame reception is suspended because of an error of some kind, the E- performs write-back to the relevant descriptor (RFP = 11 or 01), and then ends the rec processing. The E-DMAC then reads the next descriptor and enters the receive-standb again.

Note: To receive frames continuously, the receive enable control bit (RNC) must be the receive control register (RCR). After initialization, this bit is cleared to 0.

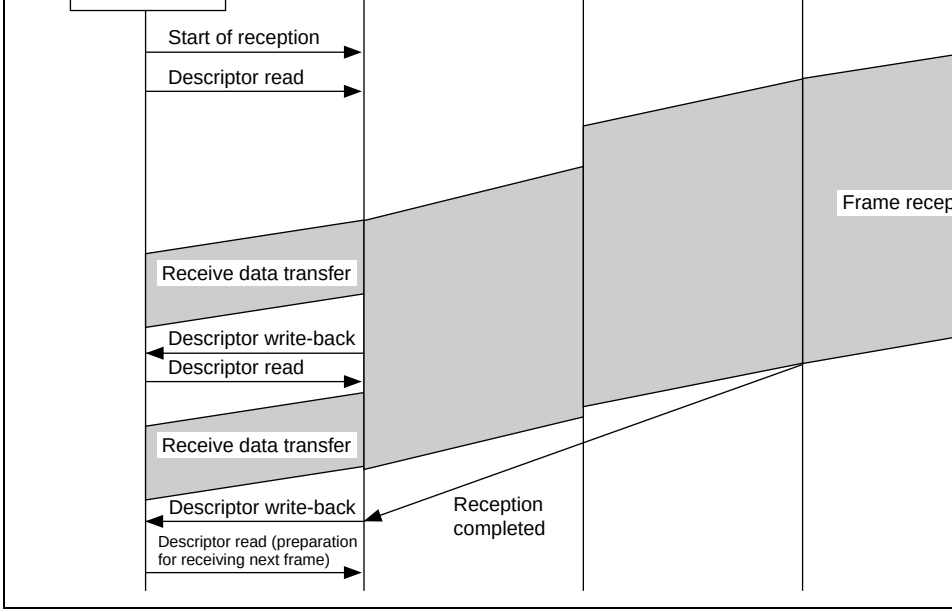


Figure 10.5 Sample Reception Flowchart

bit cleared to 0, immediately. The next descriptor is then read, and the position within
frame is determined on the basis of bits TFP1 and TFP0 (continuing [00] or end [01]).
of a continuing descriptor, the TACT bit is cleared to 0, only, and the next descriptor i
immediately. If the descriptor is the final descriptor, not only is the TACT bit cleared
write-back is also performed to the TFE and TFS bits at the same time. Data in the bu
transmitted between the occurrence of an error and write-back to the final descriptor.
interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EE
interrupt is generated immediately after the final descriptor write-back.

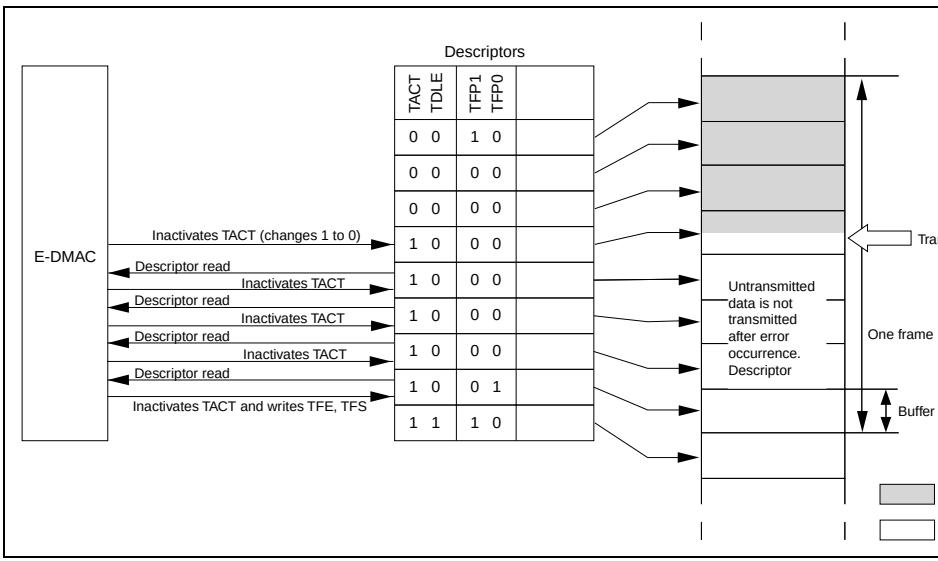


Figure 10.6 E-DMAC Operation after Transmit Error

If error interrupts are enabled in the EtherC/E-DMAC status interrupt permission regist
(EESIPR), an interrupt is generated immediately after the write-back. If there is a new f
receive request, reception is continued from the buffer after that in which the error occu

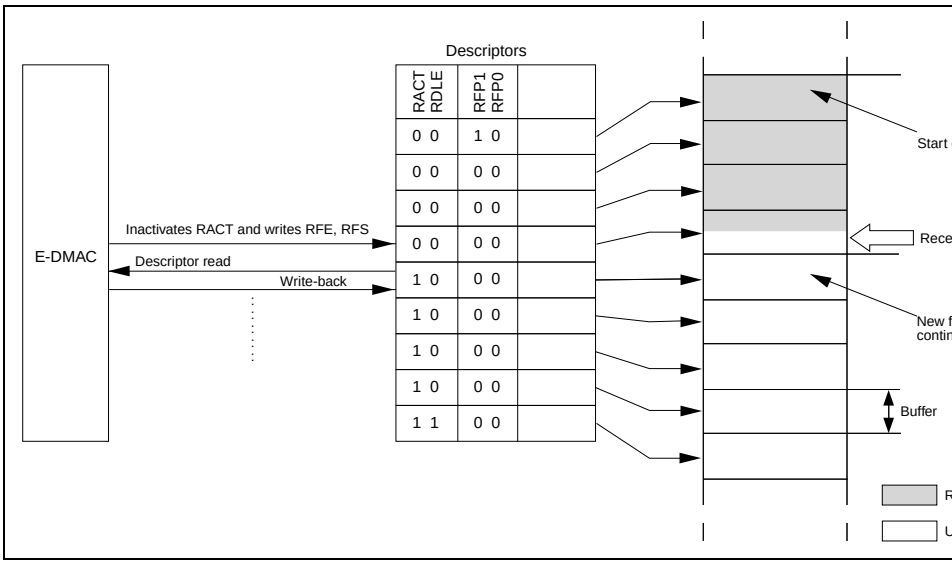


Figure 10.7 E-DMAC Operation after Receive Error

When the timing of clear TR/RR request bit and set TR/RR request bit by user's firmw
matched, E-DMAC can't recognize the exact condition of TR/RR bit.

Condition: When TR/RR request bit is always set by the firmware without checking t
TR/RR request bit.

Countermeasures: Please check the TR/RR request bit is cleared by E-DMAC first, a
the TR/RR request bit by user's firmware.

(1) There are two ways to check TR request bit that is cleared by E-DMAC.
   (a) Possible to check read "0" of TR bit of E-DMAC directly.
   (b) Possible to check read "1" of TDE (Transmit Descriptor Exhausted) in EESR r
   the interrupt on.
(2) There are two ways to check RR request bit that is cleared by E-DMAC.
   (a) Possible to check read "0" of RR bit of E-DMAC directly.
   (b) Possible to check read "1" of RDE (Receive Descriptor Exhausted) in EESR re
   the interrupt on.

the operating efficiency of the chip as a whole.

11.1.1 Features

The DMAC has the following features:

- Two channels
- Address space: Architecturally 4 Gbytes
- Choice of data transfer unit: Byte, word (2-byte), longword (4-byte) or 16-byte uni
  byte transfer, four longword reads are executed, followed by four longword writes
- Maximum of 16,777,216 (16M) transfers
- In the event of a cache hit, CPU instruction processing and DMA operation can be
  parallel
- Single address mode transfers: Either the transfer source or transfer destination (pe
  device) is accessed by a DACK signal (selectable) while the other is accessed by a
  transfer unit of data is transferred in one bus cycle.

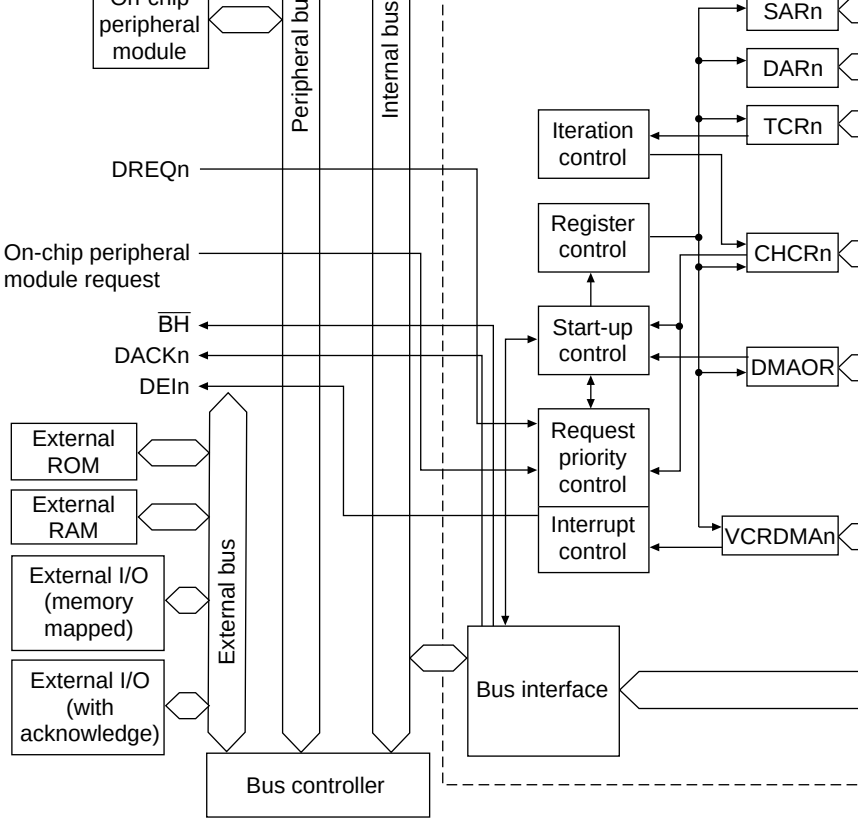
  Possible transfer devices: External devices with DACK and memory-mapped exte
  (including external memory)
- Dual address mode transfer: Both the transfer source and transfer destination are a
  address. One transfer unit of data is transferred in two bus cycles.

  Possible transfer devices:
  - Two external memories
  - External memory and memory-mapped external device
  - Two memory-mapped external devices
  - External memory and on-chip peripheral module (excluding DMAC, BSC, UB
    memory, E-DMAC, and EtherC)
  - Memory-mapped external device and on-chip peripheral module (excluding DI
    UBC, cache-memory, E-DMAC, and EtherC)
  - Two on-chip peripheral modules (excluding DMAC, BSC, UBC, cache-memo
    DMAC, and EtherC)
  - On-chip memory and memory-mapped external device

16-bit timer pulse unit (TPU), serial I/O (SIO)

— Auto-request: the transfer request is generated automatically within the DMAC

- Choice of bus mode
  - — Cycle steal mode
  - — Burst mode
- Choice of channel priority order
  - — Fixed mode
  - — Round robin mode
- An interrupt request can be sent to the CPU on completion of data transfer



DMAOR: DMA operation register
SARn: DMA source address register
DARn: DMA destination address register
TCRn: DMA transfer count register
CHCRn: DMA channel control register
VCRDMAn: DMA vector number register
DEIn: DMA transfer end interrupt request to CPU
On-chip peripheral module request: Interrupt transfer request from on-chip SCIF, SIO, TPU
$\overline{BH}$: Burst hint
n: 0, 1

Figure 11.1 DMAC Block Diagram

	DMA transfer request acknowledge	DACK0	Output	DMA transfer request acknowledge from channel 0 to external device
1	DMA transfer request	DREQ1	Input	DMA transfer request input from device to channel 1
	DMA transfer request acknowledge	DACK1	Output	DMA transfer request acknowledge from channel 1 to external device
All	Burst hint	$\overline{BH}$	Output	Burst transfer in 16-byte transfer

	DMA destination address register 0	DAR0	R/W	Undefined	H'FFFFFF
	DMA transfer count register 0	TCR0	R/W	Undefined	H'FFFFFF
	DMA channel control register 0	CHCR0	R/(W)*1	H'00000000	H'FFFFFF
	DMA vector number register 0	VCRDMA0	R/W	Undefined	H'FFFFFF
	DMA request/response selection control register 0	DRCR0	R/W	H'00	H'FFFFFE
1	DMA source address register 1	SAR1	R/W	Undefined	H'FFFFFF
	DMA destination address register 1	DAR1	R/W	Undefined	H'FFFFFF
	DMA transfer count register 1	TCR1	R/W	Undefined	H'FFFFFF
	DMA channel control register 1	CHCR1	R/(W)*1	H'00000000	H'FFFFFF
	DMA vector number register 1	VCRDMA1	R/(W)	Undefined	H'FFFFFF
	DMA request/response selection control register 1	DRCR1	R/(W)	H'00	H'FFFFFE
All	DMA operation register	DMAOR	R/(W)*2	H'00000000	H'FFFFFF

Notes: 1. Only 0 can be written to bit 1 of CHCR0 and CHCR1, after reading 1, to cle
2. Only 0 can be written to bits 1 and 2 of the DMAOR, after reading 1, to clea
3. Access DRCR0 and DRCR1 in byte units. Access all other registers in long

DMA source address registers 0 and 1 (SAR0 and SAR1) are 32-bit read/write registers specify the source address of a DMA transfer. During a DMA transfer, these registers i next source address. (In single-address mode, SAR is ignored in transfers from external with DACK to memory-mapped external devices or external memory). In 16-byte unit always set the value of the source address to a 16-byte boundary (16n address). Operati cannot be guaranteed if other values are used. Transmission in 16-byte units can be set auto-request mode and at edge detection in external request mode. Values are retained i standby mode, and when the module standby function is used.

11.2.2 DMA Destination Address Registers 0 and 1 (DAR0, DAR1)

Bit:	31	30	29	…	3	2	1
				…			
Initial value:	—	—	—	…	—	—	—
R/W:	R/W	R/W	R/W	…	R/W	R/W	R/W

DMA destination address registers 0 and 1 (DAR0 and DAR1) are 32-bit read/write reg specify the destination address of a DMA transfer. During a DMA transfer, these regist the next destination address. (In single-address mode, DAR is ignored in transfers from mapped external devices or external memory to external devices with DACK). In 16-by transfers, always set the value of the source address to a 16-byte boundary (16n address Operation results cannot be guaranteed if other values are used. Transmission in 16-byt be set only in auto-request mode and at edge detection in external request mode. Value retained in a reset, in standby mode, and when the module standby function is used.

If synchronous DRAM is accessed when performing 16-byte-unit transfer, a 16-byte bo (address 16n) value must be set for the destination address.

				…			
Initial value:	—	—	—	…	—	—	—
R/W:	R/W	R/W	R/W	…	R/W	R/W	R/W

DMA transfer count registers 0 and 1 (TCR0 and TCR1) are 32-bit read/write register
specify the DMA transfer count. The lower 24 of the 32 bits are valid. The value is wr
bits, including the upper eight bits. The number of transfers is 1 when the setting is H'
16,777,215 when the setting is H'00FFFFFF and 16, 777,216 (the maximum) when H'
set. During a DMA transfer, these registers indicate the remaining transfer count.

Set the initial value as the write value in the upper eight bits. These bits always read 0
retained in a reset, in standby mode, and when the module standby function is used. F
transfers, set the count to 4 times the number of transfers. Operation is not guaranteed
incorrect value is set.

	DM1	DM0	SM1	SM0	TS1	TS0	AR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	AL	DS	DL	TB	TA	IE	TE
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*

Note: * Only 0 can be written, to clear the flag.

DMA channel control registers 0 and 1 (CHCR0 and CHCR1) are 32-bit read/write reg
control the DMA transfer mode. They also indicate the DMA transfer status. Only the l
the 32 bits are valid. They should be read and written as 32-bit values, including the up
The registers are initialized to H'00000000 by a reset and in standby mode. Values are r
during a module standby.

Bits 31 to 16—Reserved: These bits are always read as 0. The write value should alway

Bits 15 and 14—Destination Address Mode Bits 1, 0 (DM1, DM0): Select whether the
destination address is incremented, decremented or left fixed (in single address mode, D
DM0 are ignored when transfers are made from a memory-mapped external device, or
memory to an external device with DACK). DM1 and DM0 are initialized to 00 by a re
standby mode. Values are retained during a module standby.

Bits 13 and 12—Source Address Mode Bits 1, 0 (SM1, SM0): Select whether the DM
address is incremented, decremented or left fixed. (In single address mode, SM1 and S
ignored when transfers are made from an external device with DACK to a memory-ma
external device, or external memory.) For a 16-byte transfer, the address is incremente
regardless of the SM1 and SM0 values. SM1 and SM0 are initialized to 00 by a reset a
standby mode. Values are retained during a module standby.

Bit 13: SM1	Bit 12: SM0	Description
0	0	Fixed source address (+16 for 16-byte transfer si (
	1	Source address is incremented (+1 for byte trans for word transfer size, +4 for longword transfer siz 16-byte transfer size)
1	0	Source address is decremented (–1 for byte trans for word transfer size, –4 for longword transfer siz 16-byte transfer size)
	1	Reserved (setting prohibited)

Bits 11 and 10—Transfer Size Bits (TS1, TS0): Select the DMA transfer size. When 1
bits TS1 and TS0 (in the 16-byte unit), request mode is available only in auto-request
edge detection in external request mode. When 11 is set to bits TS1 and TS0 (in the 16
and level detection in external request mode and internal peripheral-module request m
system operations are not guaranteed. TS1 and TS0 are initialized to 00 by a reset and
mode. Values are retained during a module standby.

Bit 11: TS1	Bit 10: TS0	Description
0	0	Byte unit (
	1	Word (2-byte) unit
1	0	Longword (4-byte) unit
	1	16-byte unit (4 longword transfers)

Bit 8—Acknowledge/Transfer Mode Bit (AM): In dual address mode, this bit selects w
DACKn signal is output during the data read cycle or write cycle. In single-address mo
selects whether to transfer data from memory to device or from device to memory. The
initialized to 0 by a reset and in standby mode. Its value is retained during a module sta

Bit 8: AM	Description
0	DACKn output in read cycle (dual address mode)/transfer fro to device (single address mode) (Ir
1	DACKn output in write cycle (dual address mode)/transfer fro to memory (single address mode)

Bit 7—Acknowledge Level Bit (AL): Selects whether the DACKn signal is an active-h
or an active-low signal. The AL bit is initialized to 0 by a reset and in standby mode. Its
retained during a module standby.

Bit 7: AL	Description
0	DACKn is an active-low signal (Ir
1	DACKn is an active-high signal

Bit 6—DREQn Select Bit (DS): Selects the DREQn input detection used. When 0 (leve
is set to bit DS, set 0 (cycle-steal mode) to the transfer bus mode bit (TB). When 0 is se
and 1 (burst mode) is set to bit TB, system operations are not guaranteed. The DS bit is
to 0 by a reset and in standby mode. Its value is retained during a module standby.

Bit 6: DS	Description
0	Detected by level (Ir Can be set only in cycle-steal mode
1	Detected by edge

Bit 4—Transfer Bus Mode Bit (TB): Selects the bus mode for DMA transfers. When 1
mode) is set to bit TB, set 1 (edge detection) to the DREQ select bit (DS). When 1 is s
and 0 (level detection) is set to bit DS, system operations are not guaranteed. The TB b
initialized to 0 by a reset and in standby mode. Its value is retained during a module st

Bit 4: TB	Description	
0	Cycle-steal mode	(
1	Burst mode	

Bit 3—Transfer Address Mode Bit (TA): Selects the DMA transfer address mode. The
initialized to 0 by a reset and in standby mode. Its value is retained during a module st

Bit 3: TA	Description	
0	Dual address mode	(
1	Single address mode	

Bit 2—Interrupt Enable Bit (IE): Determines whether or not to request a CPU interrup
of a DMA transfer. When the IE bit is set to 1, an interrupt (DEI) request is sent to the
the TE bit is set. The IE bit is initialized to 0 by a reset and in standby mode. Its value
during a module standby.

Bit 2: IE	Description	
0	Interrupt request disabled	(
1	Interrupt request enabled	

Bit 1—Transfer-End Flag Bit (TE): Indicates that the transfer has ended. When the va
DMA transfer count register (TCR) becomes 0, the DMA transfer ends normally and t
set to 1. When TCR is not 0, the TE bit is not set if the transfer ends because of an NM
or DMA address error, or because the DME bit in the DMA operation register (DMAO
DE bit was cleared. To clear the TE bit, read 1 from it and then write 0. When the TE
setting the DE bit to 1 will not enable a transfer. The TE bit is initialized to 0 by a rese
standby mode. Its value is retained during a module standby.

module request mode, the transfer begins when the DMA transfer request is received fr
relevant device or on-chip peripheral module, provided this bit and the DME bit are set
with the auto-request mode, the TE bit and the NMIF and AE bits in DMAOR must all
The transfer can be stopped by clearing this bit to 0. The DE bit is initialized to 0 by a r
standby mode. Its value is retained during a module standby.

Bit 0: DE	Description
0	DMA transfer disabled (In
1	DMA transfer enabled

11.2.5 DMA Vector Number Registers 0 and 1 (VCRDMA0, VCRDMA1)

Bit:	31	30	29	…	11	10	9
	—	—	—	…	—	—	—
Initial value:	0	0	0	…	0	0	0
R/W:	R	R	R	…	R	R	R

Bit:	7	6	5	4	3	2	1
	VC7	VC6	VC5	VC4	VC3	VC2	VC1
Initial value:	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DMA vector number registers 0 and 1 (VCRDMA0, VCRDMA1) are 32-bit read/write
that set the DMAC transfer-end interrupt vector number. Only the lower eight bits of th
valid. They are written as 32-bit values, including the upper 24 bits. Values are retained
in standby mode, and when the module standby function is used.

Bits 31 to 8—Reserved: These bits are always read as 0. The write value should always

	—	—	—	RS4	RS3	RS2	RS1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W

DMA request/response selection control registers 0 and 1 (DRCR0, DRCR1) are 8-bit
registers that set the DMAC transfer request source. They are written as 8-bit values. T
initialized to H'00 by a reset, but retain their values in standby mode and a module sta

Bits 7 to 5—Reserved: These bits are always read as 0. The write value should always

Bits 4 to 0—Resource Select Bits 4 to 0 (RS4 to RS0): Specify which transfer request
the DMAC. Changing the transfer request source must be done when the DMA enable
0. See section 11.3.4, DMA Transfer Types, for the possible setting combinations.

Bits RS4 to RS0 are initialized to 001 by a reset.

Bit 4: RS4	**Bit 3: RS3**	**Bit 2: RS2**	**Bit 1: RS1**	**Bit 0: RS0**	**Description**
0	0	0	0	0	DREQ (external request) (
				1	Reserved (setting prohibited)
			1	0	Reserved (setting prohibited)
				1	Reserved (setting prohibited)
		1	0	0	Reserved (setting prohibited)
				1	SCIF channel 1 RXI (on-chip SCI with F 1 receive-data-full interrupt request)*1
			1	0	SCIF channel 1 TXI (on-chip SCI with FI 1 transmit-data-empty interrupt request)
				1	Reserved (setting prohibited)

		1	0		interrupt request)*1
				1	TPU TGI0B (on-chip TPU input capture c interrupt request)*1
			1	0	TPU TGI0C (on-chip TPU input capture c interrupt request)*1
				1	TPU TGI0D (on-chip TPU input capture c interrupt request)*1
1	0	0	0	0	Reserved (setting prohibited)
				1	SIO channel 0 RDFI (on-chip SIO channe receive-data-full interrupt request)*1
			1	0	SIO channel 0 TDEI (on-chip SIO channe transmit-data-empty interrupt request)*1
				1	Reserved (setting prohibited)
		1	0	0	Reserved (setting prohibited)
				1	SIO channel 1 RDFI (on-chip SIO channe receive-data-full interrupt request)*1
			1	0	SIO channel 1 TDEI (on-chip SIO channe transmit-data-empty interrupt request)*1
				1	Reserved (setting prohibited)
	1	0	0	0	Reserved (setting prohibited)
				1	SIO channel 2 RDFI (on-chip SIO channe receive-data-full interrupt request)*1
			1	0	SIO channel 2 TDEI (on-chip SIO channe transmit-data-empty interrupt request)*1
				1	Reserved (setting prohibited)
		1	*	*	Reserved (setting prohibited)

Note: * Don’t care

1. When a transfer request is generated by an on-chip module, select cycle-ste bus mode, dual transfer as the transfer mode, and falling edge detection for setting.

	—	—	—	—	PR	AE	NMI
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/(W)*	R/(W

Note: * Only 0 can be written, to clear the flag.

The DMA operation register (DMAOR) is a 32-bit read/write register that controls the transfer mode. It also indicates the DMA transfer status. Only the lower four of the 32 valid. DMAOR is written as a 32-bit value, including the upper 28 bits. DMAOR is in H'00000000 by a reset and in standby mode. It retains its value when the module stand is used.

Bits 31 to 4—Reserved: These bits are always read as 0. The write value should alway

Bit 3—Priority Mode Bit (PR): Specifies whether a fixed channel priority order or rou mode is to be used there are simultaneous transfer requests for multiple channels. It is 0 by a reset and in standby mode. It retains its value when the module standby functio

Bit 3: PR	Description
0	Fixed priority (channel 0 > channel 1) (
1	Round-robin (Top priority shifts to bottom after each transf priority for the first DMA transfer after a reset is channel 1

Bit 2—Address Error Flag Bit (AE): This flag indicates that an address error has occu DMAC. When the AE bit is set to 1, DMA transfer cannot be enabled even if the DE b DMA channel control register (CHCR) is set to 1. To clear the AE bit, read 1 from it a write 0. Operation is performed up to the DMAC transfer being executed when the ad occurred. AE is initialized to 0 by a reset and in standby mode. It retains its value whe module standby function is used.

then write 0. Operation is completed up to the end of the DMAC transfer being execute
NMI was input. When the NMI interrupt is input while the DMAC is not operating, the
is set to 1. The NMIF bit is initialized to 0 by a reset or in the standby mode. It retains i
when the module standby function is used.

Bit 1: NMIF	Description	
0	No NMIF interrupt To clear the NMIF bit, read 1 from it and then write 0	(Ir
1	NMIF interrupt has occurred	

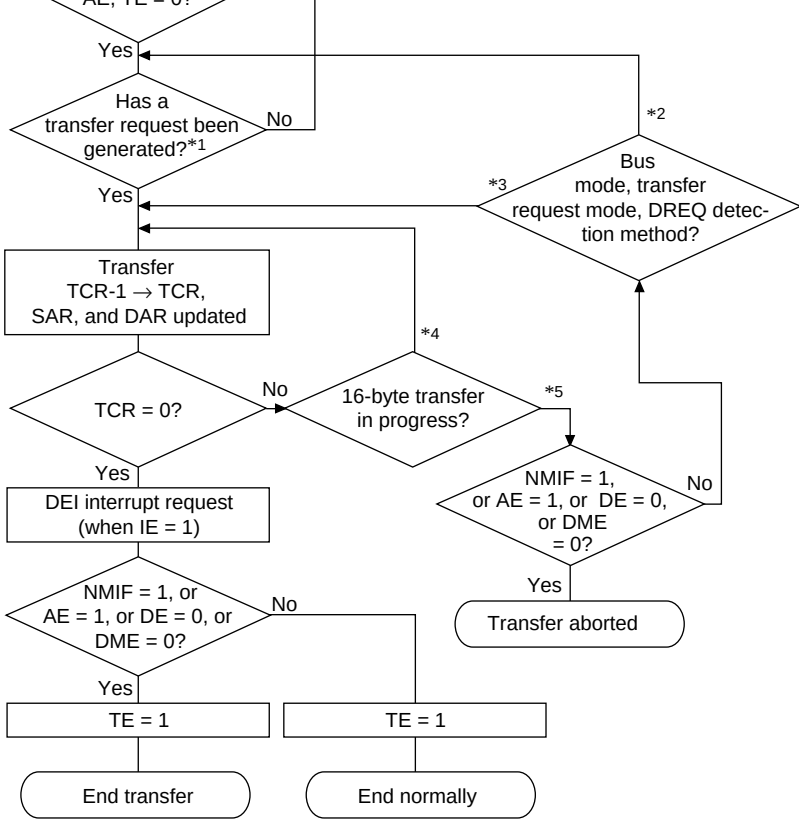
Bit 0—DMA Master Enable Bit (DME): Enables or disables DMA transfers on all char
DMA transfer becomes enabled when the DE bit in the CHCR and the DME bit are set
this to be effective, the TE bit in CHCR and the NMIF and AE bits must all be 0. When
bit is cleared, all channel DMA transfers are aborted. DME is initialized to 0 by a reset
standby mode. It retains its value when the module standby function is used.

Bit 0: DME	Description	
0	DMA transfers disabled on all channels	(Ir
1	DMA transfers enabled on all channels	

After the DMA source address registers (SAR), DMA destination address registers (D
transfer count registers (TCR), DMA channel control registers (CHCR), DMA vector
registers (VCRDMA), DMA request/response selection control registers (DRCR), and
operation register (DMAOR) are initialized (initializing sets each register so that ultim
condition (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0) is satisfied), the DMAC tran
according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0
2. When a transfer request occurs and transfer is enabled, the DMAC transfers 1 tran
   data. (In auto-request mode, the transfer begins automatically after register initializ
   TCR value will be decremented by 1.) The actual transfer flows vary depending on
   mode and bus mode.
3. When the specified number of transfers have been completed (when TCR reaches
   transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrup
   sent to the CPU.
4. When an address error occurs in the DMAC or an NMI interrupt is generated, the t
   aborted. Transfers are also aborted when the DE bit in CHCR or the DME bit in D
   changed to 0.

Figure 11.2 shows a flowchart illustrating this procedure.



Notes: 1. In auto-request mode, the transfer will start when the NMIF, AE, and TE bits
and the DE and DME bits are then set to 1.
2. Cycle-steal mode.
3. In burst mode, DREQ = edge detection (external request), or auto-request m
burst mode.
4. 16-byte transfer cycle in progress.
5. End of a 16-byte transfer cycle.

Figure 11.2 DMA Transfer Flow

Table 11.3 Selecting the DMA Transfer Request Using the AR and RS Bits

CHCR	DRCR						
AR	**RS4**	**RS3**	**RS2**	**RS1**	**RS0**	**Request Mode**	**Resource Selec**
0	0	0	0	0	0	Module request mode	DREQ (external r
			1	0	1		SCIF channel 1 R
				1	0		SCIF channel 1 T
		1	0	0	1		SCIF channel 2 R
				1	0		SCIF channel 2 T
			1	0			TPU TGI0A
					1		TPU TGI0B
				1	0		TPU TGI0C
					1		TPU TGI0D
	1	0	0	0			SIO channel 0 RI
				1	0		SIO channel 0 TD
			1	0	1		SIO channel 1 RI
				1	0		SIO channel 1 TD
		1	0	0	1		SIO channel 2 RI
				1	0		SIO channel 2 TD
1	*	*	*	*	*	Auto-request mode	

Note: * Don't care

Auto-Request Mode: When there is no transfer request signal from an external sourc memory-to-memory transfer, the auto-request mode allows the DMAC to automatical transfer request signal internally. When the DE bits in CHCR0 and CHCR1 and the D the DMA operation register (DMAOR) are set to 1, the transfer begins (so long as the CHCR0 and CHCR1 and the NMIF and AE bits in DMAOR are all 0).

0	0	Dual address mode	DACKn output in read cycle	Any*	Any*
	1	Dual address mode	DACKn output in write cycle	Any*	Any*
1	0	Single address mode	Data transferred from memory to device	External memory or memory-mapped external device	External with DA
	1	Single address mode	Data transferred from device to memory	External device with DACK	External or memo mapped device

Note: * External memory, memory-mapped external device, and on-chip peripheral (excluding DMAC, BSC, UBC, cache memory, E-DMAC, and EtherC).

Choose to detect DREQn either by the falling edge or by level using the DS and DL bit CHCR0 and CHCR1 (DS = 0 is level detection, DS = 1 is edge detection; DL = 0 is act DL = 1 is active-high). The source of the transfer request does not have to be the data tr source or destination.

When 0 (level detection) is set to the DS bit of CHCR0 and CHCR1, set the TB bit to 0 steal mode) and set the TS1 and TS0 bits of CHCR0 and CHCR1 to either 00 (byte unit unit), or 10 (long word unit).

When 0 is set to the DS bit of CHCR0 and CHCR1, when 1 (burst mode) is set to the T CHCR0 and CHCR1, and when 11 (16 byte unit) is set to the TS1 and TS0 bits of CHC CHCR1, operation is not guaranteed.

On-Chip Module Request Mode: In this mode, transfers are started by a transfer req
(interrupt request signal) from an on-chip peripheral module. Transfer request signals
SCIF and SIO receive-data-full interrupts (RXI, RDFI), SCIF and SIO transmit-data-e
interrupts (TXI, TDEI), and TPU general registers (table 11.6). If DMA transfer is ena
1, DME = 1, TE = 0, NMIF = 0, AE = 0), DMA transfer starts upon input of a transfer
signal.

When RXI or RDFI (transfer request due to an SCIF or SIO receive-data-full conditio
transfer request, the transfer source must be the receive data register of the correspond
(SCFRDR or SIRDR). When TXI or TDEI (transfer request due to an SCIF or SIO tra
empty condition) is set as a transfer request, the transfer destination must be the transm
register of the corresponding module (SCFTDR or SITDR).

These restrictions do not apply to TPU transfer requests.

When on-chip module request mode is used, an access size permitted by the periphera
register used as the transfer source or transfer destination must be set in bits TS1 and T
CHCR0 and CHCR1.

	1	0	0	0	1	SCIF channel 2 receiver	RXI	SCFRDR2	Any	Cycle-steal
				1	0	SCIF channel 2 transmitter	TXI	Any	SCFTDR2	Cycle-steal
			1	0	0	TPU channel 0A	TGI0A	Any (excluding on-chip RAM)	Any (excluding on-chip RAM)	Cycle-steal
					1	TPU channel 0B	TGI0B	Any (excluding on-chip RAM)	Any (excluding on-chip RAM)	Cycle-steal
				1	0	TPU channel 0C	TGI0C	Any (excluding on-chip RAM)	Any (excluding on-chip RAM)	Cycle-steal
					1	TPU channel 0D	TGI0D	Any (excluding on-chip RAM)	Any (excluding on-chip RAM)	Cycle-steal
1	0	0	0	0	1	SIO channel 0 receiver	RDFI	SIRDR0	Any	Cycle-steal
				1	0	SIO channel 0 transmitter	TDEI	Any	SITDR0	Cycle-steal
			1	0	1	SIO channel 1 receiver	RDFI	SIRDR1	Any	Cycle-steal
				1	0	SIO channel 1 transmitter	TDEI	Any	SITDR1	Cycle-steal
		1	0	0	1	SIO channel 2 receiver	RDFI	SIRDR2	Any	Cycle-steal
				1	0	SIO channel 2 transmitter	TDEI	Any	SITDR2	Cycle-steal

Note: * Do not perform transfers between on-chip peripheral modules.

transfer request (interrupt request) from any module will be cleared at the first transfe

11.3.3 Channel Priorities

When the DMAC receives simultaneous transfer requests on two channels, it selects a according to a predetermined priority order. There is a choice of two priority modes, f round-robin. The mode is selected by the priority bit, PR, in the DMA operation regist (DMAOR).

Fixed Priority Mode: In this mode, the relative channel priority levels are fixed. Whe to 0, channel 0 has higher priority than channel 1. Figure 11.3 shows an example of a burst mode.

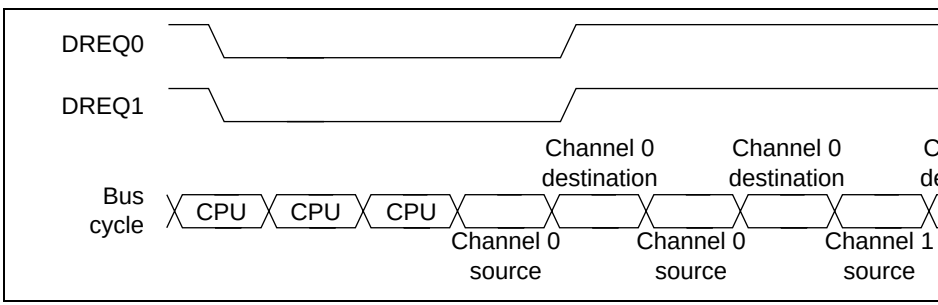


Figure 11.3 Fixed Mode DMA Transfer in Burst Mode (Dual Address, DREQn Falling-Edge Detection)

In cycle-steal mode, once a channel 0 request is accepted, channel 1 requests are also until the next request is accepted, which makes more effective use of the bus cycle. If come simultaneously for channel 0 and channel 1 when DMA operation is starting, th transmitted with channel 0, and thereafter channel 1 and channel 0 transfers are perfor alternately.

Figure 11.4 Fixed Mode DMA Transfer in Cycle-Steal Mode (Dual Address, DREQn Low-Level Detection)

Round-Robin Mode: Switches the priority of channel 0 and channel 1, shifting their al receive transfer requests. Each time one transfer ends on one channel, the priority shifts other channel. The channel on which the transfer just finished is assigned low priority. channel 1 has higher priority than channel 0.

Figure 11.5 shows how the priority changes when channel 0 and channel 1 transfers are simultaneously and another channel 0 transfer is requested after the first two transfers e DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 1 and 0.
2. Channel 1 has the higher priority, so the channel 1 transfer begins first (channel 0 w transfer).
3. When the channel 1 transfer ends, channel 1 becomes the lower-priority channel.
4. The channel 0 transfer begins.
5. When the channel 0 transfer ends, channel 0 becomes the lower-priority channel.
6. A channel 0 transfer is requested.
7. The channel 0 transfer begins.
8. When the channel 0 transfer ends, channel 0 is already the lower-priority channel, s remains the same.

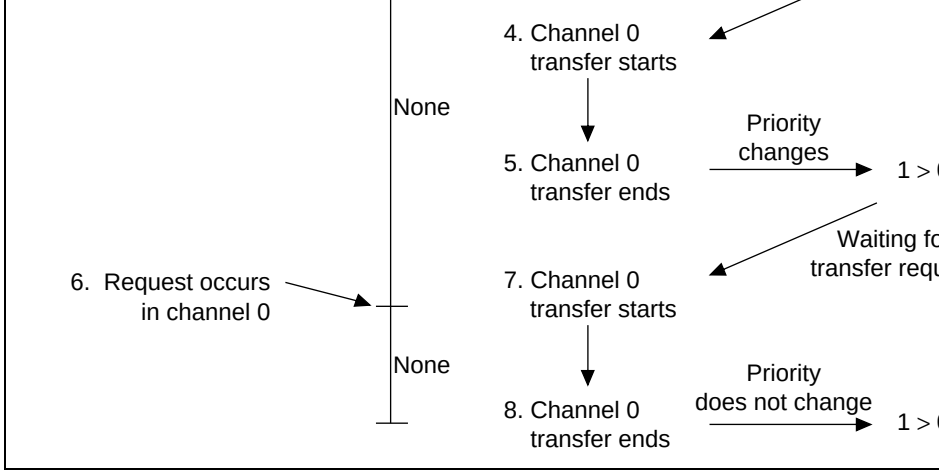


Figure 11.5 Channel Priority in Round-Robin Mode

	Destination				
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**	**O M**
External device with DACK	Not available	Single	Single	Not available	N
External memory	Single	Dual	Dual	Dual*	D
Memory-mapped external device	Single	Dual	Dual	Dual*	D
On-chip peripheral module	Not available	Dual*	Dual*	Dual*	D
On-chip memory	Not available	Dual	Dual	Dual*	D

Single: Single address mode

Dual: Dual address mode

Note: * Access size permitted by peripheral module register used as transfer source destination (excluding DMAC, BSC, UBC, cache memory, E-DMAC, and Eth

Address Modes:

- Single Address Mode

  In single address mode, both the transfer source and destination are external; one (s accessed by a DACKn signal while the other is accessed by address. In this mode, t performs the DMA transfer in one bus cycle by simultaneously outputting a transfer acknowledge DACKn signal to one external device to access it, while outputting an the other end of the transfer. Figure 11.6 shows an example of a transfer between ex memory and external device with DACK. That data is written in external memory i bus cycle while the external device outputs data to the data bus.

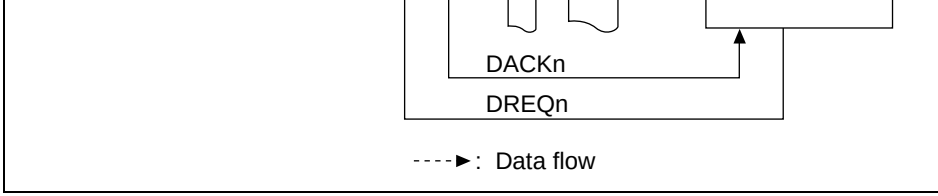


Figure 11.6 Data Flow in Single Address Mode

Two types of transfers are possible in single address mode: 1) transfers between ex
devices with DACK and memory-mapped external devices; and 2) transfers betwe
devices with DACK and external memory. For both of them, transfer must be requ
external request signal (DREQn). For the combination of the specifiable setting to
data transfer using an external request (DREQn), see table 11.9. Figure 11.7 shows
transfer timing for single address mode.

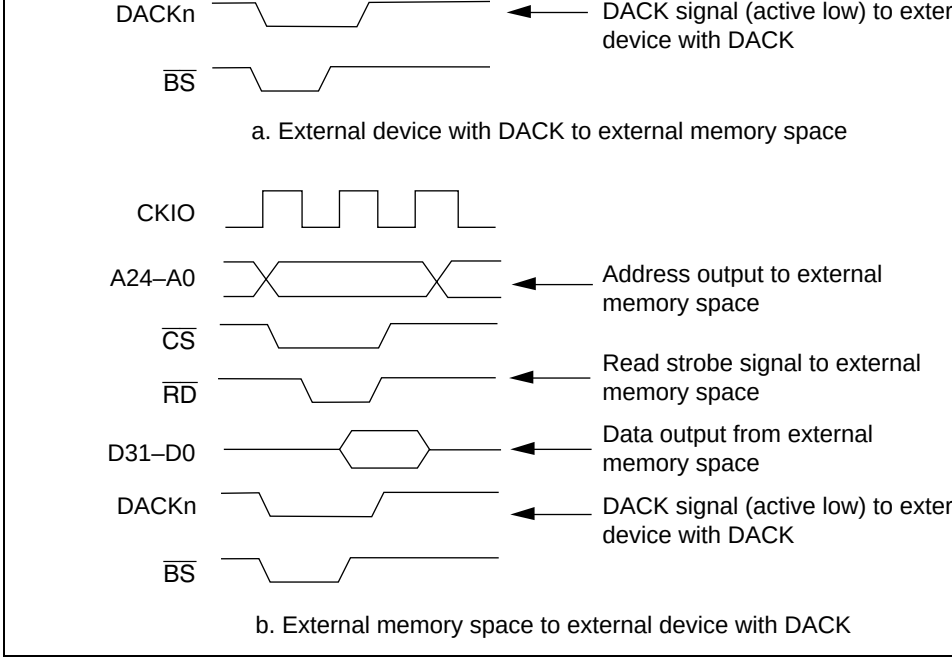


Figure 11.7 DMA Transfer Timing in Single Address Mode

- Dual Address Mode

  In dual address mode, both the transfer source and destination are accessed (selectal address. The source and destination can be located externally or internally. The DM accesses the source in the read cycle and the destination in the write cycle, so the tra performed in two separate bus cycles. The transfer data is temporarily stored in the Figure 11.8 shows an example of a transfer between two external memories in whic read from one external memory in the read cycle and written to the other external m the following write cycle.

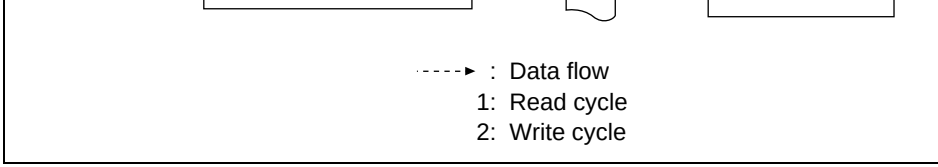


Figure 11.8 Data Flow in Dual Address Mode

In dual address mode transfers, external memory and memory-mapped external de
mixed without restriction. Specifically, this enables transfers between the followin

— Transfer between external memory and external memory
— Transfer between external memory and memory-mapped external device
— Transfer between memory-mapped external device and memory-mapped exter
— Transfer between external memory and on-chip peripheral module (excluding
BSC, UBC, cache, E-DMAC, and EtherC)*
— Transfer between memory-mapped external device and on-chip peripheral mod
(excluding DMAC, BSC, UBC, cache, E-DMAC, and EtherC)*
— Transfer between on-chip memory and on-chip memory
— Transfer between on-chip memory and memory-mapped external device
— Transfer between on-chip memory and on-chip peripheral module (excluding
BSC, UBC, cache, E-DMAC, and EtherC)*
— Transfer between on-chip memory and external memory
— Transfer between on-chip peripheral module (excluding DMAC, BSC, UBC, c
DMAC, and EtherC) and on-chip peripheral module (excluding DMAC, BSC,
E-DMAC, and EtherC)*

Note: * Access size permitted by peripheral module register used as transfer so
transfer destination (excluding DMAC, BSC, UBC, cache, E-DMAC, a

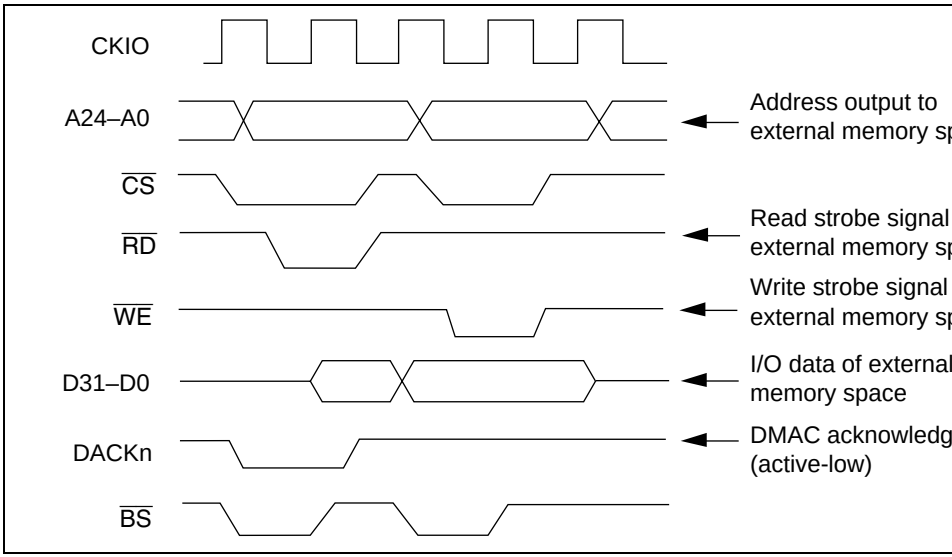


Figure 11.9 DMA Transfer Timing in Dual Address Mode
(External Memory Space → External Memory Space, DACKn Output in Read

continues to hold the bus)

Cycle-steal mode can be used with all categories of transfer destination, transfer so
transfer request source. (with the exception of transfers between on-chip peripheral

The CPU may take the bus twice when an acknowledge signal is output during the
or in single address mode. Figure 11.10 shows an example of DMA transfer timing
steal mode. The transfer conditions for the example in the figure are as shown belo

When the transfer request source is an external request mode with level detection i
steal mode, set the TS1 and TS0 bits of CHCR0 and CHCR1 to either 00 (byte uni
unit), or 01 (longword unit). If the TS1 and TS0 bits of CHCR0 and CHCR1 are se
byte transfer), operation is not guaranteed.

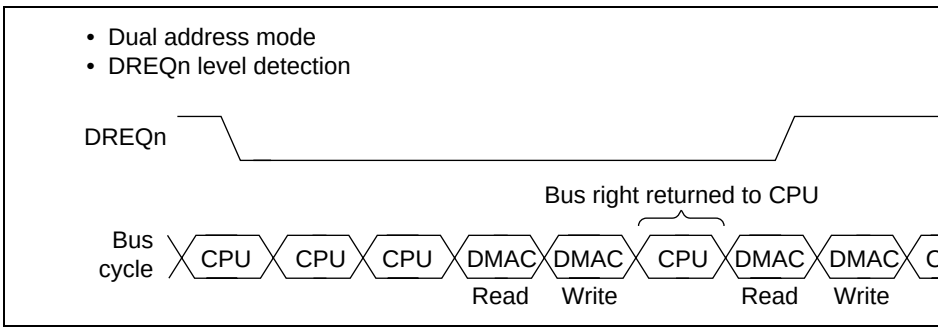


Figure 11.10 DMA Transfer Timing in Cycle-Steal Mode (Dual Address Mode, DREQn Low Level Detection)

for the example in the figure are as shown below.

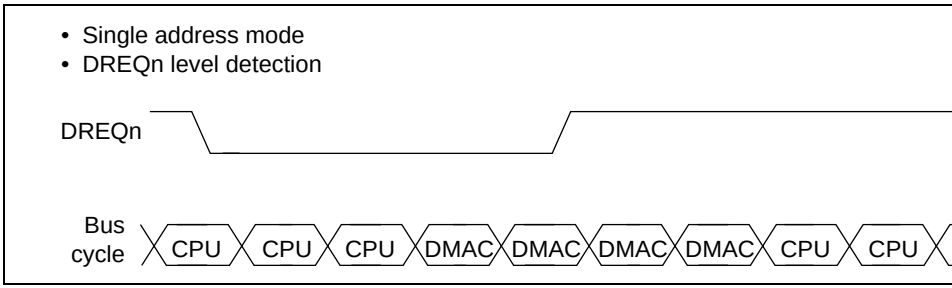


Figure 11.11 DMA Transfer Timing in Burst Mode (Single Address, DREQn Falling-Edge Detection)

Refreshes cannot be performed during a burst transfer, so ensure that the number of
satisfies the refresh request period when a memory requiring refreshing is used. Wh
transfer request source is an external request (DREQn) in burst mode, set the DS bit
and CHCR1 to 1 (edge detection). If the DS bits of CHCR0 and CHCR1 are set to 0
detection), operation is not guaranteed.

	Between external device with DACK, and memory mapped external device	External	B/C
Dual	Between external memories	External	B/C
		Automatic	B/C
		Internal peripheral module*1	C
	Between external memory and memory mapped external device	External	B/C
		Automatic	B/C
		Internal peripheral module*1	C
	Between memory mapped external devices	External	B/C
		Automatic	B/C
		Internal peripheral module*1	C
	Between external memory and internal peripheral module	External	B/C
		Automatic	B/C
		Internal peripheral module*2	C
	Between memory mapped external device and internal peripheral module	External	B/C
		Automatic	B/C
		Internal peripheral module*2	C
	Between internal memories	Automatic	B/C
	Between internal memory and memory mapped external device*5	External	B/C
		Automatic	B/C
		Internal peripheral module*1	C

Between internal peripheral modules	External	B/C	1
	Automatic	B/C	1
	Internal peripheral module*2	C	1

Notes: B: Burst mode

C: Cycle steal mode

1. For on-chip peripheral module requests, do not specify SCIF and SIO as a tr request source.
2. When the transfer request source is SCIF or SIO, the transfer source or tran destination must be SCIF and SIO, respectively.
3. When the request mode is set to internal peripheral module request, set the the DL bit of CHCR0 and CHCR1 to 1 and 0, respectively (detection at the fa of DREQn). In addition, the bus mode can only be set to cycle-steal mode.
4. Specify the access size that is allowed by the internal peripheral-module reg which are a transfer source or a transfer destination.
5. When transferring data from internal memory to a memory mapped external DACKn to write-time output. When transferring from a memory mapped exte to internal memory, set DACKn to read-time output.
6. When transferring data from internal memory to external memory, set DACK time output. When transferring from external memory to internal memory, se read-time output.
7. When B (burst mode) is set in the external request mode, set the DS bits of and CHCR1 to 1 (edge detection). If they are set to 0 (level detection), opera be guaranteed.
8. Transfer in units of 16 bytes is enabled only when edge detection has been s transfer is attempted in units of 16 bytes when level detection has been spec operation cannot be guaranteed.

External request	Level detection*1	Byte	—	O	—	O
		Word	—	O	—	O
		Longword	—	O	—	O
		16-byte unit	—	—	—	—
	Edge detection*2	Byte	O	O	O	O
		Word	O	O	O	O
		Longword	O	O	O	O
		16-byte unit	O	O	O	O

Notes: O: Can be set
—: Cannot be set
1. The same for high-level and low-level detection.
2. The same for rising-edge detection and falling-edge detection.

Bus Mode and Channel Priority: When a given channel (1) is transferring in burst m
there is a transfer request to a channel (0) with a higher priority, the transfer of the cha
higher priority (0) will begin immediately. When channel 0 is also operating in the bu
channel 1 transfer will continue as soon as the channel 0 transfer has completely finish
channel 0 is in cycle-steal mode, channel 1 will begin operating again after channel 0
the transfer of one transfer unit, but the bus will then switch between the two in the or
1, channel 0, channel 1, channel 0. Since channel 1 is in burst mode, it will not give th
CPU. This example is illustrated in figure 11.12.

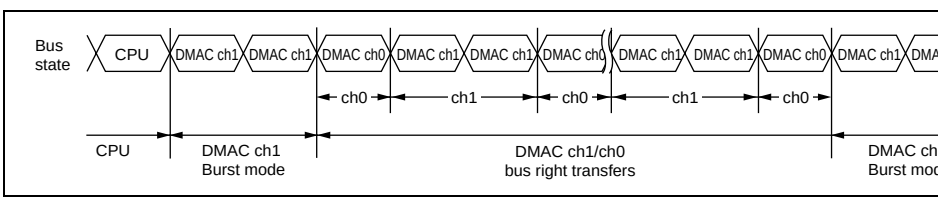


Figure 11.12 Bus Status when Multiple Channels are Operating
(when priority order is ch0 > ch1, ch1 is set to burst mode, and ch0 to cycle-st

output specified by the channel control register AM bit of the address bus. Normally, th
acknowledge signal becomes valid when DMA address output begins, and becomes inv
cycles before the address output ends. (See figure 11.13.) The output timing of the ackn
signal varies with the settings of the connected memory space. The output timing of ack
signals in the memory spaces is shown in figure 11.13.

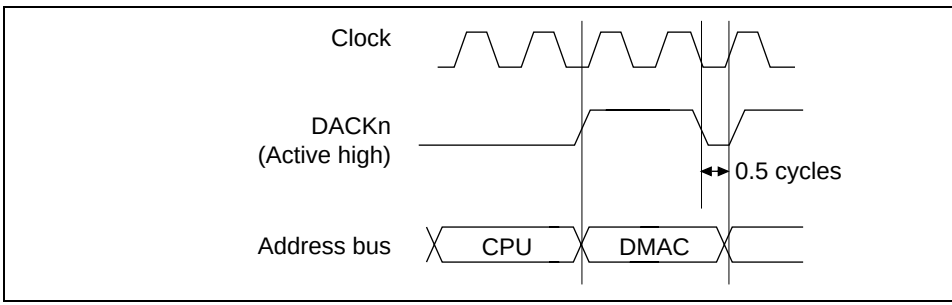


Figure 11.13 Example of DACKn Output Timing

Acknowledge Signal Output when External Memory Is Set as Ordinary Memory S

The timing at which the acknowledge signal is output is the same in the DMA read and
cycles specified by the AM bit (figures 11.14 and 11.15). When DMA address output b
acknowledge signal becomes valid; 0.5 cycles before address output ends, it becomes in
wait is inserted in this period and address output is extended, the acknowledge signal is
extended.

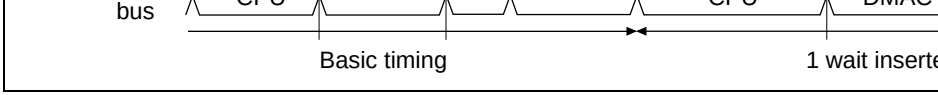


Figure 11.14 DACKn Output in Ordinary Space Accesses (AM = 0)

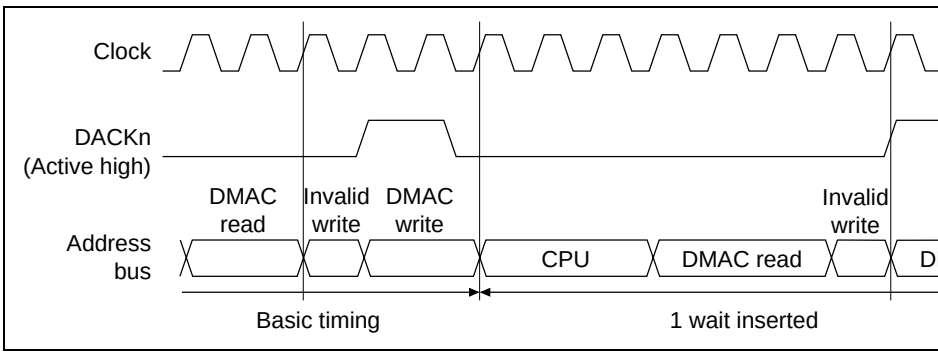


Figure 11.15 DACKn Output in Ordinary Space Accesses (AM = 1)

In a longword access of a 16-bit external device (figure 11.16) or an 8-bit external dev
11.17), or a word access of an 8-bit external device (figure 11.18), the lower and uppe
are output 2 and 4 times in each DMAC access in order to align the data. For all of the
addresses, the acknowledge signal becomes valid simultaneous with the start of outpu
signal becomes invalid 0.5 cycles before the address output ends. When multiple addr
output in a single access to align data for synchronous DRAM, DRAM, or burst ROM
acknowledge signal is output to those addresses as well.

Basic timing

Notes: 1. H: MSB side
2. L: LSB side

Figure 11.16 DACKn Output in Ordinary Space Accesses (AM = 0, Longword Access to 16-Bit External Device)

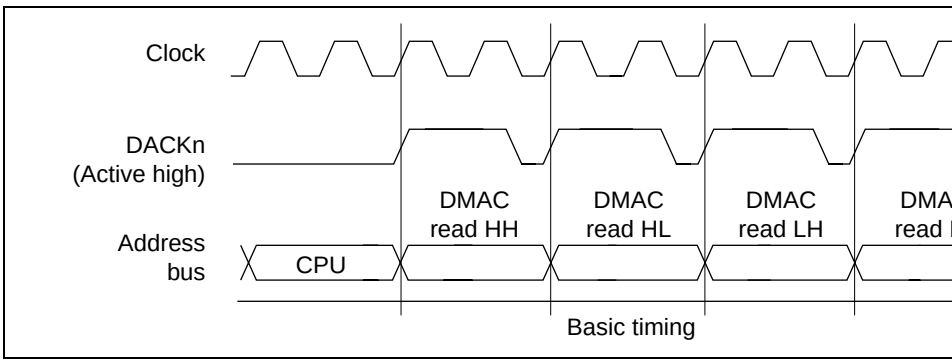


Figure 11.17 DACKn Output in Ordinary Space Accesses (AM = 0, Longword Access to 8-Bit External Device)

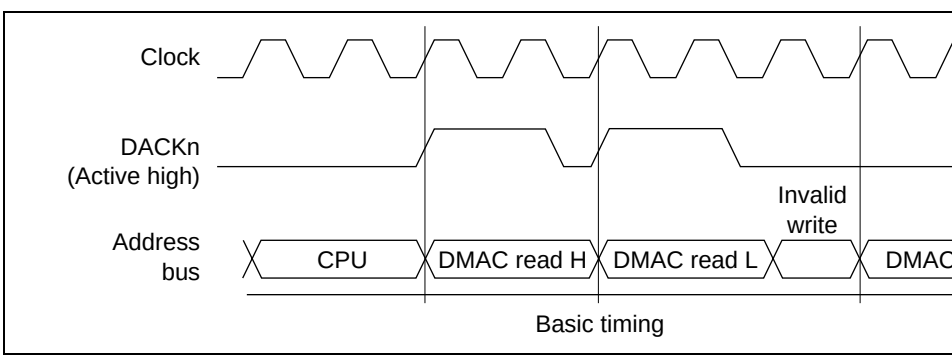


Figure 11.18 DACKn Output in Ordinary Space Accesses (AM = 0, Word Access to 8-Bit External Device)

output after the invalid read. A synchronous DRAM burst read is performed in the cas
transfer. As 16-byte transfer is enabled only in auto-request mode and in external requ
with edge detection, when using on-chip peripheral module requests or external reque
level detection, byte, word, or longword should be set as the transfer unit. Operation is
guaranteed if a 16-byte unit is set when using on-chip peripheral module requests or e
request mode with level detection. When AM = 1, the acknowledge signal is output ac
address and column address of the DMAC write (figure 11.21).

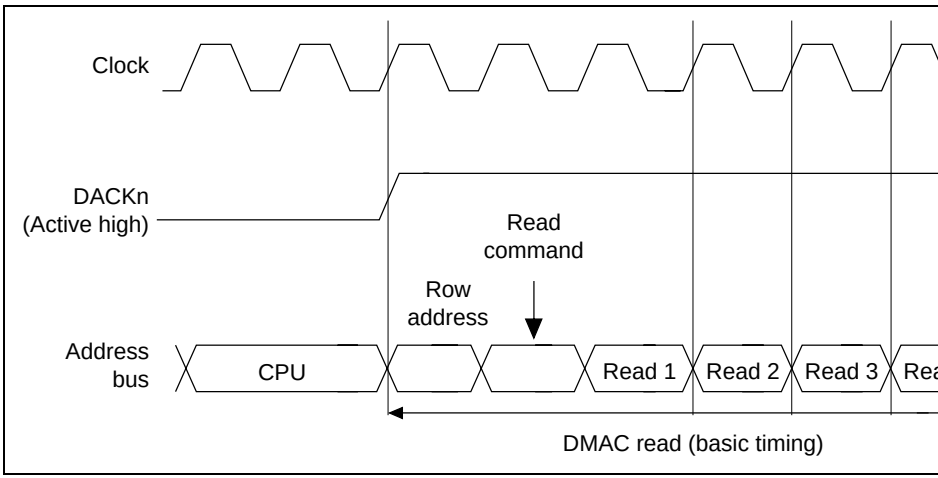


Figure 11.19 DACKn Output in Synchronous DRAM Burst Read (Auto-Precharge, AM = 0)

Figure 11.20 DACKn Output in Synchronous DRAM Single Read (Auto-Precharge, AM = 0)

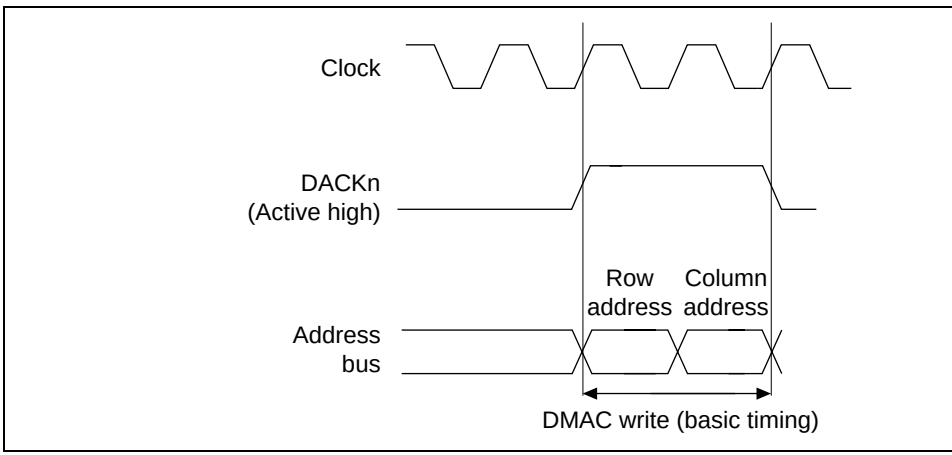


Figure 11.21 DACKn Output in Synchronous DRAM Write (Auto-Precharge, AM = 1)

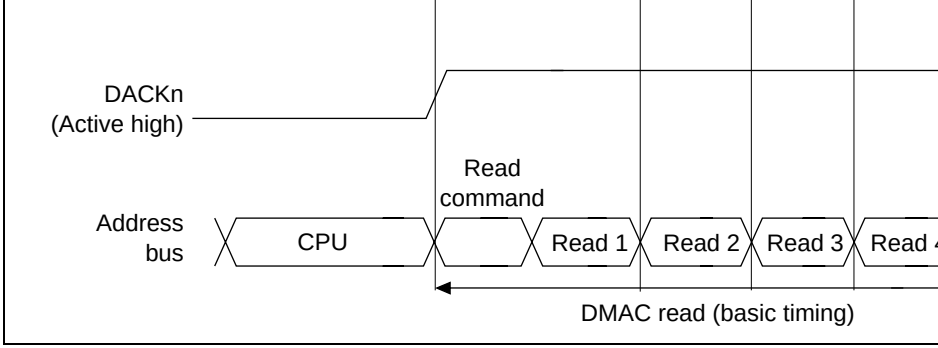


Figure 11.22 DACKn Output in Synchronous DRAM Burst Read (Bank Active, Same Row Address, AM = 0)

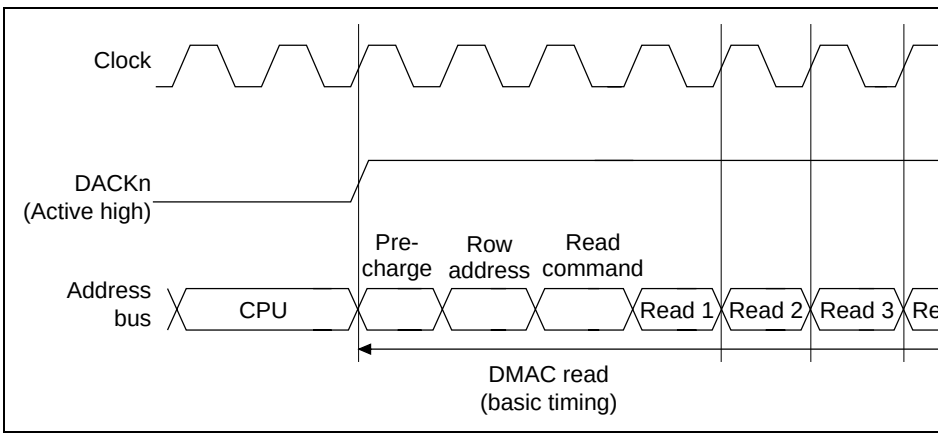


Figure 11.23 DACKn Output in Synchronous DRAM Burst Read (Bank Active, Different Row Address, AM = 0)

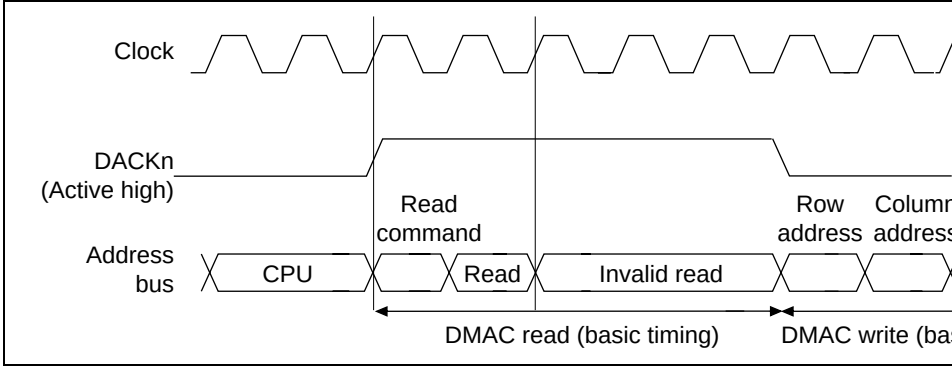


Figure 11.24 DACKn Output in Synchronous DRAM Single Read (Bank Active, Same Row Address, AM = 0)

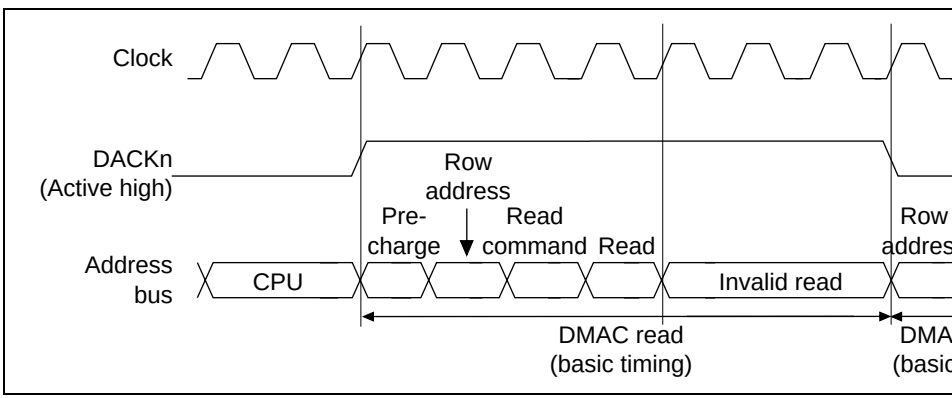


Figure 11.25 DACKn Output in Synchronous DRAM Single Read (Bank Active, Different Row Address, AM = 0)

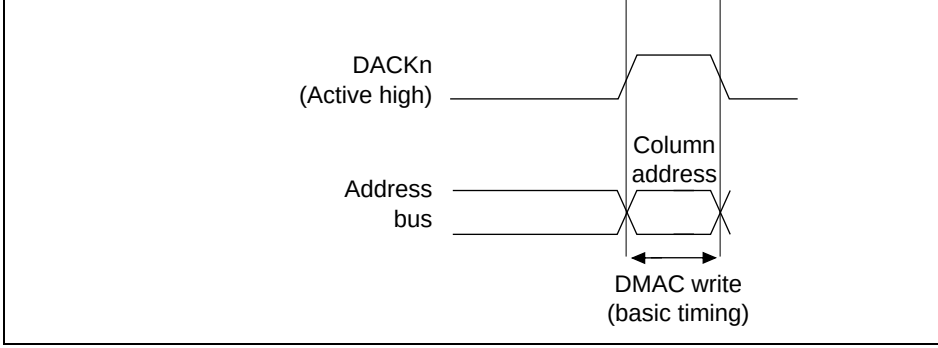


Figure 11.26 DACKn Output in Synchronous DRAM Write (Bank Active, Same Row Address, AM = 1)

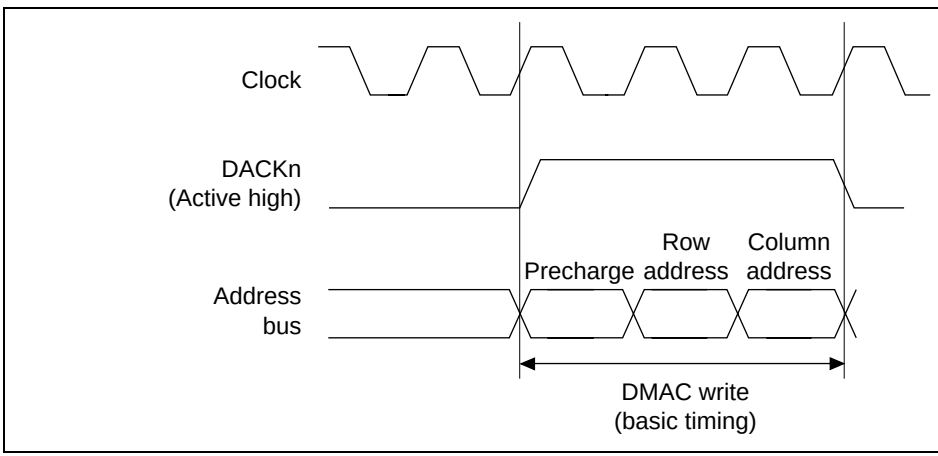


Figure 11.27 DACKn Output in Synchronous DRAM Write (Bank Active, Different Row Address, AM = 1)

Notes: 1. Do not set a 16-byte unit; operation is not guaranteed if this setting is made.
2. Cycle-steal mode must be set when DREQ is level-detected.

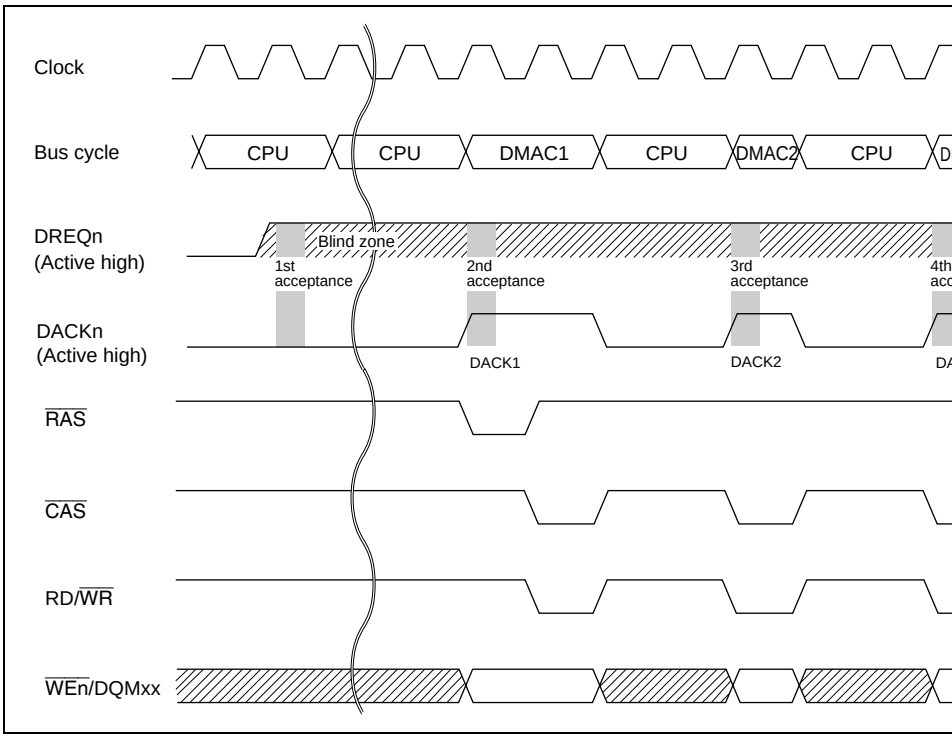


Figure 11.28 (a) Synchronous DRAM One-Cycle Write Timing

Transfer Width	Byte/Word/Longword Transfer	DREQn Detection Method	Edge D
Transfer bus mode	Burst mode	DACKn output timing	Write D
Transfer address mode	Single mode	Bus cycle	Basic bu

Note: * Edge detection must be set when burst mode is selected as the transfer bus

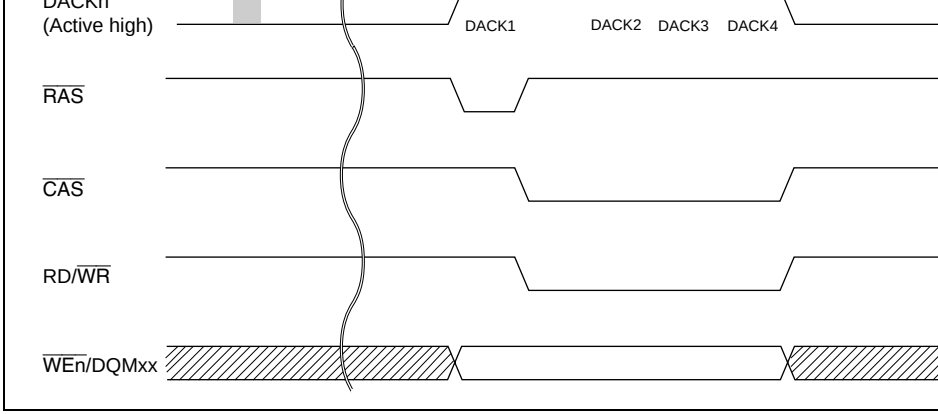


Figure 11.28 (b) Synchronous DRAM One-Cycle Write Timing

Acknowledge Signal Output when External Memory Is Set as DRAM: When exte
is set as DRAM and a row address is output during a read or write, the acknowledge s
output across the row address and column address (figures 11.29 to 11.31).

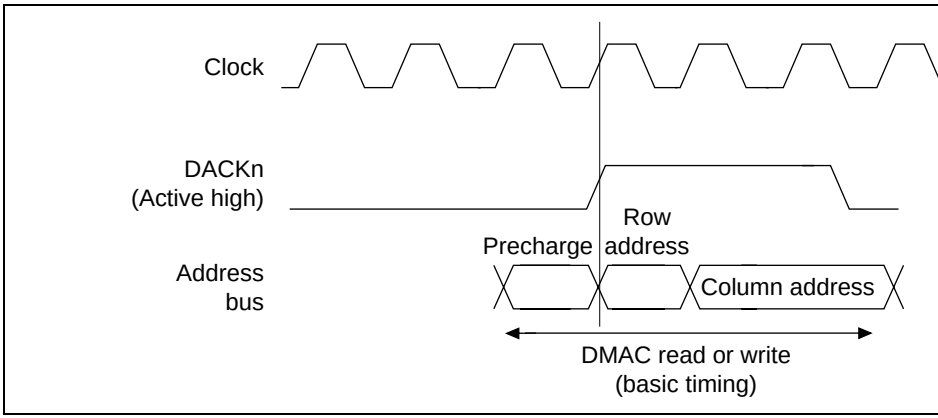


Figure 11.29 DACKn Output in Normal DRAM Accesses (AM = 0 or

(basic timing)

Figure 11.30 DACKn Output in DRAM Burst Accesses (Same Row Address, AM = 0 or 1)

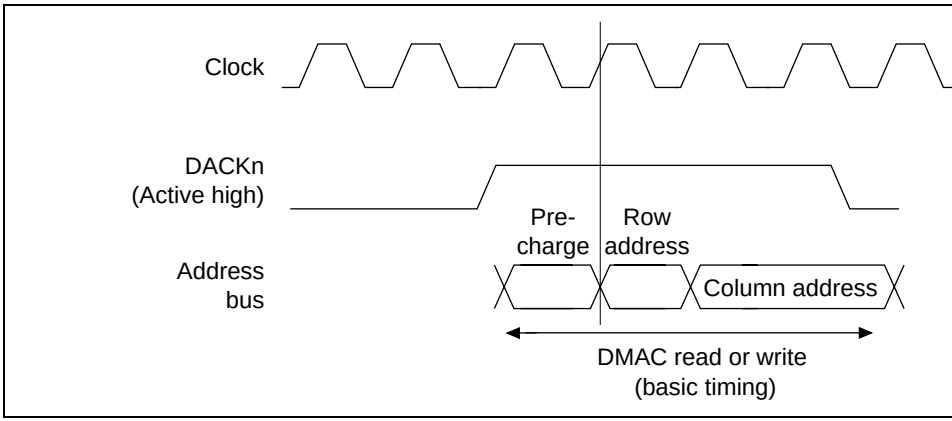


Figure 11.31 DACKn Output in DRAM Burst Accesses (Different Row Address, AM = 0 or 1)

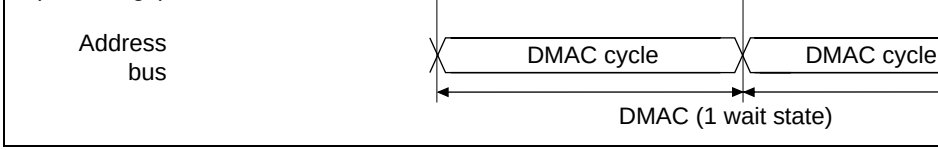


Figure 11.32 DACKn Output in Nibble Accesses of Burst ROM

11.3.7 DREQn Pin Input Detection Timing

In external request mode, DREQn pin signals are usually detected at the falling edge o
pulse (CKIO). When a request is detected, a DMAC bus cycle is produced four cycles
earliest and a DMA transfer performed. After the request is detected, the timing of the
detection varies with the bus mode, address mode, DREQn input detection, and the mo
connected.

DREQn Pin Input Detection Timing in Cycle-Steal Mode: In cycle-steal mode, one
is detected from the DREQn pin, the request signal is not detected until DACKn signa
the next external bus cycle. In cycle-steal mode, request detection is performed from D
signal output until a request is detected.

Once a request has been accepted, it cannot be canceled midway.

The timing from the detection of a request until the next time requests are detectable is
below.

- Cycle-Steal Mode Edge Detection

  When transfer control is performed using edge detection, perform DREQn/DACK
  handshaking as shown in figure 11.33, and perform DREQn input control so that t
  one-to-one relationship between DREQn and DACKn. Operation is not guarantee
  is input before the corresponding DACKn is output.

  If the DACKn signal is output a number of times, the first DACKn signal for the i
  signal indicates the request acceptance start timing, and subsequently each clock e
  sampled.

Figure 11.33 DREQn/DACKn Handshaking

- Edge Detection—1/2/4-Byte Transfer

Transfer Width	Byte/Word/Longword	DREQn Detection Method	Edge Det
Transfer bus mode	Cycle-steal mode	DACKn output timing	Read DAC DACK
Transfer address mode	Dual/single mode	Bus cycle	Basic bus

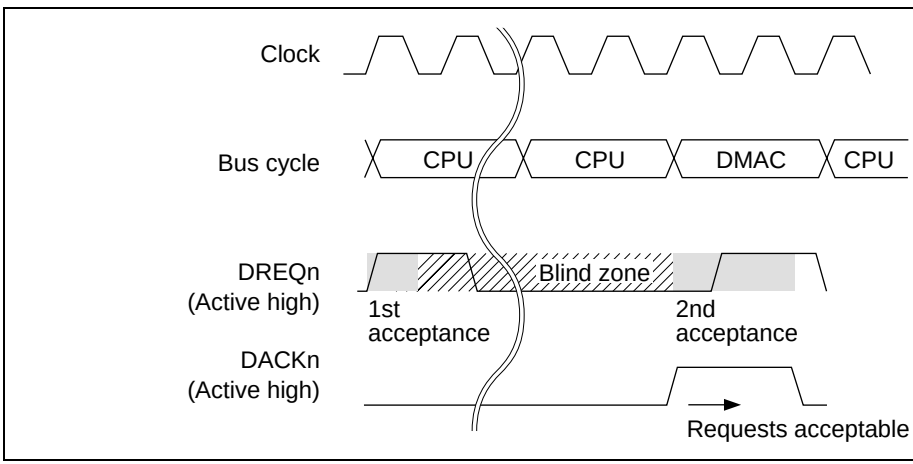


Figure 11.34 DREQn Pin Input Detection Timing in Cycle-Steal Mode with Edge

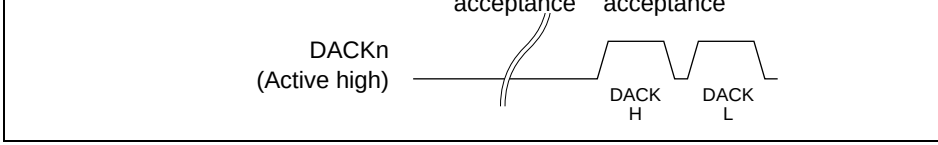


Figure 11.35 When a16-Bit External Device is Connected (Edge Detecti

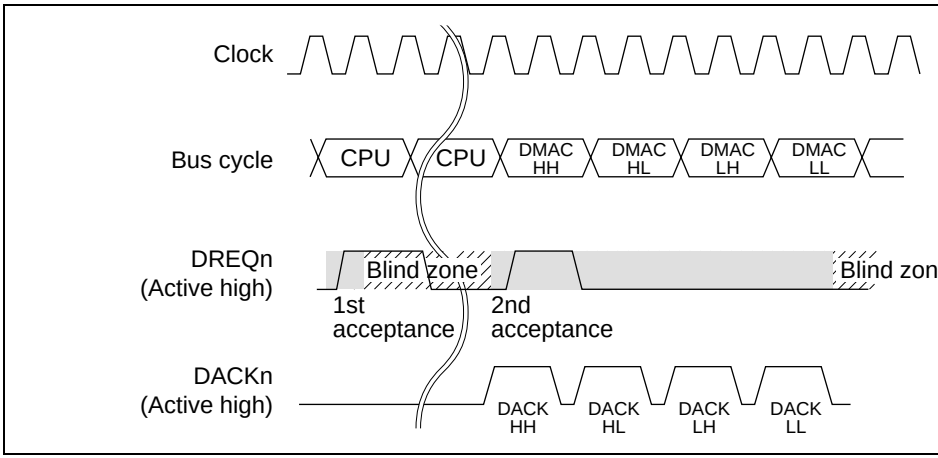
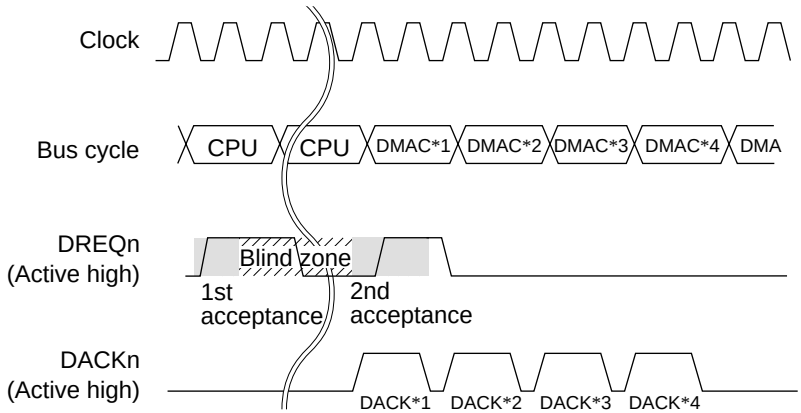


Figure 11.36 When an 8-Bit External Device is Connected (Edge Detect

Transfer address mode	Dual/single mode	Bus cycle	Basic bus c



Note: * n is the nth 16-byte transfer.

Figure 11.37 DREQn Pin Input Detection Timing in Cycle-Steal Mode with Edge (16-Byte Transfer Setting)

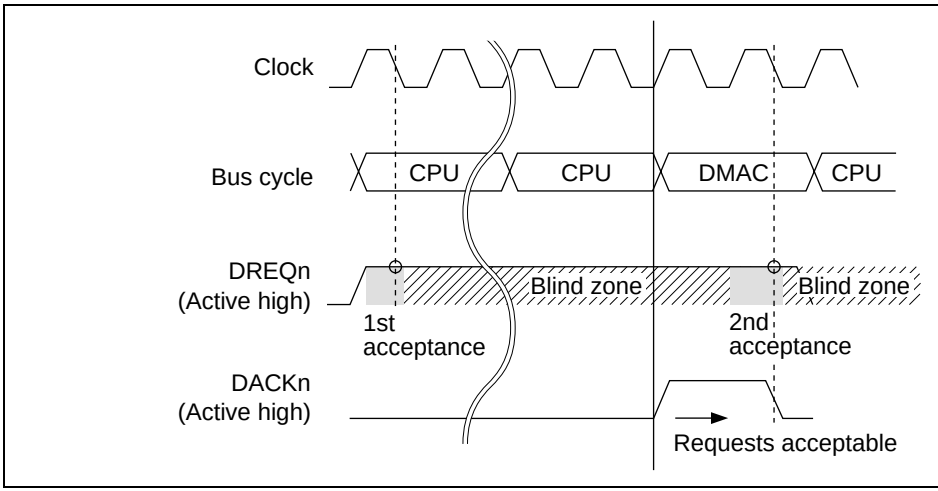


Figure 11.38 DREQn Pin Input Detection Timing in Cycle-Steal Mode with Lev (Byte/Word/Longword Setting)

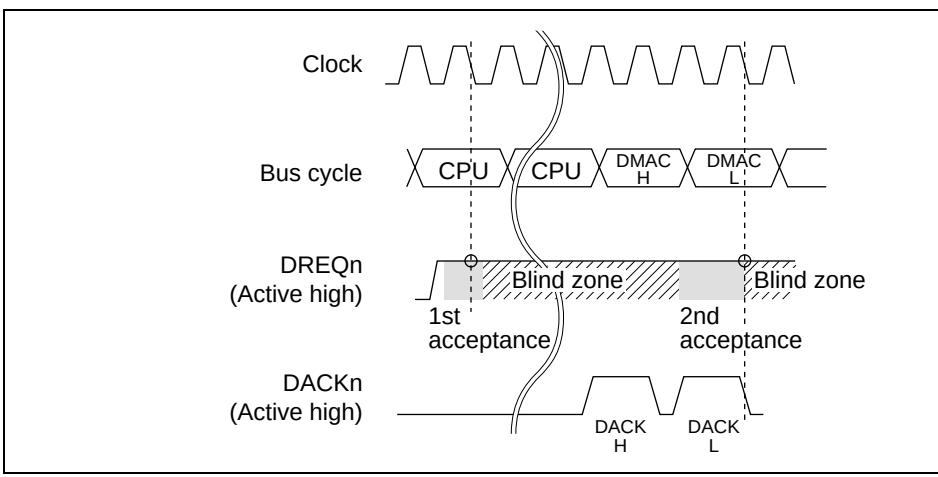


Figure 11.39 When a 16-Bit External Device is Connected (Level Detect

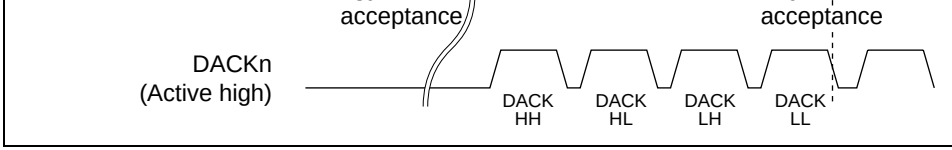


Figure 11.40 When an 8-Bit External Device is Connected (Level Detecti

DREQn Pin Input Detection Timing in Burst Mode: In burst mode, only edge detec
for DREQn input. Operation is not guaranteed if level detection is set.

With edge detection of DREQn input, once a request is detected, DMA transfer continu
transfer end condition is satisfied, regardless of the state of the DREQn pin. Request de
not performed during this time. When the transfer start conditions are fulfilled after the
transfer, request detection is performed again every cycle.

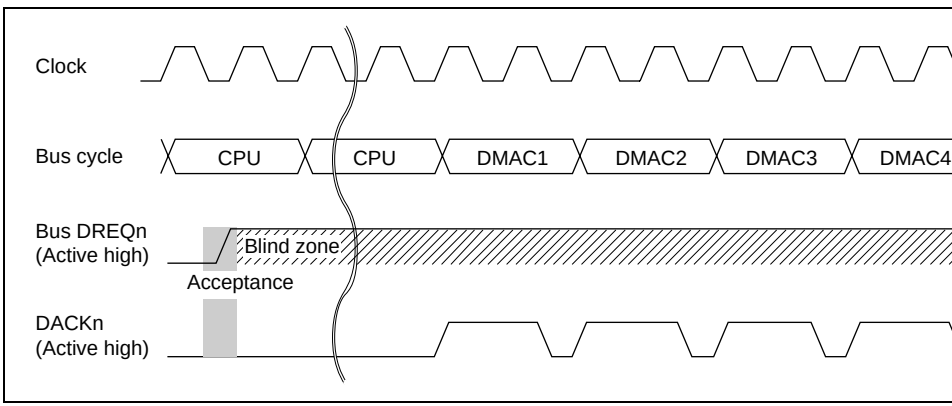


Figure 11.41 DREQn Pin Input Detection Timing in Burst Mode with Edge D

The DMA enable bit (DE) of the DMA channel control register (CHCR) is cleare

- Transfer end when TCR = 0

  When the TCR value becomes 0, the DMA transfer for that channel ends and the t flag bit (TE) is set in CHCR. If the IE (interrupt enable) bit has already been set, a interrupt (DEI) request is sent to the CPU. For 16-byte transfer, set the number of Operation is not guaranteed if an incorrect value is set.

  A 16-byte transfer is valid only in auto-request mode or in external request mode v detection. When using an external request with level detection or on-chip peripher request, do not specify a 16-byte transfer.

- Transfer end when DE = 0 in CHCR

  When the DMA enable bit (DE) in CHCR is cleared, DMA transfers in the affecte are halted. The TE bit is not set when this happens.

Conditions for Both Channels Ending Simultaneously: Transfers on both channels either of the following conditions is met:

The NMIF (NMI flag) bit or AE (address error flag) bit in DMAOR is set to 1.
The DMA master enable (DME) bit is cleared to 0 in DMAOR.

- Transfer end when NMIF = 1 or AE = 1 in DMAOR

  When an NMI interrupt or DMAC address error occurs and the NMIF or AE bit is DMAOR, all channels stop their transfers. The DMA source address register (SAR destination address register (DAR), and transfer count register (TCR) are all updat transfer immediately preceding the halt. When this transfer is the final transfer, TE transfer ends. To resume transfer after NMI interrupt exception handling or addres exception handling, clear the appropriate flag bit. When the DE bit is then set to 1, on that channel will restart. To avoid this, keep its DE bit at 0. In dual address mod transfer will be halted after the completion of the following write cycle even when error occurs in the initial read cycle. SAR, DAR and TCR are updated by the final

- Transfer end when DME = 0 in DMAOR

  Clearing the DME bit in DMAOR forcibly aborts the transfers on both channels at the current bus cycle. When the transfer is the final transfer, TE = 1 and the transfe

Due to these properties of the PCI bus, it is necessary to use Grew logic externally to co
present address and the next address and determine whether burst transfer is possible. H
the size of the external Grew logic increases if address comparisons are required, and th
the possibility that delays may interfere with timing requirements.

The specifications for $\overline{\text{BH}}$ have therefore been updated in order to solve these problems
burst transfer is possible using the present address this information is passed to the exte
logic. This provides enhanced support for PCI bus connections.

Register Settings When Using $\overline{\text{BH}}$ Pin: $\overline{\text{BH}}$ is output from only when the 16-byte tran
is selected using the DMAC built into the SH7615. However, it is not output when SDR
DRAM are accessed. When using the 16-byte transfer mode, specify auto-request mode
external request mode with edge detection. If external request mode with level detectio
chip module request mode is specified, operation is not guaranteed.

To use $\overline{\text{BH}}$, the settings for the CHCR0 register or CHCR1 register in the on-chip DMA
SH7615 must be as shown in figure 11.43. $\overline{\text{BH}}$ is not output unless the settings for the C
register or CHCR1 register are as indicated in figure 11.42.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit name	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Setting	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	DM1	DM0	SM1	SM0	TS1	TS0	AR	AM	AL	DS	DL	TB	TA	IE	TE	DE
Setting	0	1	0	1	1	1	*	*	*	*	*	*	*	*	*	1

16-byte unit (four long words transferred)
Source address is incremented
Destination address is incremented
* Don't care

Figure 11.42 Register Settings When Using $\overline{\text{BH}}$

11.4 Usage Examples

11.4.1 Example of DMA Data Transfer Between On-chip SCIF and External M

In this example data received by the serial communication interface (SCIF) with on-ch
sent to external memory using DMAC channel 1. Table 11.9 lists the transfer conditio
register setting values.

**Table 11.9 Transfer Conditions and Register Setting Values for Data Transfer
On-chip SCIF and External Memory**

Transfer Condition	Register	Setting Value
Transfer source: SCFRDR1 in on-chip SCIF	SAR1	H'FFFFFCCC
Transfer destination: External memory (word space)	DAR1	Transfer destinat
Number of transfers: 64	TCR1	H'0040
Transfer destination address: Increment	CHCR1	H'4045
Transfer source address: Fixed		
Bus mode: Cycle-steal		
Transfer unit: Byte		
DEI interrupt request at end of transfer DE = 1		
Channel priority: Fixed (0 > 1) DME = 1	DMAOR	H'0001
Transfer request source (transfer request signal): SCIF (RXI)	DRCR1	H'05

Note: Make sure the SCIF settings have interrupts enabled and the appropriate CPU
level.

3. When the DMAC is not operating, the NMIF bit in DMAOR is set even when an N interrupt is input.

4. The DMAC cannot access the cache memory.

5. Before changing the frequency or changing to standby mode, set the DME bit of DM and stop operation of the DMAC.

6. Do not use the DMAC, BSC, UBC, E-DMAC, and EtherC for on-chip peripheral m transfers.

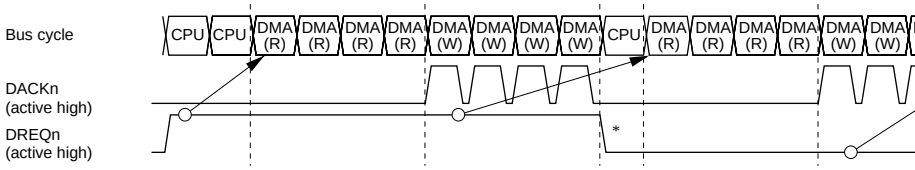
7. Do not access the cache (address array, data array, associative purge area).

8. Note that when level detection of the request signal is used in single address mode, signal may be detected before DACKn is output.

9. When Eϕ exceeds 31.25 MHz, do not use transfer involving DACKn output on ordi for word or longword access with an 8-bit bus width, or longword access with a 16- width.

10. When DMA transfer is performed in response to a DMA transfer request signal fro peripheral module, if clearing of the DMA transfer request signal from the peripher by the DMA transfer is not completed before the next transfer request signal from t subsequent DMA transfers may not be possible.

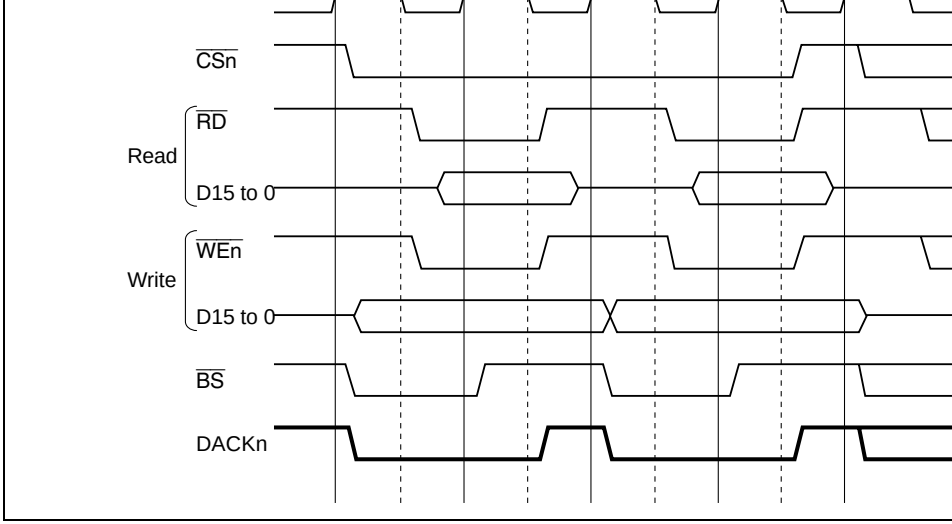
DACK output in read cycle

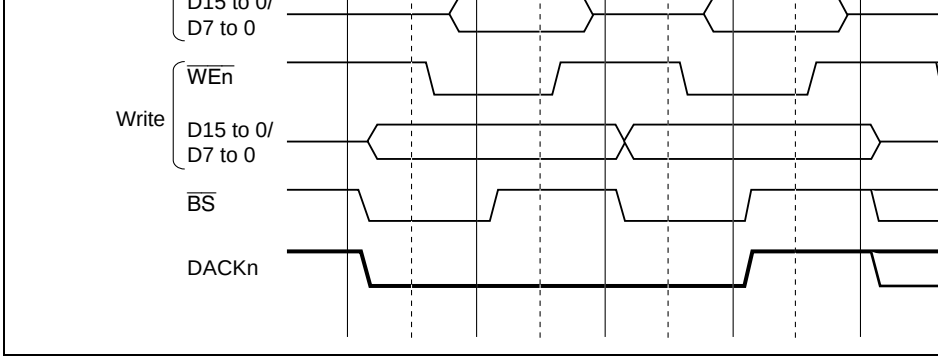


DACK output in write cycle

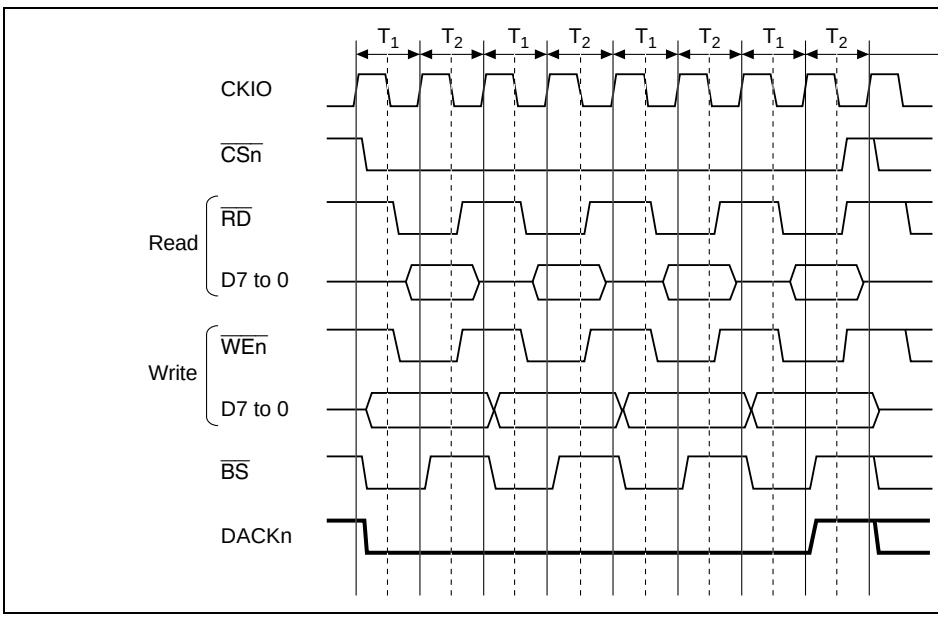
Note: * In addition to CPU cycles, E-DMAC cycles may be inserted in some cases.

b. When external request DREQ edge detection is set, if DREQn is input continu
DMAC continues to operate without insertion of a CPU cycle. (However, a CP
begin if there is no request from DREQn.)


CSn
Read
RD
D15 to 0
Write
WEn
D15 to 0
BS
DACKn



Error timing 2:



timing.

(1) Iϕ:Eϕ = 1:1

(2) DMA transfer to an ordinary space or burst ROM space

(3) 16-byte or longword DMA transfer to a 16-bit width space or 16-byte, longwor
DMA transfer to an 8-bit width space, which generates multiple bus cycles

Countermeasures:

This problem is avoided by any of the following countermeasures.

(1) Specify a clock ratio except t_{Ecyc}:t_{Pcyc} 1:1.

(2) Use 32-bit bus width.

(3) When the bus width is 16 bits, perform word or byte DMA transfer.

(4) When the bus width is 8 bits, perform byte DMA transfer.

13. DMAC does not perform DMA transfer on channel 1 by an on-chip peripheral mod

Phenomenon:

(1) DMAC does not perform DMA transfer on channel 1 by an on-chip peripheral r
request.

When channel 0 of the on-chip DMAC is set to cycle-steal mode and channel 1
chip peripheral module request mode, the DMAC may not perform DMA transf
channel 1.

Conditions:

(1) Conditions for malfunction in DMA transfer on channel 1 by an on-chip periphe
request

When the following conditions are all satisfied, the DMAC does not perform DM
on channel 1 by an on-chip peripheral module request.

(a) DMAC channels 0 and 1 are both enabled.

(b) DMAC channel 0 is set to cycle-steal mode.

(c) DMAC channel 1 is set to cycle-steal mode, dual address mode, and on-chip
module request mode.

(d) Round-robin mode is specified as the DMAC priority mode.

Phenomenon:

(1) DMAC does not perform DMA transfer between on-chip memory and on-chip
module by an external request.

When the on-chip DMAC is set to external request (DREQ) mode and cycle-st
and DMA transfer is attempted between on-chip memory and on-chip peripher
between on-chip peripheral modules, the DMAC may not perform DMA transf
second and later DREQ inputs.

Conditions:

(1) Conditions for malfunction in DMA transfer between on-chip memory and on-
peripheral module by an external request

When the following conditions are all satisfied, the DMAC does not perform D
between on-chip memory and on-chip peripheral module or between on-chip p
modules by an external request.

(a) The external request (DREQ) is selected for the transfer request source.

(b) The DMA transfer between on-chip memory and on-chip peripheral modul
on-chip peripheral modules is selected.

(c) Cycle-steal mode is used.

Countermeasures:

(1) Countermeasure against malfunction in DMA transfer between on-chip memor
chip peripheral module by an external request

This problem is avoided by the following counter measure.

(a) Do not select the external request (DREQ) for the transfer request source.

15. Data bus collision during single-address DMAC transfer

Phenomenon:

(1) Data bus collision during DMA transfer in single address mode

In the system which includes the SH7615, an external device with DACK, and
synchronous DRAM (SDRAM), if single-address DMA transfer is performed f
external device with DACK to SDRAM immediately after the SH7615 writes d
SDRAM, the SH7615 may erroneously drive data bus during the single-addres
transfer , and the erroneously driven data may collide with the DMA transfer d

Countermeasures:

(1) Countermeasure against data bus collision during single-address DMA transfer

This problem is avoided by any of the following countermeasures.

(a) Specify a clock ratio other than external clock (Eϕ):internal clock (Iϕ) = 1:1.

(b) Do not write to SDRAM from CPU, Ethernet controller direct memory acces controller (E-DMAC), or another channel of the DMAC during single-addre transfer from the external device with DACK to SDRAM.

16. DMAC DACK error output

Phenomenon:

(1) DACK error

When DMAC channels 0 and 1 are both set to external request (DREQ0 and DR mode, the DMAC may execute DMA transfer with DACK1 output on channel 1 DREQ1 is not input.

Conditions:

(1) Conditions for DACK error

When the following conditions are all satisfied, the DMAC executes DMA trans DACK1 output on channel 1 while the DREQ1 is not input.

(a) DMAC channels 0 and 1 are both enabled.

(b) DMAC channels 0 and 1 both select the external request (DREQ0 and DREQ transfer request source.

(c) DMAC channels 0 and 1 are both set to cycle-steal mode.

(d) Round-robin mode is specified as the DMAC priority mode.

Countermeasures:

(1) Countermeasure against DACK error

This problem is avoided by any of the following countermeasures.

(a) Set either DMAC channel 0 or 1 to burst mode.

(b) Set the DMAC priority mode to fixed priority mode.

12.1.1 Features

The FRT has the following features:

- Choice of four counter input clocks

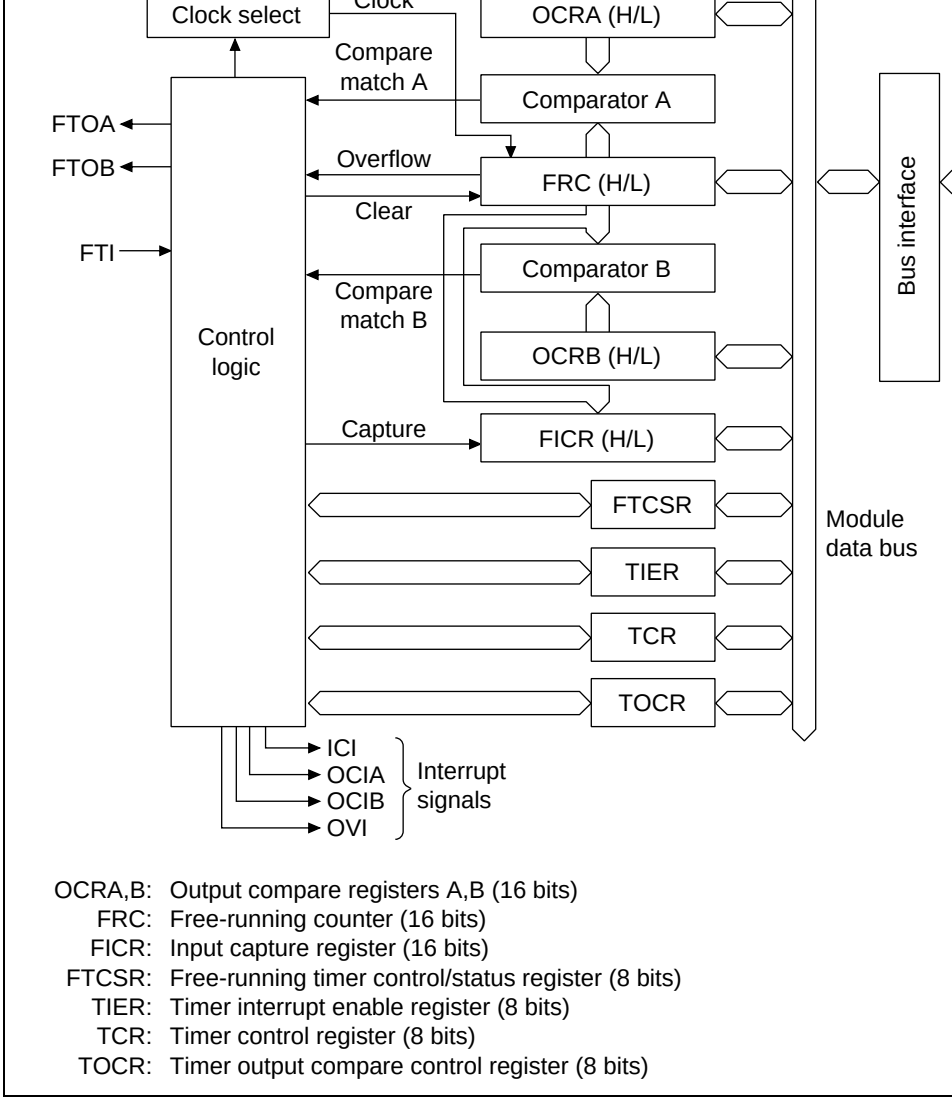
  The counter input clock can be selected from three internal clocks (Pϕ/8, Pϕ/32, Pϕ
  an external clock (enabling external event counting).
- Two independent comparators

  Two waveform outputs can be generated.
- Input capture

  Choice of rising edge or falling edge
- Counter clear specification

  The counter value can be cleared by compare match A.
- Four interrupt sources

  Two compare match sources, one input capture source, and one overflow source ca
  requests independently.



OCRA,B: Output compare registers A,B (16 bits)
FRC: Free-running counter (16 bits)
FICR: Input capture register (16 bits)
FTCSR: Free-running timer control/status register (8 bits)
TIER: Timer interrupt enable register (8 bits)
TCR: Timer control register (8 bits)
TOCR: Timer output compare control register (8 bits)

Figure 12.1 FRT Block Diagram

Output compare B output pin	FTOB	Output	Output pin for output compa
Input capture input pin	FTI	Input	Input pin for input capture

12.1.4 Register Configuration

Table 12.2 shows the FRT register configuration.

Table 12.2 Register Configuration

Register	Abbreviation	R/W	Initial Value	Add
Timer interrupt enable register	TIER	R/W	H'01	HFF
Free-running timer control/status register	FTCSR	R/(W)*1	H'00	HFF
Free-running counter H	FRC H	R/W	H'00	HFF
Free-running counter L	FRC L	R/W	H'00	HFF
Output compare register A H	OCRA H	R/W	H'FF	HFF
Output compare register A L	OCRA L	R/W	H'FF	HFF
Output compare register B H	OCRB H	R/W	H'FF	HFF
Output compare register B L	OCRB L	R/W	H'FF	HFF
Timer control register	TCR	R/W	H'00	HFF
Timer output compare control register	TOCR	R/W	H'E0	HFF
Input capture register H	FICR H	R	H'00	HFF
Input capture register L	FICR L	R	H'00	HFF

Notes: Use byte-size access for all registers.

1. Bits 7 to 1 are read-only. The only value that can be written is a 0, which is clear flags. Bit 0 can be read or written.
2. OCRA and OCRB have the same address. The OCRS bit in TOCR is used between them.

FRC is a 16-bit read/write register. It increments upon input of a clock. The input clock
selected using clock select bits 1 and 0 (CKS1, CKS0) in TCR. FRC can be cleared upo
match A.

When FRC overflows (H'FFFF → H'0000), the overflow flag (OVF) in FTCSR is set to
can be read or written to by the CPU, but because it is 16 bits long, data transfers involv
CPU are performed via a temporary register (TEMP). See section 12.3, CPU Interface,
detailed information.

FRC is initialized to H'0000 by a reset, in standby mode, and when the module standby
used.

12.2.2 Output Compare Registers A and B (OCRA and OCRB)

Bit:	15	14	13	…	3	2	1
				…			
Initial value:	1	1	1	…	1	1	1
R/W:	R/W	R/W	R/W	…	R/W	R/W	R/W

OCR is composed of two 16-bit read/write registers (OCRA and OCRB). The contents
always compared to the FRC value. When the two values are the same, the output comp
in FTCSR (OCFA and OCFB) are set to 1.

When the OCR and FRC values are the same (compare match), the output level values
output level bits (OLVLA and OLVLB) are output to the output compare pins (FTOA a
After a reset, FTOA and FTOB output 0 until the first compare match occurs.

Because OCR is a 16-bit register, data transfers involving the CPU are performed via a
register (TEMP). See section 12.3, CPU Interface, for more detailed information.

OCR is initialized to H'FFFF by a reset, in standby mode, and when the module standb
is used.

(FTI pin) is detected, the current FRC value is transferred to FICR. At the same time, capture flag (ICF) in FTCSR is set to 1. The edge of the input signal can be selected u input edge select bit (IEDG) in TCR.

Because FICR is a 16-bit register, data transfers involving the CPU are performed via register (TEMP). See section 12.3, CPU Interface, for more detailed information. To e the input capture operation is reliably performed, set the pulse width of the input captu signal to six system clocks (Pϕ) or more.

FICR is initialized to H'0000 by a reset, in standby mode, and when the module standl is used.

12.2.4 Timer Interrupt Enable Register (TIER)

Bit:	7	6	5	4	3	2	1
	ICIE	—	—	—	OCIAE	OCIBE	OVI
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W

TIER is an 8-bit read/write register that controls enabling of all interrupt requests. TIE initialized to H'01 by a reset, in standby mode, and when the module standby function

Bit 7—Input Capture Interrupt Enable (ICIE): Selects enabling/disabling of the ICI in request when the input capture flag (ICF) in FTCSR is set to 1.

Bit 7: ICIE	Description
0	Interrupt request (ICI) caused by ICF disabled (
1	Interrupt request (ICI) caused by ICF enabled

Bits 6 to 4—Reserved: These bits are always read as 0. The write value should always

interrupt request when the output compare flag B (OCFB) in FTCSR is set to 1.

Bit 2: OCIBE	Description
0	Interrupt request (OCIB) caused by OCFB disabled (Ir
1	Interrupt request (OCIB) caused by OCFB enabled

Bit 1—Timer Overflow Interrupt Enable (OVIE): Selects enabling/disabling of the OV
request when the overflow flag (OVF) in FTCSR is set to 1.

Bit 1: OVIE	Description
0	Interrupt request (OVI) caused by OVF disabled (ir
1	Interrupt request (OVI) caused by OVF enabled

Bit 0—Reserved: This bit is always read as 1. The write value should always be 1.

12.2.5 Free-Running Timer Control/Status Register (FTCSR)

Bit:	7	6	5	4	3	2	1
	ICF	—	—	—	OCFA	OCFB	OVF
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R/(W)*	R/(W)*

Note: * For bits 7, and 3 to 1, the only value that can be written is 0 (to clear the flag

FTCSR is an 8-bit register that selects counter clearing and controls interrupt request si
FTCSR is initialized to H'00 by a reset, in standby mode, and when the module standby
used. See section 12.4, Operation, for the timing.

Bit 7—Input Capture Flag (ICF): Status flag that indicates that the FRC value has been
FICR by the input capture signal. This flag is cleared by software and set by hardware.
be set by software.

Bit 3—Output Compare Flag A (OCFA): Status flag that indicates when the values of
and OCRA match. This flag is cleared by software and set by hardware. It cannot be s
software.

Bit 3: OCFA	Description
0	[Clearing condition] When OCFA is read while set to 1, and then 0 is written to (
1	[Setting condition] When the FRC value becomes equal to OCRA

Bit 2—Output Compare Flag B (OCFB): Status flag that indicates when the values of
OCRB match. This flag is cleared by software and set by hardware. It cannot be set by

Bit 2: OCFB	Description
0	[Clearing condition] When OCFB is read while set to 1, and then 0 is written to (
1	[Setting condition] When the FRC value becomes equal to OCRB

Bit 1—Timer Overflow Flag (OVF): Status flag that indicates when FRC overflows (f
to H'0000). This flag is cleared by software and set by hardware. It cannot be set by so

Bit 1: OVF	Description
0	[Clearing condition] When OVF is read while set to 1, and then 0 is written to it (
1	[Setting condition] When the FRC value changes from H'FFFF to H'0000

Bit:	7	6	5	4	3	2	1
	IEDG	—	—	—	—	—	CKS1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W

TCR is an 8-bit read/write register that selects the input edge for input capture and sele
input clock for FRC. TCR is initialized to H'00 by a reset, in standby mode, and when t
standby function is used.

Bit 7—Input Edge Select (IEDG): Selects whether to capture the input capture input (F
falling edge or rising edge.

Bit 7: IEDG	Description
0	Input captured on falling edge (In
1	Input captured on rising edge

Bits 6 to 2—Reserved: These bits are always read as 0. The write value should always b

Bits 1 and 0—Clock Select (CKS1, CKS0): These bits select whether to use an externa
one of three internal clocks for input to FRC. The external clock is counted at the rising

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	Internal clock: count at $\phi/8$ (In
	1	Internal clock: count at $\phi/32$
1	0	Internal clock: count at $\phi/128$
	1	External clock: count at rising edge

switching between access of output compare registers A and B. TOCR is initialized to
reset, in standby mode, and when the module standby function is used.

Bits 7 to 5—Reserved: These bits are always read as 1. The write value should always

Bit 4—Output Compare Register Select (OCRS): OCRA and OCRB share the same a
OCRS bit controls which register is selected when reading/writing to this address. It d
affect the operation of OCRA and OCRB.

Bit 4: OCRS	Description	
0	OCRA register selected	(
1	OCRB register selected	

Bits 3 and 2—Reserved: These bits are always read as 0. The write value should alway

Bit 1—Output Level A (OLVLA): Selects the level output to the output compare A ou
upon compare match A (signal indicating match of FRC and OCRA).

Bit 1: OLVLA	Description	
0	0 output on compare match A	(
1	1 output on compare match A	

Bit 0—Output Level B (OLVLB): Selects the level output to the output compare B ou
upon compare match B (signal indicating match of FRC and OCRB).

Bit 0: OLVLB	Description	
0	0 output on compare match B	(
1	1 output on compare match B	

The upper byte is written, which results in the upper byte of data being stored in TE
lower byte is then written, which results in 16 bits of data being written to the regist
combined with the upper byte value in TEMP.

- Reading from 16-bit Registers

  The upper byte of data is read, which results in the upper byte value being transferre
  CPU. The lower byte value is transferred to TEMP. The lower byte is then read, wh
  in the lower byte value in TEMP being sent to the CPU.

When registers of these three types are accessed, two byte accesses should always be pe
first to the upper byte, then the lower byte. If only the upper byte or lower byte is acces
data will not be transferred properly.

Figure 12.2 and 12.3 show the flow of data when FRC is accessed. Other registers func
same way. When reading OCRA and OCRB, however, both upper and lower-byte data
transferred directly to the CPU without passing through TEMP.

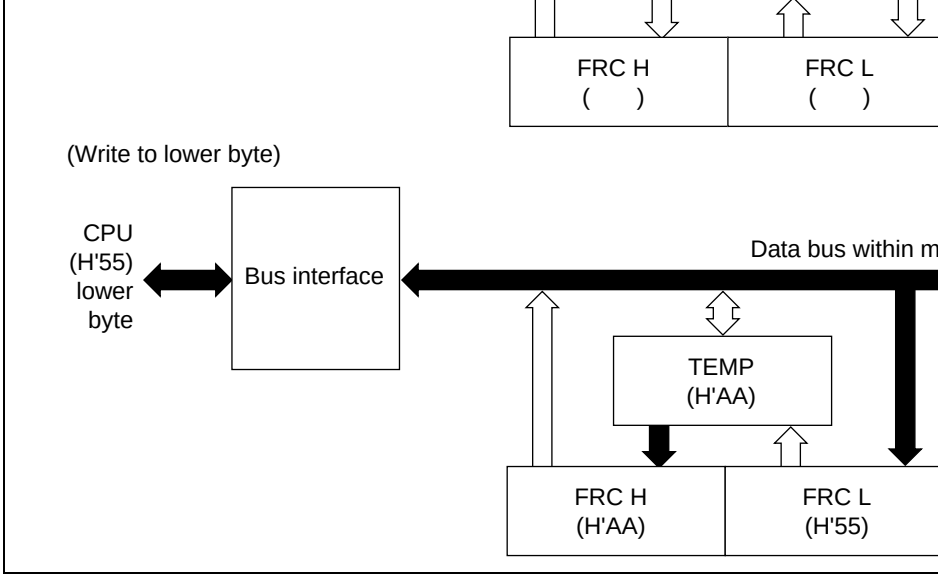


Figure 12.2 FRC Access Operation (CPU Writes H'AA55 to FRC)

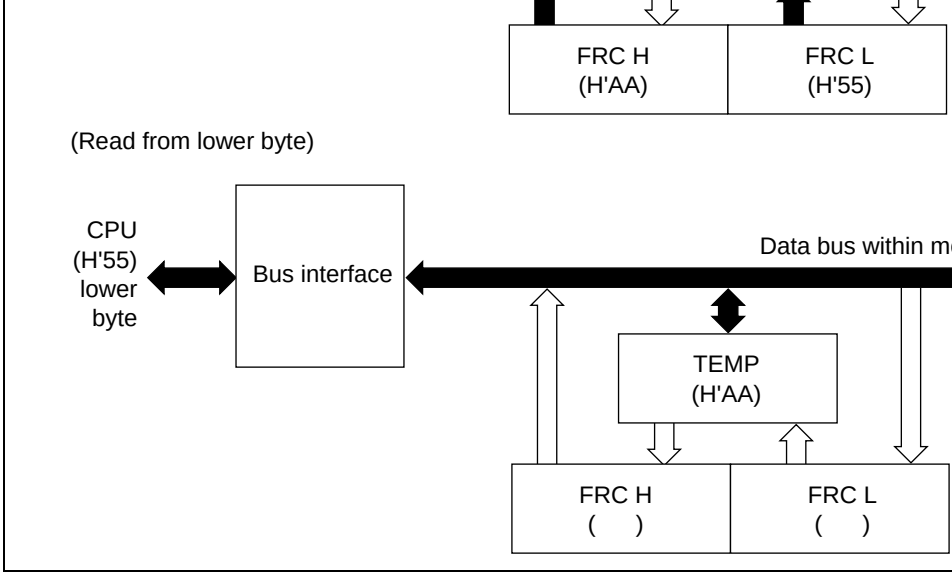


Figure 12.3 FRC Access Operation (CPU Reads H'AA55 from FRC)

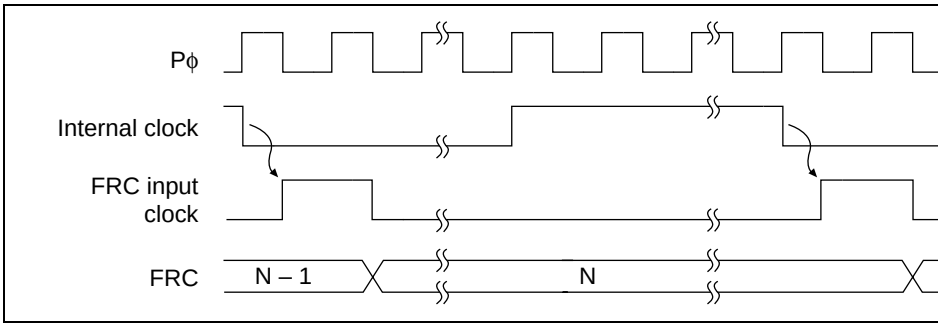


Figure 12.4 Count Timing (Internal Clock Operation)

External Clock Operation: Set the CKS1 and CKS0 bits in TCR to select the externa
External clock pulses are counted on the rising edge. The pulse width of the external c
at least 6 system clocks (ϕ). A smaller pulse width will result in inaccurate operation.
shows the timing.

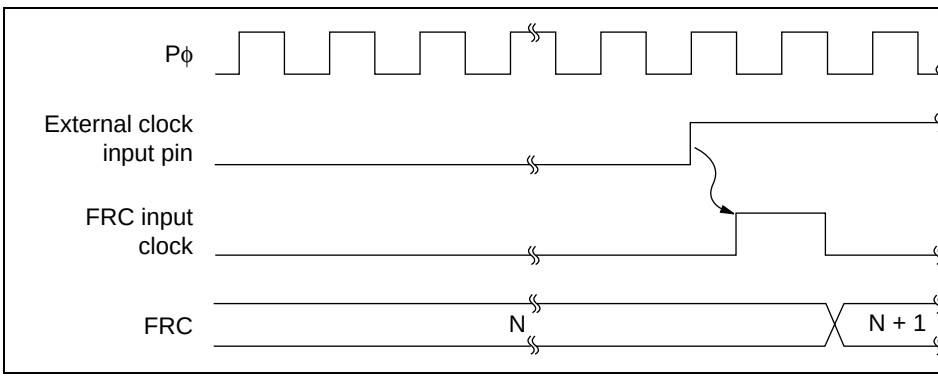


Figure 12.5 Count Timing (External Clock Operation)

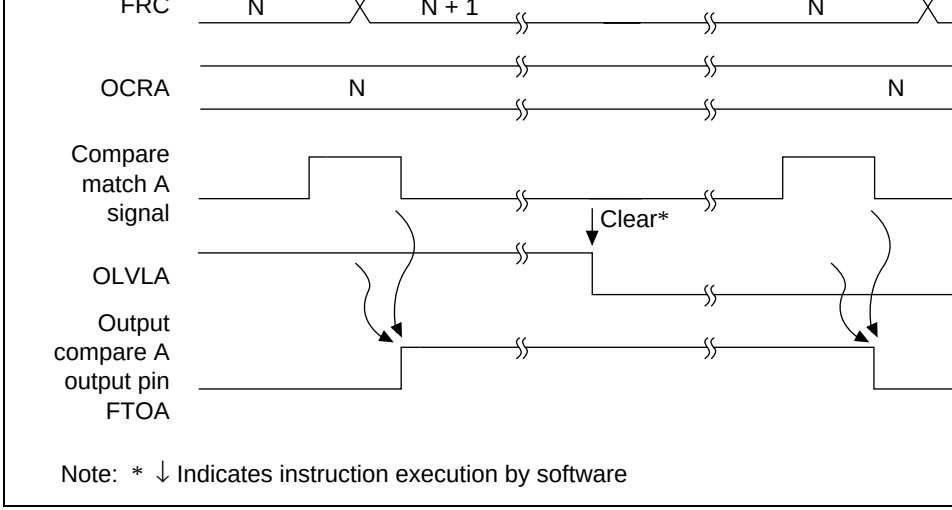


Figure 12.6 Output Timing for Output Compare A

12.4.3 FRC Clear Timing

FRC can be cleared on compare match A. Figure 12.7 shows the timing.

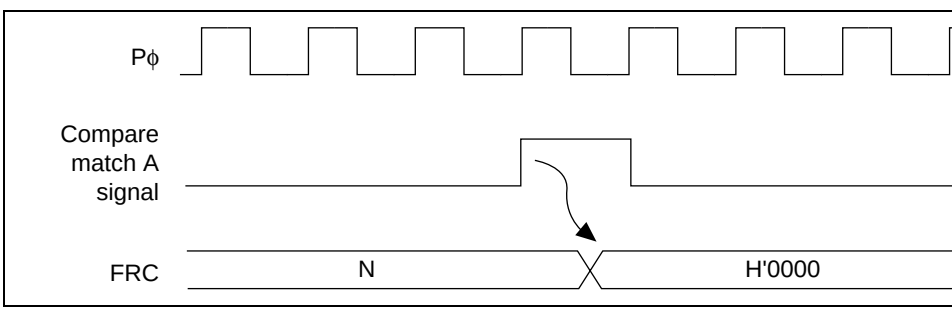


Figure 12.7 Compare Match A Clear Timing

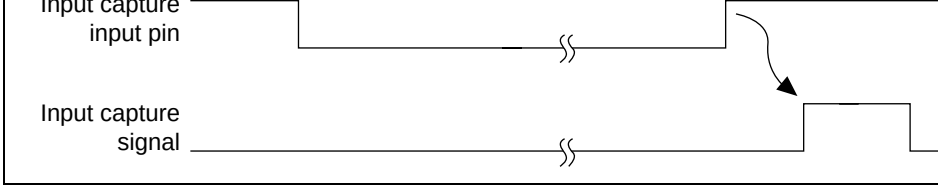


Figure 12.8 Input Capture Signal Timing (Normal)

When the input capture signal is input when FICR is read (upper-byte read), the input signal is delayed by one cycle of Pϕ. Figure 12.9 shows the timing.

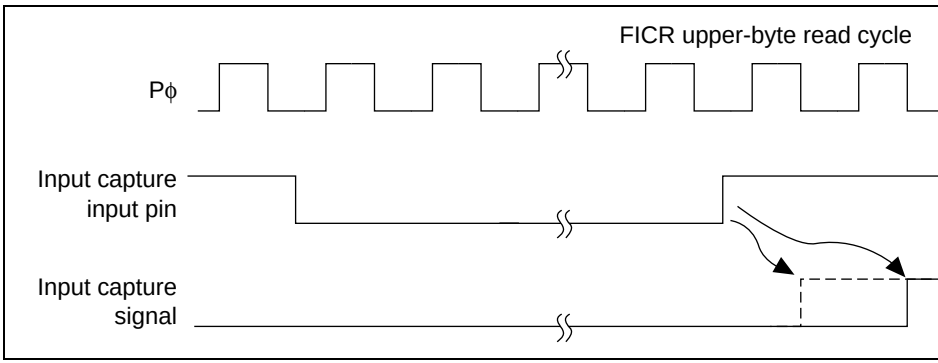


Figure 12.9 Input Capture Signal Timing (Input Capture Input when FICR

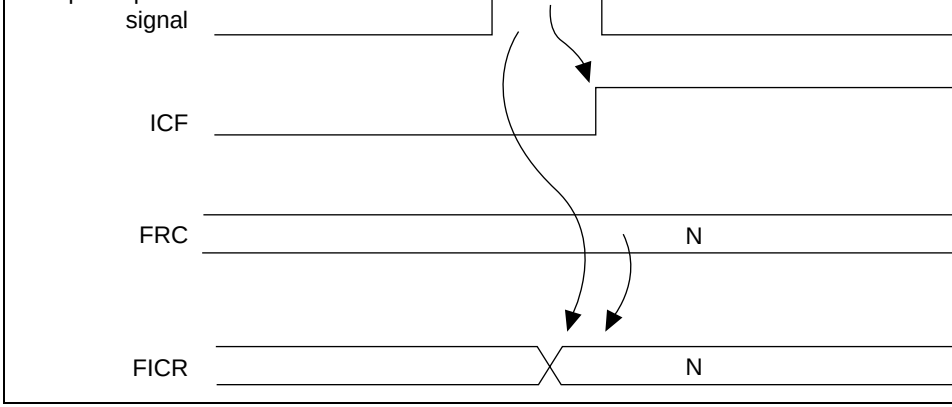


Figure 12.10 ICF Setting Timing

12.4.6 Output Compare Flag (OCFA, OCFB) Setting Timing

The compare match signal output (when OCRA or OCRB matches the FRC value) sets
compare flag OCFA or OCFB to 1. The compare match signal is generated in the last s
which the values matched (at the timing for updating the count value that matched the I
OCRA or OCRB matches the FRC, no compare match is generated until the next incre
occurs. Figure 12.11 shows the timing for setting OCFA and OCFB.

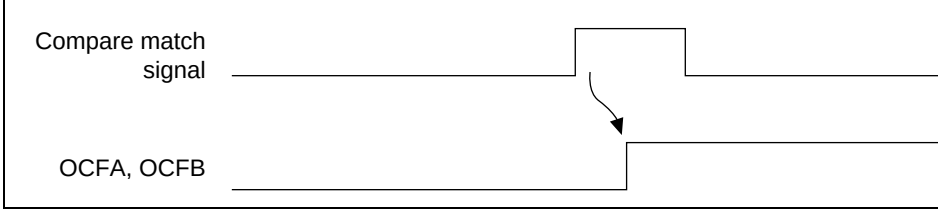


Figure 12.11 OCF Setting Timing

12.4.7 Timer Overflow Flag (OVF) Setting Timing

FRC overflow (from H'FFFF to H'0000) sets the timer overflow flag (OVF) to 1. Figu
shows the timing.

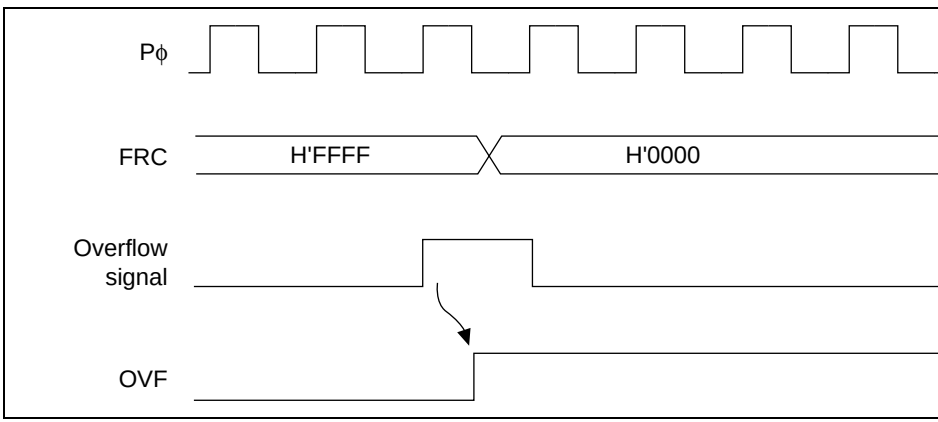


Figure 12.12 OVF Setting Timing

Interrupt Source	Description	Priority
ICI	Interrupt by ICF	High
OCIA, OCIB	Interrupt by OCFA or OCFB	↕
OVI	Interrupt by OVF	Low

12.6 Example of FRT Use

Figure 12.13 shows an example in which pulses with a 50% duty factor and arbitrary pl
relationship are output. The procedure is as follows:

1. Set the CCLRA bit in FTCSR to 1.
2. The OLVLA and OLVLB bits are inverted by software whenever a compare match

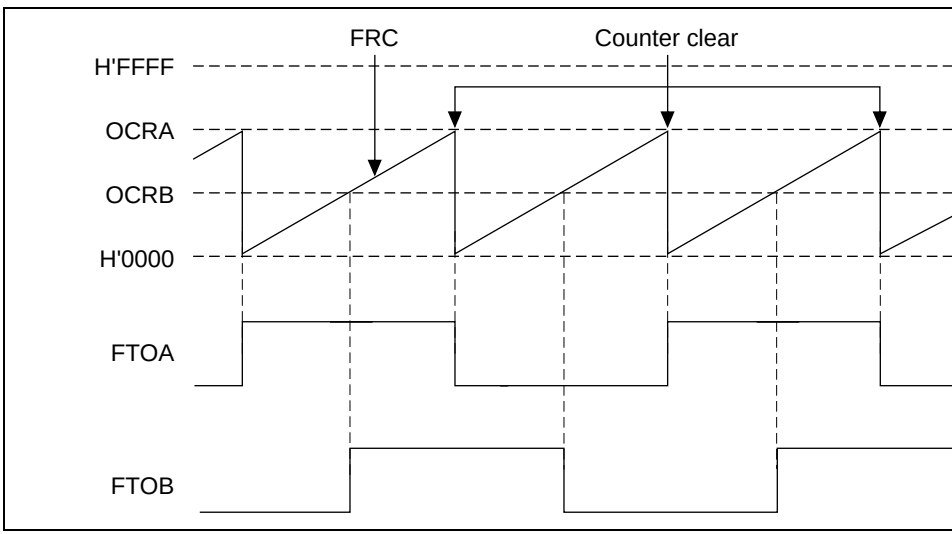


Figure 12.13 Example of Pulse Output

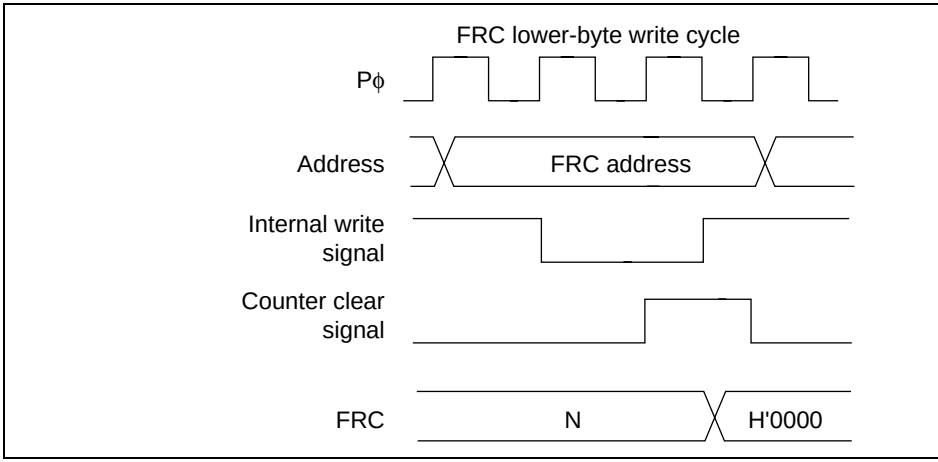


Figure 12.14 Contention between FRC Write and Clear

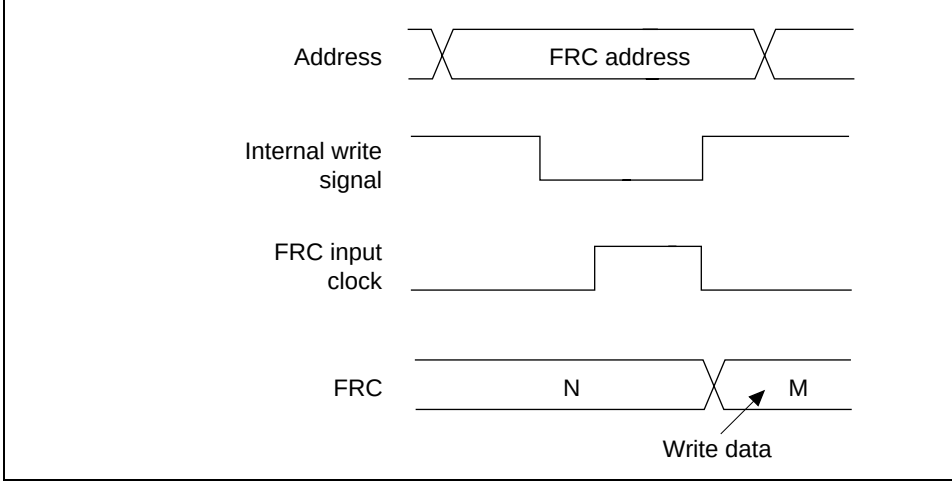


Figure 12.15 Contention between FRC Write and Increment

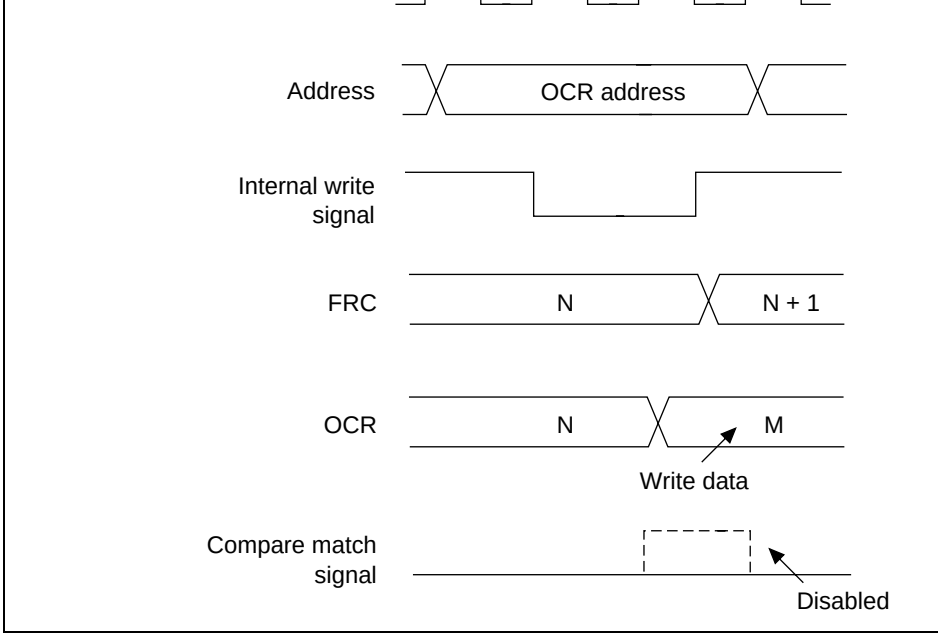
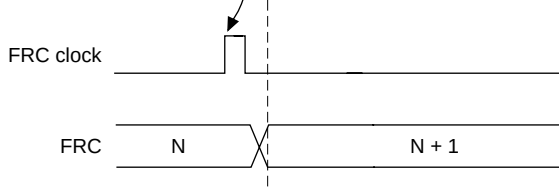


Figure 12.16 Contention between OCR and Compare Match

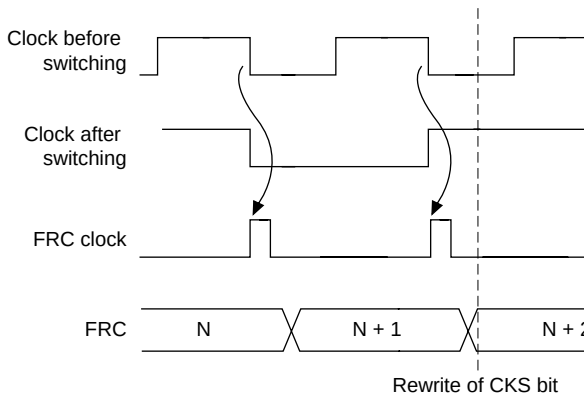
12.7.4 Internal Clock Switching and Counter Operation

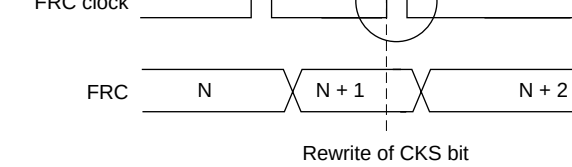
FRC will sometimes begin incrementing because of the timing of switching between i
clocks. Table 12.4 shows the relationship between internal clock switching timing (CK
CKS0 bit rewrites) and FRC operation.

When an internal clock is used, the FRC clock is generated when the falling edge of a
clock (created by dividing the system clock (ϕ)) is detected. When a clock is switched
before the switching and to low after switching, as shown in case 3 in table 12.4, the s
considered a falling edge and an FRC clock pulse is generated, causing FRC to increm
may also increment when switching between an internal clock and an external clock.

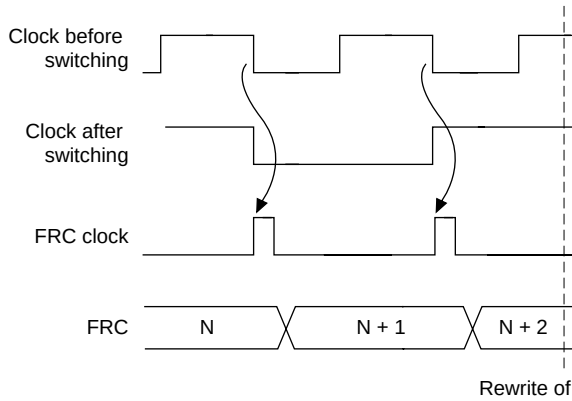


2 Low-to-high switch





4 High-to-high switch



Note: Because the switchover is considered a falling edge, FRC starts counting up.

12.7.5 Timer Output (FTOA, FTOB)

During a power-on reset, the timer outputs (FTOA, FTOB) will be unreliable until the
stabilizes. The initial value is output after the oscillation settling time has elapsed.

When this watchdog function is not needed, the WDT can be used as an interval timer
interval timer operation, an interval timer interrupt is generated at each counter overfl
WDT is also used when recovering from standby mode, in modifying a clock frequen
clock pause mode.

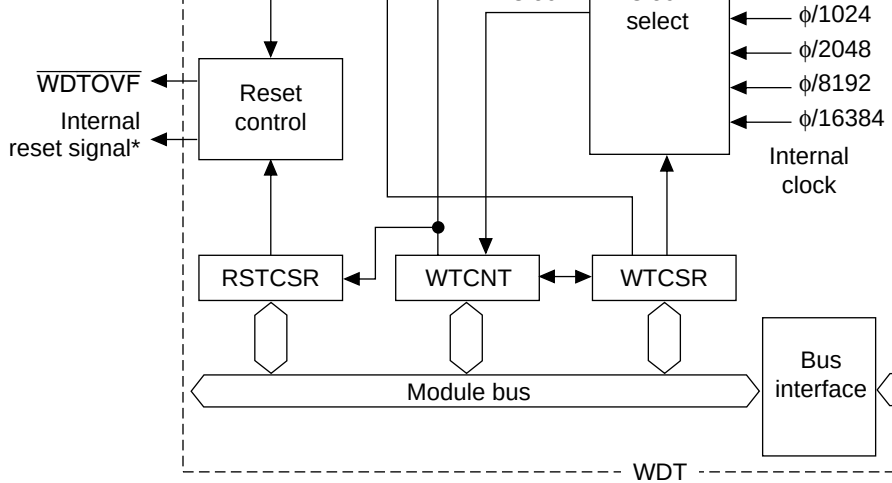
13.1.1 Features

The WDT includes the following features.

- Can be switched between watchdog timer mode and interval timer mode.
- $\overline{\text{WDTOVF}}$ output in watchdog timer mode

  The $\overline{\text{WDTOVF}}$ signal is output externally when the counter overflows, and a simu
  internal reset of the chip can also be selected (either a power-on reset or manual re
  specified).
- Interrupt generation in interval timer mode

  An interval timer interrupt is generated when the counter overflows.
- Used when standby mode is cleared or the clock frequency is changed, and in cloc
  mode.
- Choice of eight counter input clocks



φ: See figure 3.1, Block Diagram of Clock Pulse Generator Circuit.

WTCSR: Watchdog timer control/status register
WTCNT: Watchdog timer counter
RSTCSR: Reset control/status register

Note: The internal reset signal can be generated by a register setting. The type of res
selected (power-on or manual reset).

Figure 13.1 WDT Block Diagram

13.1.3 Input/Output Pin

Table 13.1 shows the pin configuration.

Table 13.1 Pin Configuration

Pin	Abbreviation	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs the counter overflow si watchdog timer mode

Watchdog timer control/status register	WTCSR	R/(W)*2	H'18	H'FFFFFE80	H'F
Watchdog timer counter	WTCNT	R/W	H'00	H'FFFFFE80	H'F
Reset control/status register	RSTCSR	R/(W)*3	H'1E	H'FFFFFE82	H'F

Notes: 1. Write by word access. It cannot be written by byte or longword access.
2. Read by byte access. The correct value cannot be read by word or longwor
3. Only 0 can be written in bit 7 to clear the flag.

13.2 Register Descriptions

13.2.1 Watchdog Timer Counter (WTCNT)

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WTCNT is an 8-bit read/write register. The method of writing to WTCNT differs from most other registers to prevent inadvertent rewriting. See section 13.2.4, Notes on Reg for details. When the timer enable bit (TME) in the watchdog timer control/status regi (WTCSR) is set to 1, the watchdog timer counter starts counting pulses of an internal selected by clock select bits 2 to 0 (CKS2 to CKS0) in WTCSR. When the value of W overflows (changes from H'FF to H'00), a watchdog timer overflow signal ($\overline{\text{WDTOVF}}$ timer interrupt (ITI) is generated, depending on the mode selected in the WT/$\overline{\text{IT}}$ bit in WTCNT is initialized to H'00 by a reset and when the TME bit is cleared to 0. It is no in standby mode, when the clock frequency is changed, or in clock pause mode.

The watchdog timer control/status register (WTCSR) is an 8-bit read/write register. Its
include selecting the timer mode and clock source. Bits 7 to 5 are initialized to 000 by a
standby mode, when the clock frequency is changed, and in clock pause mode. Bits 2 to
initialized to 000 by a reset, but are not initialized in standby mode, when the clock freq
changed, or in clock pause mode.

Bit 7—Overflow Flag (OVF): Indicates that WTCNT has overflowed from H'FF to H'0
interval timer mode. It is not set in watchdog timer mode.

Bit 7: OVF	**Description**
0	No overflow of WTCNT in interval timer mode (I Cleared by reading OVF, then writing 0 in OVF
1	WTCNT overflow in interval timer mode

Bit 6—Timer Mode Select (WT/$\overline{\text{IT}}$): Selects whether to use the WDT as a watchdog tim
interval timer. When WTCNT overflows, the WDT either generates an interval timer in
(ITI) or generates a $\overline{\text{WDTOVF}}$ signal, depending on the mode selected.

Bit 6: WT/$\overline{\text{IT}}$	**Description**
0	Interval timer mode: interval timer interrupt (ITI) request to th when WTCNT overflows (I
1	Watchdog timer mode: $\overline{\text{WDTOVF}}$ signal output externally wh overflows. Section 13.2.3, Reset Control/Status Register (RS describes in detail what happens when WTCNT overflows in timer mode

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight intern sources for input to WTCNT. The clock signals are obtained by dividing the frequency system clock (ϕ).

			Description	
Bit 2: CKS2	**Bit 1: CKS1**	**Bit 0: CKS0**	**Clock Source**	**Overflow Interval* (**
0	0	0	ϕ/4 (Initial value)	17.0 µs
		1	ϕ/128	544 µs
	1	0	ϕ/256	1.1 ms
		1	ϕ/512	2.2 ms
1	0	0	ϕ/1024	4.4 ms
		1	ϕ/2048	8.7 ms
	1	0	ϕ/8192	34.8 ms
		1	ϕ/16384	69.6 ms

Note: * The overflow interval listed is the time from when the WTCNT begins count until an overflow occurs.

13.2.3 Reset Control/Status Register (RSTCSR)

Bit:	7	6	5	4	3	2	1
	WOVF	RSTE	RSTS	—	—	—	—
Initial value:	0	0	0	1	1	1	1
R/W:	R/(W)*	R/W	R/W	R	R	R	R

Note: * Only 0 can be written in bit 7, to clear the flag.

RSTCSR is an 8-bit read/write register that controls output of the reset signal generate watchdog timer counter (WTCNT) overflow and selects the internal reset signal type. of writing to RSTCSR differs from that of most other registers to prevent inadvertent r See section 13.2.4, Notes on Register Access, for details. RSTCR is initialized to H'1E

1	Set by WTCNT overflow in watchdog timer mode

Bit 6—Reset Enable (RSTE): Selects whether to reset the chip internally if WTCNT ov
watchdog timer mode.

Bit 6: RSTE	Description
0	Not reset when WTCNT overflows (In LSI not reset internally, but WTCNT and WTCSR reset withir
1	Reset when WTCNT overflows

Bit 5—Reset Select (RSTS): Selects the type of internal reset generated if WTCNT ove
watchdog timer mode.

Bit 5: RSTS	Description
0	Power-on reset (In
1	Manual reset

Bits 4 to 1—Reserved: These bits are always read as 1. The write value should always b

Bit 0— Reserved: This bit is always read as 0. The write value should always be 0.

13.2.4 Notes on Register Access

The watchdog timer's WTCNT, WTCSR, and RSTCSR registers differ from other regi
they are more difficult to write. The procedures for writing and reading these registers a
below.

Writing to WTCNT and WTCSR: These registers must be written by a word transfer
instruction. They cannot be written by byte or longword transfer instructions. WTCNT
WTCSR both have the same write address. The write data must be contained in the low
the written word. The upper byte must be H'5A (for WTCNT) or H'A5 (for WTCSR) (f
This transfers the write data from the lower byte to WTCNT or WTCSR.

Figure 13.2 Writing to WTCNT and WTCSR

Writing to RSTCSR: RSTCSR must be written by a word access to address H'FFFF
cannot be written by byte or longword transfer instructions. Procedures for writing 0 i
(bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figu
write 0 in the WOVF bit, the write data must be H'A5 in the upper byte and H'00 in th
This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write to
and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data
of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respective
WOVF bit is not affected.

Writing 0 to the WOVF bit

		15	8 7
Address:	H'FFFFFE82	H'A5	H'00

Writing to the RSTE and RSTS bits

		15	8 7
Address:	H'FFFFFE82	H'5A	Write data

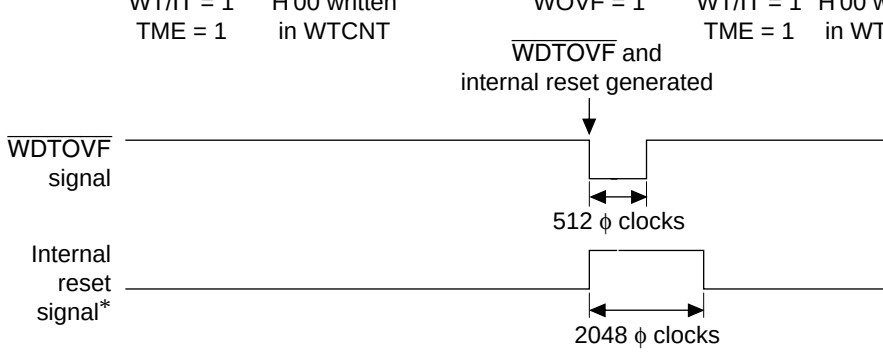
Figure 13.3 Writing to RSTCSR

Reading from WTCNT, WTCSR, and RSTCSR: WTCNT, WTCSR, and RSTCSR
like other registers. Use byte transfer instructions. The read addresses are H'FFFFFE8
WTCSR, H'FFFFFE81 for WTCNT, and H'FFFFFE83 for RSTCSR.

$\overline{\text{WDTOVF}}$ signal is output for 512 ϕ clock cycles.

If the RSTE bit in RSTCSR is set to 1, a signal to reset the chip will be generated intern
simultaneously with the $\overline{\text{WDTOVF}}$ signal when WTCNT overflows. Either a power-on
manual reset can be selected by the RSTS bit. The internal reset signal is output for 204
cycles.

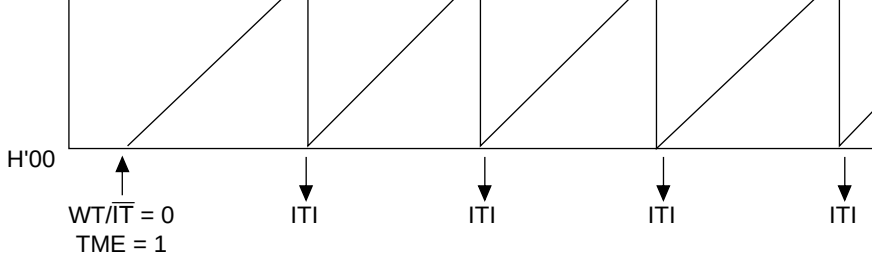
If a reset due to the input signal from the $\overline{\text{RES}}$ pin and a reset due to WDT overflow oc
simultaneously, the $\overline{\text{RES}}$ reset takes priority and the WOVF bit in RSTCSR is cleared t



WT/$\overline{\text{IT}}$: Timer mode select bit
TME: Timer enable bit

Note: * Internal reset signal is generated only when the RSTE bit is set to 1.

Figure 13.4 Operation in Watchdog Timer Mode



ITI: Interval timer interrupt request generation

Figure 13.5 Operation in Interval Timer Mode

13.3.3 Operation when Standby Mode is Cleared

The watchdog timer has a special function to clear standby mode with an NMI interrup
using standby mode, set the WDT as described below.

Transition to Standby Mode: The TME bit in WTCSR must be cleared to 0 to stop th
timer counter before it enters standby mode. The chip cannot enter standby mode while
bit is set to 1. Set bits CKS2 to CKS0 in WTCSR so that the counter overflow interval i
or longer than the oscillation settling time. See section 21, Electrical Characteristics, fo
oscillation settling time.

Recovery from Standby Mode: When an NMI request signal is received in standby m
clock oscillator starts running and the watchdog timer starts counting at the rate selecte
CKS2 to CKS0 before standby mode was entered. When WTCNT overflows (changes
to H'00) the system clock (ϕ) is presumed to be stable and usable; clock signals are sup
entire chip and standby mode ends.

For details on standby mode, see section 20, Power Down Modes.

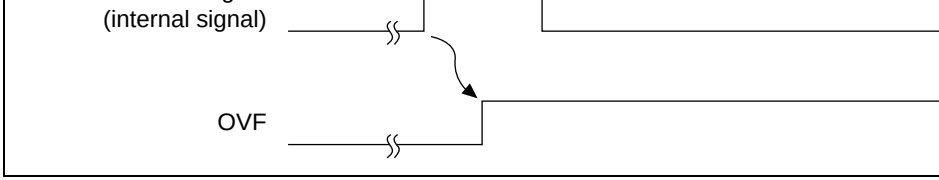


Figure 13.6 Timing of OVF Setting

13.3.5 Timing of Watchdog Timer Overflow Flag (WOVF) Setting

When WTCNT overflows the WOVF flag in RSTCSR is set to 1 and a $\overline{\text{WDTOVF}}$ sig
When the RSTE bit is set to 1, WTCNT overflow enables an internal reset signal to be
for the entire chip (figure 13.7).

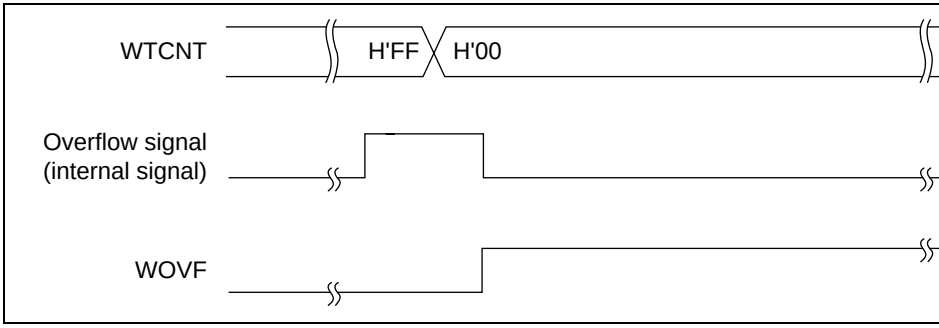


Figure 13.7 Timing of WOVF Setting

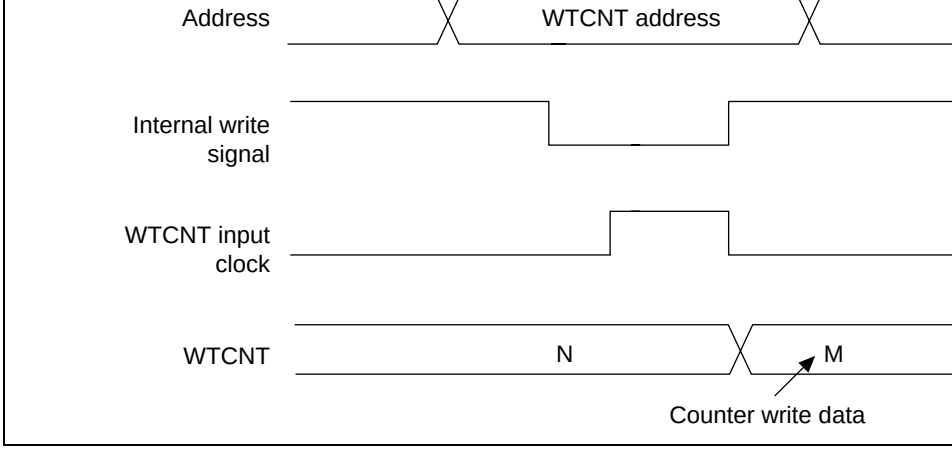


Figure 13.8 Contention between WTCNT Write and Increment

13.4.2 Changing CKS2 to CKS0 Bit Values

If the values of bits CKS2 to CKS0 are altered while the WDT is running, the count ma
increment incorrectly. Always stop the watchdog timer (by clearing the TME bit to 0) b
changing the values of bits CKS2 to CKS0.

13.4.3 Switching between Watchdog Timer Mode and Interval Timer Mode

The WDT may not operate correctly if it is switched between watchdog timer mode and
timer mode while it is running.

To ensure correct operation, always stop the watchdog timer (by clearing the TME bit t
switching between watchdog timer mode and interval timer mode.

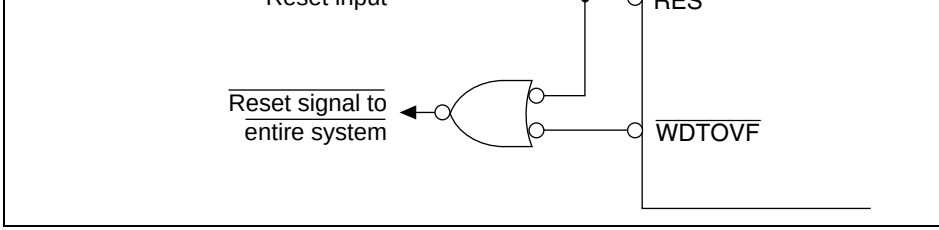


Figure 13.9 Example of Circuit for System Reset with $\overline{\text{WDTOVF}}$ Sign

13.4.5 Internal Reset in Watchdog Timer Mode

If the RSTE bit is cleared to 0 in watchdog timer mode, the chip will not reset internal
WTCNT overflow occurs, but WTCNT and WTCSR in the WDT will reset.

13.4.6 Internal Reset by Watchdog Timer (WDT) in Sleep Mode

When the watchdog time counter (WTCNT) overflows in watchdog timer mode, the S
resets (power-on reset or manual reset) the chip internally. However, if WTCNT overf
sleep mode, internal reset is not executed properly and exception handling by the reset
start.

Conditions:

- In sleep mode
- WDT.WTCSR.WT/$\overline{\text{IT}}$ bit = 1 (watchdog timer mode)
- WDT.RSTCSR.RSTE bit = 1 (internal reset enabled)
- WTCNT overflows

Countermeasures: This problem can be avoided by the following countermeasures.

- When sleep mode is not used, use this internal reset function in watchdog timer m
- When sleep mode is used, reset by an external $\overline{\text{RES}}$ signal instead of the internal re
  Note that the $\overline{\text{WDTOVF}}$ output signal must not be logically input to the $\overline{\text{RES}}$ pin o

(multiprocessor communication function).

An on-chip Infrared Data Association (IrDA) interface based on the IrDA 1.0 system i
provided, enabling infrared communication.

Sixteen-stage FIFO registers are provided for both transmission and reception, enablin
efficient, and continuous communication.

14.1.1 Features

The SCIF has the following features:

- Choice of synchronous or asynchronous serial communication mode
  - — Asynchronous mode

    Serial data communication is executed using an asynchronous system in which
    synchronization is achieved character by character. Serial data communication
    carried out with standard asynchronous communication chips such as a Univers
    Asynchronous Receiver/Transmitter (UART) or Asynchronous Communicatio
    Adapter (ACIA). A multiprocessor communication function is also provided th
    serial data communication with a number of processors.

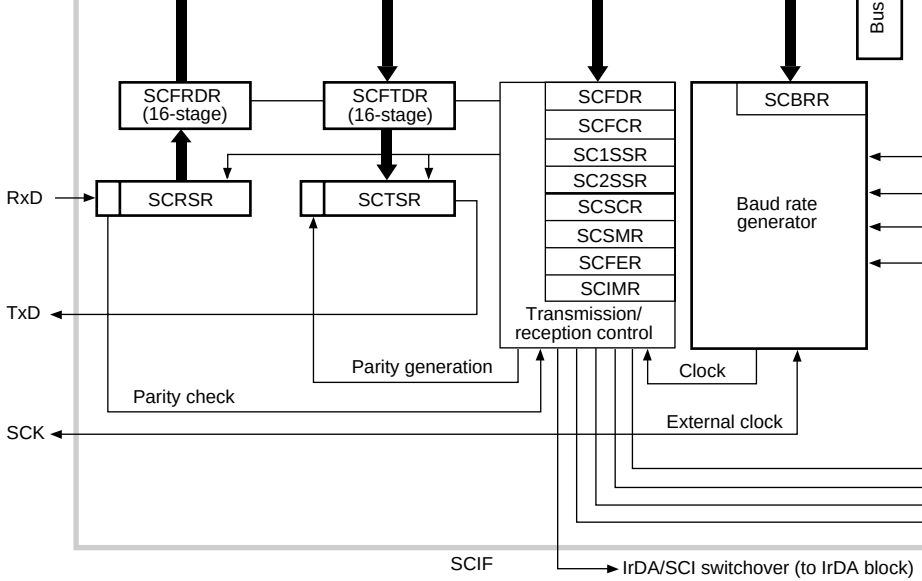
    There is a choice of 12 serial data communication formats.
    - Data length: 7 or 8 bits
    - Stop bit length: 1 or 2 bits
    - Parity: Even/odd/none
    - Multiprocessor bit: 1 or 0
    - Receive error detection: Parity, overrun, and framing errors
    - Automatic break detection

The transmitter and receiver are mutually independent, enabling transmission and re
be executed simultaneously. In addition, the transmitter and receiver both have a 16
FIFO buffer structure, enabling continuous serial data transmission and reception.

(However, IrDA communication is carried out in half-duplex mode.)

- Built-in baud rate generator allows a choice of bit rates.
- Choice of transmit/receive clock source: internal clock from baud rate generator or
  clock from SCK pin
- Four interrupt sources

  There are four interrupt sources—transmit-FIFO-data-empty, break, receive-FIFO-d
  and receive-error—that can issue requests independently. The transmit-FIFO-data-e
  receive-FIFO-data-full interrupts can activate the on-chip DMAC to execute data tra
- When not in use, the SCIF can be stopped by halting its clock supply to reduce pow
  consumption.
- Choice of LSB-first or MSB-first mode
- In asynchronous mode, operation can be selected on a base clock of 4, 8, or 16 time
  rate.
- Built-in modem control functions ($\overline{\text{RTS}}$ and $\overline{\text{CTS}}$)



SCRSR:	Receive shift register	SC1SSR:	Serial status 1 register
SCFRDR:	Receive FIFO data register	SC2SSR:	Serial status 2 register
SCTSR:	Transmit shift register	SCBRR:	Bit rate register
SCFTDR:	Transmit FIFO data register	SCFCR:	FIFO control register
SCSMR:	Serial mode register	SCFDR:	FIFO data count register
SCSCR:	Serial control register	SCFER:	FIFO error register
		SCIMR:	IrDA mode register

Figure 14.1 Block Diagram of SCIF

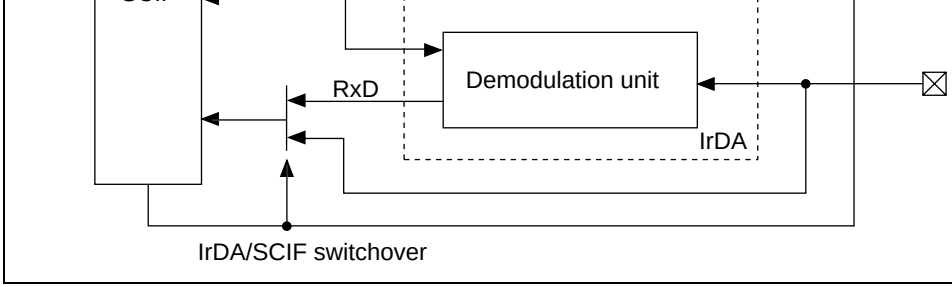


Figure 14.2 Diagram of IrDA Block

14.1.3 Input/Output Pins

The SCIF has the serial pins shown in table 14.1.

Table 14.1 Pin Configuration

Channel	Name	Abbreviation	I/O	Function
1	Serial clock pin	SCK1	Input/ output	Clock input/output
	Receive data pin	RxD1	Input	Receive data input
	Transmit data pin	TxD1	Output	Transmit data outp
	Transmit request pin	$\overline{\text{RTS}}$	Output	Transmit request
	Transmit enable pin	$\overline{\text{CTS}}$	Input	Transmit enable
2	Serial clock pin	SCK2	Input/ output	Clock input/output
	Receive data pin	RxD2	Input	Receive data input
	Transmit data pin	TxD2	Output	Transmit data outp

1	Serial mode register	SCSMR1	R/W	H'00	H'FFFFFC
	Bit rate register	SCBRR1	R/W	H'FF	H'FFFFFC
	Serial control register	SCSCR1	R/W	H'00	H'FFFFFC
	Transmit FIFO data register	SCFTDR1	W	—	H'FFFFFC
	Serial status 1 register	SC1SSR1	R/(W)*	H'0060	H'FFFFFC
	Serial status 2 register	SC2SSR1	R/(W)*	H'20	H'FFFFFC
	Receive FIFO data register	SCFRDR1	R	Undefined	H'FFFFFC
	FIFO control register	SCFCR1	R/W	H'00	H'FFFFFC
	FIFO data count register	SCFDR1	R	H'0000	H'FFFFFC
	FIFO error register	SCFER1	R	H'0000	H'FFFFFC
	IrDA mode register	SCIFMR1	R/W	H'00	H'FFFFFC
2	Serial mode register	SCSMR2	R/W	H'00	H'FFFFFC
	Bit rate register	SCBRR2	R/W	H'FF	H'FFFFFC
	Serial control register	SCSCR2	R/W	H'00	H'FFFFFC
	Transmit FIFO data register	SCFTDR2	W	—	H'FFFFFC
	Serial status 1 register	SC1SSR2	R/(W)*	H'0060	H'FFFFFC
	Serial status 2 register	SC2SSR2	R/(W)*	H'20	H'FFFFFC
	Receive FIFO data register	SCFRDR2	R	Undefined	H'FFFFFC
	FIFO control register	SCFCR2	R/W	H'00	H'FFFFFC
	FIFO data count register	SCFDR2	R	H'0000	H'FFFFFC
	FIFO error register	SCFER2	R	H'0000	H'FFFFFC
	IrDA mode register	SCIMR2	R/W	H'00	H'FFFFFC

Note: * Only 0 can be written, to clear flags. Use byte access on registers with an a
of 8, and word access on registers with an access size of 16.

R/W:	—	—	—	—	—	—	—

The receive shift register (SCRSR) is the register used to receive serial data.

The SCIF sets serial data input from the RxD pin in SCRSR in the order received, starti
LSB (bit 0) or MSB (bit 7), and converts it to parallel data. When one byte of data has b
received, it is transferred to the receive FIFO data register (SCFRDR) automatically.

SCRSR cannot be read or written to directly.

14.2.2 Receive FIFO Data Register (SCFRDR)

Bit:	7	6	5	4	3	2	1
R/W:	R	R	R	R	R	R	R

The receive FIFO data register (SCFRDR) is a 16-stage FIFO register (8 bits per stage)
received serial data.

When the SCIF has received one byte of serial data, it transfers the received data from S
SCFRDR where it is stored, and completes the receive operation. SCRSR is then enabl
reception, and consecutive receive operations can be performed until the receive FIFO d
register is full (16 data bytes).

SCFRDR is a read-only register, and cannot be written to.

If a read is performed when there is no receive data in the receive FIFO data register, an
value will be returned. When the receive FIFO data register is full of receive data, subs
serial data is lost.

SCTSR, then sends the data to the TxD pin starting with the LSB (bit 0) or MSB (bit 7

When transmission of one byte is completed, the next transmit data is transferred from
to SCTSR, and transmission started, automatically.

SCTSR cannot be read or written to directly.

14.2.4 Transmit FIFO Data Register (SCFTDR)

Bit:	7	6	5	4	3	2	1
R/W:	W	W	W	W	W	W	W

The transmit FIFO data register (SCFTDR) is a 16-stage FIFO register (8 bits per stag
data for serial transmission.

When the SCIF detects that SCTSR is empty, it transfers the transmit data written in S
SCTSR and starts serial transmission. Serial transmission is performed continuously u
no transmit data left in SCFTDR.

SCFTDR is a write-only register, and cannot be read.

The next data cannot be written when SCFTDR is filled with 16 bytes of transmit data
written in this case is ignored.

The serial mode register (SCSMR) is an 8-bit register used to set the SCIF's serial com
format and select the baud rate generator clock source. In IrDA communication mode, i
select the output pulse width.

SCSMR can be read or written to by the CPU at all times.

SCSMR is initialized to H'00 by a reset, by the module standby function, and in standb

Bit 7—Communication Mode (C/$\overline{A}$): Selects asynchronous mode or synchronous mode
SCIF operating mode. In IrDA communication mode, this bit must be cleared to 0.

Bit 7: C/$\overline{A}$	**Description**
0	Asynchronous mode (Ir
1	Synchronous mode

Bit 6—Character Length (CHR)/IrDA Clock Select 3 (ICK3): Selects 7 or 8 bits as the
in asynchronous mode. In synchronous mode, a fixed data length of 8 bits is used regar
CHR setting,

Bit 6: CHR	**Description**
0	8-bit data (Ir
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data registe
(SCFTDR) is not transmitted.

In IrDA communication mode, bit 6 is the IrDA clock select 3 (ICK3) bit, enabling app
clock pulses to be generated according to its setting. See Pulse Width Selection, in secti
Operation in IrDA Mode, for details.

transmit data before transmission. In reception, the parity bit is checked for
(even or odd) specified by the O/$\overline{E}$ bit.

In IrDA communication mode, bit 5 is the IrDA clock select 2 (ICK2) bit, enabling ap
clock pulses to be generated according to its setting. See Pulse Width Selection, in sec
Operation in IrDA Mode, for details.

Bit 4—Parity Mode (O/$\overline{E}$)/IrDA Clock Select 1 (ICK1): Selects either even or odd par
parity addition and checking. The O/$\overline{E}$ bit setting is only valid when the PE bit is set to
parity bit addition and checking, in asynchronous mode. The O/$\overline{E}$ bit setting is invalid
synchronous mode, and when parity addition and checking is disabled in asynchronou

Bit 4: O/$\overline{E}$	**Description**
0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is performed in transmission so t
number of 1-bits in the transmit character plus the parity bit is even. In rece
check is performed to see if the total number of 1-bits in the receive charac
parity bit is even.

2. When odd parity is set, parity bit addition is performed in transmission so th
number of 1-bits in the transmit character plus the parity bit is odd. In recep
is performed to see if the total number of 1-bits in the receive character plus
bit is odd.

In IrDA communication mode, bit 4 is the IrDA clock select 1 (ICK1) bit, enabling ap
clock pulses to be generated according to its setting. See Pulse Width Selection, in sec
Operation in IrDA Mode, for details.

2. In transmission, two 1-bits (stop bits) are added to the end of a transmit char
before it is sent.

In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the
stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next tra
character.

In IrDA communication mode, bit 3 is the IrDA clock select 0 (ICK0) bit, enabling app
clock pulses to be generated according to its setting. See Pulse Width Selection, in secti
Operation in IrDA Mode, for details.

Bit 2—Multiprocessor Mode (MP): Selects a multiprocessor format. When a multiproc
format is selected, the PE bit and O/$\overline{E}$ bit parity settings are invalid. The MP bit setting
valid in asynchronous mode; it is invalid in synchronous mode and IrDA mode.

For details of the multiprocessor communication function, see section 14.3.3, Multiproc
Communication Function.

Bit 2: MP	Description
0	Multiprocessor function disabled (Ir
1	Multiprocessor format selected

0	0	Pϕ clock	(
	1	Pϕ/4 clock	
1	0	Pϕ/16 clock	
	1	Pϕ/64 clock	

Note: Pϕ = peripheral clock

14.2.6 Serial Control Register (SCSCR)

Bit:	7	6	5	4	3	2	1
	TIE	RIE	TE	RE	MPIE	—	CKE1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R/W

The serial control register (SCSCR) performs enabling or disabling of SCIF transmit/r
operations, serial clock output in asynchronous mode, and interrupt requests, and selec
transmit/receive clock source.

SCSCR can be read or written to by the CPU at all times.

SCSCR is initialized to H'00 by a reset, by the module standby function, and in standb

Note: * TXI interrupt requests can be cleared by writing transmit data exceeding the trigger set number to SCFTDR, reading 1 from the TDFE flag, then clearing clearing the TIE bit to 0. When transmit data is written to SCFTDR using the DMAC, the TDFE flag is cleared automatically.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables generation of receive-FIFO interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests when, receive data is transferred from the receive shift register (SCRSR) to the receive FIFO register (SCFRDR), the number of data bytes in SCFRDR reaches or exceeds the receiv set number, and the RDF flag is set to 1 in SC1SSR.

Bit 6: RIE	Description
0	Receive-FIFO-data-full interrupt (RXI) request, receive-error interrup request, and break interrupt (BRI) request disabled* (In
1	Receive-FIFO-data-full interrupt (RXI) request, receive-error interrup request, and break interrupt (BRI) request enabled*

Note: * RXI, ERI, and BRI interrupt requests can be cleared by reading 1 from the R flag, the FER, PER, ORER, or ER flag, or the BRK flag, then clearing the fla clearing the RIE bit to 0. With the RDF flag, read receive data from SCFRDR number of receive data bytes is less than the receive trigger set number, the from the RDF flag and clear it to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of serial transmission by the

Bit 5: TE	Description
0	Transmission disabled*1 (In
1	Transmission enabled*2

Notes: 1. The TDRE flag in SC1SSR is fixed at 1.
2. Serial transmission is started when transmit data is written to SCFTDR in thi Serial mode register (SCSMR) and FIFO control register (SCFCR) settings r made, the transmission format decided, and the transmit FIFO reset, before is set to 1.

SCSMR settings must be made to decide the reception format before settin
to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor in
The MPIE bit setting is only valid in asynchronous mode when the MP bit in SCSMR

The MPIE bit setting is invalid in synchronous mode and IrDA mode, and when the M

Bit 3: MPIE	Description
0	Multiprocessor interrupts disabled (normal reception performed) ([Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled* Receive interrupt (RXI) requests, receive-error interrupt (ERI) requ setting of the RDF and FER in SC1SSR and ORER in SC2SSR ar until data with the multiprocessor bit set to 1 is received.

Note: * Receive data transfer from SCRSR to SCFRDR, receive error detection, ar
the RDF and FER in SC1SSR and ORER flags in SC2SSR, is not performe
receive data with MPB = 1 is received, the MPB flag in SC2SSR is set to 1,
is cleared to 0 automatically, and generation of RXI and ERI (when the RIE
SCSCR is set to 1) and FER and ORER flag setting is enabled.

Bit 2—Reserved: This bit is always read as 0. The write value should always be 0.

the SCIF's operating mode with SCSMR.

For details of clock source selection, see table 14.9 in section 14.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description	
0	0	Asynchronous mode	Internal clock/SCK pin functions as input signal ignored)*1
		Synchronous mode	Internal clock/SCK pin functions as serial output*1
	1	Asynchronous mode	Internal clock/SCK pin functions as clock
		Synchronous mode	Internal clock/SCK pin functions as serial output
1	*	Asynchronous mode	External clock/SCK pin functions as cloc
		Synchronous mode	External clock/SCK pin functions as seria input

*

Notes: 1. Initial value
2. Outputs a clock with a frequency of 16/8/4 times the bit rate.
3. Inputs a clock with a frequency of 16/8/4 times the bit rate.

	ER	TEND	TDFE	BRK	FER	PER	RDF
Initial value:	0	1	1	0	0	0	0
R/W:	R/(W)*	R	R/(W)*	R/(W)*	R	R	R/(W)

Note: * Only 0 can be written, to clear the flag.

The serial status 1 register (SC1SSR) is a 16-bit register in which the lower 8 bits cons
flags that indicate the operating status of the SCIF, and the upper 8 bits indicate the nu
receive errors in the data in the receive FIFO register.

SC1SSR can be read or written to at all times. However, 1 cannot be written to the ER
BRK, RDF, and DR status flags. Also note that in order to clear these flags to 0, they
read as 1. The TEND, FER, and PER flags are read-only and cannot be modified.

SC1SSR is initialized to H'0084 by a reset, by the module standby function, and in sta

Bits 15 to 12—Parity Error Count 3 to 0 (PER3 to PER0): These bits indicate the num
bytes in which a parity error occurred in the receive data in the receive FIFO data regi

These bits are cleared by reading all the receive data in the receive FIFO data register,
setting the RFRST bit to 1 in SCFCR and resetting the receive FIFO data register to th
state.

Bits 11 to 8—Framing Error Count 3 to 0 (FER3 to FER0): These bits indicate the nu
bytes in which a framing error occurred in the receive data in the receive FIFO data re

These bits are cleared by reading all the receive data in the receive FIFO data register,
setting the RFRST bit to 1 in SCFCR and resetting the receive FIFO data register to th
state.

	[Setting conditions] • When the SCIF checks whether the stop bit at the end of the rec 1 when reception ends, and the stop bit is 0*2 • When, in reception, the number of 1-bits in the receive data plus bit does not match the parity setting (even or odd) specified by t in the serial mode register (SCSMR) • When the next serial receive operation is completed while there receive data bytes in SCFRDR

Notes: 1. The ER flag is not affected and retains its previous state when the RE bit in S cleared to 0. When a framing error or parity error occurs, the receive data is transferred to SCFRDR, and reception is then halted or continued according setting of the EI bit. When an overrun error occurs, the receive data is not tra SCFRDR and reception cannot be continued.

2. In 2-stop-bit mode, only the first stop bit is checked for a value of 1; the seco is not checked.

Bit 6—Transmit End (TEND): Indicates that there is no valid data in SCFTDR when th the transmit character is sent, and transmission has been ended.

Bit 6: TEND	Description
0	Transmission is in progress [Clearing condition] When data is written to SCFTDR while TE = 1
1	Transmission has been ended (In [Setting conditions] • In a reset or in standby mode • When the TE bit in SCSCR is 0 • When there is no transmit data in SCFTDR on transmission of th a 1-byte serial transmit character

	• When transmit data exceeding the transmit trigger set number SCFTDR, and 0 is written to TDFE after reading TDFE = 1 • When transmit data exceeding the transmit trigger set number SCFTDR by the on-chip DMAC
1	The number of transmit data bytes in SCFTDR does not exceed th trigger set number ([Setting conditions] • In a reset or in standby mode • When the number of SCFTDR transmit data bytes falls to or be transmit trigger set number as the result of a transmit operatior

Note: * As SCFTDR is a 16-byte FIFO register, the maximum number of bytes that written when TDFE = 0 is {16 – (transmit trigger set number)}. Data written this will be ignored. The number of data bytes in SCFTDR is indicated by th bits of SCFDR.

Bit 4—Break Detect (BRK): Indicates that a receive data break signal has been detect

Bit 4: BRK	Description
0	A break signal has not been received ([Clearing conditions] • In a reset or in standby mode • When 0 is written to BRK after reading BRK = 1
1	A break signal has been received [Setting condition] When data with a framing error is received, and a framing error als the next receive data (all space “0”)

Note: When a break is detected, transfer to SCFRDR of the receive data (H'00) follow detection is halted. When the break ends and the receive signal returns to mark data transfer is resumed.

1	There is a framing error in the receive data read from SCFRDR [Setting condition] When there is a framing error in SCFRDR read data

Bit 2—Parity Error (PER): In asynchronous mode, indicates a parity error in the data re
the receive FIFO data register (SCFRDR).

Bit 2: PER	Description
0	There is no parity error in the receive data read from SCFRDR (Ir [Clearing conditions] • In a reset or in standby mode • When there is no parity error in SCFRDR read data
1	There is a parity error in the receive data read from SCFRDR [Setting condition] When there is a parity error in SCFRDR read data

Bit 1—Receive Data Register Full (RDF): Indicates that the received data has been tran
the receive FIFO data register (SCFRDR), and the number of receive data bytes in SCF
equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in th
control register (SCFCR).

	• When SCFRDR is read by the on-chip DMAC until the number data bytes in SCFRDR falls below the receive trigger set numb
1	The number of receive data bytes in SCFRDR is equal to or greate receive trigger set number [Setting condition] When SCFRDR contains at least the receive trigger set number of bytes

Note: SCFRDR is a 16-byte FIFO register. When RDF = 1, at least the receive trigger of data bytes can be read. If all the data in SCFRDR is read and another read is the data value will be undefined. The number of receive data bytes in SCFRDR by the lower 8 bits of SCFDR.

Bit 0—Receive Data Ready (DR): Indicates that there are fewer than the receive trigg number of data bytes in the receive FIFO data register (SCFRDR), and no further data for at least 16 etu after the stop bit of the last data received.

Bit 0: DR	Description
0	Reception is in progress or has ended normally and there is no rec in SCFRDR ([Clearing conditions] • In a reset or in standby mode • When 0 is written to DR after all the remaining receive data has read*1
1	No further receive data has arrived, and SCFRDR contains fewer t receive trigger set number of data bytes [Setting condition] When SCFRDR contains fewer than the receive trigger set numbe data bytes, and no further data has arrived for at least 16 etu after of the last data received*2

Notes: 1. All remaining receive data should be read before clearing the DR flag.
2. Equivalent to 1.6 frames when using an 8-bit, 1-stop-bit format.
etu: Elementary time unit = sec/bit

The serial status 2 register (SC2SSR) is an 8-bit register.

SC2SSR can be read or written to at all times. However, 1 cannot be written to the ORI
Also note that in order to clear this flag to 0, they must first be read as 1.

SC2SSR is initialized to H'20 by a reset, by the module standby function, and in standb

Bit 7—Transmit LSB/MSB-First Select (TLM): Selects LSB-first or MSB-first mode i
transmission.

Bit 7: TLM	Description	
0	LSB-first transmission	(Ir
1	MSB-first transmission	

Bit 6—Receive LSB/MSB-First Select (RLM): Selects LSB-first or MSB-first mode in
reception.

Bit 6: RLM	Description	
0	LSB-first reception	(Ir
1	MSB-first reception	

Bits 5 and 4—Clock Bit Rate Ratio (N1, N0): These bits select the ratio of the base clo
rate.

Bit 5: N1	Bit 4: N0	Description	
0	0	SCIF operates on base clock of 4 times the bit rate	
	1	SCIF operates on base clock of 8 times the bit rate	
1	0	SCIF operates on base clock of 16 times the bit rate	(Ir
	1	Setting prohibited	

format.

Bit 2—Multiprocessor Bit Transfer (MPBT): When transmission is performed using a multiprocessor format in asynchronous mode, MPBT stores the multiprocessor bit to b the transmit data.

The MPBT bit setting is invalid in synchronous mode and IrDA mode, when a multipr format is not used, and when the operation is not transmission.

Bit 2: MPBT	Description
0	Data with a 0 multiprocessor bit is transmitted (
1	Data with a 1 multiprocessor bit is transmitted

Bit 1—Receive Data Error Ignore Enable (EI): Selects whether or not the receive oper be continued when a framing error or parity error occurs in receive data (ER = 1).

Bit 1: EI	Description
0	Receive operation is halted when framing error or parity error occu reception (ER = 1) (
1	Receive operation is continued when framing error or parity error o reception (ER = 1)

Note: When EI = 0, only the last data in SCFRDR is treated as data containing an err = 1, receive data is sent to SCFRDR even if it contains an error.

1	An overrun error occurred during reception [Setting condition] When the next serial receive operation is completed while there are data bytes in SCFRDR

Notes: 1. The ORER flag is not affected and retains its previous state when the RE bit
is cleared to 0.

2. The receive data prior to the overrun error is retained in SCFRDR, and the d
received subsequently is lost. Serial reception cannot be continued while the
is set to 1. Also, serial transmission cannot be continued in synchronous mo

14.2.9 Bit Rate Register (SCBRR)

Bit:	7	6	5	4	3	2	1
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The bit rate register (SCBRR) is an 8-bit register that sets the serial transmit/receive bit
accordance with the baud rate generator operating clock selected by bits CKS1 and CK
serial mode register (SCSMR).

SCBRR can be read or written to by the CPU at all times.

SCBRR is initialized to H'FF by a reset, by the module standby function, and in standb

The SCBRR setting is found from the following equations.

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, 1, 2, or 3)
(See the table below for the relation between n and the clock.)

		SCSMR Settings	
n	**Clock**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4		1
2	Pϕ/16	1	0
3	Pϕ/64		1

The bit rate error in asynchronous mode is found from the following equations:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

(When operating on a base clock of 16 times the b

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$

(When operating on a base clock of 8 times the bi

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 16 \times 2^{2n-1}} - 1 \right\} \times 100$$

(When operating on a base clock of 4 times the bi

(Bits/s)	n	N	(%)	n	N	(%)	n	N	(%)	n	N
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	21
150	1	103	0.16	1	108	0.21	1	127	0.00	1	15
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77
600	0	103	0.16	0	108	0.21	0	127	0.00	0	15
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19
9600	0	6	–6.99	0	6	–2.48	0	7	0.00	0	9
19200	0	2	8.51	0	2	13.78	0	3	0.00	0	4
31250	0	1	0.00	0	1	4.86	0	1	22.88	0	2
38400	0	1	–18.62	0	1	–14.67	0	1	0.00	—	—

	Pϕ (MHz)										
	3.6864			4			4.9152				
Bit Rate (Bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64
300	1	95	0.00	1	103	0.16	1	127	0.00	1	12
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	12
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15
19200	0	5	0.00	0	6	–6.99	0	7	0.00	0	7
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3

1200	0	155	0.16	0	159	0.00	0	191	0.00	0	2
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	1
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	5
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	2
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	1
31250	0	5	0.00	0	5	2.40	0	6	5.33	0	7
38400	0	4	–2.34	0	4	0.00	0	5	0.00	0	6

	Pϕ (MHz)										
	9.8304			10			12				
Bit Rate (Bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	2
150	2	127	0.00	2	129	0.16	2	155	0.16	2	1
300	1	255	0.00	2	64	0.16	2	77	0.16	2	7
600	1	127	0.00	1	129	0.16	1	155	0.16	1	1
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	7
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	1
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	7
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	3
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	1
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	1
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9

1200	1	95	0.00	1	103	0.16	1	194	0.16
2400	0	191	0.00	0	207	0.16	1	97	–0.35
4800	0	95	0.00	0	103	0.16	0	194	0.16
9600	0	47	0.00	0	51	0.16	0	97	–0.35
19200	0	23	0.00	0	25	0.16	0	48	–0.35
31250	0	14	–1.70	0	15	0.00	0	29	0.00
38400	0	11	0.00	0	12	0.16	0	23	1.73

1 k	1	249	2	124	2	249	3
2.5 k	1	99	1	199	2	99	2
5 k	0	199	1	99	1	199	2
10 k	0	99	0	199	1	99	1
25 k	0	39	0	79	0	159	1
50 k	0	19	0	39	0	79	0
100 k	0	9	0	19	0	39	0
250 k	0	3	0	7	0	15	0
500 k	0	1	0	3	0	7	0
1 M	0	0*	0	1	0	3	0
2 M			0	0*	0	1	0

Note: As far as possible, the setting should be made so that the error is within 1%.

[Legend]

Blank: No setting is available.

—: A setting is available but error occurs.

* Continuous transmission/reception is not possible.

2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.66080	614400	0	0
20	625000	0	0
24	750000	0	0
24.57600	768000	0	0
28	896875	0	0
30	937500	0	0

4	1.0000	62500
4.9152	1.2288	76800
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
30	7.5000	468750

Table 14.7 Maximum Bit Rate with External Clock Input (Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bits/s
8	1.3333	1333333.3
16	2.6667	2666666.7
30	5.0	5000000.0

for the transmit and receive FIFO registers, and also contains a loopback test enable bit

SCFCR can be read or written to at all times.

SCFCR is initialized to H'00 by a reset, by the module standby function, and in standby

Bits 7 and 6—Receive FIFO Data Number Trigger (RTRG1, RTRG0): These bits are u
the number of receive data bytes that sets the receive data full (RDF) flag in the serial s
register (SC1SSR).

The RDF flag is set when the number of receive data bytes in the receive FIFO data reg
(SCFRDR) is equal to or greater than the trigger set number shown in the following tab

Bit 7: RTRG1	Bit 6: RTRG0	Receive Trigger Number
0	0	1*
	1	4
1	0	8
	1	14

Note: * Initial value

Bits 5 and 4—Transmit FIFO Data Number Trigger (TTRG1, TTRG0): These bits are u
the number of remaining transmit data bytes that sets the transmit FIFO data register en
(TDFE) flag in the serial status 1 register (SC1SSR).

The TDFE flag is set when the number of transmit data bytes in the transmit FIFO data
(SCFTDR) is equal to or less than the trigger set number shown in the following table.

Bit 5: TTRG1	Bit 4: TTRG0	Transmit Trigger Number
0	0	8 (8)*
	1	4 (12)
1	0	2 (14)
	1	1 (15)

Note: * Initial value. Figures in parentheses are the number of empty bytes in SCFT
the flag is set.

Bit 2—Transmit FIFO Data Register Reset (TFRST): Invalidates the transmit data in t
FIFO data register and resets it to the empty state.

Bit 2: TFRST	Description
0	Reset operation disabled (I
1	Reset operation enabled

Note: A reset operation is performed in the event of a reset, module standby, or in sta

Bit 1—Receive FIFO Data Register Reset (RFRST): Invalidates the receive data in the
FIFO data register and resets it to the empty state.

Bit 1: RFRST	Description
0	Reset operation disabled (I
1	Reset operation enabled

Note: A reset operation is performed in the event of a reset, module standby, or in sta

Bit 0—Loopback Test (LOOP): Internally connects the transmit output pin (TxD) and
input pin (RxD), enabling loopback testing.

Bit 0: LOOP	Description
0	Loopback test disabled (I
1	Loopback test enabled

SCFDR is initialized to H'0000 by a reset, by the module standby function, and in stanc
It is also initialized to H'00 by setting the TFRST and RFRST bits to 1 in SCFCR to res
SCFTDR and SCFRDR to the empty state.

Upper 8 bits:	15	14	13	12	11	10	9
	—	—	—	T4	T3	T2	T1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 15 to 13—Reserved: These bits are always read as 0. The write value should alway

Bits 12 to 8—Transmit FIFO Data Count 4 to 0 (T4 to T0): These bits show the numbe
untransmitted data bytes in SCFTDR.

A value of H'00 indicates that there is no transmit data, and a value of H'10 indicates th
SCFTDR is full of transmit data. The value is cleared to H'00 by transmitting all the da
as by the above initialization conditions.

Lower 8 bits:	7	6	5	4	3	2	1
	—	—	—	R4	R3	R2	R1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 7 to 5—Reserved: These bits are always read as 0. The write value should always b

Bits 4 to 0—Receive FIFO Data Count 4 to 0 (R4 to R0): These bits show the number c
data bytes in SCFRDR.

A value of H'00 indicates that there is no receive data, and a value of H'10 indicates tha
is full of receive data. The value is cleared to H'00 by reading all the receive data from
as well as by the above initialization conditions.

Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Lower 8 bits:	7	6	5	4	3	2	1
	ED7	ED6	ED5	ED4	ED3	ED2	ED1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bits 15 to 0—Error Data Flags 15 to 0 (ED15 to ED0): These flags indicate the data lo
receive FIFO data register at which an error occurred. When data in the nth stage of th
contains an error, the nth bit is set to 1. Note that this register is not cleared by setting
bit to 1 in SCFCR.

Bits 15 to 0: ED15 to ED0	**Description**
0	No parity or framing error in data in corresponding stage of registe ([illegible]
1	Parity or framing error present in data in corresponding stage of re

Note: A reset operation is performed in the event of a reset, when the module standb
used, or in standby mode. These flags are also cleared by reading the data in w
parity error or framing error occurred from SCFRDR.

14.2.13 IrDA Mode Register (SCIMR)

The IrDA mode register (SCIFMR) allows selection of the IrDA mode and the IrDA c
width, and inversion of the IrDA receive data polarity.

SCIMR can be read and written to at all times.

SCIMR is initialized to H'00 by a reset, by the module standby function, and in standb

0	Operation as SCIF is selected	(Ir
1	Operation as IrDA is selected*	

Note: * When operation as an IrDA interface is selected, bit 7 (C/$\overline{A}$) of the serial mod (SCSMR) must be cleared to 0.

Bit 6—Output Pulse Width Select (PSEL): Selects either 3/16 of the bit length set by bi ICK0 in the serial mode register (SCSMR), or 3/16 of the bit length corresponding to th baud rate, as the IrDA output pulse width. The setting is shown together with bits 6 to 3 ICK0) of the serial mode register (SCSMR).

Serial Mode Register (SCSMR)				SCIMR	
Bit 6: ICK3	**Bit 5: ICK2**	**Bit 4: ICK1**	**Bit 3: ICK0**	**Bit 2: PSEL**	**Description**
ICK3	ICK2	ICK1	ICK0	1	Pulse width: 3/16 of bit length set in b ICK0
Don't care	Don't care	Don't care	Don't care	0	Pulse width: 3/16 of bit length set in S (Ir

Note: A fixed clock pulse signal, IRCLK, must be generated by multiplying the Pϕ clock 2 (where N is determined by the value set in ICK3 to ICK0). For details, see sect Pulse Width Selection.

Bit 5—IrDA Receive Data Inverse (RIVS): Allows inversion of the receive data polarit selected in IrDA communication.

Bit 5: RIVS	**Description**
0	Receive data polarity inverted in reception (Ir
1	Receive data polarity not inverted in reception

Note: Make the selection according to the characteristics of the IrDA modulation/demo module.

Bits 4 to 0—Reserved: These bits are always read as 0. The write value should always b

executed by connecting an infrared transmission/reception unit.

Sixteen-stage FIFO buffers are provided for both transmission and reception, reducing overhead and enabling fast, continuous communication to be performed.

Selection of asynchronous, synchronous, or IrDA mode and the transmission format is means of the serial mode register (SCSMR) and IrDA mode register (SCIMR) as show 14.8. The SCIF clock source is determined by a combination of the C/$\overline{A}$ bit in SCSMR IRMOD bit in SCIMR, and the CKE1 and CKE0 bits in the serial control register (SC shown in table 14.9.

- Asynchronous Mode
  - Data length: Choice of 7 or 8 bits
  - Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 st combination of these parameters determines the transmit/receive format and ch length)
  - Detection of framing, parity, and overrun errors, receive FIFO data full and rec ready conditions, and breaks, during reception
  - Detection of transmit FIFO data empty condition during transmission
  - Choice of internal or external clock as SCIF clock source

    When internal clock is selected: The SCIF operates on a clock with a frequency 4 times the bit rate of the baud rate generator, and can output this operating clo

    When external clock is selected: A clock with a frequency of 16, 8, or 4 times must be input (the built-in baud rate generator is not used).
- Synchronous Mode
  - Transmit/receive format: Fixed 8-bit data
  - Detection of overrun errors during reception
  - Choice of internal or external clock as SCIF clock source

    When internal clock is selected: The SCIF operates on the baud rate generator can output a serial clock to external devices.

    When external clock is selected: The on-chip baud rate generator is not used, a operates on the input serial clock.

SCIMR	SCSMR Settings						SCIF Transmit/Receiv		
Bit 7: IRMOD	**Bit 7: C/$\overline{A}$**	**Bit 6: CHR**	**Bit 2: MP**	**Bit 5: PE**	**Bit 3: STOP**	**Mode**	**Data Length**	**MP Bit**	**Parity Bit**
0	0	0	0	0	0	Asynchronous mode	8-bit data	Absent	Abser
					1				
				1	0				Preser
					1				
		1		0	0		7-bit data		Abser
					1				
				1	0				Preser
					1				
		0	1	*	0	Asynchronous mode (multi-processor format)	8-bit data	Present	Abser
				*	1				
		1		*	0		7-bit data		
				*	1				
0	1	*	*	*	*	Synchronous mode	8-bit data	Absent	Abser
1	0	ICK3	ICK2	ICK1	ICK0	IrDA mode	8-bit data	Absent	Abser
	1	*	*	*	*	Setting prohibited	—	—	—

Note: * Don’t care

		1			times bit rate
1	0	0	Synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

so that data can be read or written during transmission or reception, enabling continuou
transfer.

Figure 14.3 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the communication line is usually held in the m
(high level). The SCIF monitors the line, and when it goes to the space state (low level)
recognizes a start bit and starts serial communication.

One serial communication character consists of a start bit (low level), followed by data
or MSB-first order selectable), a parity bit or multiprocessor bit (high or low level), and
one or two stop bits (high level).

In asynchronous mode, the SCIF performs synchronization at the falling edge of the sta
reception. The SCIF samples the data on the eighth (fourth, second) pulse of a clock wi
frequency of 16 (8, 4) times the length of one bit, so that the transfer data is latched at t
each bit.

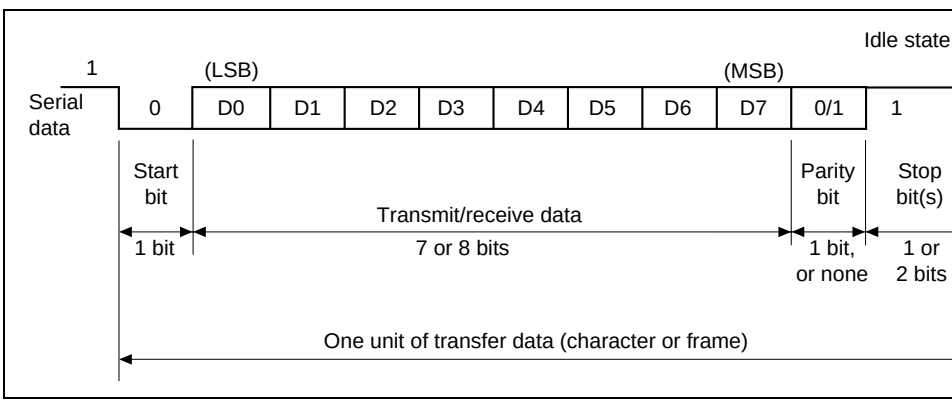


Figure 14.3 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits, LSB-First Transfer)

			1	S	8-bit data		STOP	
	1		0	S	8-bit data		P	
			1	S	8-bit data		P	
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	
0	*	1	0	S	8-bit data		MPB	
	*		1	S	8-bit data		MPB	
1	*		0	S	7-bit data	MPB	STOP	
	*		1	S	7-bit data	MPB	STOP	

Note: * Don’t care

[Legend]

S: Start bit

STOP: Stop bit

P: Parity bit

MPB: Multiprocessor bit

frequency of the clock output in this case is 16, 8, or 4 times the bit rate.

Data Transmit/Receive Operations

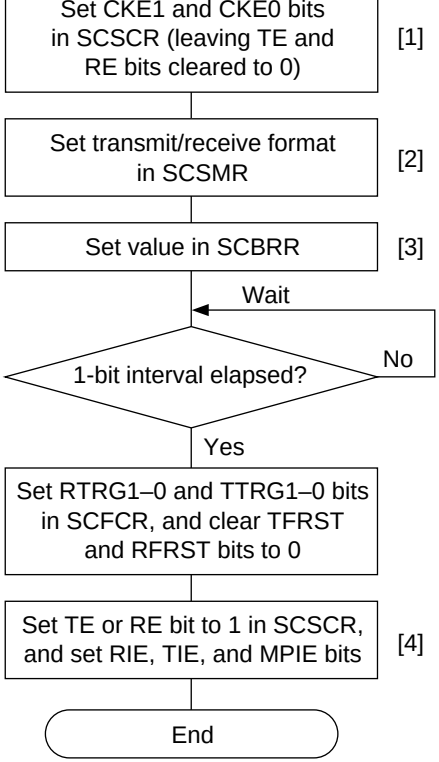
- SCIF Initialization (Asynchronous Mode)

  Before transmitting and receiving data, it is necessary to clear the TE and RE bits to
  SCSCR, then initialize the SCIF as described below.

  When the operating mode, communication format, etc., is changed, the TE and RE b
  cleared to 0 before making the change using the following procedure. When the TE
  cleared to 0, the transmit shift register (SCTSR) is initialized. Note that clearing the
  bits to 0 does not change the contents of the serial status 1 register (SC1SSR), the tr
  FIFO data register (SCFTDR), or the receive FIFO data register (SCFRDR). The TE
  not be cleared to 0 until all transmit data has been transmitted and the TEND flag ha
  in SC1SSR. It is possible to clear the TE bit to 0 during transmission, but the data b
  transmitted will go to the high-impedance state after TE is cleared. Also, before star
  transmission by setting TE again, the TFRST bit should first be set to 1 in SCFCR t
  SCFTDR.

  When an external clock is used the clock should not be stopped during operation, in
  initialization, since operation will be unreliable in this case.

  Figure 14.4 shows a sample SCIF initialization flowchart.



SCSMR.
When using IrDA mode, also s
SCIFMR.

[3] Write a value corresponding to
rate into the bit rate register (S
(Not necessary if an external c
used.)

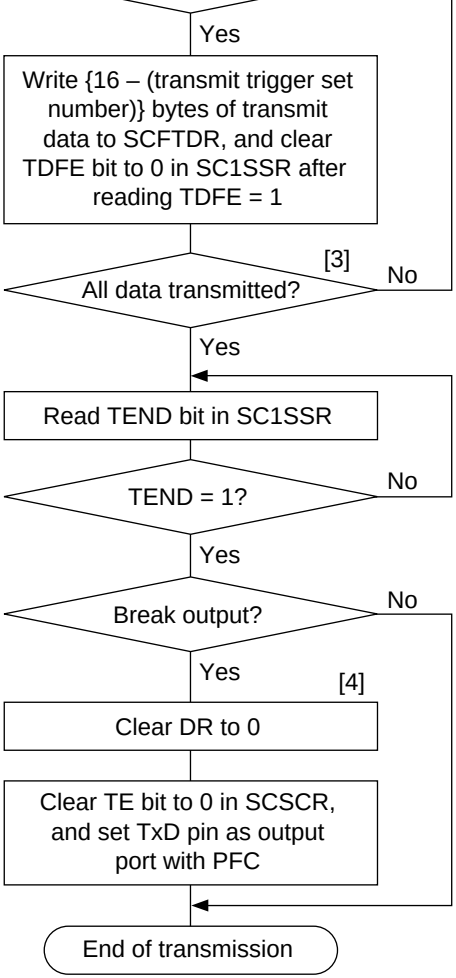
[4] Wait at least one bit interval, th
the TE bit or RE bit in SCSCR
Also set the RIE, TIE, and MPI
Setting the TE and RE bits ena
the TxD and RxD pins to be us
When transmitting, the SCIF w
the mark state; when receiving
go to the idle state, waiting for
bit.

Figure 14.4 Sample SCIF Initialization Flowchart

- Serial Data Transmission (Asynchronous Mode)

  Figure 14.5 shows a sample flowchart for serial transmission.

  Use the following procedure for serial data transmission after enabling the SCIF f
  transmission.



started by writing transmit data.

The number of data bytes that car written is {16 – (transmit trigger se number)}.

[3] Serial transmission continuation procedure: To continue serial transmission, read 1 from the TDF confirm that writing is possible, the data to SCFTDR, and then clear t TDFE bit to 0. (Checking and clea the TDFE bit is automatic when th DMAC is activated by a transmit-F data-empty interrupt (TXI) request data is written to SCFTDR.)

[4] Break output at the end of serial transmission: To output a break in transmission, clear the port data r (DR) to 0, then clear the TE bit to SCSCR, and set the TxD pin as a port with the PFC.

In steps 2 and 3, the number of transr bytes that can be written can be asce from the number of transmit data byte SCFTDR indicated in the upper 8 bits FIFO data count register (SCFDR).

Figure 14.5 Sample Serial Transmission Flowchart

number of data bytes in SCFTDR falls to or below the transmit trigger number set
control register (SCFCR) during transmission, the TDFE flag is set. If the TE bit s
serial control register (SCSCR) is 1 at this time, a transmit-FIFO-data-empty intern
requested.

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 or MSB-first order acco
setting of the TLM bit in SC2SSR.
c. Parity bit or multiprocessor bit: One parity bit (even or odd parity), or one mult
bit is output. (A format in which neither a parity bit nor a multiprocessor bit is
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 transm
sent.

3. The SCIF checks for transmit data in SCFTDR at the timing for sending the stop b
data in SCFTDR, it is transferred to SCTSR, the stop bit is sent, and then serial tra
the next frame is started.

   If there is no transmit data in SCFTDR, the TEND flag is set to 1 in the serial statu
(SC1SSR), the stop bit is sent, and then the line goes to the mark state in which 1 i
continuously.

   Figure 14.6 shows an example of the operation for transmission in asynchronous m

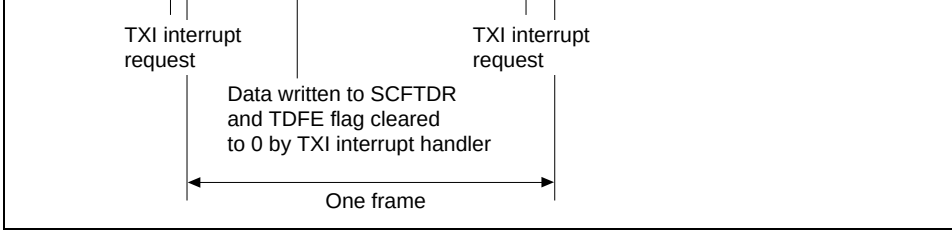


Figure 14.6 Example of Transmit Operation in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit, LSB-First Transfer)

4. When modem control is enabled, transmission can be stopped and restarted in accor
the $\overline{\text{CTS}}$ input value. When $\overline{\text{CTS}}$ is set to 1, if transmission is in progress, the line go
mark state after transmission of one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit
output starting from the start bit.

   Figure 14.7 shows an example of the operation when modem control is used.

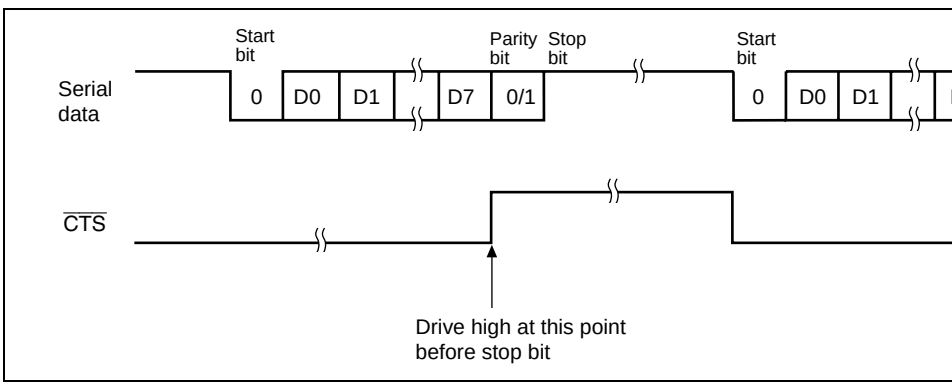
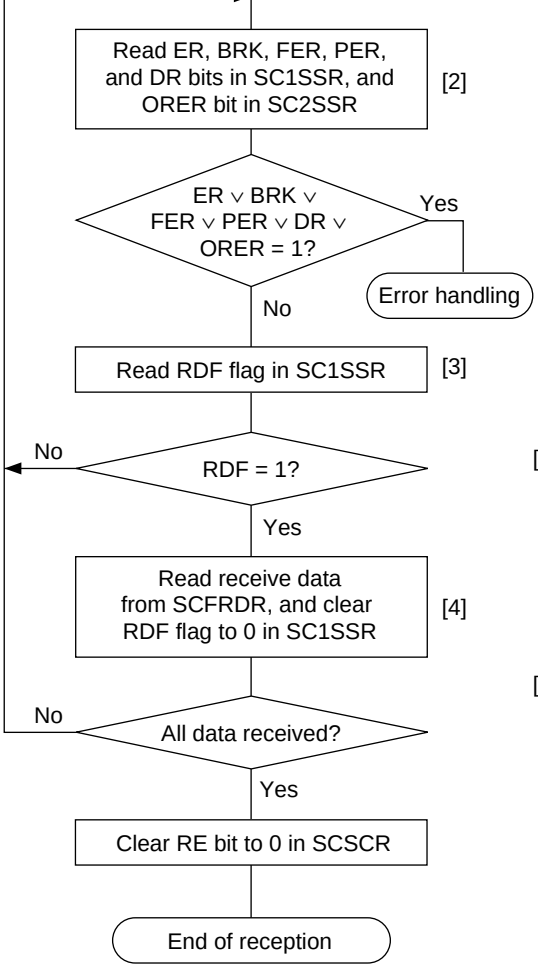


Figure 14.7 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

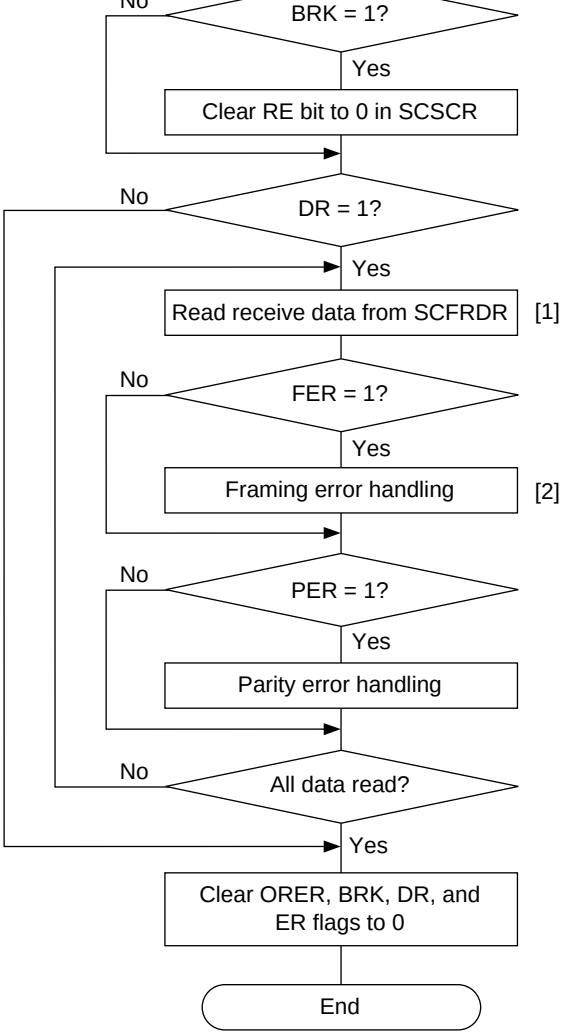


error has occurred.
If a receive error has occurred, ER, BRK, FER, PER, and DR fl SC1SSR and the ORER flag in to identify the error. After perfor appropriate error handling, ensu the ORER, BRK, DR, and ER b cleared to 0. Reception cannot resumed if the ORER bit is set t setting of the EI bit in SC2SSR determines whether reception is continued or halted when any o PER3–0 or FER3–0 is set to 1.
In the case of a framing error, a can be detected by reading the the RxD pin.

[3] SCIF status check and receive read: Read the serial status 1 re (SC1SSR) and check that RDF read receive data from the rece data register (SCFRDR) and cle RDF bit to 0. Transition of the R from 0 to 1 can also be identifie means of an RXI interrupt.

[4] Serial reception continuation pr To continue serial reception, rea least the receive trigger set num data bytes from SCFRDR, and the RDF flag after reading 1 fro number of receive data bytes in SCFRDR can be ascertained by the lower bits of the FIFO data register (SCFDR). (The RDF bit cleared automatically when the activated by an RXI interrupt an SCFRDR value is read.)

Figure 14.8 Sample Serial Reception Flowchart (1)



note that the H'00 break
in which a framing error
occurred is stored as th
data in SCFRDR.

Figure 14.8 Sample Serial Reception Flowchart (2)

After receiving these bits, the SCIF carries out the following checks:

a. Parity check: The SCIF checks whether the number of 1-bits in the receive data
the parity (even or odd) set in the O/$\overline{E}$ bit in the serial mode register (SCSMR).
b. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop
the first is checked.
c. Status check: The SCIF checks whether receive data can be transferred from th
shift register (SCRSR) to SCFRDR.
d. Break check: The SCIF checks that the BRK flag is 0, indicating no break.

If all the above checks are passed, the receive data is stored in SCFRDR. If a recei
detected in the error check, the operation is as shown in table 14.11.

Note: No further receive operations can be performed when an overrun error has occ
setting of the EI bit in SC2SSR determines whether reception is continued or l
a framing error or parity error occurs.
Also, as the RDF flag is not set to 1 when receiving, the error flags must be cl

4. If the RIE bit setting in SCSCR is 1 when the RDF or DR flag is set to 1, a receive
full interrupt (RXI) is requested.

If the RIE bit setting in SCSCR is 1 when the ORER, PER, or FER flag is set to 1,
error interrupt (ERI) is requested.

If the RIE bit setting in SCSCR is 1 when the BRK flag is set to 1, a break-receive
(BRI) is requested.

	from that (even or odd) set in SCSMR	SCRSR to SCFRDR

Figure 14.9 shows an example of the operation for reception in asynchronous mode

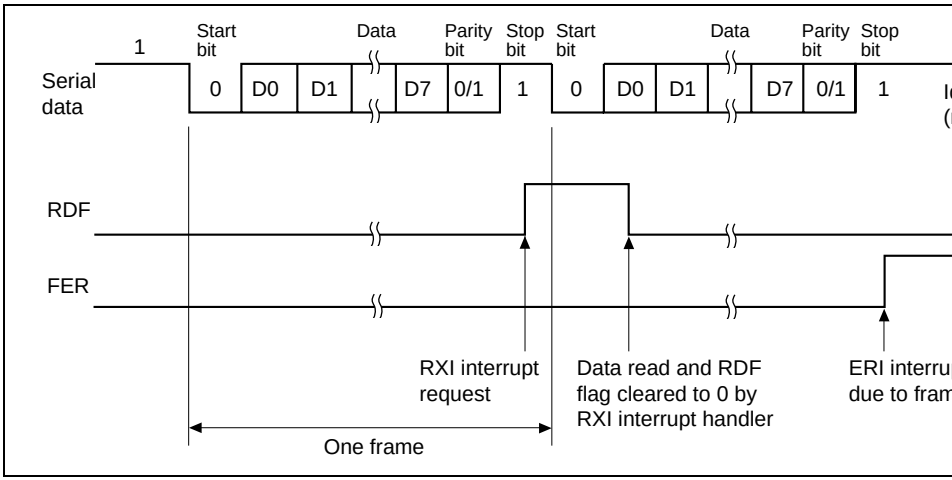


Figure 14.9 Example of SCIF Receive Operation
(Example with 8-Bit Data, Parity, One Stop Bit, LSB-First Transfer)

5. When modem control is enabled, the $\overline{\text{RTS}}$ signal is output when SCFRDR is empty. $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that SCFRDR is full a reception is not possible.

   Figure 14.10 shows an example of the operation when modem control is used.

14.3.3 Multiprocessor Communication Function

The multiprocessor communication function performs serial communication using a multiprocessor format, in which a multiprocessor bit is added to the transfer data, in as mode. Use of this function enables data transfer to be performed among a number of p sharing a serial communication line.

When multiprocessor communication is carried out, each receiving station is addresse unique ID code.

The serial communication cycle consists of two cycles: an ID transmission cycle whic the receiving station, and a data transmission cycle. The multiprocessor bit is used to d between the ID transmission cycle and the data transmission cycle.

The transmitting station first sends the ID of the receiving station with which it wants serial communication as data with a 1 multiprocessor bit added. It then sends transmit with a 0 multiprocessor bit added.

The receiving stations skip the data until data with a 1 multiprocessor bit is sent. Wher 1 multiprocessor bit is received, each receiving stations compares that data with its ow station whose ID matches then receives the data sent next. Stations whose ID does not continue to skip the data until data with a 1 multiprocessor bit is again received. In thi communication is carried out among a number of processors.

Figure 14.11 shows an example of inter-processor communication using a multiproces

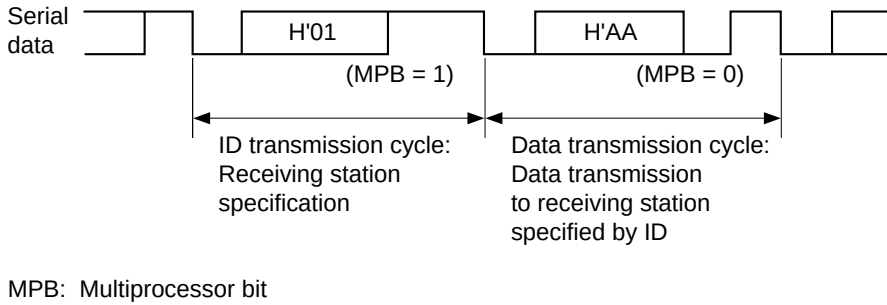


Figure 14.11 Example of Inter-Processor Communication Using Multiprocesso (Transmission of Data H'AA to Receiving Station A)

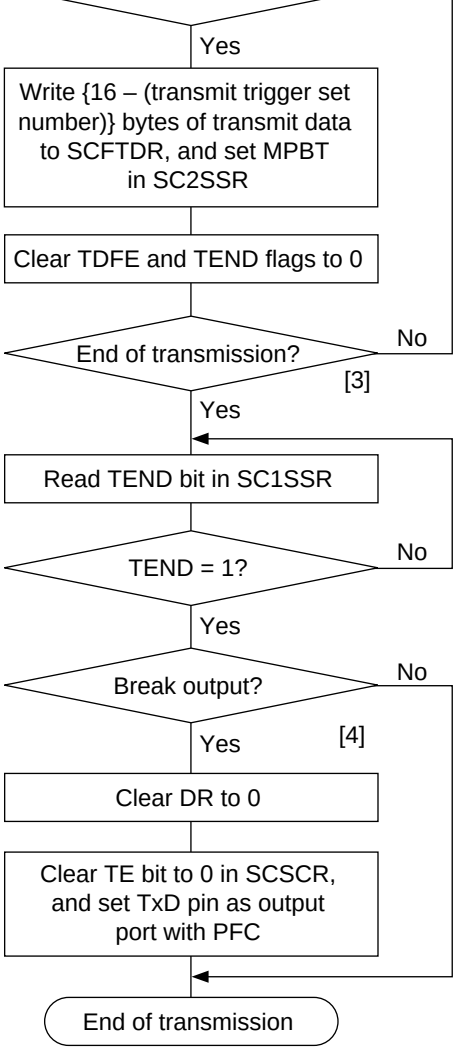
Transmit/Receive Formats: There are four transmit/receive formats. When the multip format is specified, the parity bit specification is invalid. For details, see table 14.10.

Clock: See the section on asynchronous mode.

Data Transmit/Receive Operations

- SCI Initialization
  See the section on asynchronous mode.

- Multiprocessor Serial Data Transmission
  Figure 14.12 shows a sample flowchart for multiprocessor serial data transmission.
  Use the following procedure for multiprocessor serial data transmission after enabli SCIF for transmission.



and TEND flags to 0 after read
from them.
The number of data bytes that
written is {16 – (transmit trigge
number)}.

[3] Serial transmission continuatio
procedure: To continue serial
transmission, read 1 from the T
to confirm that writing is possib
write data to SCFTDR, and the
the TDFE bit to 0. (Checking a
clearing of the TDFE bit is auto
when the DMAC is activated by
transmit-FIFO-data-empty inter
(TXI) request, and data is writte
SCFTDR.)

[4] Break output at the end of seria
transmission: To output a brea
serial transmission, clear the p
register (DR) to 0, then clear th
to 0 in SCSCR, and set the TxD
an output port with the PFC.

In steps 2 and 3, the number of tra
data bytes that can be written can
ascertained from the number of tra
data bytes in SCFTDR indicated in
upper 8 bits of the FIFO data coun
(SCFDR).

Figure 14.12 Sample Multiprocessor Serial Transmission Flowchart

number of data bytes in SCFTDR falls to or below the transmit trigger number set i
during transmission, the TDFE flag is set to 1. If the TIE bit setting in SCSCR is 1 a
a transmit-FIFO-data-empty interrupt (TXI) is requested.

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 or MSB-first order accor
setting of the TLM bit in SC2SSR.
c. Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
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 transmi
sent.

3. The SCIF checks for transmit data in SCFTDR at the timing for sending the stop bit
data in SCFTDR, it is transferred to SCTSR, the stop bit is sent, and then serial tran
the next frame is started.

   If there is no transmit data in SCFTDR, the TEND flag is set to 1 in SC1SSR, the st
sent, and then the line goes to the mark state in which 1 is output continuously.

   Figure 14.13 shows an example of SCIF operation for transmission using a multipro
format.

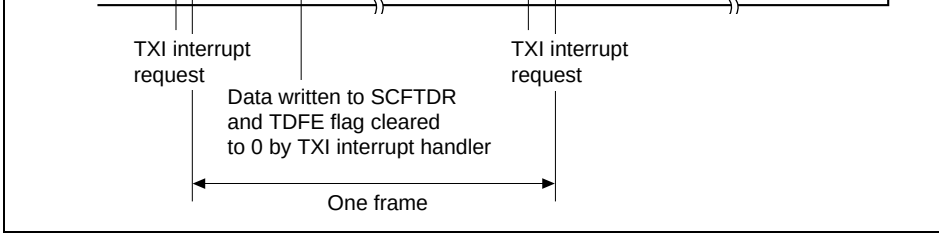
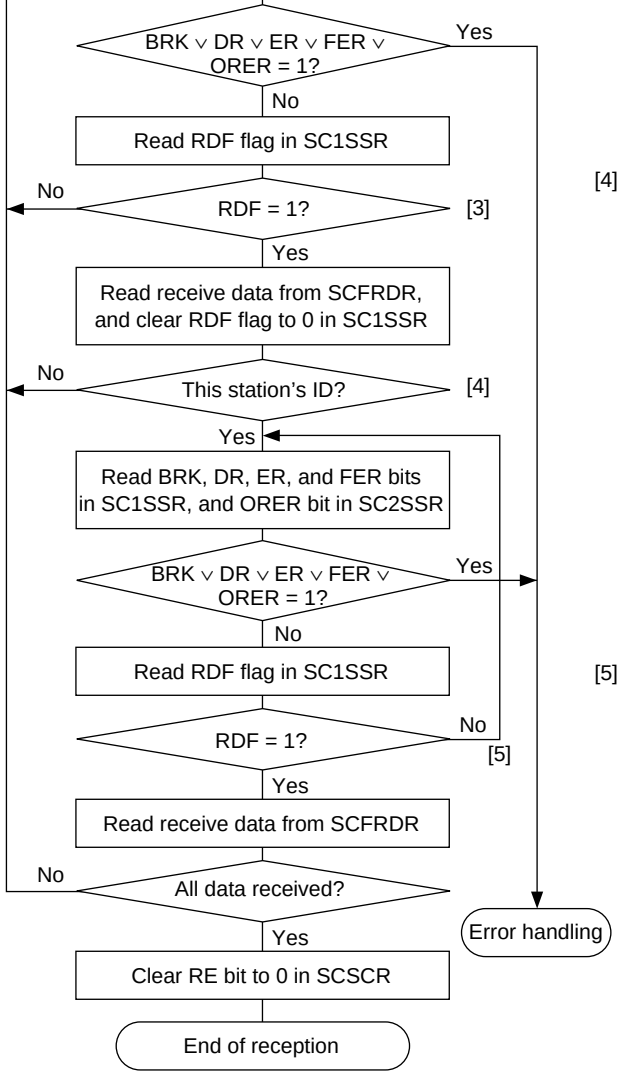


Figure 14.13 Example of SCIF Transmit Operation
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit, LSB-First Tra

- Multiprocessor Serial Data Reception

  Figure 14.14 shows a sample flowchart for multiprocessor serial reception.

  Use the following procedure for multiprocessor serial data reception after enabling
  for reception.

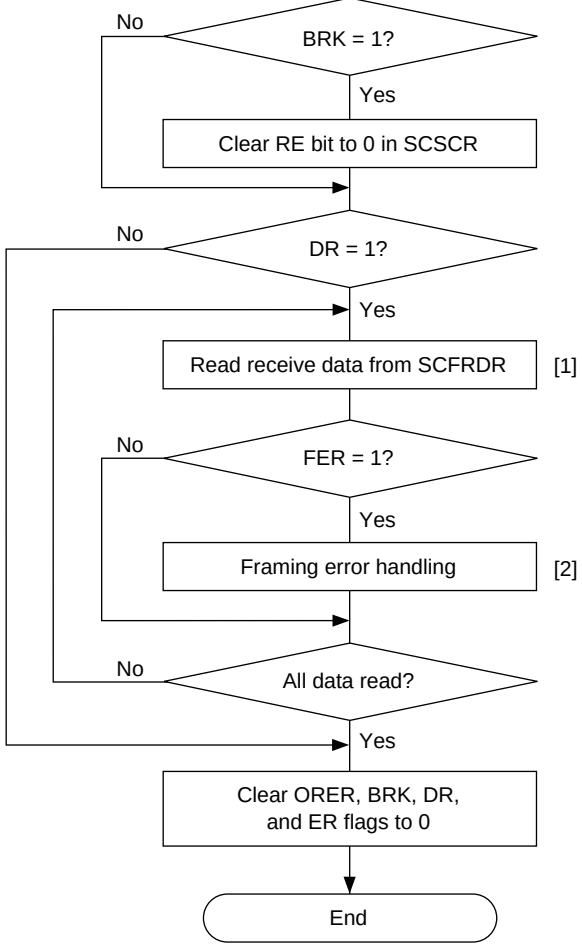


and compare it with this statio
If the data is not this station's
the MPIE bit to 1 again, and c
RDF bit to 0.
If the data is this station's ID,
the RDF bit to 0.

[4] Receive error handling and b
detection: Read the ER, BRK
and DR flags in SC1SSR and
ORER flag in SC2SSR to che
whether a receive error has c
If a receive error has occurre
the ER, BRK, FER, and DR fl
SC1SSR and the ORER flag
SC2SSR to identify the error.
performing the appropriate er
handling, ensure that ER, BR
and ORER are all cleared to
setting of the EI bit in SC2SS
determines whether reception
continued or halted when the
bit is set to 1. In the case of a
error, a break can be detecte
reading the value of the RxD

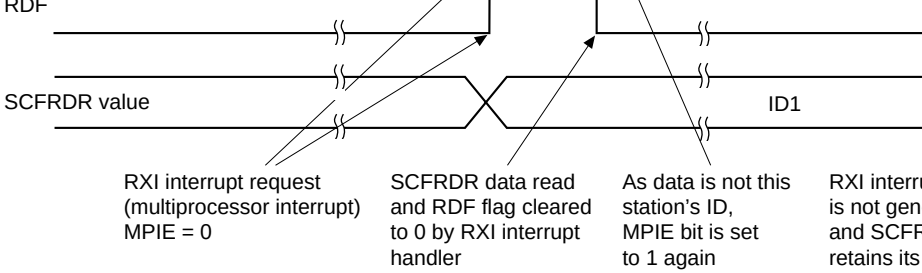
[5] SCIF status check and receiv
read: Read the serial status 1
(SC1SSR) and check that RD
then read receive data from t
receive FIFO data register
(SCFRDR).

Figure 14.14 Sample Multiprocessor Serial Reception Flowchart (1)

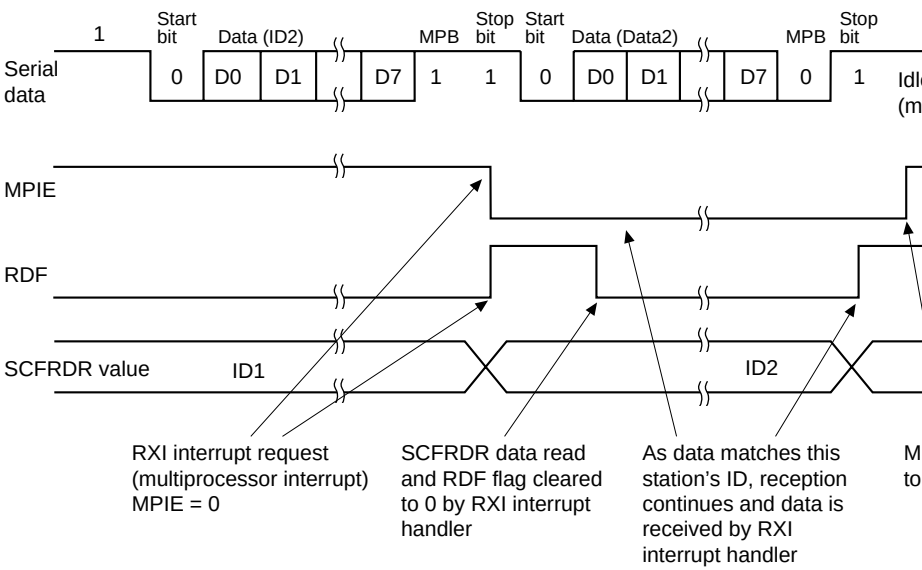


error occurred is stored.
note that the H'00 break
which a framing error occ
stored as the last data in
SCFRDR.

Figure 14.14 Sample Multiprocessor Serial Reception Flowchart (2)



(a) Data does not match station's ID



(b) Data matches station's ID

Figure 14.15 Example of SCIF Receive Operation
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit, LSB-First Tran

Figure 14.16 shows the general format for synchronous serial communication.

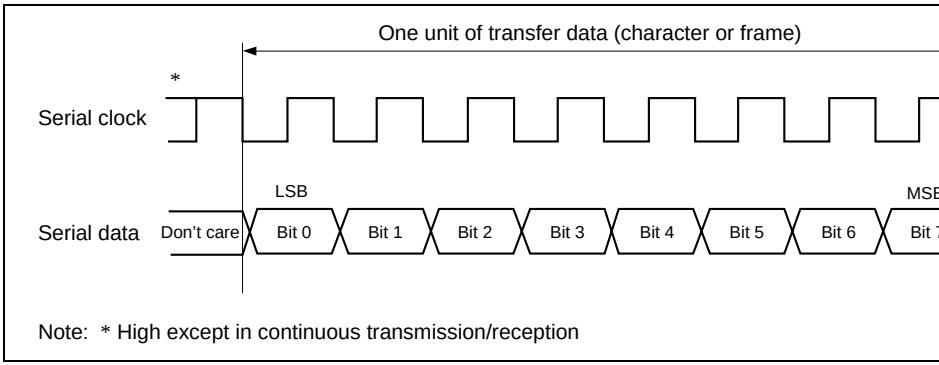


Figure 14.16 Data Format in Synchronous Communication (Example of LSB-First Transfer)

In synchronous serial communication, data on the communication line is output from
the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clo

In serial communication, each character is output starting with the LSB and ending wi
or vice versa, according to the setting of the TLM bit in the serial status 2 register (SC
After the last data is output, the communication line remains in the state of the last dat

In synchronous mode, the SCIF receives data in synchronization with the rising edge
clock.

transmission/reception is performed the clock is fixed high. In receive-only operation, h
the SCIF receives two characters as one unit, and so a 16-pulse serial clock is output. T
single-character receive operations, an external clock should be selected as the clock so

Transmit/Receive Operations

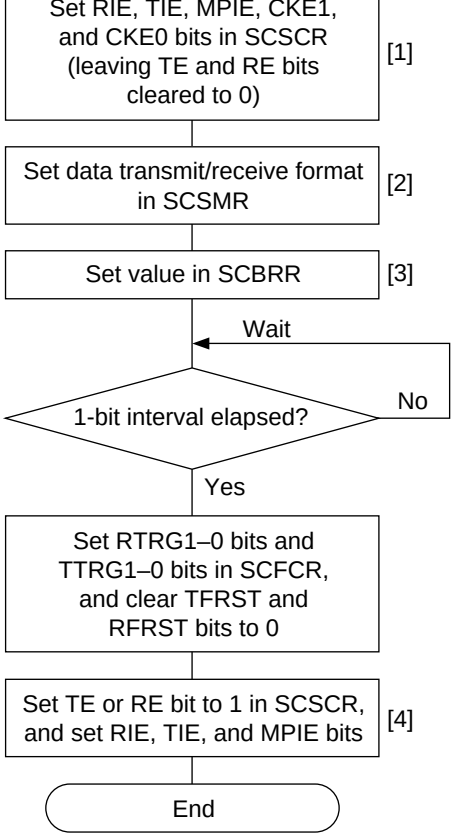
- SCIF Initialization (Synchronous Mode)

  Before transmitting and receiving data, it is necessary to clear the TE and RE bits to
  serial control register (SCSCR), then initialize the SCIF as described below.

  When the operating mode, communication format, etc., is changed, the TE and RE b
  cleared to 0 before making the change using the following procedure. When the TE
  cleared to 0, the TDFE flag is set to 1 and the transmit shift register (SCTSR) is init

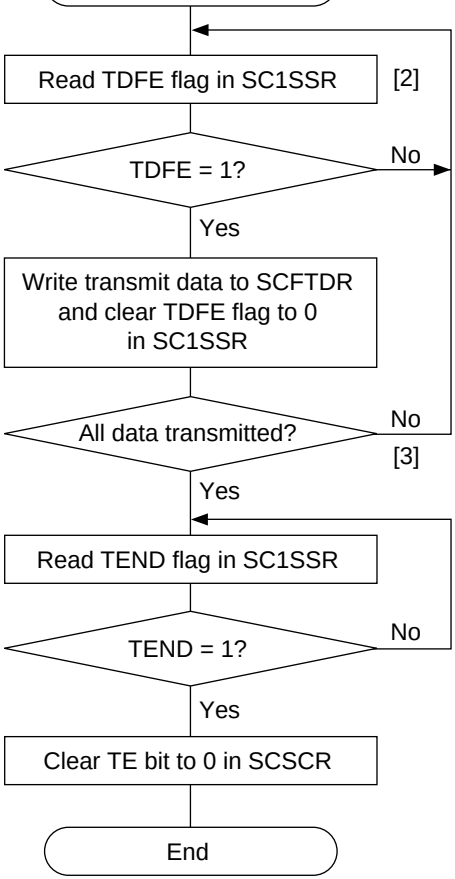
  Note that clearing the RE bit to 0 does not change the contents of the RDF, PER, FE
  ORER flags, or the receive FIFO data register (SCFRDR).

  Figure 14.17 shows a sample SCIF initialization flowchart.



[4] Wait at least one bit interval, set the TE bit or RE bit to 1 in SCSCR.
Also set the RIE, TIE, and MP bits. Setting the TE and RE b simultaneously enables the T and RxD pins to be used.

Figure 14.17 Sample SCIF Initialization Flowchart



write: Read SC1SSR and chec
TDFE =1, then write transmit d
the transmit FIFO data register
(SCFTDR) and clear the TDFE
to 0.

[3] Serial transmission continuatio
procedure: To continue serial
transmission, read 1 from the T
flag to confirm that writing is po
then write data to SCFTDR, an
clear the TDFE flag to 0.

Figure 14.18 Sample Serial Transmission Flowchart

operations are performed continually until there is no transmit data left in SCFTDR number of data bytes in SCFTDR falls to or below the transmit trigger number set control register (SCFCR) during transmission, the TDFE flag is set. If the TIE bit s serial control register (SCSCR) is 1 at this time, a transmit-FIFO-data-empty intern requested.

When clock output mode has been set, the SCIF outputs eight serial clock pulses f of data.

When use of an external clock has been specified, data is output in synchronizatio input clock.

The serial transmit data is sent from the TxD pin starting with the LSB (bit 0) or M according to the setting of the TLM bit in the serial status 2 register (SC2SSR).

3. The SCIF checks for transmit data in SCFTDR at the timing for sending the last bi transmit data in SCFTDR, it is transferred to SCTSR and then serial transmission o frame is started. If there is no transmit data in SCFTDR, the TEND flag is set to 1 status 1 register (SC1SSR), the last bit is sent, and then the transmit data pin (TxD state.

4. After completion of serial transmission, the SCK pin is fixed high.

   Figure 14.19 shows an example of SCIF operation in transmission.

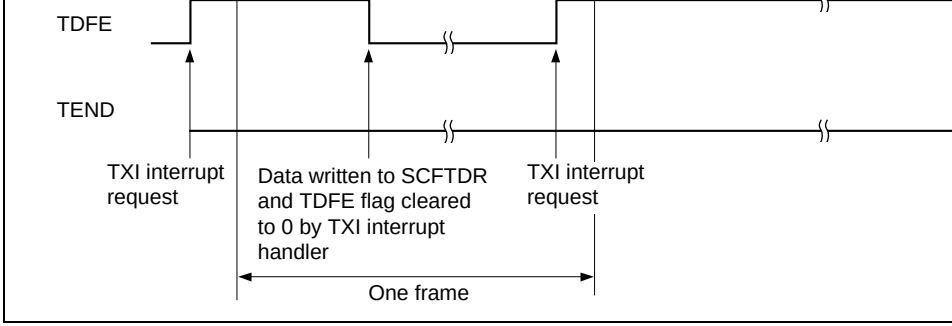


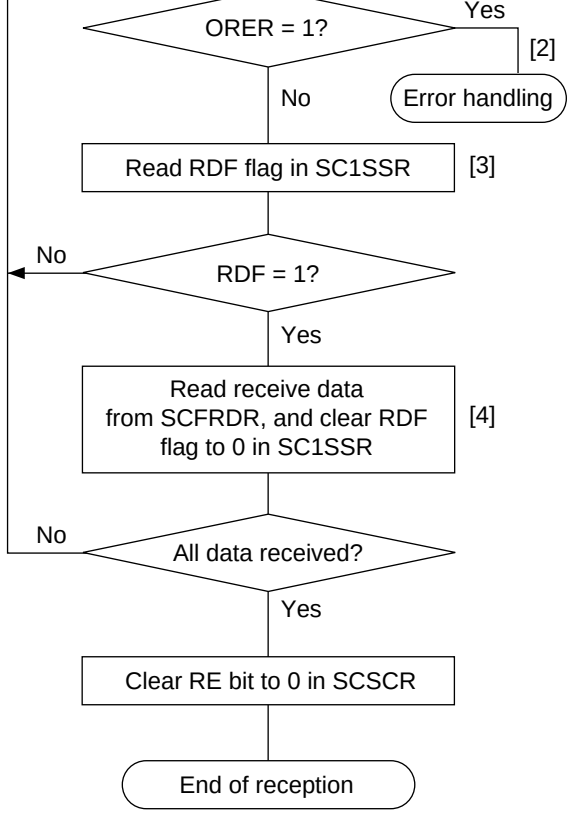
Figure 14.19 Example of SCIF Transmit Operation (Example of LSB-First Tr

- Serial Data Reception (Synchronous Mode)

  Figure 14.20 shows a sample flowchart for serial reception.

  Use the following procedure for serial data reception after enabling the SCIF for rec

  When changing the operating mode from asynchronous to synchronous without rese SCFRDR and SCFTDR by means of SCIF initialization, be sure to check that the O PER3 to PER0, and FER3 to FER0 flags are all cleared to 0. The RDF flag will not any of flags FER3 to FER0 or PER3 to PER0 are set to 1, and neither transmit nor r operations will be possible.



set to 1.

[3] SCIF status check and re
data read: Read the seria
register (SC1SSR) and cl
RDF = 1, then read receiv
from the receive FIFO da
register (SCFRDR) and c
RDF flag to 0. Transition
RDF flag from 0 to 1 can
identified by an RXI intern

[4] Serial reception continuat
procedure: To continue s
reception, read at least th
trigger set number of data
from SCFRDR, and write
RDF flag after reading 1 f
The number of receive da
in SCFRDR can be ascer
reading the lower 8 bits o
FIFO data count register
(SCFDR). (The RDF bit is
automatically when the D
activated by an RXI intern
the SCFRDR value is rea

Figure 14.20 Sample Serial Reception Flowchart (1)

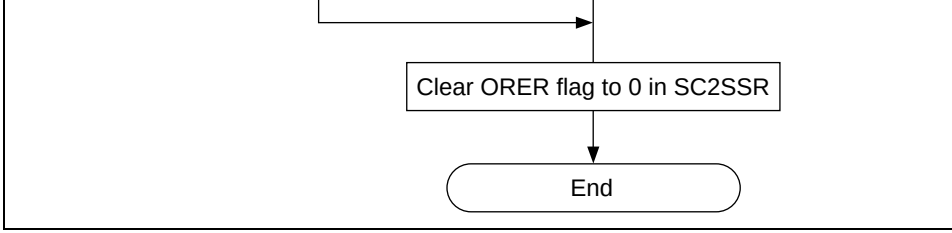


Figure 14.20 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF performs internal initialization in synchronization with serial clock input

2. The received data is stored in the receive shift register (SCRSR) in LSB-to-MSB or MSB-to-LSB order according to the setting of the RLM bit in SC2SSR.

   After reception, the SCIF checks whether the receive data can be transferred from S the receive FIFO data register (SCFRDR). If this check is passed, the receive data is SCFRDR.

   If a receive error is detected in the error check, the operation is as shown in table 14 Neither transmit nor receive operations can be performed subsequently when a rece has been found in the error check.

   Also, as the RDF flag is not set to 1 when receiving, the flag must be cleared to 0.

3. If the RIE bit setting in the serial control register (SCSCR) is 1 when the RDF flag i receive-FIFO-data-full interrupt (RXI) is requested. If the RIE bit setting in SCRSR the ORER flag is set to 1, a receive-error interrupt (ERI) is requested.

   Figure 14.21 shows an example of SCIF operation in reception.

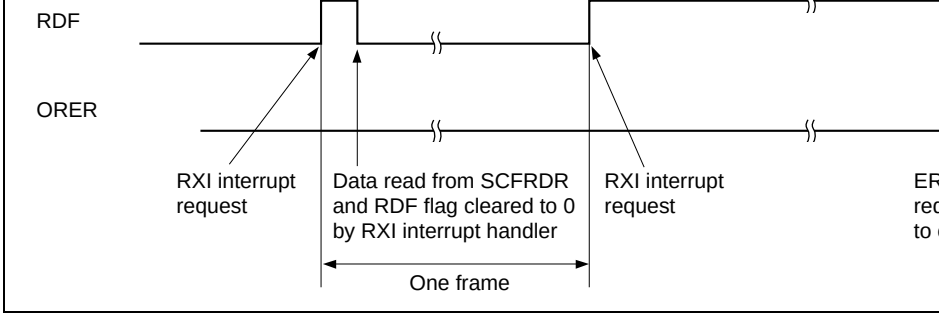
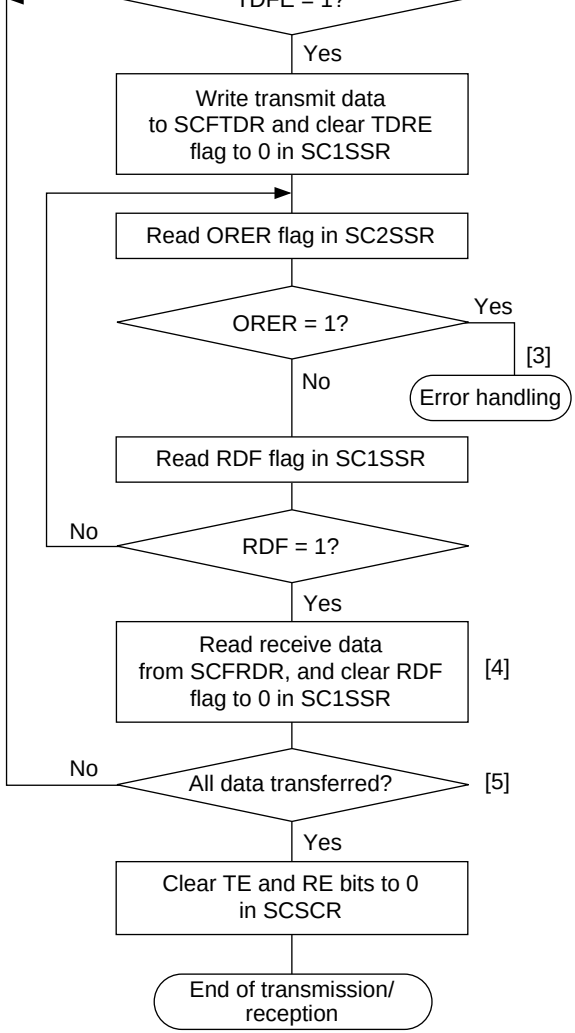


Figure 14.21 Example of SCIF Receive Operation (Example of LSB-First T

- Simultaneous Serial Data Transmission and Reception (Synchronous Mode)

  Figure 14.22 shows a sample flowchart for simultaneous serial transmit and receiv

  Use the following procedure for simultaneous serial data transmit and receive oper enabling the SCIF for transmission and reception.



[3] Receive error handling: If a r
error occurs, read the ORER
SC2SSR, and after performi
appropriate error handling, c
ORER flag to 0.
Transmission/reception cann
resumed if the ORER flag is

[4] SCIF status check and recei
read: Read SC1SSR and ch
the RDF flag is set to 1, ther
receive data from SCFRDR
clear the RDF flag to 0. Tran
the RDF flag from 0 to 1 can
identified by an RXI interrupt

[5] Serial transmission/reception
continuation procedure: To c
serial transmission/reception
reading the RDF flag, readin
SCFRDR, and clearing the R
to 0, before the MSB (bit 7) c
current frame is received. Al
before the MSB (bit 7) of the
frame is transmitted, read 1
TDFE flag to confirm that wr
possible, then write data to S
and clear the TDFE flag to 0

Note: When switching from tran
or receiving to simultaneo
transmitting and receiving
clear the TE bit and RE b
then set the TE bit and R
1 simultaneously.

Figure 14.22 Sample Flowchart for Serial Data Transmission and Recept

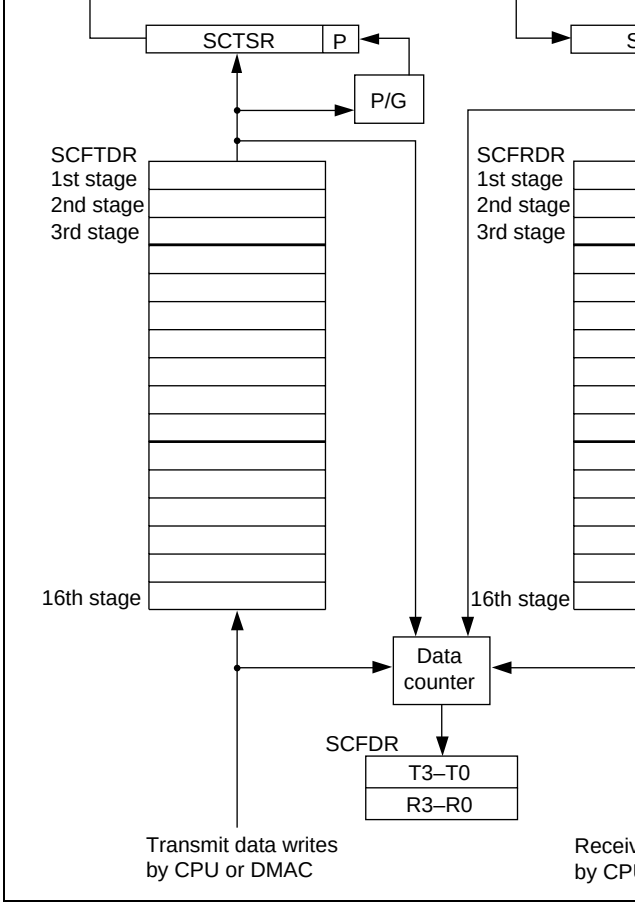


Figure 14.23 Transmit/Receive FIFO Configuration

reading bits T4 to T0 in SCFDR.

A value of H'10 in bits T4 to T0 means that data has been written into all 16 stages of t FIFO. If additional data is written to the FIFO in this state, bits T4 to T0 will not be inc and the written data will be lost.

When the transmit trigger number is set and transmit data is written to the FIFO by the care must be taken not to write data exceeding the number of empty bytes in SCFTDR by the FIFO control register (SCFCR) (see section 14.2.10).

In Serial Data Receive Operations: In reception, serial data input from the RxD pin is captured in the receive shift register (SCRSR) in the order specified by the RLM bit in status 2 register (SC2SSR). A parity bit check is carried out, and if there is a parity erro (parity error) flag for that data is set to 1. A stop bit check is also performed, and if a fr is found the F (framing error) flag for that data is set to 1. The receive FIFO buffer has configuration, with the P and F flags for each 8-bit data unit stored together with that da

- Receive FIFO Control in Normal Operation

  Receive data held in the receive FIFO buffer is read by the CPU or DMAC.

  Each time data is transferred from SCRSR to the receive FIFO, the value in bits R4 SCFDR is incremented, and each time the CPU or DMAC reads receive data from t FIFO, the value in bits R4 to R0 is decremented. The current number of data bytes i receive FIFO can thus be found by reading bits R4 to R0 in SCFDR.

  A value of H'10 in bits R4 to R0 means that receive data has been transferred to all of the receive FIFO. If the next serial receive operation is completed before the CPU DMAC reads data from the receive FIFO, an overrun error will result and the serial be lost. If receive FIFO data is read when the value of bits R4 to R0 is H'00, an und value will be returned.

FER are cleared when data with no parity error or framing error is read from the re

This data is transferred to the receive FIFO even if it contains a parity error or fran
Whether or not the receive operation is to be continued at this point can be specifie
EI bit in SC2SSR. If the EI bit is set to 1, specifying continuation of the receive op
receive data is still transferred sequentially to the receive FIFO after an error occur
of the 16-stage FIFO buffer in which the data with the error is located can be deter
reading bits ED15 to ED0 in the FIFO error register (SCFER).

When the receive trigger number is set and receive data is read from the receive FI
DMAC, care must be taken not to read data exceeding the receive trigger number i
the FIFO control register (SCFCR) (see section 14.2.10).

- Receive FIFO Control by DR Flag

  When a number of data bytes equal to or exceeding the receive trigger number hav
  received, a receive data read request is issued to the CPU or DMAC by means of a
  interrupt (RDF only). However, an RXI interrupt is not requested if all reception h
  completed with fewer than the receive trigger number of data bytes having been re
  this case, the DR flag is set and an ERI interrupt is requested 16 etu after reception
  data is completed. The CPU should therefore read bits R4 to R0 in SCFDR to find
  of data bytes left in the receive FIFO, and read all the data in the FIFO.

Note: With an 8-bit, 1-stop-bit format, one etu is equivalent to 1.6 frames.
etu: Elementary time unit = sec/bit

automatically in this module. When executing communication, therefore, it is necessary the communication speed and have the appropriate speed set in this module by software

Note: In IrDA mode, reception is not possible when the TE bit is set to 1 (enabling communication) in the serial control register (SCSCR). When performing recep TE bit in SCSCR must be cleared to 0.

Transmission: In the case of a serial output signal (UART frame) from the SCIF, the v corrected and the signal is converted to an IR frame serial output signal by the IrDA mo shown in figure 14.24.

When the serial data is 0, if the PSEL bit is 0 in the IrDA mode register (SCIMR) a pul the IR frame bit width is generated and output, and if the PSEL bit is 1 a pulse of 3/16 t width of the bit rate set in bits ICK3 to 0 in the serial mode register (SCSMR) is genera output. When the serial data is 1, a pulse is not output.

An infrared LED is driven by a signal demodulated to a 3/16 width.

Reception: Pulses of 3/16 the received IR frame bit width are converted to UART fram demodulation as shown in figure 14.24.

Demodulation to 0 is executed for pulse output and demodulation to 1 when there is no output.

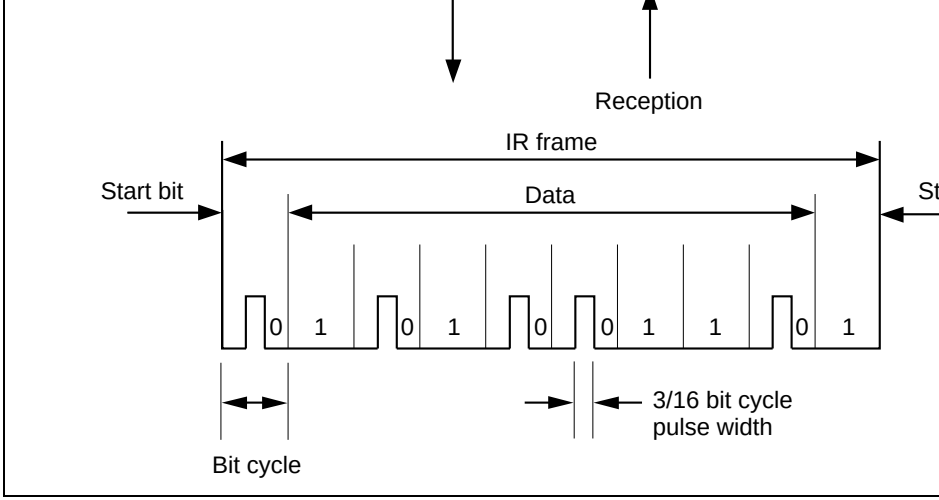


Figure 14.24 IrDA Mode Transmit/Receive Operations

Pulse Width Selection: In transmission, the IR frame pulse width can be selected as e
the transmission bit rate or a smaller pulse width by means of the PSEL bit in the IrDA
register (SCIMR).

The SCIF includes a baud rate generator that generates the transmit frame bit rate and
generator that generates the IRCLK signal for varying the pulse width.

When the PSEL bit is cleared to 0 in SCIMR, a width of 3/16 the bit rate set in the bit
(SCBRR) is output as the IR frame pulse width. As the pulse width is the direct infrar
time; if the user wishes to minimize the pulse width in order to reduce power consump
PSEL bit should be set to 1 in SCIMR and a setting should also be made in bits ICK3
the serial mode register (SCSMR) to generate the IRCLK signal, resulting in output w
minimum settable pulse width.

For example, when Pϕ = 20 MHz, N = 10.

Table 14.12 shows the settings of bits ICK3 to ICK0 that can be used to obtain the mini
width for various operating frequencies.

Table 14.12 Bits ICK3 to ICK0 and Operating Frequencies in IrDA mode (When PSEL = 1)

Operating Frequency Pϕ (MHz)	Setting of Bits ICK3 to ICK0 in SCSMR			
	ICK3	ICK2	ICK1	ICK0
2	0	0	0	0
3				1
5			1	0
6				1
8		1	0	0
10				1
12			1	0
14				1
16	1	0	0	0
18				1
20			1	0
21				1
22				1
23		1	0	0
24				1
25				1
26			1	0
27				0
28				1

When the TDFE flag is set to 1 in the serial status 1 register (SC1SSR), a TXI interrup
requested. A TXI interrupt request can activate the on-chip DMAC to perform data tra
TDFE bit is cleared to 0 automatically when all writes to the transmit FIFO data regist
(SCFTDR) by the DMAC are completed.

When the RDF flag is set to 1 in SC1SSR, an RXI interrupt is requested. An RXI inter
can activate the on-chip DMAC to perform data transfer. The RDF bit is cleared to 0 a
when all receive FIFO data register (SCFRDR) reads by the DMAC are completed.

When the ER flag is set to 1, an ERI interrupt is requested. The on-chip DMAC canno
activated by an ERI interrupt request.

When the BRK flag is set to 1, a BRI interrupt is requested. The on-chip DMAC cann
activated by a BRI interrupt request.

A TXI interrupt indicates that transmit data can be written, and an RXI interrupt indic
there is receive data in SCFRDR.

Table 14.13 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Pri Res
ERI	Receive error (ER)	Not possible	Hig
RXI	Receive data full (RDF) or data ready (DR)	Possible (RDF only)	↕
BRI	Break (BRK)	Not possible	
TXI	Transmit data FIFO empty (TDFE)	Possible	Low

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 the upper 8 bits of th
data count register (SCFDR).

Simultaneous Multiple Receive Errors: If a number of receive errors occur at the sam
state of the status flags in SC1SSR and SC2SSR is as shown in table 14.14. If there is a
error, data is not transferred from the receive shift register (SCRSR) to the receive FIFO
register (SCFRDR), and the receive data is lost.

Table 14.14 SC1SSR/SC2SSR Status Flags and Transfer of Receive Data

	SC1SSR/SC2SSR Status Flags				Receive Data
Receive Errors	**RDF**	**ORER**	**FER**	**PER**	**SCRSR → S**
Overrun error	1	1	0	0	×
Framing error	0	0	1	0	O
Parity error	0	0	0	1	O
Overrun error + framing error	1	1	1	0	×
Overrun error + parity error	1	1	0	1	×
Framing error + parity error	0	0	1	1	O
Overrun error + framing error + parity error	1	1	1	1	×

Note: O: Receive data is transferred from SCRSR to SCFRDR.
×: Receive data is not transferred from SCRSR to SCFRDR.

are determined by the I/O port data register (DR) and the control register (CR) of the p
controller (PFC). This fact can be used to send a break signal.

The DR value substitutes for the mark state until the PFC setting is made. The initial s
should therefore be as an output port outputting 1.

To send a break signal during serial transmission, clear DR, then set the TxD pin as an
with the PFC.

When the TE bit is cleared to 0, the transmitter is initialized regardless of the current t
state.

Receive Error Flags and Transmit Operations (Synchronous Mode Only): Transm
cannot be started when any of the receive error flags (ORER, PER3 to PER0, FER3 to
set to 1, even if the TDFE flag is set to 1. Be sure to clear the receive error flags to 0 b
starting transmission.

Note also that the receive error flags are not cleared to 0 by clearing the RE bit to 0.

Receive Data Sampling Timing and Receive Margin in Asynchronous Mode: In a
mode, the SCIF operates on a base clock with a frequency of 16, 8, or 4 times the tran

In reception, the SCIF synchronizes internally with the falling edge of the start bit, wh
samples on the base clock. Receive data is latched at the rising edge of the eighth, fou
second base clock pulse. The timing is shown in figure 14.25.

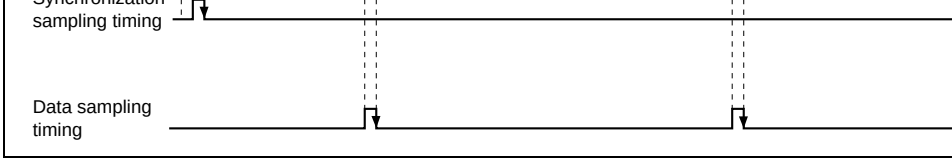


Figure 14.25 Receive Data Sampling Timing in Asynchronous Mode (Using base clock with frequency of 16 times the transfer rate, sampled in 8th cl

The receive margin in asynchronous mode can therefore be expressed as shown in equa

$$M = \left|\left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\% \quad \text{.....}$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16, 8, or 4)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equ

When D = 0.5, F = 0, and N = 16:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\% \text{.....}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 3

When Using Synchronous External Clock Mode

- Do not set TE or RE to 1 until at least 4 peripheral operating clock cycles after exte SCK has changed from 0 to 1.
- Only set both TE and RE to 1 when external clock SCK is 1.
- In reception, note that if RE is cleared to 0 from 2.3 to 3.5 peripheral operating cloc after the rising edge of the RxD D7 bit SCK input, RDF will be set to 1 but copying SCFRDR will not be possible.

When performing SCFRDR reads by the DMAC, be sure to set the relevant SCIF rece
data-full interrupt (RXI) as an activation source.

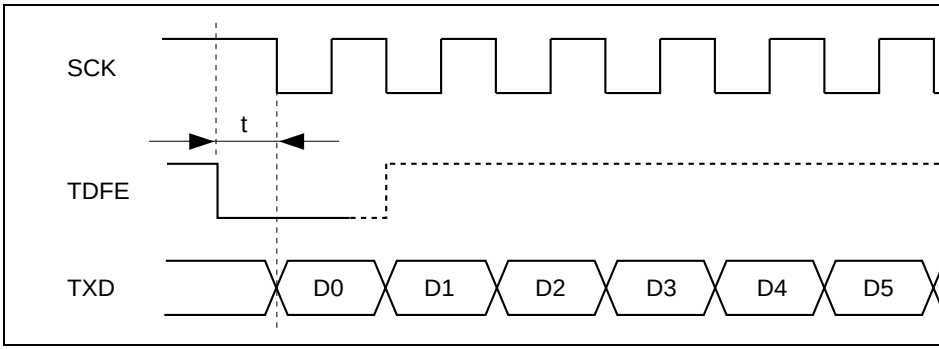


Figure 14.26 Example of Synchronous Transmission by DMAC

SCFRDR Reading and the RDF Flag: The RDF flag in the serial status 1 register (S
set when the number of receive data bytes in the receive FIFO data register (SCFRDR
become equal to or greater than the receive trigger number set by bits RTRG1 and RT
FIFO control register (SCFCR). After RDF is set, receive data equivalent to the trigge
can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes in SCFRDR is equal to or greater than the trigge
the RDF flag will be set to 1 again if it is cleared to 0. RDF should therefore be cleare
being read as 1 after receive data has been read to reduce the number of data bytes in S
less than the trigger number.

The number of receive data bytes in SCFRDR can be found from the lower 8 bits of th
count register (SCFDR).

SCIF Initialization Flowchart and Receive-FIFO-Data-Full Interrupt (RXI) Requ

Phenomenon:

When the SCIF function is used and the operation in the SCIF initialization flowchart e figure 14.4 is executed two or more times consecutively, the SCIF receive FIFO data fu (RXI) request may be set in the second or later initialization operations, even if there is received data.

Condition:

Figure 14.27 shows an example of SCIF initialization flowchart with 2nd initialization. request may be set at the trigger (RTRG1, RTRG0) setting [2] of 2nd initialization whe reset the value of the Receive FIFO Data Number Trigger (RTRG1, RTRG0) setting [1 initialization.

Countermeasures:

Please apply any of the following countermeasures, if the write-access occurs at the Re Data Number Trigger setting [2] of 2nd initialization.

(1) Read out SCFCR and write the same value with the Receive FIFO Data Number Tr (RTRG1, RTRG0).

(2) Set Receive Interrupt Enable (RIE) bit to “0” in SCSCR before changing the value o Receive FIFO Data Number Trigger (RTRG1, RTRG0). Mask the RXI request. Aft the SCFCR, clear the interrupt request to Receive Data Register Full (RDF). Set Re Interrupt Enable (RIE) bit to “1” in SCSCR to terminate the mask-setting.

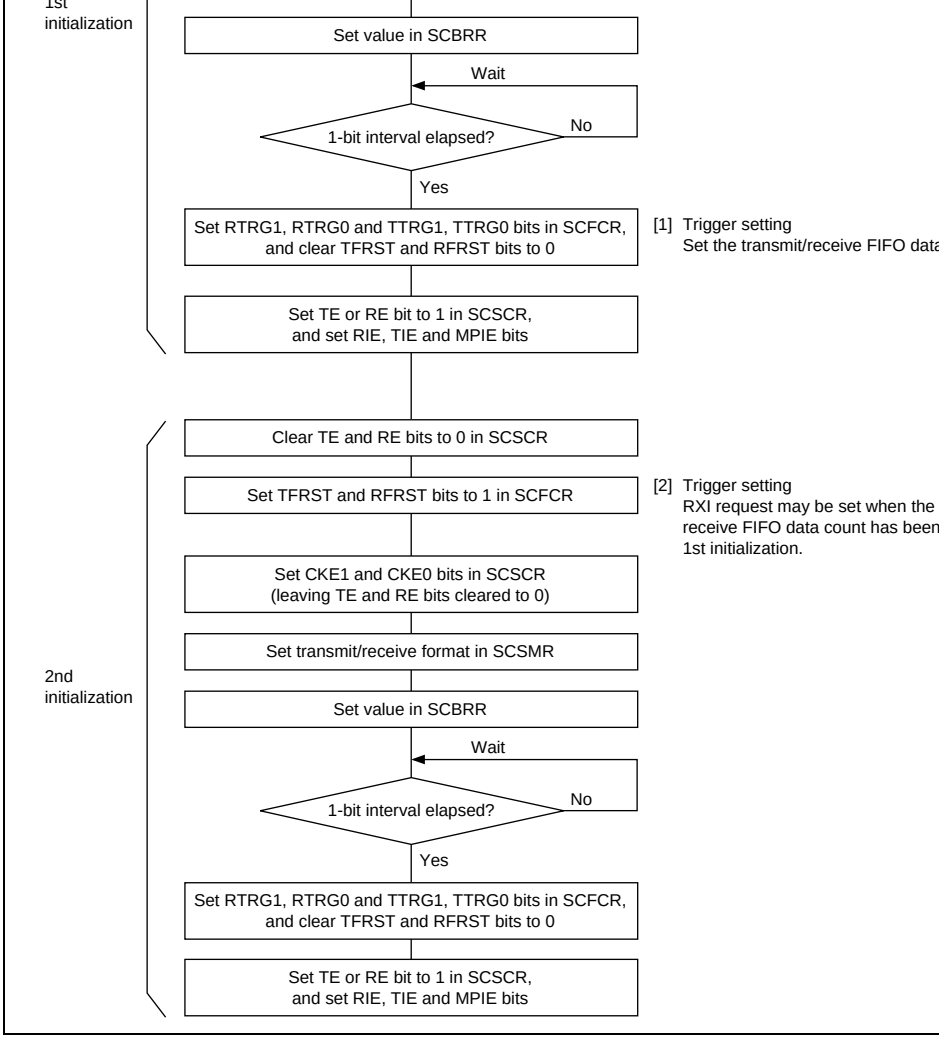
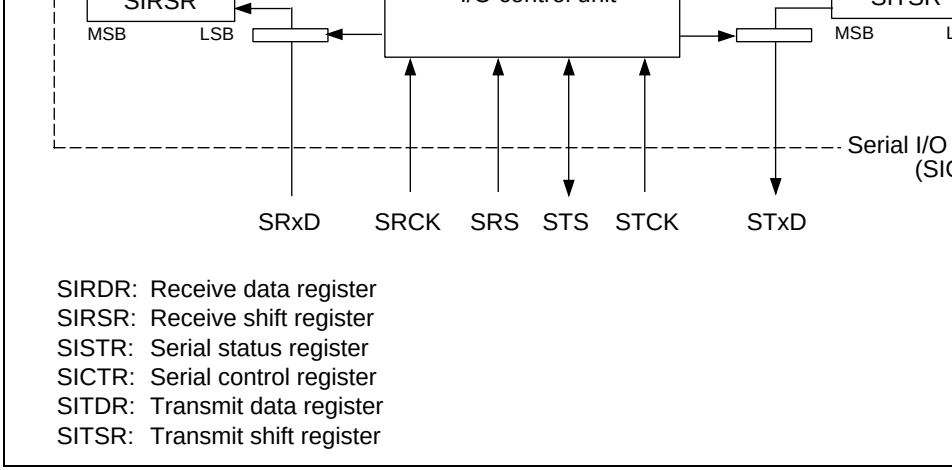


Figure 14.27 Example of SCIF Initialization Flowchart

The serial I/O has the following features:

- Full-duplex operation
  Independent transmit/receive registers and independent transmit/receive clocks
- Double-buffered transmit/receive ports
  Continuous data transmission/reception possible
- Interval transfer mode and continuous transfer mode
- Memory-mapped receive register, transmit register, control register, and status reg
  With the exception of SIRSR and SITSR, these registers are memory-mapped and accessed by a MOV instruction.
- Choice of 8- or 16-bit data length
- Data transfer communication by means of polling or interrupts
  Data transfer can be monitored by polling the receive data register full flag (RDRF transmit data register empty flag (TDRE) in the serial status register. Interrupt requ generated during data transfer by setting the receive interrupt request flag and tran interrupt request flag.
- MSB-first transfer between SIO and data I/O

Figure 15.1 shows a block diagram of the serial I/O.



SIRDR: Receive data register
SIRSR: Receive shift register
SISTR: Serial status register
SICTR: Serial control register
SITDR: Transmit data register
SITSR: Transmit shift register

Figure 15.1 SIO Block Diagram

	input pin			synchronization
	Serial transmit data output pin	STxD0	Output	Serial data outp
	Serial transmit clock input pin	STCK0	Input	Serial transmit c
	Serial transmission synchronization input/output pin	STS0	I/O	Serial transmiss synchronization port 0
1	Serial receive data input pin	SRxD1	Input	Serial data input
	Serial receive clock input pin	SRCK1	Input	Serial receive cl
	Serial reception synchronization input pin	SRS1	Input	Serial reception synchronization
	Serial transmit data output pin	STxD1	Output	Serial data outp
	Serial transmit clock input pin	STCK1	Input	Serial transmit c
	Serial transmission synchronization input/output pin	STS1	I/O	Serial transmiss nization input/ou
2	Serial receive data input pin	SRxD2	Input	Serial data input
	Serial receive clock input pin	SRCK2	Input	Serial receive cl
	Serial reception synchronization input pin	SRS2	Input	Serial reception synchronization
	Serial transmit data output pin	STxD2	Output	Serial data outp
	Serial transmit clock input pin	STCK2	Input	Serial transmit c
	Serial transmission synchronization input/output pin	STS2	I/O	Serial transmiss synchronization port 2

Note: In a reset, all pins are initialized to the high-impedance state.

	Receive data register 0	SIRDR0	R	H'0000	H'FFFFFC00	8,
	Transmit shift register 0	SITSR0	—	—	—	—
	Transmit data register 0	SITDR0	R/W	H'0000	H'FFFFFC02	8,
	Serial control register 0	SICTR0	R/W	H'0000	H'FFFFFC04	8,
	Serial status register 0	SISTR0	R/(W)*	H'0002	H'FFFFFC06	8,
1	Receive shift register 1	SIRSR1	—	—	—	—
	Receive data register 1	SIRDR1	R	H'0000	H'FFFFFC10	8,
	Transmit shift register 1	SITSR1	—	—	—	—
	Transmit data register 1	SITDR1	R/W	H'0000	H'FFFFFC12	8,
	Serial control register 1	SICTR1	R/W	H'0000	H'FFFFFC14	8,
	Serial status register 1	SISTR1	R/(W)*	H'0002	H'FFFFFC16	8,
2	Receive shift register 2	SIRSR2	—	—	—	—
	Receive data register 2	SIRDR2	R	H'0000	H'FFFFFC20	8,
	Transmit shift register 2	SITSR2	—	—	—	—
	Transmit data register 2	SITDR2	R/W	H'0000	H'FFFFFC22	8,
	Serial control register 2	SICTR2	R/W	H'0000	H'FFFFFC24	8,
	Serial status register 2	SISTR2	R/(W)*	H'0002	H'FFFFFC26	8,

Note: * Only 0 should be written, to clear flags (after reading 1 from the flag).

SRxD pin in synchronization with the fall of the serial receive clock (SRCK), and is si
SIRSR. The data length is set by the transmit/receive data length select bit (DL) in the
corresponding serial control register (SICTR). When data transfer to SIRSR is comple
contents are automatically transferred to the receive data register (SIRDR), and the rec
register full flag (RDRF) is set in the serial status register (SISTR).

If the next data word input operation ends before the RDRF flag is cleared, an overrun
occurs, the receive overrun error flag (RERR) is set in SISTR, and an overrun error sig
to the interrupt controller (INTC). The data in SIRSR overwrites the data in SIRDR.

15.2.2 Receive Data Register (SIRDR)

Bit:	15	14	13	...	3	2	1
				...			
Initial value:	0	0	0	...	0	0	0
R/W:	R	R	R	...	R	R	R

SIRDR is a 16-bit register that stores serial receive data. When data is transferred from
SIRDR, the receive data register full flag (RDRF) is set in the serial status register (SI
receive interrupt enable flag (RIE) is set in SICTR, a receive-data-full interrupt (RDFI
sent to the interrupt controller (INTC) and the DMA controller (DMAC). When the fla
this interrupt request signal is not generated. When SIRDR is read by the DMAC, the
is cleared automatically. SIRDR is initialized to H'0000 by a reset.

MSB-first order in synchronization with the rising edge of the serial transmit clock (ST
output from the STxD pin. The transfer data length is set by the transmit/receive data le
bit (DL) in the serial control register (SICTR). When the DL bit is cleared to 0 (8-bit da
the lower 8 bits of SITDR are output. When the serial transmission synchronization sig
goes high, or the last data transmission ends without the synchronization enable (SE) bi
in SICTR, the contents of the transmit data register (SITDR) are transferred to SITSR, a
TDRE is 0, TDRE is then set. If output of the next data begins before TDRE is cleared,
error occurs, the transmit overrun error flag (TERR) is set in SISTR, and a transmit ove
interrupt request is sent to the INTC.

15.2.4 Transmit Data Register (SITDR)

Bit:	15	14	13	...	3	2	1
				...			
Initial value:	0	0	0	...	0	0	0
R/W:	R/W	R/W	R/W	...	R/W	R/W	R/W

SITDR is a 16-bit register that stores serial transmit data. Data should be written to SIT
the transmit data register empty flag (TDRE) is set to 1 in SISTR. If data is written to S
when TDRE is 0, the previous data will be overwritten. When STS goes high or data ou
transmit shift register SITSR ends with the SE bit cleared to 0 in SICTR, the data in SIT
automatically transferred to SITSR, and if TDRE is 0, TDRE is then set. If the transmit
enable flag (TIE) is set, a transmit-data-empty interrupt (TDEI) request is sent to the IN
DMAC. When TIE is cleared, this interrupt request is not generated. When the DMAC
SITDR, the TDRE flag is cleared automatically. The TDRE flag is set only by hardwar
initialized to H'0000 by a reset.

	—	TM	SE	DL	TIE	RIE	TE
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

SICTR is a 16-bit register used to set parameters for serial port control. SICTR is initi
H'0000 by a reset.

When modifying bit 4, 5, or 6 (TM, SE, or DL), TE and RE should be cleared to 0 bef

Bits 15 to 7—Reserved: These bits are always read as 0. The write value should alway

Bit 6—Transfer Mode Control (TM): Specifies whether the transmission synchronizat
to be input from an external source or generated internally by the chip. When this flag
the transmission synchronization signal is STS pin input. When this flag is set, the tra
synchronization signal is generated by the chip, and is output to an external device fro
pin. This bit does not affect reception.

Bit 6: TM	Description
0	External signal input from STS pin is used as transmission start i (
1	Internal signal output from STS pin is used as transmission start i

Bit 5—Synchronization Signal Enable (SE): Specifies whether the synchronization sig
be used for all serial data transfers, or only for the first transfer.

When this bit is cleared to 0, the synchronization signals (SRS and STS) are necessary
first data transfer, and are not required for subsequent transfers. When this bit is set to
synchronization signals are necessary for all data transfers.

Bit 5: SE	Description
0	Continuous mode: SRS and STS are used only for the first data t (
1	Interval mode: SRS and STS are used for all data transfers

Bit 3—Transmit Interrupt Enable (TIE): Enables the transmit-data-empty interrupt. The
value of this bit is 0.

Bit 3: TIE	Description
0	Transmit interrupt disabled (Ir
1	Transmit interrupt enabled

Bit 2—Receive Interrupt Enable (RIE): Enables the receive-data-full interrupt. The init
this bit is 0.

Bit 2: RIE	Description
0	Receive interrupt disabled (Ir
1	Receive interrupt enabled

Bit 1—Transmit Enable (TE): Enables data transmission. When this flag is cleared, the
STCK, and STS pins go to the high-impedance state.

Bit 1: TE	Description
0	Transmission disabled: STxD, STCK, and STS pins go to high-imp state (Ir
1	Transmission enabled

Bit 0—Receive Enable (RE): Enables data reception. When this flag is cleared, the SRx
and SRS pins go to the high-impedance state.

Bit 0: RE	Description
0	Reception disabled: SRxD, SRCK, and SRS pins go to high-imped (Ir
1	Reception enabled

SISTR is a 16-bit register that indicates the status of the serial I/O module. SISTR is i
H'0002 by a reset.

Bits 15 to 4—Reserved: These bits are always read as 0. The write value should alway

Bit 3—Transmit Underrun Error (TERR): Flag that indicates the occurrence of a trans
underrun.

Bit 3: TERR	Description
0	Transmission is in progress, or has ended normally ([Clearing conditions] • When 0 is written to the TERR bit after reading TERR = 1 • When the processor enters the reset state
1	A transmit underrun error has occurred TERR is set to 1 if data transmission is started while TDRE = 1

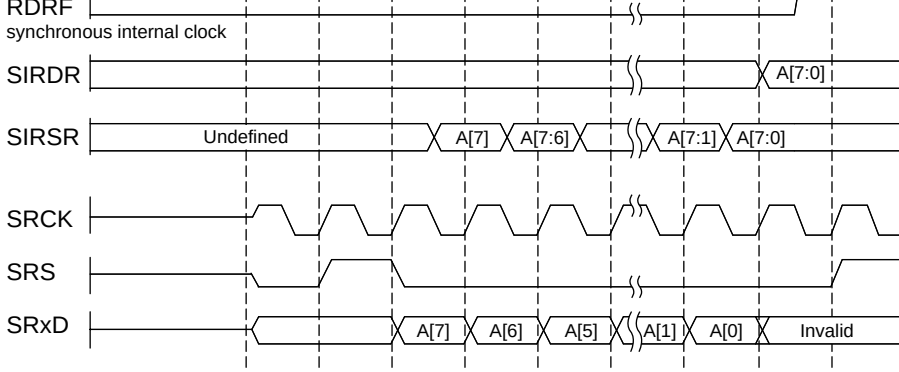
Bit 2—Receive Overrun Error (RERR): Flag that indicates the occurrence of a receive

Bit 2: RERR	Description
0	Reception is in progress, or has ended normally ([Clearing conditions] • When 0 is written to the RERR bit after reading RERR = 1 • When the processor enters the reset state
1	A receive overrun error has occurred RERR is set to 1 if data reception ends while RDRE = 1

1	SITDR transmit data is invalid (In TDRE is set to 1 in the following cases: • When data is transferred from SITDR to SITSR • When the TE bit is cleared to 0 in the serial control register (SI • When the processor enters the reset state

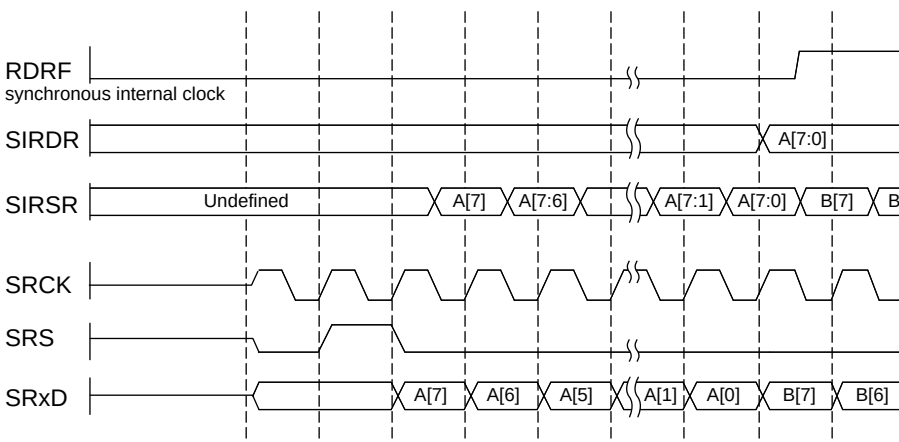
Bit 0—Receive Data Register Full (RDRF): Flag that indicates that SIRDR receive data

Bit 0: RDRF	Description
0	SIRDR receive data is invalid (In [Clearing conditions] • When the DMAC reads data from SIRDR • When 1 is read from RDRF and 0 is written • When the RE bit is cleared to 0 in the serial control register (SI • When the processor enters the reset state
1	SIRDR receive data is valid RDRF is set to 1 when serial data reception ends normally and the transferred from SIRSR to SIRDR



Note: DL = 0: 8-bit data transfer
SE = 1: Synchronous transfer in start signal mode

Figure 15.2 Reception: Interval Transfer Mode



Note: DL = 0: 8-bit data transfer
SE = 0: Asynchronous transfer, no start signal mode

Figure 15.3 Reception: Continuous Transfer Mode

Figure 15.7 shows continuous transfer mode (SE cleared to 0 in SICTR) when TM is se
SICTR.

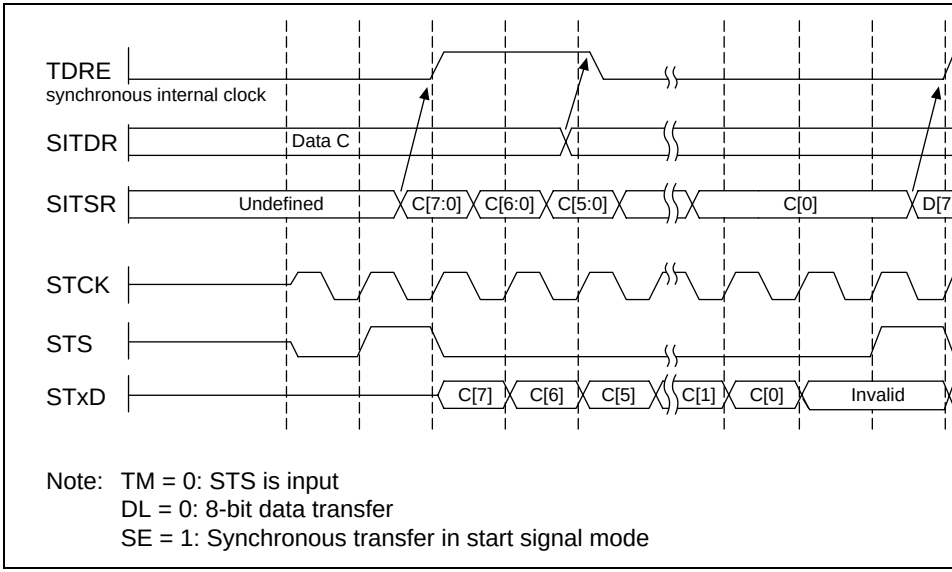
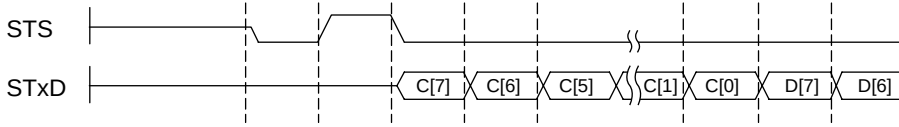
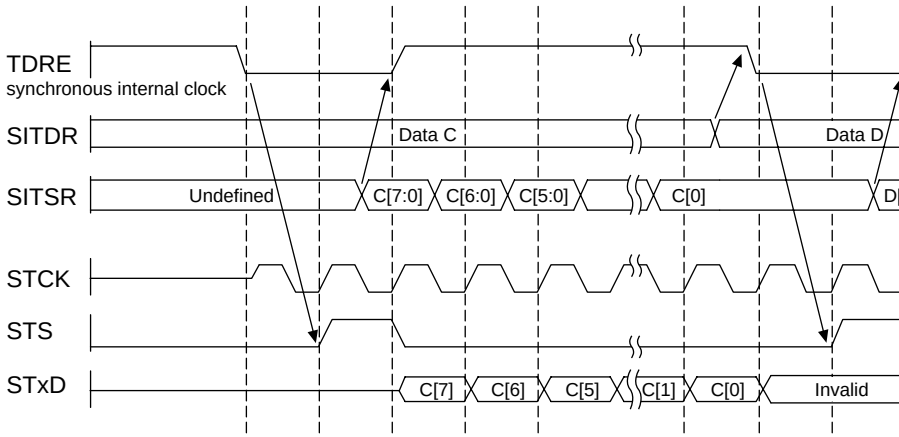


Figure 15.4 Transmission: Interval Transfer Mode (TM = 0 Mode)



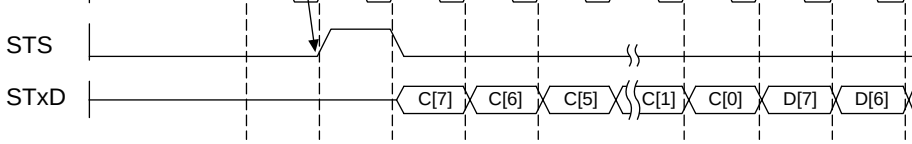
Note: TM = 0: STS is input
DL = 0: 8-bit data transfer
SE = 0: Asynchronous transfer, no start signal mode

Figure 15.5 Transmission: Continuous Transfer Mode (TM = 0 Mod



Note: TM = 1: STS is output
DL = 0: 8-bit data transfer
SE = 1: Synchronous transfer in start signal mode

Figure 15.6 Transmission: Interval Transfer Mode (TM = 1 Mode)



Note: TM = 1: STS is output
DL = 0: 8-bit data transfer
SE = 0: Asynchronous transfer, no start signal mode

Figure 15.7 Transmission: Continuous Transfer Mode (TM = 1 Mode)

the DMA controller (DMAC) to read the data in SIRDR. RDRF is cleared to 0 automa
the DMAC reads data from SIRDR.

A TDEI interrupt request is generated when the TDRE bit is set to 1 in SISTR. TDEI
the DMAC to write the next data to SITDR. TDRE is cleared to 0 automatically when
writes data to SITDR.

When TDEI and RDFI interrupt requests are handled by the DMAC, and not by the in
controller, a low priority level should be given to interrupts from the SIO to prevent th
controller from operating.

When the RERR bit is set to 1 in SISTR, an RERI interrupt request is generated.

When the TERR bit is set to 1 in SISTR, a TERI interrupt request is generated.

Channel interrupt priority levels are set by means of the IRPE register, as described in
Interrupt Controller (INTC).

Table 15.3 SIO Interrupt Sources

Interrupt Source	Description	DMAC Activation
RERI	Receive overrun error (RERR)	Not possible
TERI	Transmit underrun error (TERR)	Not possible
RDFI	Receive data register full (RDRF)	Possible
TDEI	Transmit data register empty (TDRE)	Possible

The TPU has the following features:

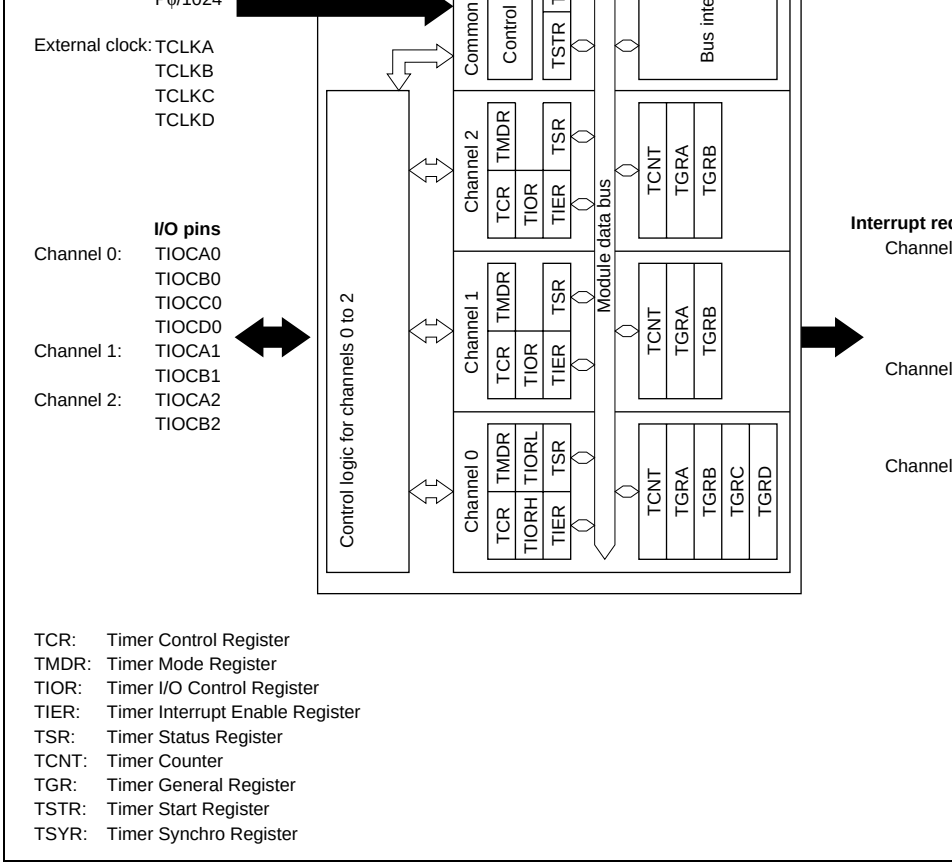
- Maximum 8-pulse input/output
- A total of eight timer general registers (TGRs) are provided (four for channel 0 an
  for channels 1, and 2).
  - Each register can be set independently as an output compare/input capture regi
  - TGRC and TGRD for channel 0 can be used as buffer registers
- Choice of seven or eight counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output by compare match: Selection of 0, 1, or toggle output
  - Input capture function: Choice of rising edge, falling edge, or both edge detect
  - Counter clear operation: Counter clearing possible by compare match or input
  - Synchronous operation: Multiple timer counters (TCNT) can be written to sim
    simultaneous clearing by compare match and input capture possible register si
    input/output possible by counter synchronous operation
  - PWM mode: Any PWM output duty can be set maximum of 7-phase PWM ou
    by combination with synchronous operation
- Buffer operation settable for channel 0
  - Input capture register double-buffering possible
  - Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 1, and 2
  - Two-phase encoder pulse up/down-count possible
- Fast access via internal 16-bit bus
  - Fast access is possible via a 16-bit bus interface
- 13 interrupt sources
  - For channel 0 four compare match/input capture dual-function interrupts and o
    interrupt can be requested independently
  - For channels 1, and 2, two compare match/input capture dual-function interrup
    overflow interrupt, and one underflow interrupt can be requested independentl

Count clock		Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 TCLKA TCLKB TCLKC TCLKD	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/1024 TCLKA TCLKB TCLKC
General registers		TGR0A TGR0B	TGR1A TGR1B	TGR2A TGR2B
General registers/ buffer registers		TGR0C TGR0D	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare or input captu
Compare match output	0 output	O	O	O
	1 output	O	O	O
	Toggle output	O	O	O
Input capture function		O	O	O
Synchronous operation		O	O	O
PWM mode		O	O	O
Phase counting mode		—	O	O
Buffer operation		O	—	—

Notes: O : Possible
— : Not possible

	• Compare match or input capture 0C • Compare match or input capture 0D • Overflow	• Overflow • Underflow	• Overflow • Underflo

Note: — : Not possible



TCR: Timer Control Register
TMDR: Timer Mode Register
TIOR: Timer I/O Control Register
TIER: Timer Interrupt Enable Register
TSR: Timer Status Register
TCNT: Timer Counter
TGR: Timer General Register
TSTR: Timer Start Register
TSYR: Timer Synchro Register

Figure 16.1 TPU Block Diagram

	Clock input B	TCLKB	Input	External clock B input pi (Channel 1 phase counti phase input)
	Clock input C	TCLKC	Input	External clock C input pi (Channel 2 phase counti phase input)
	Clock input D	TCLKD	Input	External clock D input pi (Channel 2 phase counti phase input)
0	Input capture/output compare match A0	TIOCA0	I/O	TGR0A input capture inp compare output/PWM ou
	Input capture/output compare match B0	TIOCB0	I/O	TGR0B input capture inp compare output/PWM ou
	Input capture/output compare match C0	TIOCC0	I/O	TGR0C input capture inp compare output/PWM ou
	Input capture/output compare match D0	TIOCD0	I/O	TGR0D input capture inp compare output/PWM ou
1	Input capture/output compare match A1	TIOCA1	I/O	TGR1A input capture inp compare output/PWM ou
	Input capture/output compare match B1	TIOCB1	I/O	TGR1B input capture inp compare output/PWM ou
2	Input capture/output compare match A2	TIOCA2	I/O	TGR2A input capture inp compare output/PWM ou
	Input capture/output compare match B2	TIOCB2	I/O	TGR2B input capture inp compare output/PWM ou

	Timer mode register 0	TMDR0	R/W	H'C0	H'FFFFFC51	8
	Timer I/O control register 0H	TIOR0H	R/W	H'00	H'FFFFFC52	8
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H'FFFFFC53	8
	Timer interrupt enable register 0	TIER0	R/W	H'40	H'FFFFFC54	8
	Timer status register 0	TSR0	R/(W)*	H'C0	H'FFFFFC55	8
	Timer counter 0	TCNT0	R/W	H'0000	H'FFFFFC56	1
	Timer general register 0A	TGR0A	R/W	H'FFFF	H'FFFFFC58	1
	Timer general register 0B	TGR0B	R/W	H'FFFF	H'FFFFFC5A	1
	Timer general register 0C	TGR0C	R/W	H'FFFF	H'FFFFFC5C	1
	Timer general register 0D	TGR0D	R/W	H'FFFF	H'FFFFFC5E	1
1	Timer control register 1	TCR1	R/W	H'00	H'FFFFFC60	8
	Timer mode register 1	TMDR1	R/W	H'C0	H'FFFFFC61	8
	Timer I/O control register 1	TIOR1	R/W	H'00	H'FFFFFC62	8
	Timer interrupt enable register 1	TIER1	R/W	H'40	H'FFFFFC64	8
	Timer status register 1	TSR1	R/(W)*	H'C0	H'FFFFFC65	8
	Timer counter 1	TCNT1	R/W	H'0000	H'FFFFFC66	1
	Timer general register 1A	TGR1A	R/W	H'FFFF	H'FFFFFC68	1
	Timer general register 1B	TGR1B	R/W	H'FFFF	H'FFFFFC6A	1

	Timer status register 2	TSR2	R/(W)*	H'C0	H'FFFFFC76
	Timer counter 2	TCNT2	R/W	H'0000	H'FFFFFC76
	Timer general register 2A	TGR2A	R/W	H'FFFF	H'FFFFFC78
	Timer general register 2B	TGR2B	R/W	H'FFFF	H'FFFFFC7A
All	Timer start register	TSTR	R/W	H'00	H'FFFFFC40
	Timer synchro register	TSYR	R/W	H'00	H'FFFFFC41

Note: * Only 0 can be written, to clear the flags.

16.2 Register Descriptions

16.2.1 Timer Control Register (TCR)

Channel 0: TCR0

Bit:	7	6	5	4	3	2	1
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 1: TCR1

Channel 2: TCR2

Bit:	7	6	5	4	3	2	1
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC
Initial value:	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

The TCR registers are 8-bit registers that control the TCNT channels. The TPU has th
registers, one for each of channels 0 to 2. The TCR registers are initialized to H'00 by

TCNT operation should be stopped when making TCR settings.

				capture
			1	TCNT cleared by counter clearing for channel performing synchronous clearing/synchronous operation *1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare ma capture *2
		1	0	TCNT cleared by TGRD compare ma capture *2
			1	TCNT cleared by counter clearing for channel performing synchronous clearing/synchronous operation *1

Channel	Bit 7: Reserved*3	Bit 6: CCLR1	Bit 5: CCLR0	Description
1, 2	0	0	0	TCNT clearing disabled (
			1	TCNT cleared by TGRA compare ma capture
		1	0	TCNT cleared by TGRB compare ma capture
			1	TCNT cleared by counter clearing for channel performing synchronous clea synchronous operation *1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYP
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared beca buffer register setting has priority, and compare match/input capture does no
3. Bit 7 is reserved in channels 1 and 2. It is always read as 0. The write value always be 0.

Note: Internal clock edge selection is valid when the input clock is Pϕ/4 or slower. If P
selected for the input clock, this setting is ignored and a rising-edge count is se

Bits 2 to 0—Time Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the TCNT co
The clock source can be selected independently for each channel. Table 16.4 shows th
sources that can be set for each channel.

Table 16.4 TPU Clock Sources

	Internal Clock						External Clo		
Channel	**Pϕ/1**	**Pϕ/4**	**Pϕ/16**	**Pϕ/64**	**Pϕ/256**	**Pϕ/1024**	**TCLKA**	**TCLKB**	**TCLK**
0	O	O	O	O			O	O	O
1	O	O	O	O	O		O	O	
2	O	O	O	O		O	O	O	O

Notes: O: Setting
Blank: No setting

		1	0	External clock: counts on TCLKC pin
			1	External clock: counts on TCLKD pin

Channel	Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1 (
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin
			1	External clock: counts on TCLKB pin
		1	0	Internal clock: counts on Pϕ/256
			1	Setting prohibited

Note: This setting is ignored when channel 1 is in phase counting mode.

Channel	Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
2	0	0	0	Internal clock: counts on Pϕ/1 (
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin
			1	External clock: counts on TCLKB pin
		1	0	External clock: counts on TCLKC pin
			1	Internal clock: counts on Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Channel 1:
TMDR1

Channel 2:
TMDR2

Bit:	7	6	5	4	3	2	1
	—	—	—	—	MD3	MD2	MD1
Initial value:	1	1	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W

The TMDR registers are 8-bit readable/writable registers that are used to set the opera
for each channel. The TPU has three TMDR registers, one for each channel. The TMD
are initialized to H'C0 by a reset.

TCNT operation should be stopped when making TMDR settings.

Bits 7 and 6—Reserved: These bits are always read as 1. The write value should alway

Bit 5—Buffer Operation B (BFB): Specifies whether TGRB is to operate in the norma
TGRB and TGRD are to be used together for buffer operation. When TGRD is used a
register, TGRD input capture/output compare is not generated.

In channels 1 and 2, which have no TGRD, bit 5 is reserved. It is always read as 0 and
modified.

Bit 5: BFB	Description
0	TGRB operates normally (
1	TGRB and TGRD used together for buffer operation

1	TGRA and TGRC used together for buffer operation

Bits 3 to 0—Modes 3 to 0 (MD3 to MD0): These bits are used to set the timer operating

Bit 3: MD3*1	Bit 2: MD2*2	Bit 1: MD1	Bit 0: MD0	Description
0	0	0	0	Normal operation (I
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	*	*	*	—

*

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.
2. Phase counting mode cannot be set for channel 0. In this case, 0 should alw written to MD2.

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 0: TIOR0L

Bit:	7	6	5	4	3	2	1
	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid a
register operates as a buffer register.

The TIOR registers are 8-bit registers that control the TGR registers. The TPU has fou
registers, two for channel 0 and one each for channels 1, and 2. The TIOR registers ar
to H'00 by a reset.

Note that TIOR is affected by the TMDR setting.

The initial output specified by TIOR becomes valid when the counter is halted (i.e. wh
bit is cleared to 0 in TSTR). In PWM mode 2, the output at the point at which the cou
cleared to 0 is specified.

0	0	0	0	0	TGR0B is output compare register	Output disabled	(N
				1		Initial output is 0 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at ris
				1			Input capture at fa
			1	*			Input capture at bo
		1	*	*		Setting prohibited	

*

	1	0	0		Output disabled		
			1		Initial output is 1 output	0 output at comp	
		1	0			1 output at comp	
			1			Toggle output at match	
1	0	0	0	TGR0D is input capture register*1	Capture input source is TIOCD0 pin	Input capture at r	
			1			Input capture at f	
		1	*			Input capture at l	
	1	*	*		Setting prohibited		

Note: 1. When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer regi
setting is invalid and input capture/output compare is not generated.

		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at ris
				1			Input capture at fa
			1	*			Input capture at bo
		1	*	*		Setting prohibited	

*

TIOR2

Channel	Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
2	0	0	0	0	TGR2B is output compare register	Output disabled	(
				1		Initial output is 0 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at ris
				1			Input capture at fa
			1	*			Input capture at bo

*

0	0	0	0	0	TGR0A is output compare register	Output disabled	
				1		Initial output is 0 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at r
				1			Input capture at f
			1	*			Input capture at b
		1	*	*		Setting prohibited	

	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR0C is input capture register*1	Capture input source is TIOCC0 pin	Input capture at ris
			1			Input capture at fa
		1	*			Input capture at bo
	1	*	*		Setting prohibited	

*

Note: 1. When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer regis setting is invalid and input capture/output compare is not generated.

		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at r
				1			Input capture at f
			1	*			Input capture at l
		1	*	*		Setting prohibited	

TIOR2

Channel	Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
2	0	0	0	0	TGR2A is output compare register	Output disabled	
				1		Initial output is 0 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at r
				1			Input capture at f
			1	*			Input capture at l

Channel 1: TIER1

Channel 2: TIER2

Bit:	7	6	5	4	3	2	1
	—	—	TCIEU	TCIEV	—	—	TGIEB
Initial value:	0	1	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W

The TIER registers are 8-bit registers that control enabling or disabling of interrupt requ
each channel. The TPU has three TIER registers, one for each channel. The TIER regis
initialized to H'40 by a reset.

Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.

Bit 6—Reserved: This bit is always read as 1. The write value should always be 1.

Bit 5—Underflow Interrupt Enable (TCIEU): Enables or disables interrupt requests (TC
TCFU flag when the TCFU flag in TSR is set to 1 in channels 1 and 2.

In channel 0, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5: TCIEU	Description
0	Interrupt requests (TCIU) by TCFU disabled (Ir
1	Interrupt requests (TCIU) by TCFU enabled

Bit 4—Overflow Interrupt Enable (TCIEV): Enables or disables interrupt requests (TCI
TCFV flag when the TCFV flag in TSR is set to 1.

Bit 4: TCIEV	Description
0	Interrupt requests (TCIV) by TCFV disabled (I
1	Interrupt requests (TCIV) by TCFV enabled

Bit 2—TGR Interrupt Enable C (TGIEC): Enables or disables interrupt requests (TGI
TGFC bit when the TGFC bit in TSR is set to 1 in channel 0.

In channels 1 and 2, bit 2 is reserved. It is always read as 0 and cannot be modified.

Bit 2: TGIEC	Description
0	Interrupt requests (TGIC) by TGFC bit disabled
1	Interrupt requests (TGIC) by TGFC bit enabled

Bit 1—TGR Interrupt Enable B (TGIEB): Enables or disables interrupt requests (TGI
TGFB bit when the TGFB bit in TSR is set to 1.

Bit 1: TGIEB	Description
0	Interrupt requests (TGIB) by TGFB bit disabled
1	Interrupt requests (TGIB) by TGFB bit enabled

Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables interrupt requests (TGI
TGFA bit when the TGFA bit in TSR is set to 1.

Bit 0: TGIEA	Description
0	Interrupt requests (TGIA) by TGFA bit disabled
1	Interrupt requests (TGIA) by TGFA bit enabled

Channel 1: TSR1

Channel 2: TSR2

Bit:	7	6	5	4	3	2	1
	TCFD	—	TCFU	TCFV	—	—	TGFB
Initial value:	1	1	0	0	0	0	0
R/W:	R	R	R/(W)*	R/(W)*	R	R	R/(W)*

Note: * Only 0 can be written, to clear the flags.

The TSR registers are 8-bit registers that indicate the status of each channel. The TPU h
TSR registers, one for each channel. The TSR registers are initialized to H'C0 by a rese

Bit 7—Count Direction Flag (TCFD): Status flag that shows the direction in which TCN
in channels 1, and 2.

In channel 0, bit 7 is reserved. It is always read as 1 and cannot be modified.

Bit 7: TCFD	Description
0	TCNT counts down
1	TCNT counts up (I

Bit 6—Reserved: This bit is always read as 1. The write value should always be 1.

	When the TCNT value underflows (changes from H'0000 to H'FFF

Bit 4—Overflow Flag (TCFV): Status flag that indicates that TCNT overflow has occ

Bit 4: TCFV	Description
0	[Clearing condition] (When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000

Bit 3—Input Capture/Output Compare Flag D (TGFD): Status flag that indicates the c
TGRD input capture or compare match in channel 0.

In channels 1 and 2, bit 3 is reserved. It is always read as 0 and cannot be modified.

Bit 3: TGFD	Description
0	[Clearing conditions] (• When DMAC is activated by TGID interrupt while DRCR settin TGI0D • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output com register • When TCNT value is transferred to TGRD by input capture sig TGRD is functioning as input capture register

	• When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output comp register • When TCNT value is transferred to TGRC by input capture signa TGRC is functioning as input capture register

Bit 1—Input Capture/Output Compare Flag B (TGFB): Status flag that indicates the oc TGRB input capture or compare match.

Bit 1: TGFB	Description
0	[Clearing conditions] (Ini • When DMAC is activated by TGIB interrupt while DRCR setting TGI0B • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output comp register • When TCNT value is transferred to TGRB by input capture signa TGRB is functioning as input capture register

2	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output com register • When TCNT value is transferred to TGRA by input capture sign TGRA is functioning as input capture register

16.2.6 Timer Counter (TCNT)

Channel 0: TCNT0 (up-counter)

Channel 1: TCNT1 (up/down-counter*)

Channel 2: TCNT2 (up/down-counter*)

Bit:	15	14	13	12	11	10	9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * These counters can be used as up/down-counters only in phase counting r other cases they function as up-counters.

The TCNT registers are 16-bit counters. The TPU has three TCNT counters, one for e

The TCNT counters are initialized to H'0000 by a reset.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed a unit.

Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The TGR registers are 16-bit registers with a dual function as output compare and inpu
registers. The TPU has 8 TGR registers, four for channel 0 and two each for channels 1
TGRC and TGRD for channel 0 can also be designated for operation as buffer registers
TGR registers are initialized to H'FFFF by a reset.

The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a

Note: * TGR buffer register combinations are TGRA–TGRC and TGRB–TGRD.

16.2.8 Timer Start Register (TSTR)

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	CST2	CST1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels
TSTR is initialized to H'00 by a reset.

TCNT counter operation should be stopped when setting the operating mode in TMDR
TCNT count clock in TCR.

Bits 7 to 3—Reserved: These bits are always read as 0. The write value should always b

written to when the CST bit is cleared to 0, the pin output level will be chan
set initial output value.

16.2.9 Timer Synchro Register (TSYR)

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	SYNC2	SYNC
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W

TSYR is an 8-bit readable/writable register that selects independent operation or synch
operation for the channel 0 to 2 TCNT counters. A channel performs synchronous ope
the corresponding bit in TSYR is set to 1.

TSYR is initialized to H'00 by a reset.

Bits 7 to 3—Reserved: These bits are always read as 0. The write value should always

Bits 2 to 0—Timer Synchro 2 to 0 (SYNC2 to SYNC0): These bits select whether ope
independent of or synchronized with other channels.

When synchronous operation is selected, synchronous presetting of multiple channels
synchronous clearing through counter clearing on another channel*2 are possible.

Bit n:	SYNCn	Description
0		TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels
1		TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

Notes: n = 2 to 0

1. To set synchronous operation, the SYNC bits for two channels at least mus
2. To set synchronous clearing, in addition to the SYNC bit , the TCNT clearin
must also be set by means of bits CCLR2 to CCLR0 in TCR.

An example of 16-bit register access operation is shown in figure 16.2.

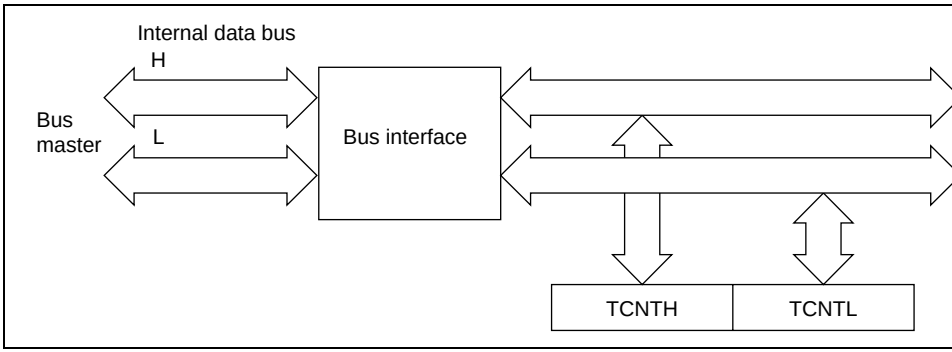


Figure 16.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 B

16.3.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits wi
registers can be read and written to in 16-bit units. They can also be read and written to
units.

Examples of 8-bit register access operation are shown in figures 16.3, 16.4, and 16.5.

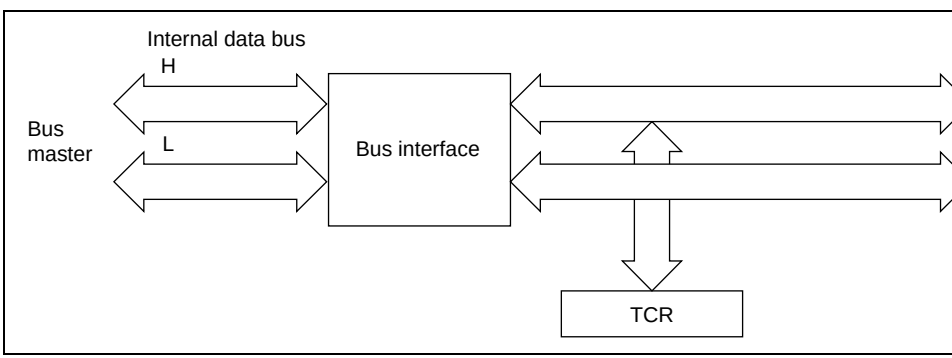


Figure 16.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8

Figure 16.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower

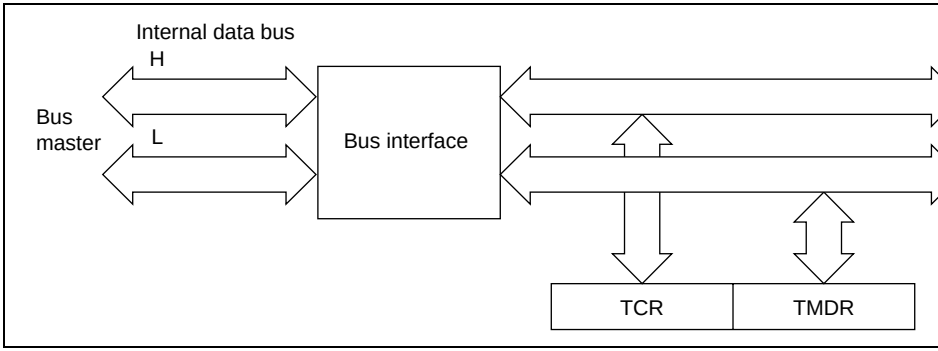


Figure 16.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR

Each TGR can be used as an input capture register or output compare register.

Synchronous Operation: The TCNT counter for a channel designated for synchronous by means of TSYR performs synchronous presetting. That is, when TCNT for a channe designated for synchronous operation is rewritten, the TCNT counters for the other cha also rewritten at the same time. Synchronous clearing of the TCNT counters is also pos setting the counter clear bits in TCR for channels designated for synchronous operation

Buffer Operation

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the relevant chan transferred to TGR.

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transfer to TGR and the value pre held in TGR is transferred to the buffer register.

PWM Mode: In this mode, a PWM waveform is output. The output level can be set by TIOR. A PWM waveform with a duty of between 0% and 100% can be output, accordi setting of each TGR register.

Phase Counting Mode: In this mode, TCNT is incremented or decremented by detecti phases of two clocks input from the external clock input pins in channels 1, and 2. Whe counting mode is set, the corresponding TCLK pin functions as the clock input, and TC performs up- or down-counting.

This can be used for two-phase encoder pulse input.

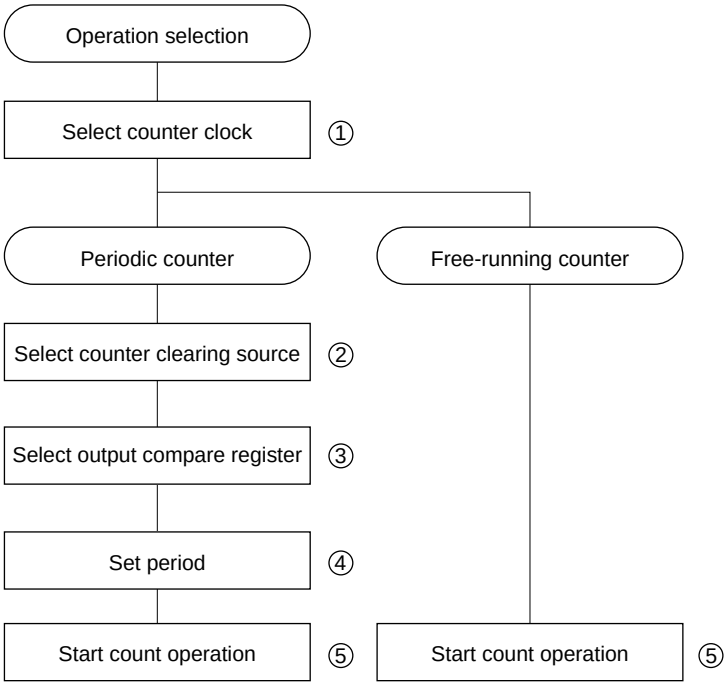


Figure 16.6 Example of Counter Operation Setting Procedure

Figure 16.7 illustrates free-running counter operation.

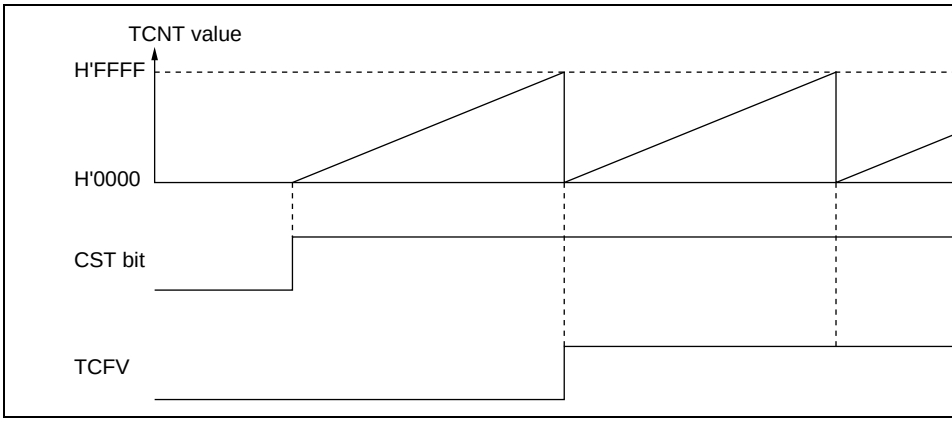


Figure 16.7 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter f
relevant channel performs periodic count operation. The TGR register for setting th
designated as an output compare register, and counter clearing by compare match is
by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TC
up-count operation as periodic counter when the corresponding bit in TSTR is set to
the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT
to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU reque
interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 16.8 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output
corresponding output pin using compare match.

- Example of setting procedure for waveform output by compare match

  Figure 16.9 shows an example of the setting procedure for waveform output by co

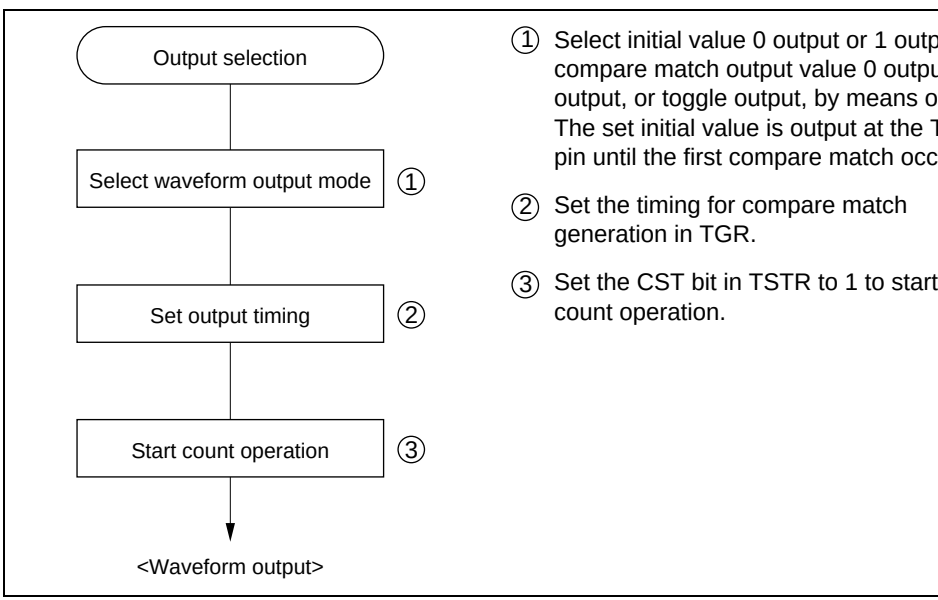


Figure 16.9 Example of Setting Procedure for Waveform Output by Compar

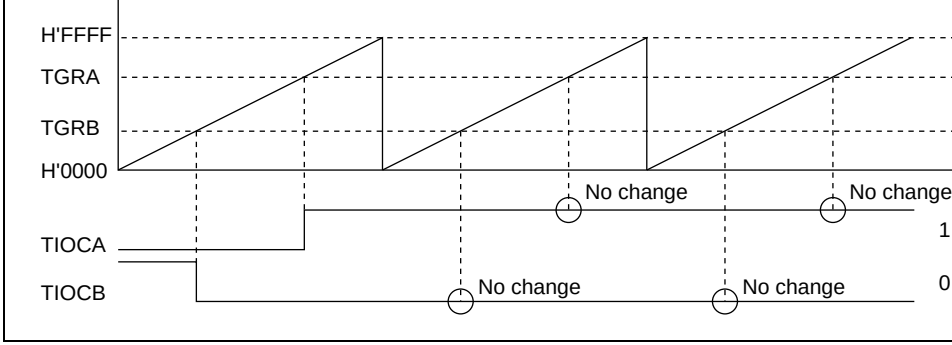


Figure 16.10 Example of 0 Output/1 Output Operation

Figure 16.11 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clear
performed by compare match B), and settings have been made so that output is togg
compare match A and compare match B.

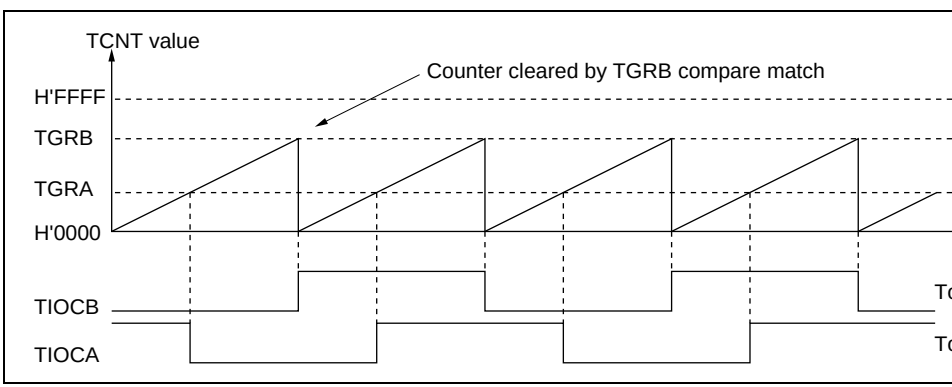
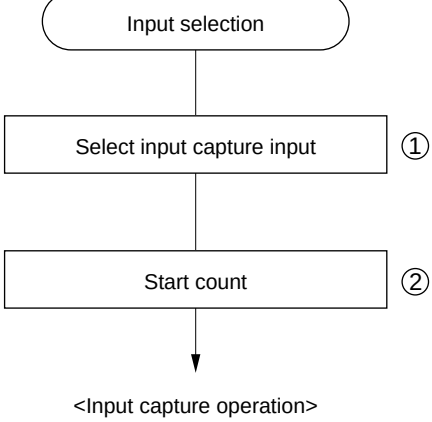


Figure 16.11 Example of Toggle Output Operation



① Designate TGR as an input capture
means of TIOR, and select rising ed
edge, or both edges as the input cap
source and input signal edge.

② Set the CST bit in TSTR to 1 to start
operation.

Figure 16.12 Example of Input Capture Operation Setting Procedur

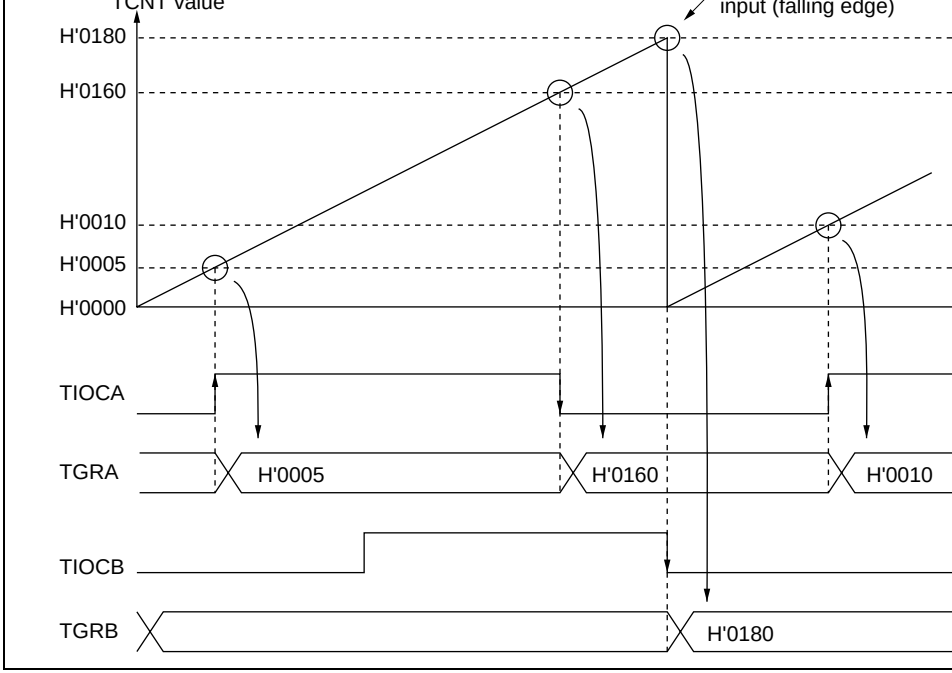
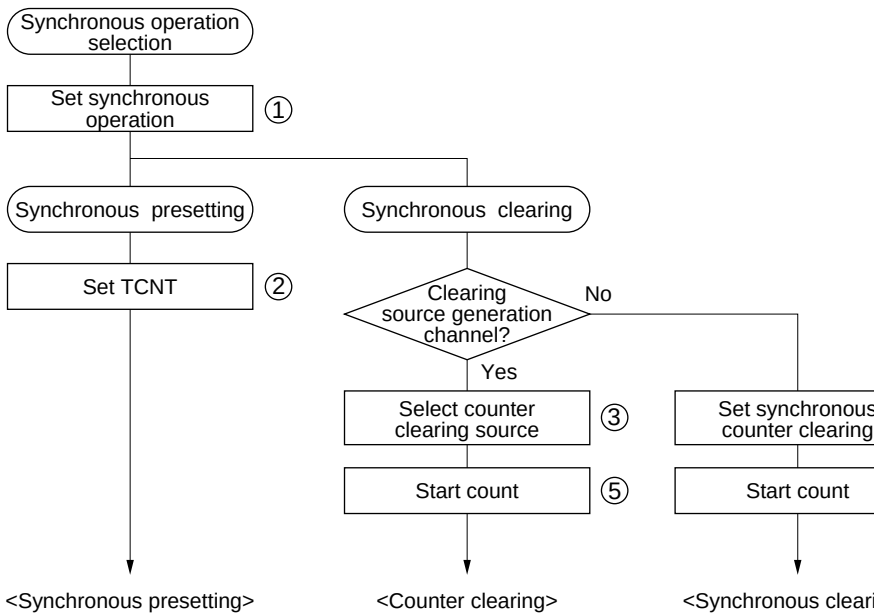


Figure 16.13 Example of Input Capture Operation

Example of Synchronous Operation Setting Procedure: Figure 16.14 shows an exa
synchronous operation setting procedure.



① Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for syn
operation.

② When the TCNT counter of any of the channels designated for synchronous ope
written to, the same value is simultaneously written to the other TCNT counters.

③ Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output c
etc.

④ Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter
source.

⑤ Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 16.14 Example of Synchronous Operation Setting Procedure

For details of PWM modes, see section 16.4.5, PWM Modes.

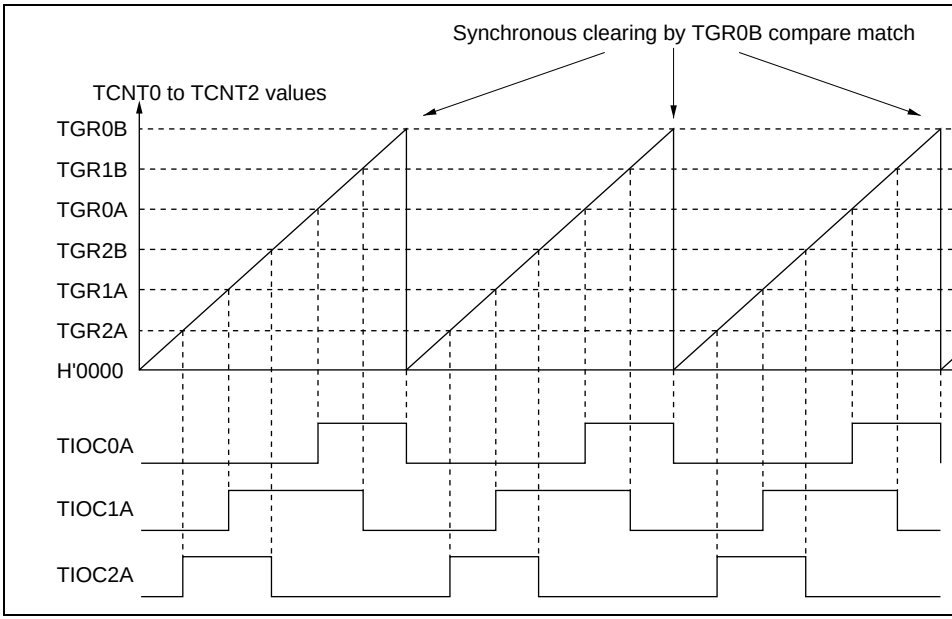


Figure 16.15 Example of Synchronous Operation

Channel	Timer General Register	Buffer Register
0	TGR0A	TGR0C
	TGR0B	TGR0D

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the correspondi
  transferred to the timer general register.

  This operation is illustrated in figure 16.16.

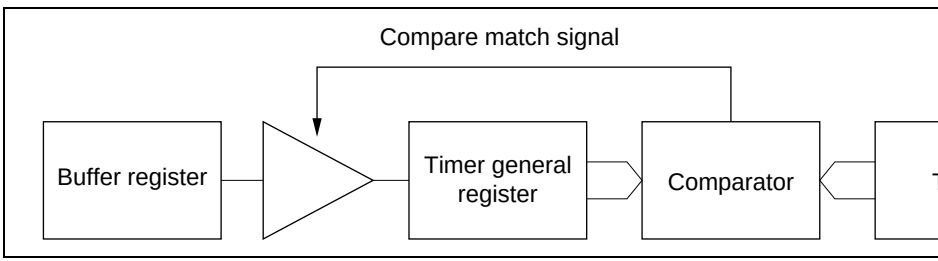


Figure 16.16 Compare Match Buffer Operation

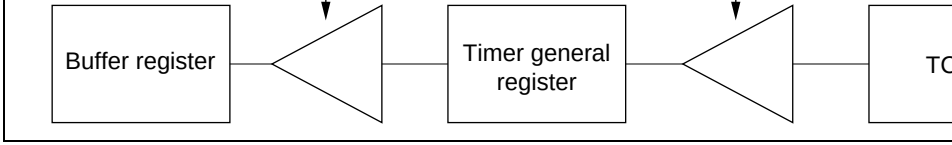


Figure 16.17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 16.18 shows an example of
operation setting procedure.

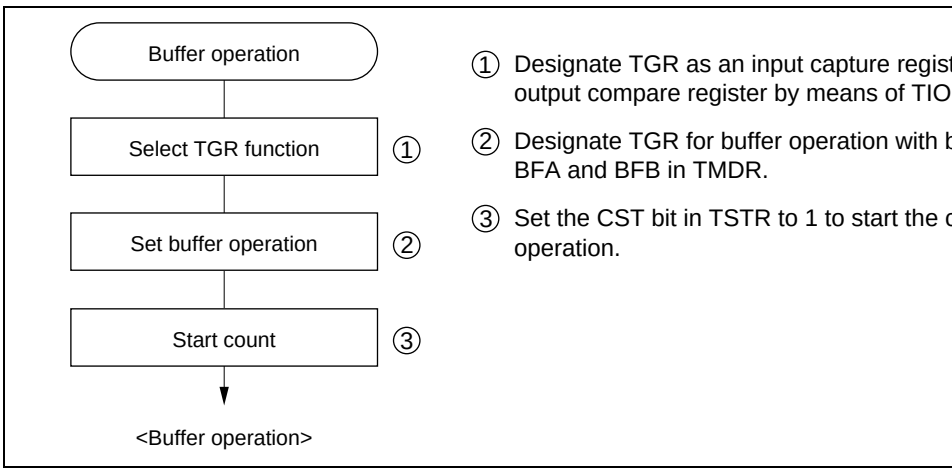


Figure 16.18 Example of Buffer Operation Setting Procedure

value in buffer register TGRC is simultaneously transferred to timer general regist
This operation is repeated each time compare match A occurs.

For details of PWM modes, see section 16.4.5, PWM Modes.

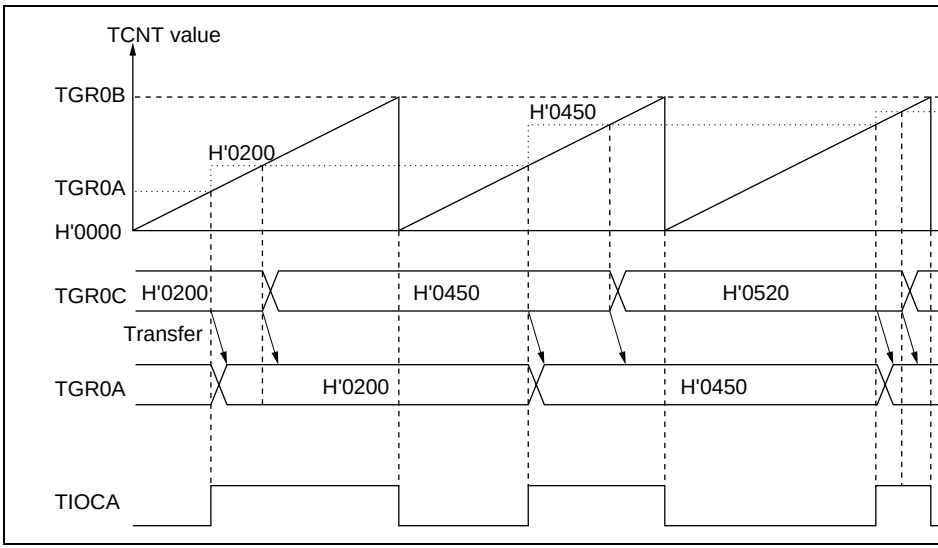


Figure 16.19 Example of Buffer Operation (1)

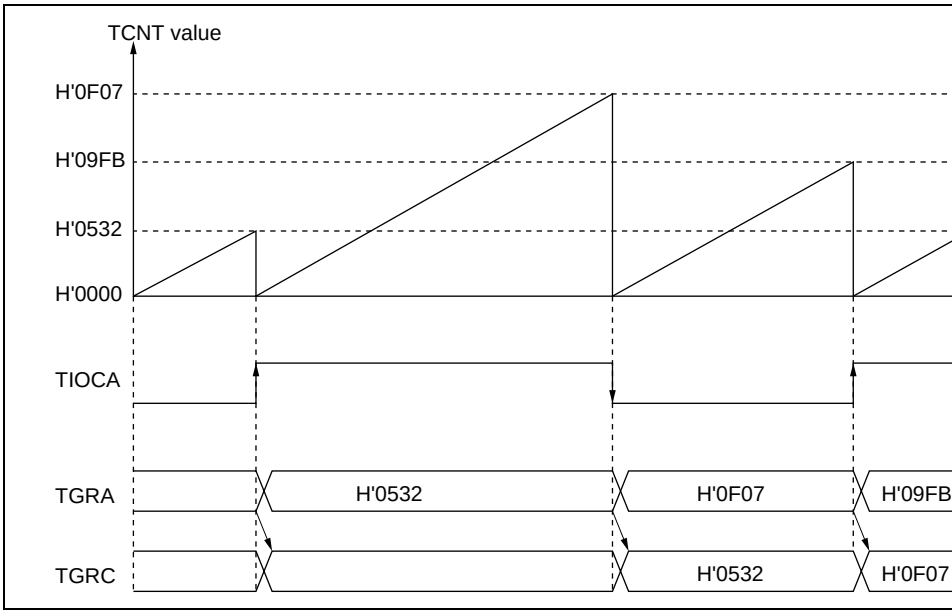


Figure 16.20 Example of Buffer Operation (2)

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated by pairing TGRA with TGRB and TGRC with TGRD. T
  specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR is performed in respon
  compare match A and C, and the output specified by bits IOB3 to IOB0 and IOD3
  TIOR in response to compare match B and D, from pins TIOCA and TIOCC. The
  value is the value set in TGRA or TGRC. If the set values of paired TGRs are iden
  output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 4-phase PWM output is possible.

- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as du
  The output specified by TIOR is performed in response to a compare match. Also,
  counter is cleared by a synchronization register compare match, pin output values
  initial values set in TIOR. If the set values of the period and duty registers are iden
  output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 7-phase PWM output is possible by combined use w
  synchronous operation.

The correspondence between PWM output pins and registers is shown in table 16.6.

1	TGR1A	TIOCA1	TIOCA1
	TGR1B		TIOCB1
2	TGR2A	TIOCA2	TIOCA2
	TGR2B		TIOCB2

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the p

Example of PWM Mode Setting Procedure: Figure 16.21 shows an example of the P setting procedure.

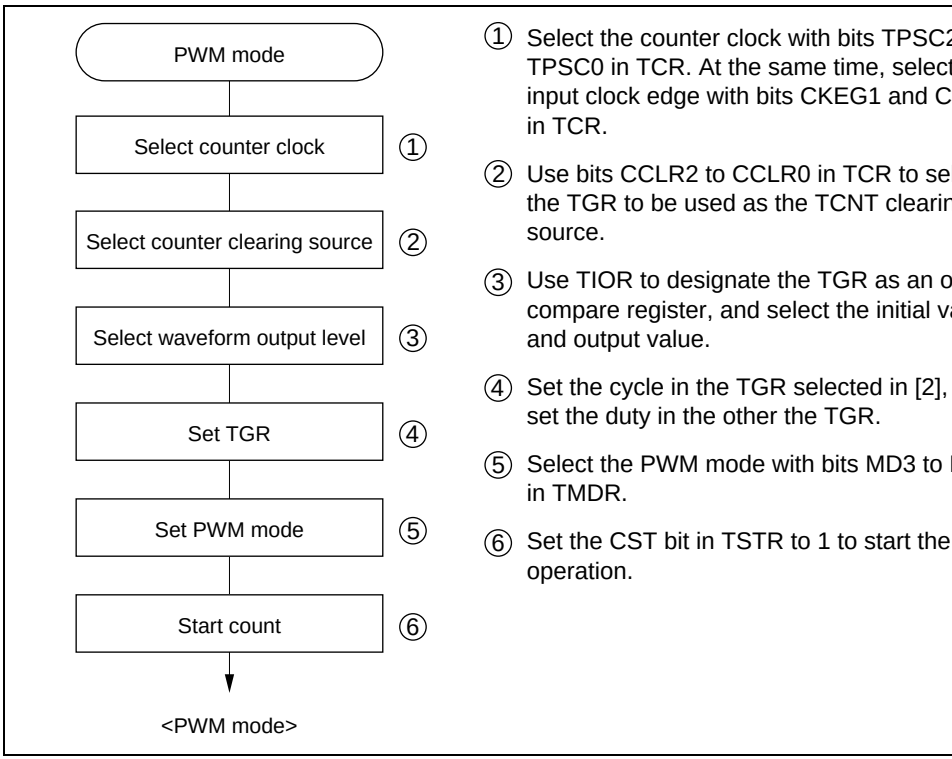


Figure 16.21 Example of PWM Mode Setting Procedure

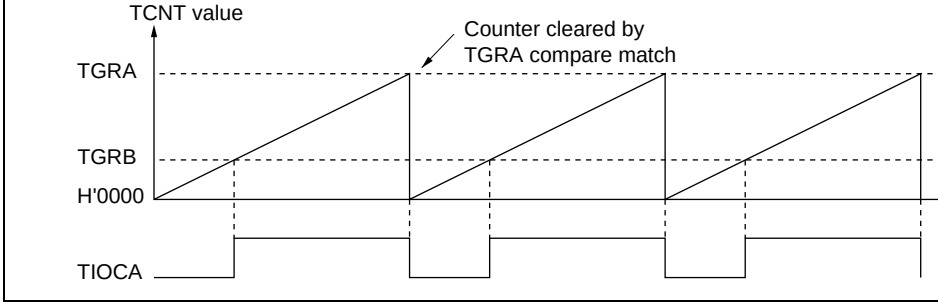


Figure 16.22 Example of PWM Mode Operation (1)

Figure 16.23 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGR1B co
match is set as the TCNT clearing source, and 0 is set for the initial output value and 1
output value of the other TGR registers, to output a 5-phase PWM waveform.

In this case, the value set in TGR1B is used as the cycle, and the values set in the othe
the duty.

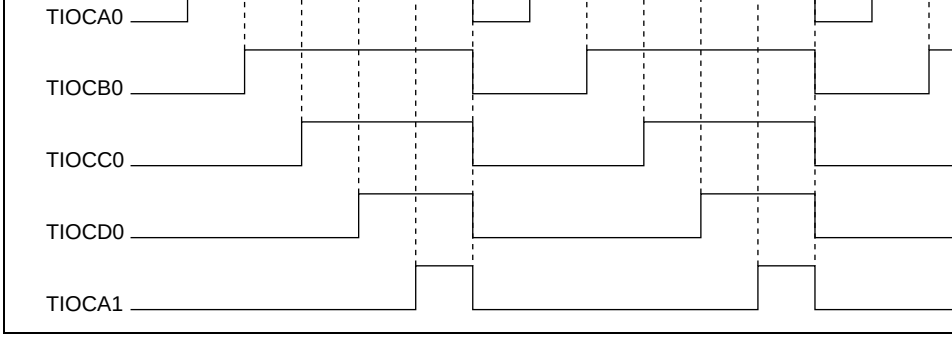


Figure 16.23 Example of PWM Mode Operation (2)

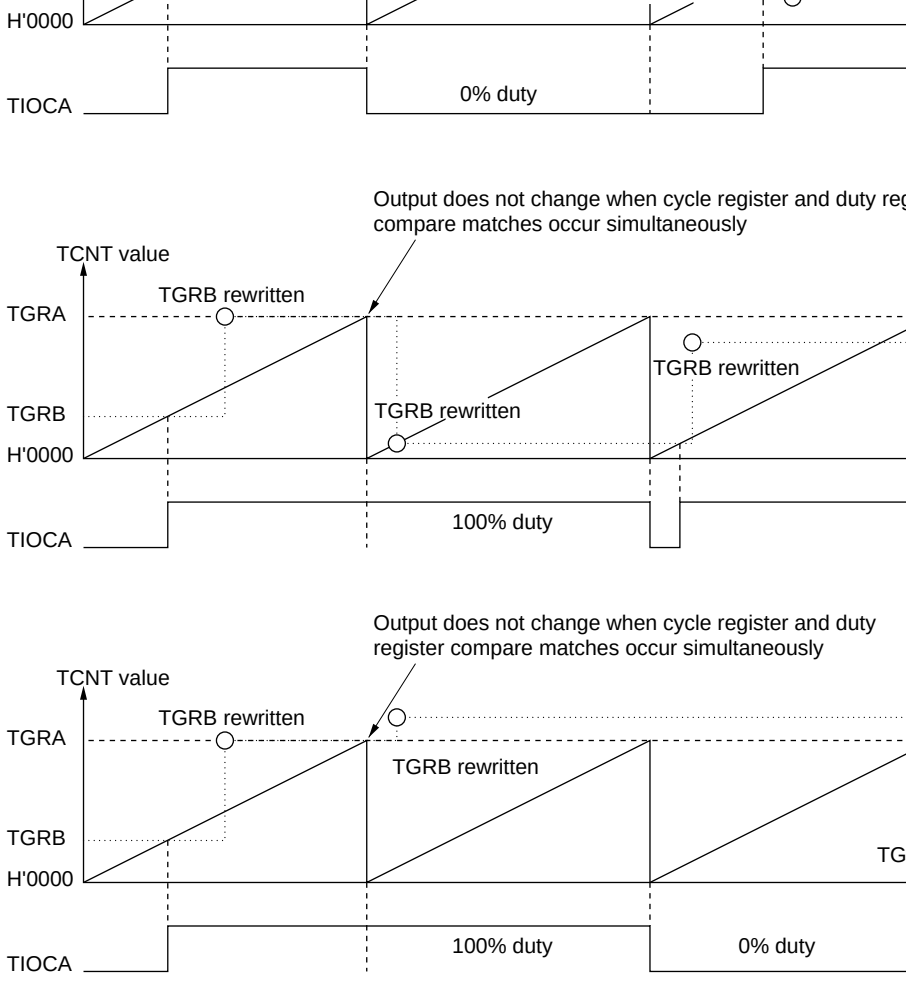


Figure 16.24 Example of PWM Mode Operation (3)

used.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when
occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an in
whether TCNT is counting up or down.

Table 16.7 shows the correspondence between external clock pins and channels.

Table 16.7 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 16.25 shows an examp
phase counting mode setting procedure.

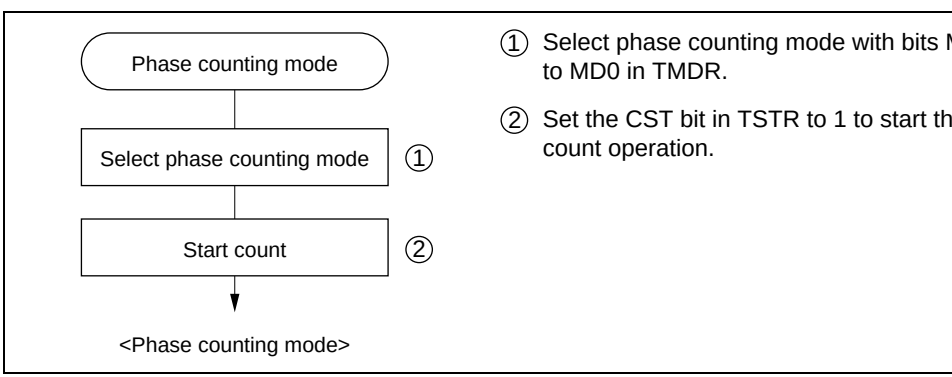


Figure 16.25 Example of Phase Counting Mode Setting Procedure

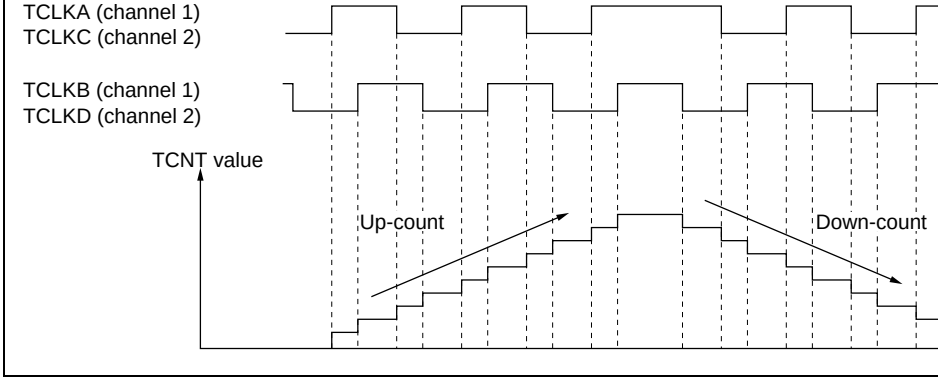


Figure 16.26 Example of Phase Counting Mode 1 Operation

Table 16.8 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-coun
Low level	↑	
↑	High level	
↓	Low level	

Notes: ↑ : Rising edge
↓ : Falling edge

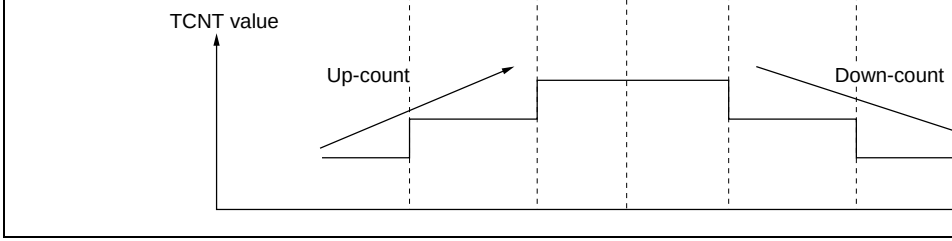


Figure 16.27 Example of Phase Counting Mode 2 Operation

Table 16.9 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

Notes: ↑ : Rising edge
↓ : Falling edge

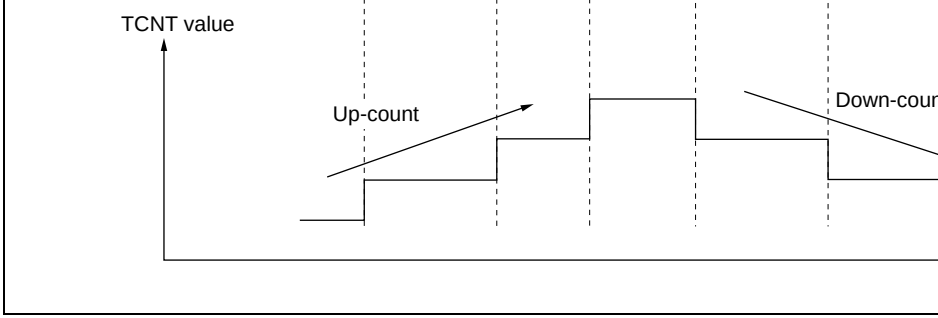


Figure 16.28 Example of Phase Counting Mode 3 Operation

Table 16.10 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don’t care
Low level	↓	Don’t care
↑	Low level	Don’t care
↓	High level	Up-count
High level	↓	Down-coun
Low level	↑	Don’t care
↑	High level	Don’t care
↓	Low level	Don’t care

Notes: ↑: Rising edge
↓: Falling edge

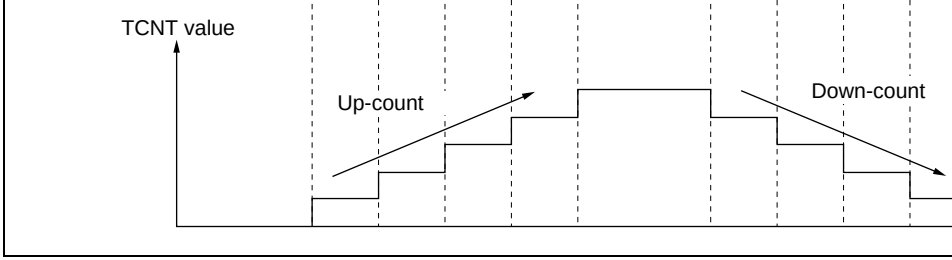


Figure 16.29 Example of Phase Counting Mode 4 Operation

Table 16.11 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

Notes: ↑ : Rising edge
↓ : Falling edge

corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is request
interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority
a channel is fixed. For details, see section 5, Interrupt Controller (INTC).

Table 16.12 lists the TPU interrupt sources.

Table 16.12 TPU Interrupts

Channel	Interrupt Source	Description	DMAC Activation
0	TGI0A	TGR0A input capture/compare match	Possible
	TGI0B	TGR0B input capture/compare match	Possible
	TGI0C	TGR0C input capture/compare match	Possible
	TGI0D	TGR0D input capture/compare match	Possible
	TCI0V	TCNT0 overflow	Not possible
1	TGI1A	TGR1A input capture/compare match	Not possible
	TGI1B	TGR1B input capture/compare match	Not possible
	TCI1V	TCNT1 overflow	Not possible
	TCI1U	TCNT1 underflow	Not possible
2	TGI2A	TGR2A input capture/compare match	Not possible
	TGI2B	TGR2B input capture/compare match	Not possible
	TCI2V	TCNT2 overflow	Not possible
	TCI2U	TCNT2 underflow	Not possible

Note: This table shows the initial state immediately after a reset. The relative channel
can be changed by the interrupt controller.

one for each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 wh
TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on channel. The in
request is cleared by clearing the TCFU flag to 0. The TPU has two underflow interrup
for channels 1 and 2.

16.5.2 DMAC Activation

The DMAC can be activated by the TGR input capture/compare match interrupt for a c
details, see section 11, Direct Memory Access Controller (DMAC).

A total of four TPU input capture/compare match interrupts can be used as DMAC acti
sources for channel 0.

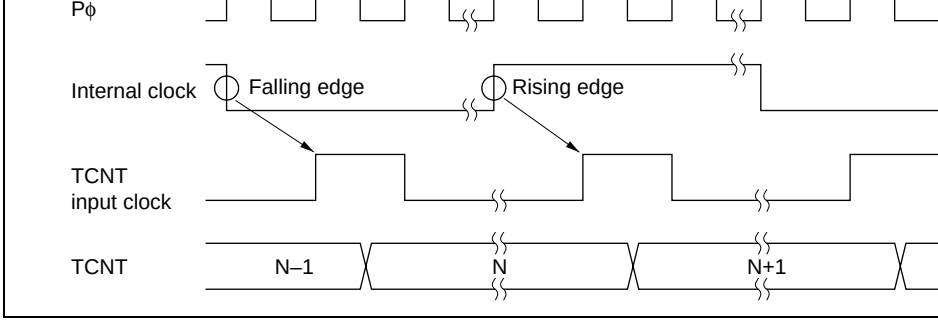


Figure 16.30 Count Timing in Internal Clock Operation

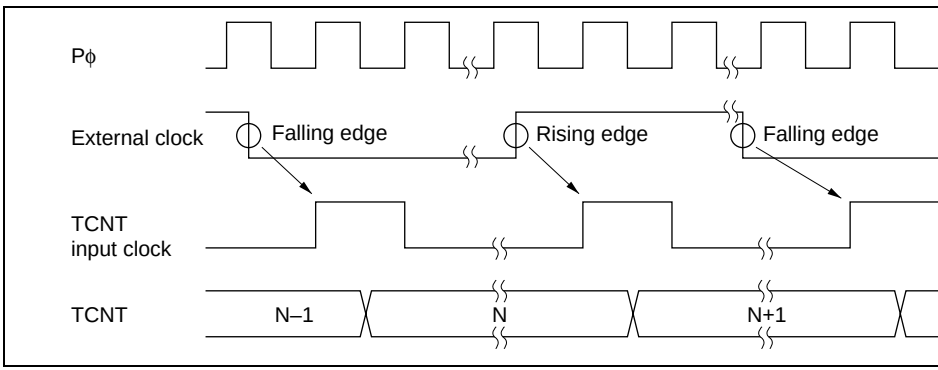


Figure 16.31 Count Timing in External Clock Operation

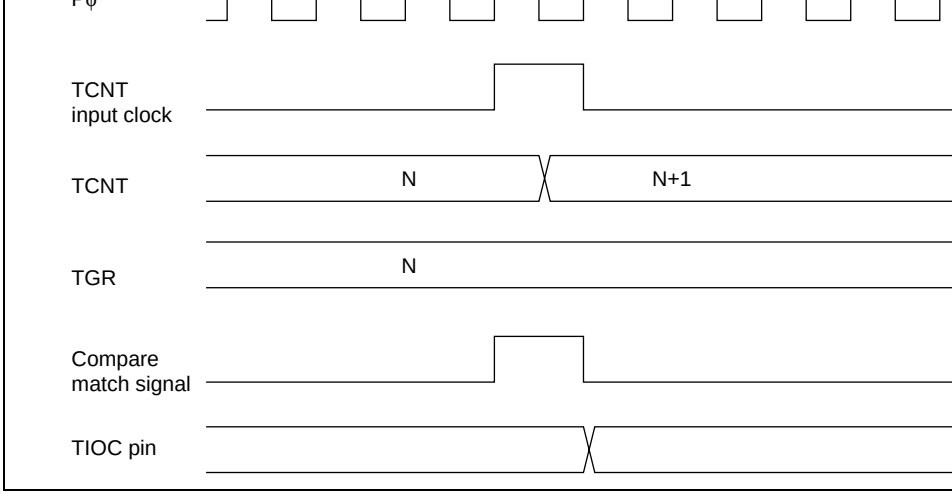


Figure 16.32 Output Compare Output Timing

Input Capture Signal Timing: Figure 16.33 shows input capture signal timing.

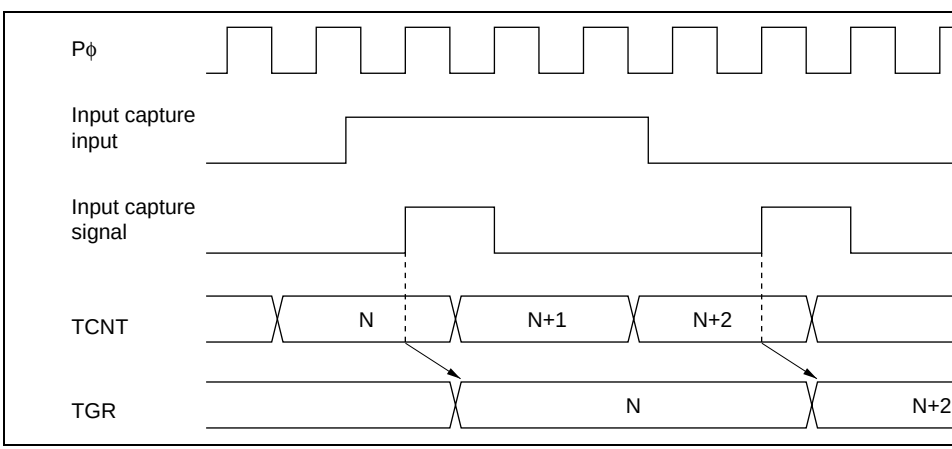


Figure 16.33 Input Capture Input Signal Timing

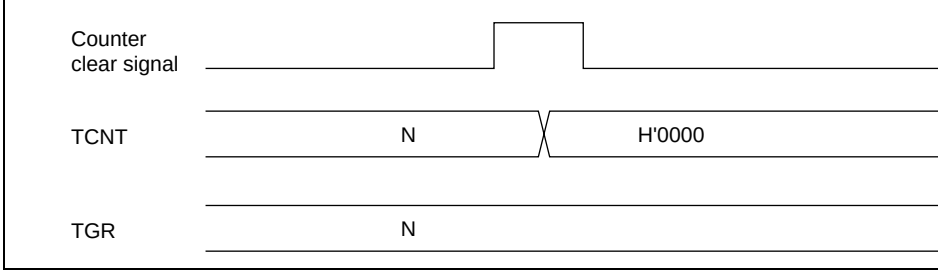


Figure 16.34 Counter Clear Timing (Compare Match)

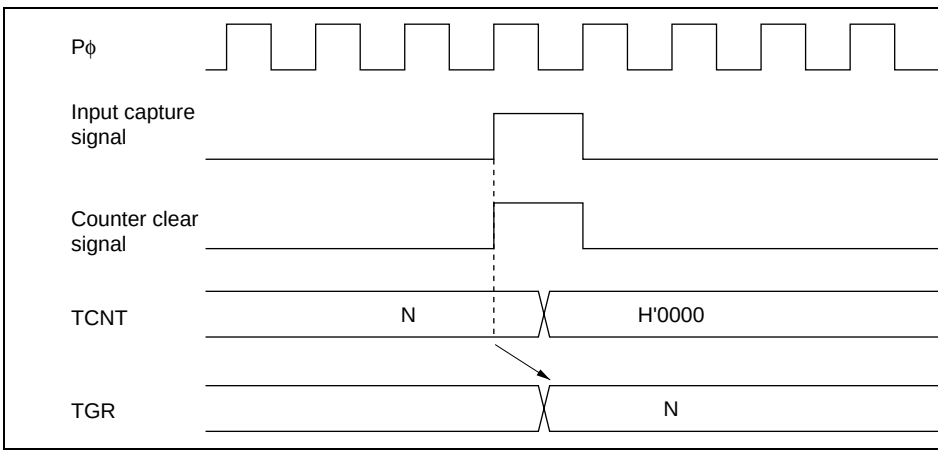


Figure 16.35 Counter Clear Timing (Input Capture)

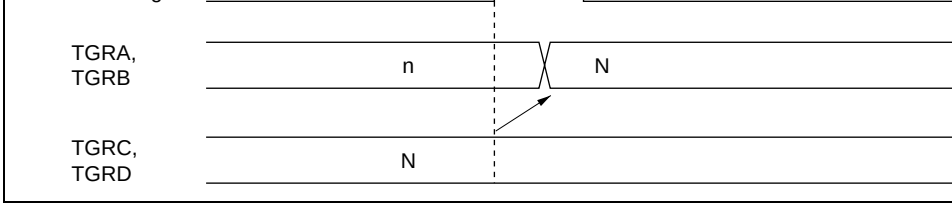


Figure 16.36 Buffer Operation Timing (Compare Match)

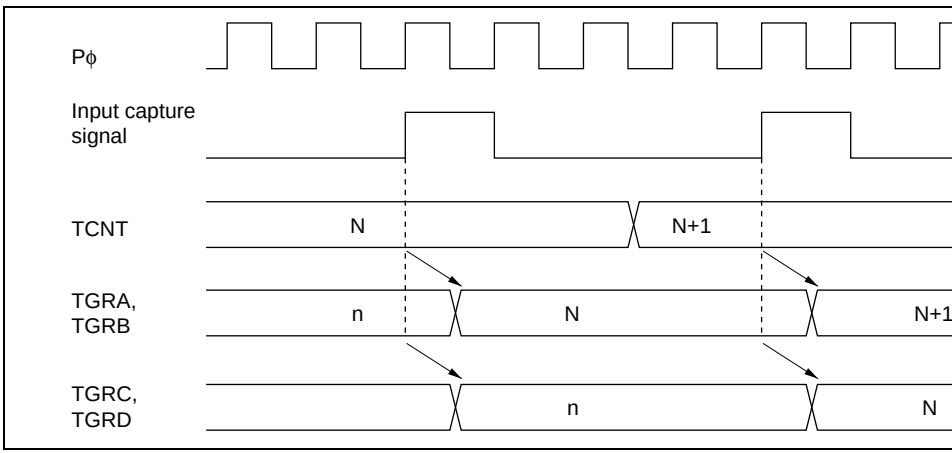


Figure 16.37 Buffer Operation Timing (Input Capture)

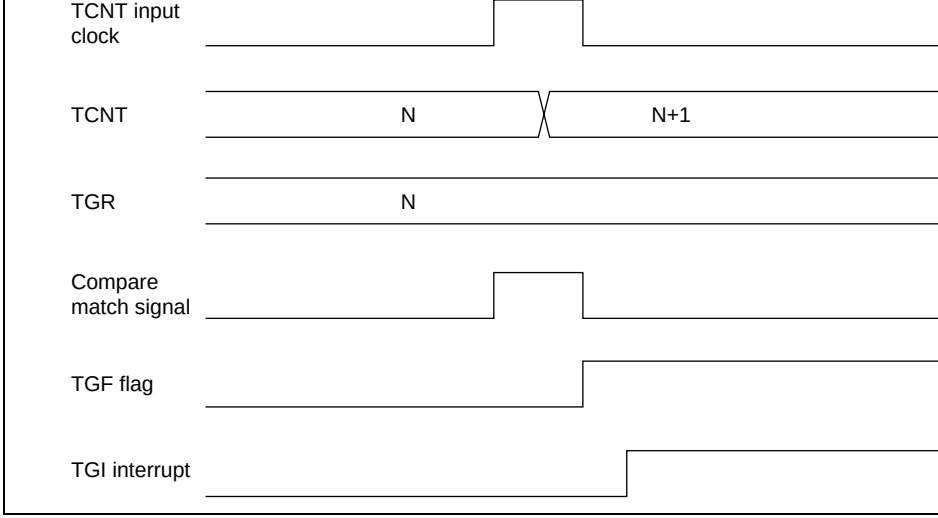


Figure 16.38 TGI Interrupt Timing (Compare Match)

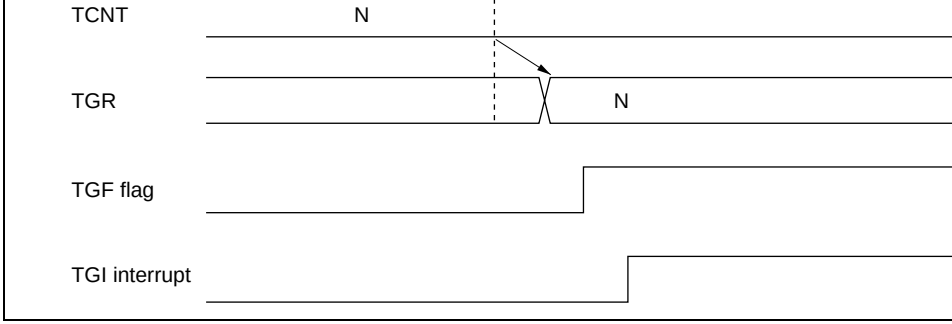


Figure 16.39 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 16.40 shows the timing for setting of flag in TSR by overflow occurrence, and TCIV interrupt request signal timing.

Figure 16.41 shows the timing for setting of the TCFU flag in TSR by underflow occur TCIU interrupt request signal timing.

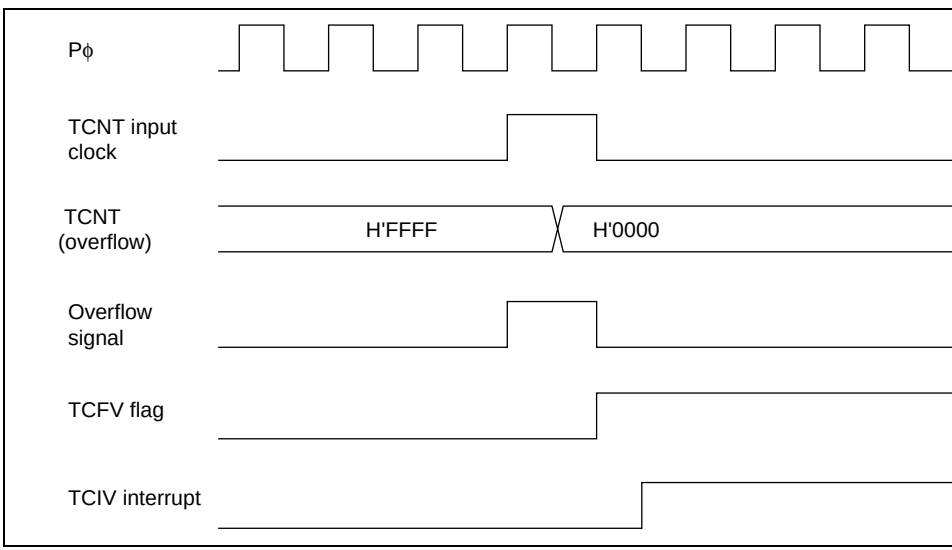


Figure 16.40 TCIV Interrupt Setting Timing

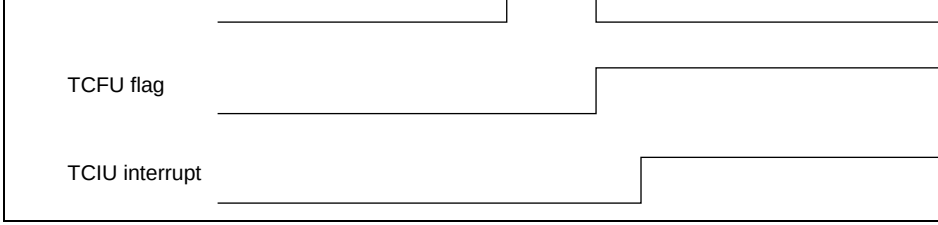


Figure 16.41 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared 0 to it. When the DMAC is activated, the flag is cleared automatically. Figure 16.42 sh timing for status flag clearing by the CPU, and figure 16.43 shows the timing for statu clearing by the DMAC.

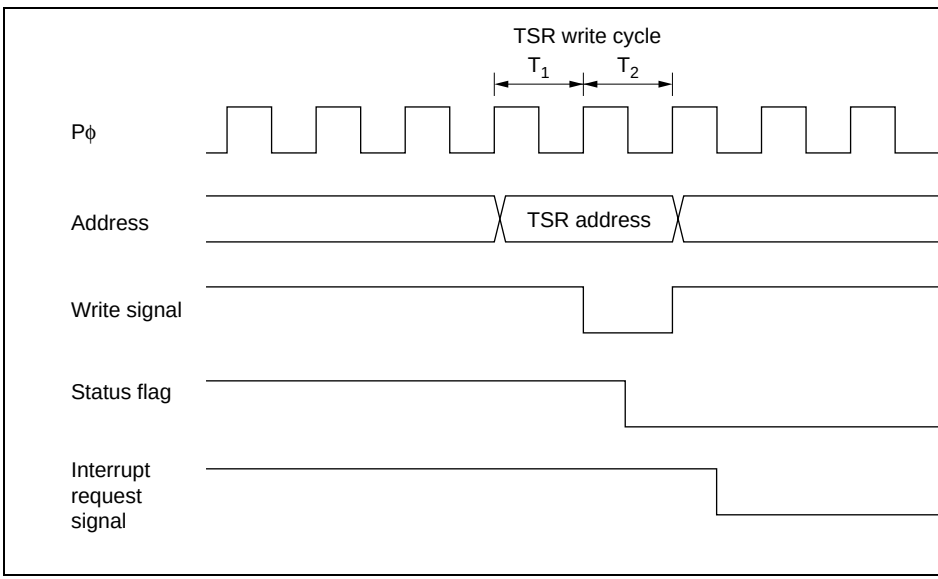


Figure 16.42 Timing for Status Flag Clearing by CPU

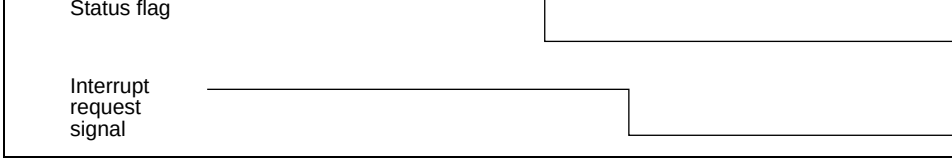


Figure 16.43 Timing for Status Flag Clearing by DMAC Activation

16.7 Usage Notes

Note that the kinds of operation and contention described below occur during TPU ope

Input Clock Restrictions: The input clock pulse width must be at least 1.5 states in th single-edge detection, and at least 2.5 states in the case of both-edge detection. The TP operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks least 1.5 states, and the pulse width must be at least 2.5 states. Figure 16.44 shows the i conditions in phase counting mode.

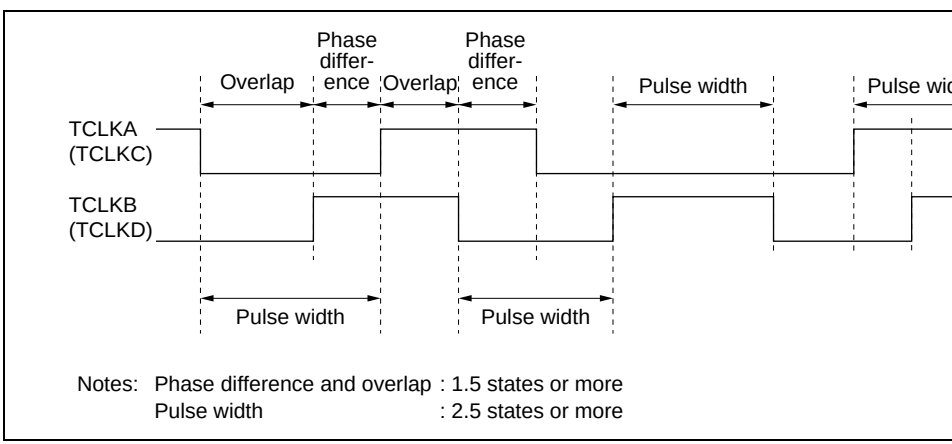


Figure 16.44 Phase Difference, Overlap, and Pulse Width in Phase Counting

N : TGR set value

Contention between TCNT Write and Clear Operations: If the counter clear signa generated in the T_2 state of a TCNT write cycle, TCNT clearing takes precedence and write is not performed.

Figure 16.45 shows the timing in this case.

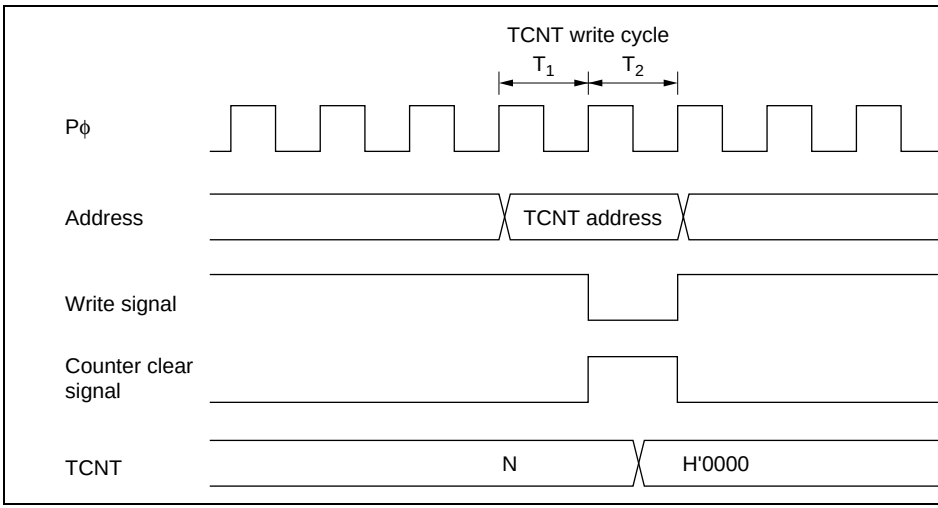


Figure 16.45 Contention between TCNT Write and Clear Operation

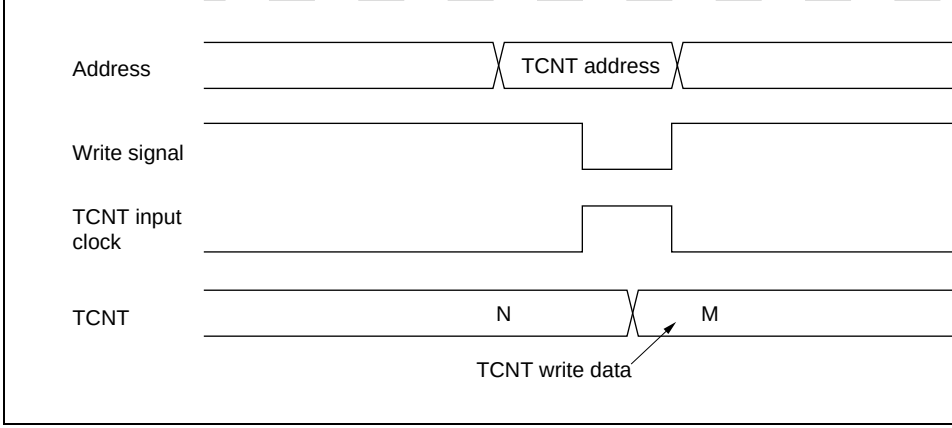


Figure 16.46 Contention between TCNT Write and Increment Operatio

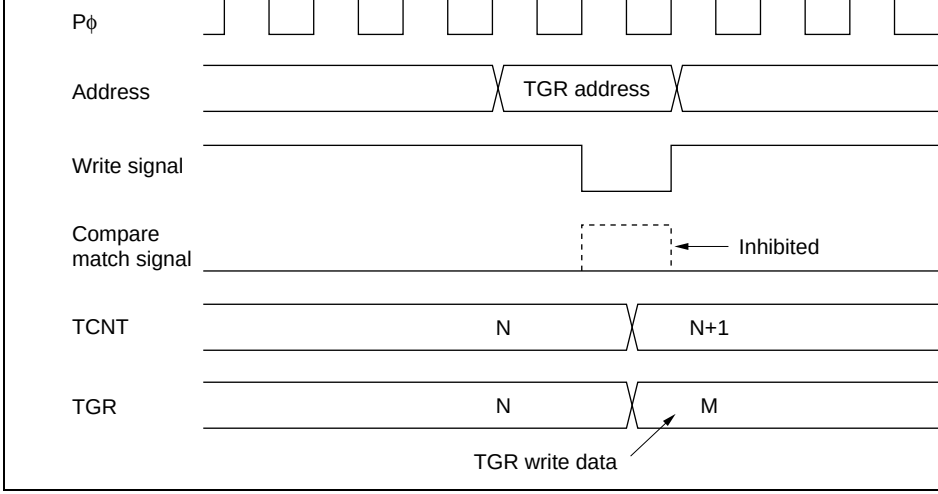


Figure 16.47 Contention between TGR Write and Compare Match

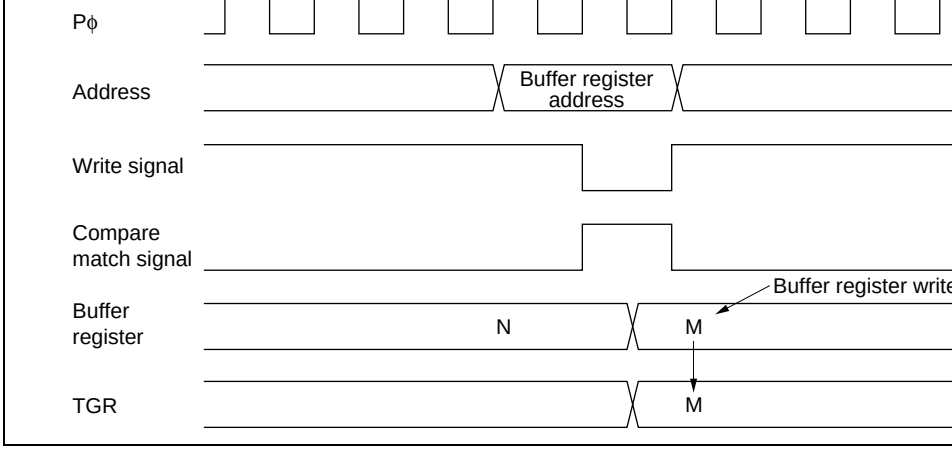


Figure 16.48 Contention between Buffer Register Write and Compare Ma

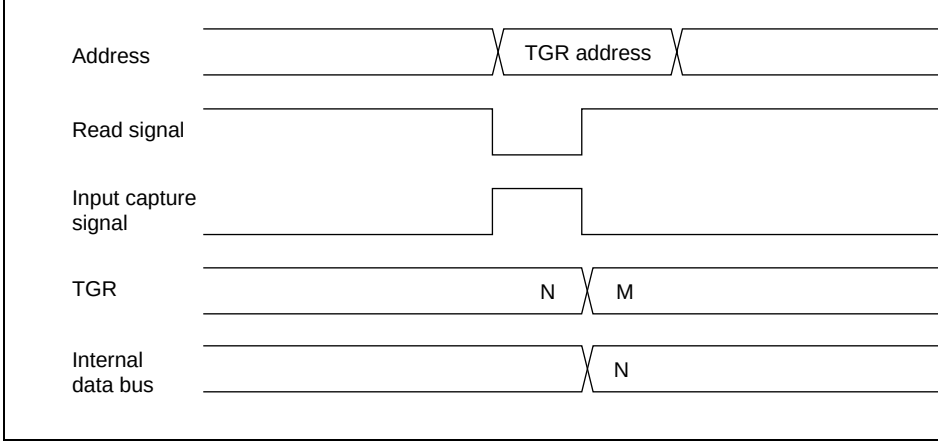


Figure 16.49 Contention between TGR Read and Input Capture

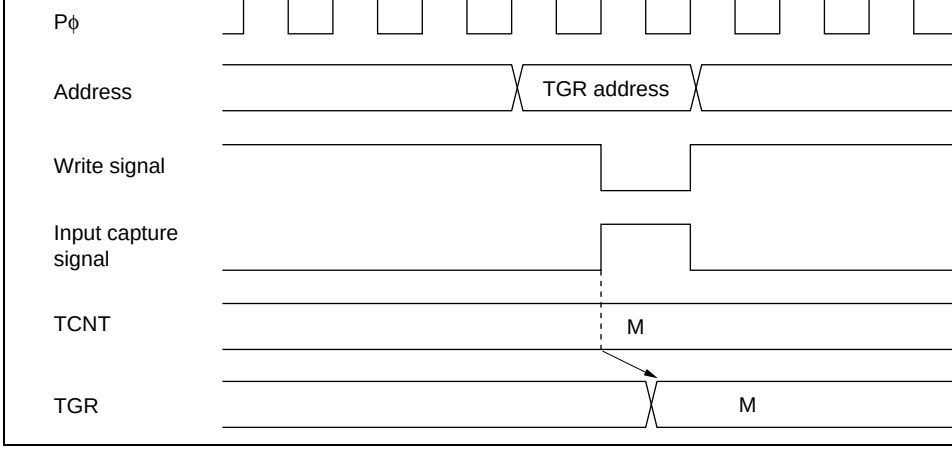


Figure 16.50 Contention between TGR Write and Input Capture

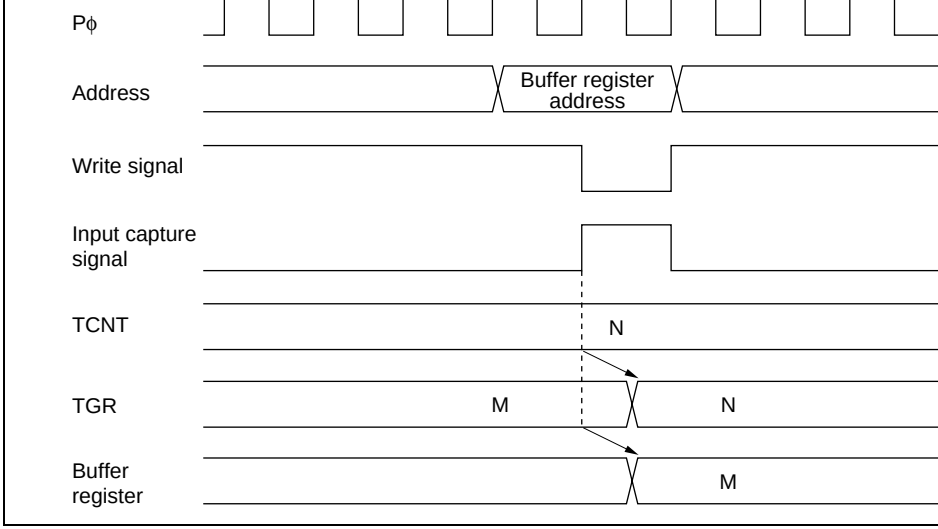


Figure 16.51 Contention between Buffer Register Write and Input Cap

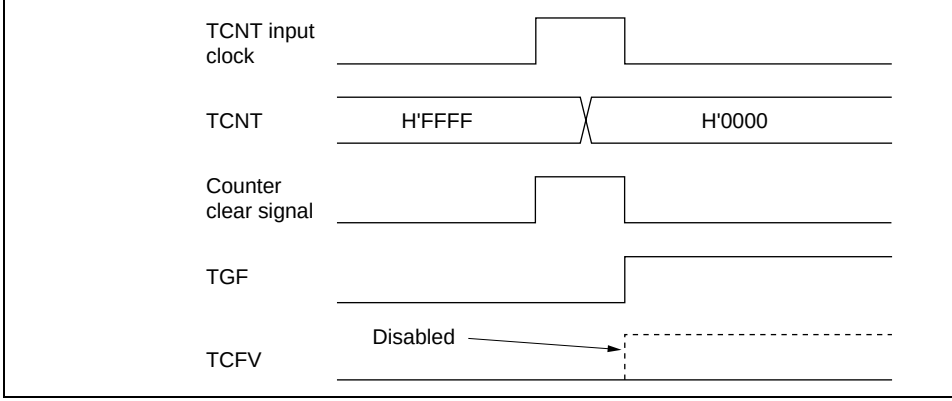


Figure 16.52 Contention between Overflow and Counter Clearing

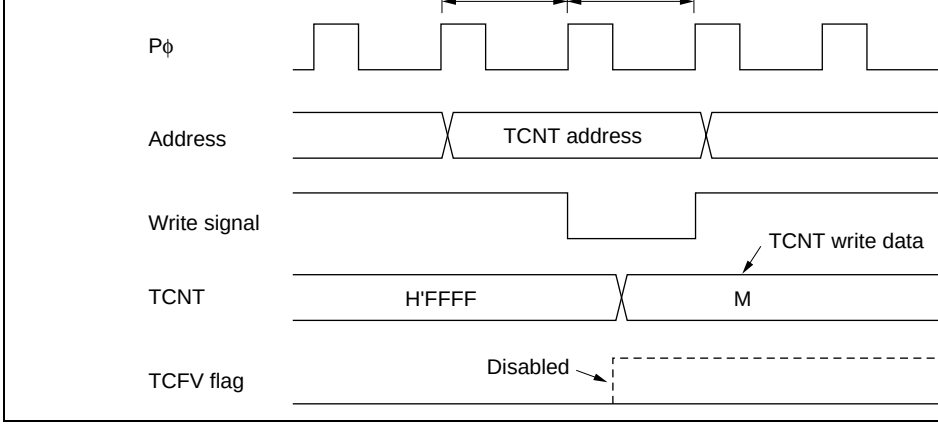


Figure 16.53 Contention between TCNT Write and Overflow

Multiplexing of I/O Pins: In the Chip, the TCLKA input pin is multiplexed with the
pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the T
pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is in
compare match output should not be performed from a multiplexed pin.

Interrupts and Module Stop Mode: If module stop mode is entered when an interru
requested, it will not be possible to clear the CPU interrupt source or DMAC activatio
Interrupts should therefore be disabled before entering module stop mode.

Either of the following measures should therefore be taken when clearing flags in TSR0

1. Execute clearing while the TPU timer is counting up.
2. If clearing when the TPU timer is stopped, write 0 to the flag again after executing c

16.8.2 DMA Transfer by TPU0

When DMA transfer is performed by means of TPU channel 0 compare match or input internal logic interrupt requests (transfer requests) may not be cleared correctly. Theref not be possible to execute DMA transfer when a subsequent transfer request is generate channel 0 compare match or input capture.

Either of the following measures should therefore be taken when performing DMA tran means of TPU channel 0 compare match or input capture.

1. Do not set on-chip RAM as the DMA transfer source or destination.
2. When on-chip RAM has not been set as the DMA transfer source or destination, exe transfer while the TPU channel 0 timer is counting up.

17.1.1 Features

The H-UDI has the following features conforming to the IEEE 1149.1 standard.

- Five test signals (TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$)
- TAP controller
- Instruction register
- Data register
- Bypass register
- Boundary scan register

The H-UDI has seven instructions.

- Bypass mode
  Test mode conforming to IEEE 1149.1
- EXTEST mode
  Test mode corresponding to IEEE1149.1.
- SAMPLE/PRELOAD mode
  Test mode corresponding to IEEE1149.1.
- CLAMP mode
  Test mode corresponding to IEEE1149.1.
- HIGHZ mode
  Test mode corresponding to IEEE1149.1.
- IDCODE mode
  Test mode corresponding to IEEE1149.1.
- H-UDI interrupt
  H-UDI interrupt request to INTC

This chip does not support test modes other than bypass mode.

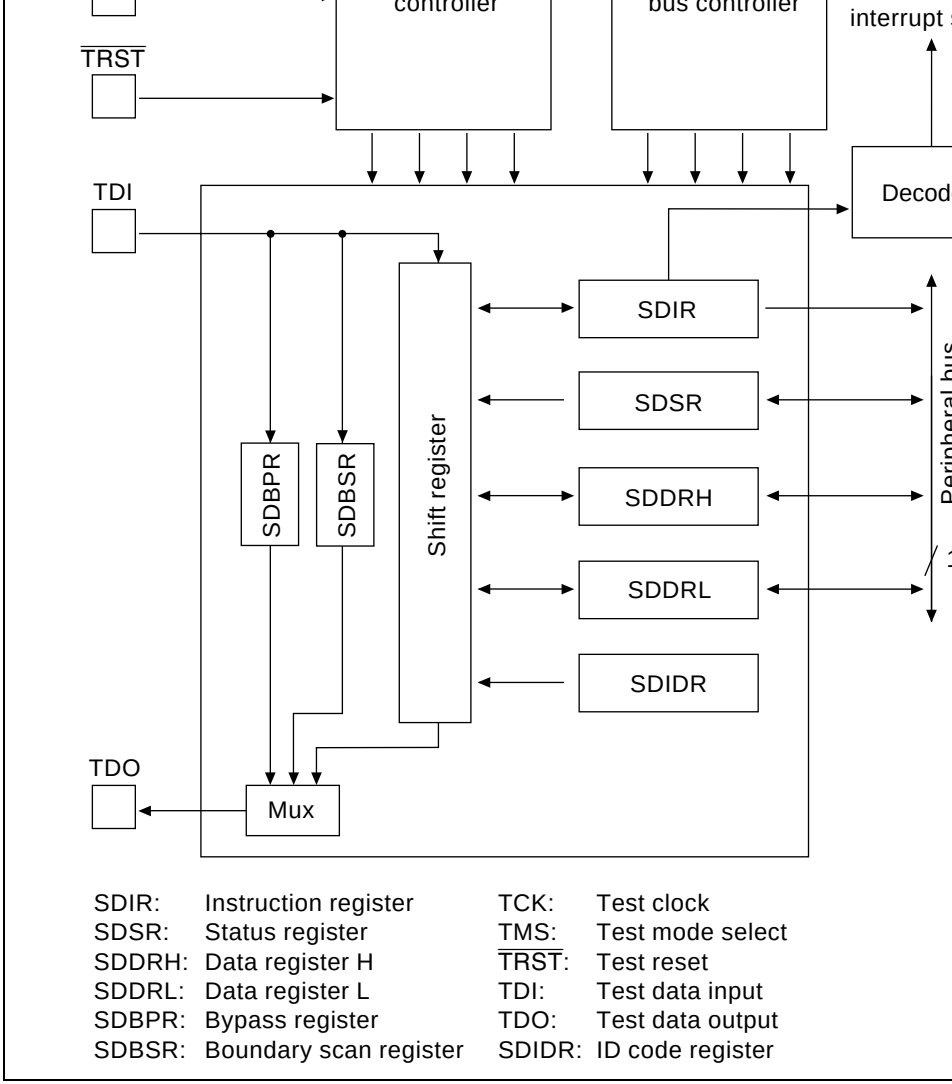


Figure 17.1 H-UDI Block Diagram

Test data input	TDI	Input	Serial data input
Test data output	TDO	Output	Serial data output
Test reset	$\overline{\text{TRST}}$	Input	Test reset input signal

17.1.4 Register Configuration

Table 17.2 shows the H-UDI registers.

Table 17.2 Register Configuration

Register	Abbreviation	R/W*1	Initial Value*2	Address	A (
Instruction register	SDIR	R	H'E000	H'FFFFFCB0	8
Status register	SDSR	R/W	H'0401	H'FFFFFCB2	8
Data register H	SDDRH	R/W	Undefined	H'FFFFFCB4	8
Data register L	SDDRL	R/W	Undefined	H'FFFFFCB6	8
Bypass register	SDBPR	—	—	—	–
Boundary scan register	SDBSR	—	—	—	–
ID code register	SDIDR	—	H'0101000F	—	–

Notes: 1. Indicates whether the register can be read/written to by the CPU.
2. Initial value when the $\overline{\text{TRST}}$ signal is input. Registers are not initialized by a (power-on or manual) or in standby mode.

Instructions and data can be input to the instruction register (SDIR) and data register (serial transfer from the test data input pin (TDI). Data from SDIR, the status register (SDDR can be output via the test data output pin (TDO). The bypass register (SDBPR) register to which TDI and TDO are connected in bypass mode. The boundary scan reg (SDBSR) is a 330-bit register, and is connected to TDI and TDO in the SAMPLE/PRI EXTEST mode. The ID code register (SDIDR) is a 32-bit register; a fixed code can be TDO in the IDCODE mode. All registers, except SDBPR, SDBSR, and SDIDR, can b from the CPU.

SDBPR	Possible	Possible
SDBSR	Possible	Possible
SDIDR	Impossible	Possible

17.2 External Signals

17.2.1 Test Clock (TCK)

The test clock pin (TCK) provides an independent clock supply to the H-UDI. As the cl
to TCK is supplied directly to the H-UDI, a clock waveform with a duty cycle close to
be input (for details, see section 21, Electrical Characteristics). If no clock is input, TCI
1 by internal pull-up.

17.2.2 Test Mode Select (TMS)

The test mode select pin (TMS) is sampled on the rise of TCK. TMS controls the intern
the TAP controller. If no signal is input, TMS is fixed at 1 by internal pull-up.

17.2.3 Test Data Input (TDI)

The test data input pin (TDI) performs serial input of instructions and data for H-UDI r
TDI is sampled on the rise of TCK. If no signal is input, TDI is fixed at 1 by internal pu

17.2.4 Test Data Output (TDO)

The test data output pin (TDO) performs serial output of instructions and data from H-U
registers. Transfer is performed in synchronization with TCK. If there is no output, TD
the high-impedance state.

Bit:	15	14	13	12	11	10	9
	TS3	TS2	TS1	TS0	—	—	—
Initial value:	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

The instruction register (SDIR) is a 16-bit register that can only be read by the CPU. F
instructions can be transferred to SDIR by serial input from TDI. SDIR can be initializ
$\overline{\text{TRST}}$ signal, but is not initialized by a reset or in standby mode.

SDIR defines 4 valid bits for instruction. If an instruction exceeding 4 bits is input, the
of the serial data will be stored in SDIR.

Operation is not guaranteed if a reserved instruction is set in this register.

Bits 15 to 12—Test Set Bits (TS3 to TS0): Table 17.4 shows the instruction configura

			1	Reserved	
		1	0	Reserved	
			1	Reserved	
1	0	0	0	Reserved	
			1	Reserved	
		1	0	H-UDI interrupt	
			1	Reserved	
	1	0	0	Reserved	
			1	Reserved	
		1	0	IDCODE mode	(Initi
			1	BYPASS mode	

Bits 11 to 0—Reserved: These bits are always read as 0. The write value should always

	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

The status register (SDSR) is a 16-bit register that can be read and written to by the C
from TDO is possible for SDSR, but serial data cannot be written to SDSR via TDI. T
bit is output by means of a 1-bit shift. In the case of a 2-bit shift, the SDTRF bit is firs
followed by a reserved bit.

SDSR is initialized by $\overline{\text{TRST}}$ signal input, but is not initialized by a reset or in standby

Bits 15 to 1—Reserved: Bits 15 to 11 and 9 to 1 are always read as 0, and the write va
always be 0. Bit 10 is always read as 1, and the write value should always be 1.

Bit 0—Serial Data Transfer Control Flag (SDTRF): Indicates whether H-UDI register
accessed by the CPU. The SDTRF bit is reset by the $\overline{\text{TRST}}$ signal , but is not initialize
or in standby mode.

Bit 0: SDTRF	**Description**
0	Serial transfer to SDDR has ended, and SDDR can be accessed
1	Serial transfer to SDDR in progress (

Bit:	7	6	5	4	3	2	1
Initial value:	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SDDRH and SDDRL are 16-bit registers that can be read and written to by the CPU. SI
connected to TDO and TDI for serial data transfer to and from an external device.

32-bit data is input and output in serial data transfer. If data exceeding 32 bits is input, c
last 32 bits will be stored in SDDR. Serial data is input starting from the MSB of SDDR
SDDRH), and output starting from the LSB (bit 0 of SDDRL).

This register is not initialized by a reset, in standby mode, or by the $\overline{\text{TRST}}$ signal.

17.3.4 Bypass Register (SDBPR)

The bypass register (SDBPR) is a one-bit shift register. In bypass mode, CLAMP mode
HIGHZ mode, SDBPR is connected to TDI and TDO, and the chip is excluded from th
when a boundary scan test is conducted. SDBPR cannot be read or written to by the CP

17.3.5 Boundary Scan Register (SDBSR)

The boundary scan register (SDBSR), a shift register that controls the I/O terminals of t
provided on the PAD.

Using the EXTEST mode or the SAMPLE/PRELOAD mode, a boundary scan test conf
the IEEE1149.1 standard can be performed.

For SDBSR, read/write by the CPU cannot be performed.

Table 17.5 shows the relationship between the terminals of the LSI and the boundary sc

		Output	325
		Output enable	324
37	D2	Input	323
		Output	322
		Output enable	321
38	D3	Input	320
		Output	319
		Output enable	318
39	D4	Input	317
		Output	316
		Output enable	315
40	D5	Input	314
		Output	313
		Output enable	312
41	D6	Input	311
		Output	310
		Output enable	309
43	D7	Input	308
		Output	307
		Output enable	306
44	D8	Input	305
		Output	304
		Output enable	303
46	D9	Input	302
		Output	301
		Output enable	300

		Output	292
		Output enable	291
51	D13	Input	290
		Output	289
		Output enable	288
53	D14	Input	287
		Output	286
		Output enable	285
54	D15	Input	284
		Output	283
		Output enable	282
55	D16	Input	281
		Output	280
		Output enable	279
56	D17	Input	278
		Output	277
		Output enable	276
57	D18	Input	275
		Output	274
		Output enable	273
59	D19	Input	272
		Output	271
		Output enable	270

		Output	262
		Output enable	261
65	D23	Input	260
		Output	259
		Output enable	258
68	D24	Input	257
		Output	256
		Output enable	255
70	D25	Input	254
		Output	253
		Output enable	252
71	D26	Input	251
		Output	250
		Output enable	249
72	D27	Input	248
		Output	247
		Output enable	246
73	D28	Input	245
		Output	244
		Output enable	243
74	D29	Input	242
		Output	241
		Output enable	240

		Output enable	232
82	A1	Output	231
		Output enable	230
83	A2	Output	229
		Output enable	228
84	A3	Output	227
		Output enable	226
85	A4	Output	225
		Output enable	224
86	A5	Output	223
		Output enable	222
87	A6	Output	221
		Output enable	220
88	A7	Output	219
		Output enable	218
90	A8	Output	217
		Output enable	216
92	A9	Output	215
		Output enable	214
93	A10	Output	213
		Output enable	212
94	A11	Output	211
		Output enable	210

		Output enable	202
100	A16	Output	201
		Output enable	200
102	A17	Output	199
		Output enable	198
103	A18	Output	197
		Output enable	196
104	A19	Output	195
		Output enable	194
105	A20	Output	193
		Output enable	192
106	A21	Output	191
		Output enable	190
107	A22	Output	189
		Output enable	188
108	A23	Output	187
		Output enable	186
111	A24	Output	185
		Output enable	184
115	$\overline{\text{WAIT}}$	Input	183
117	$\overline{\text{RAS}}$	Output	182
		Output enable	181
118	$\overline{\text{CAS}}$	Output	180
		Output enable	179

		Output enable	171
123	$\overline{\text{CAS3}}$	Output	170
		Output enable	169
124	$\overline{\text{CAS2}}$	Output	168
		Output enable	167
125	$\overline{\text{CAS1}}$	Output	166
		Output enable	165
126	$\overline{\text{CAS0}}$	Output	164
		Output enable	163
127	CKE	Output	162
		Output enable	161
128	$\overline{\text{RD}}$	Output	160
		Output enable	159
129	REFOUT	Output	158
		Output enable	157
131	$\overline{\text{BS}}$	Output	156
		Output enable	155
133	RD/$\overline{\text{WR}}$	Output	154
		Output enable	153
134	$\overline{\text{CS0}}$	Output	152
		Output enable	151
135	$\overline{\text{CS1}}$	Output	150
		Output enable	149

140	$\overline{BH}$	Output	141
		Output enable	140
141	DREQ1	Input	139
142	DREQ0	Input	138
143	DACK1	Output	137
		Output enable	136
144	DACK0	Output	135
		Output enable	134
145	$\overline{BRLS}$	Input	133
148	$\overline{BGR}$	Output	132
		Output enable	131
151	PB15	Input	130
		Output	129
		Output enable	128
152	PB14	Input	127
		Output	126
		Output enable	125
153	PB13	Input	124
		Output	123
		Output enable	122
154	PB12	Input	121
		Output	120
		Output enable	119

		Output	111
		Output enable	110
160	PB8	Input	109
		Output	108
		Output enable	107
161	PB7	Input	106
		Output	105
		Output enable	104
162	PB6	Input	103
		Output	102
		Output enable	101
163	PB5	Input	100
		Output	99
		Output enable	98
164	PB4	Input	97
		Output	96
		Output enable	95
165	PB3	Input	94
		Output	93
		Output enable	92
166	PB2	Input	91
		Output	90
		Output enable	89

		Output	81
		Output enable	80
172	PA12	Input	79
		Output	78
		Output enable	77
173	PA11	Input	76
		Output	75
		Output enable	74
174	PA10	Input	73
		Output	72
		Output enable	71
175	PA9	Input	70
		Output	69
		Output enable	68
176	PA8	Input	67
		Output	66
		Output enable	65
177	PA7	Input	64
		Output	63
		Output enable	62
178	PA6	Input	61
		Output	60
		Output enable	59

		Output enable	51
184	PA2	Input	50
		Output	49
		Output enable	48
185	PA1	Input	47
		Output	46
		Output enable	45
186	PA0	Input	44
		Output	43
		Output enable	42
187	RX–ER	Input	41
188	RX–DV	Input	40
189	COL	Input	39
190	CRS	Input	38
192	RX–CLK	Input	37
194	ERXD0	Input	36
195	ERXD1	Input	35
196	ERXD2	Input	34
197	ERXD3	Input	33
198	MDIO	Input	32
		Output	31
		Output enable	30
199	MDC	Output	29
		Output enable	28

206	ETXD2	Output	20
		Output enable	19
207	ETXD3	Output	18
		Output enable	17
208	TX–ER	Output	16
		Output enable	15
1	$\overline{\text{IRL3}}$	Input	14
2	$\overline{\text{IRL2}}$	Input	13
3	$\overline{\text{IRL1}}$	Input	12
4	$\overline{\text{IRL0}}$	Input	11
5	NMI	Input	10
13	MD4	Input	9
14	MD3	Input	8
15	MD2	Input	7
16	MD1	Input	6
17	MD0	Input	5
24	$\overline{\text{CKPREQ}}$/CKM	Input	4
25	$\overline{\text{CKPACK}}$	Output	3
		Output enable	2
27	$\overline{\text{IVECF}}$	Output	1
		Output enable	0
to TDO			

Note: The output enable signals are active-low. When an output enable signal is drive
corresponding pin is driven. The exception is the output enable signal for the M
which is active-high.

(4 bits)	(16 bits)	(11 bits)	

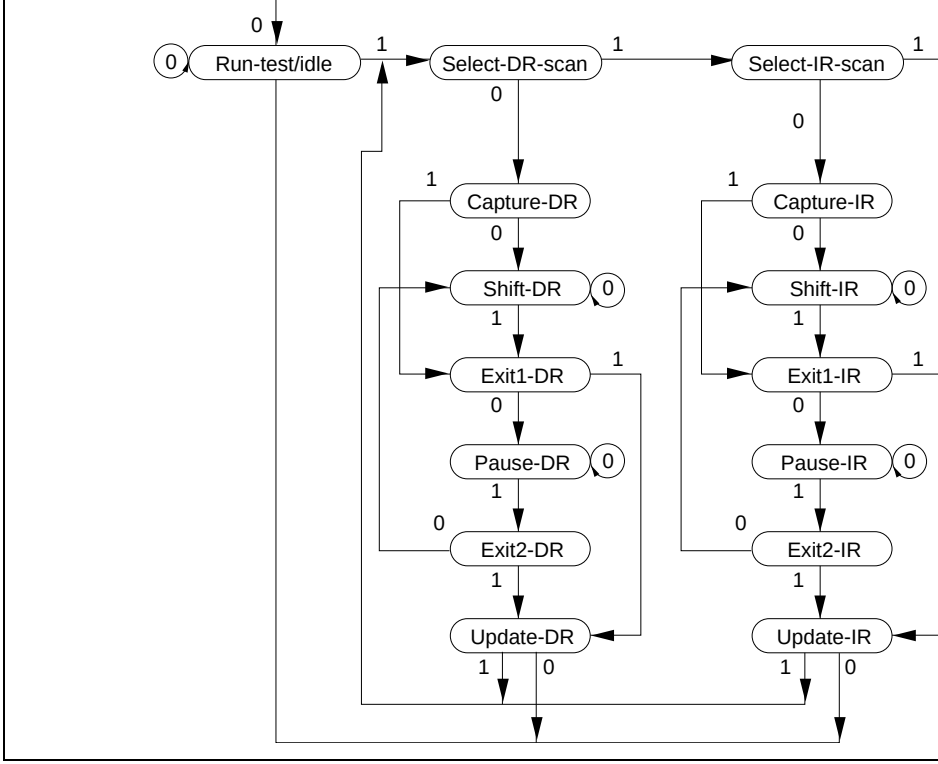
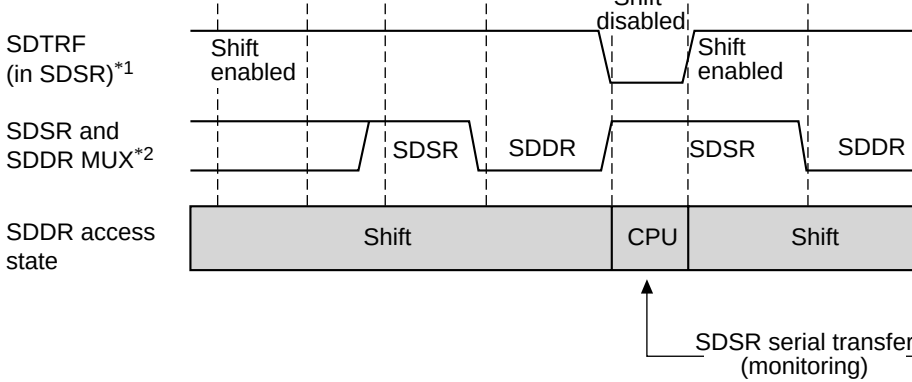


Figure 17.2 TAP Controller State Transitions

The H-UDI interrupt and serial transfer procedure is as follows.

1. An instruction is input to SDIR by serial transfer, and an H-UDI interrupt request is
2. After the H-UDI interrupt request is issued, the SDTRF bit in SDSR is monitored e
   After output of SDTRF = 1 from TDO is observed, serial data is transferred to SDD
3. On completion of the serial transfer to SDDR, the SDTRF bit is cleared to 0, and SI
   accessed by the CPU. After SDDR has been accessed, SDDR serial transfer is enab
   setting the SDTRF bit to 1 in SDSR.
4. Serial data transfer between an external device and the H-UDI can be carried out by
   monitoring the SDTRF bit in SDSR externally and internally.

Figures 17.3, 17.4, and 17.5 show the timing of data transfer between an external devic
H-UDI.



Notes: 1. SDTRF flag (in SDSR): Indicates whether SDDR access by the CPU or se
data input/output to SDDR is possible.

1	SDDR is shift-disabled. SDDR access by the CPU is enabled.
2	SDDR is shift-enabled. Do not access SDDR until SDTRF = 0.

Conditions:
- SDTRF = 1
  — When $\overline{\text{TRST}}$ = 0
  — When the CPU writes 1
  — In bypass mode
- SDTRF = 0
  — End of SDDR shift access in serial transfer

2. SDSR/SDDR (Update-DR state) internal MUX switchover timing
   - Switchover from SDSR to SDDR: On completion of serial transfer in whi
     SDTRF = 1 is output from TDO
   - Switchover from SDDR to SDSR: On completion of serial transfer to SD

Figure 17.3 Data Input/Output Timing Chart (1)

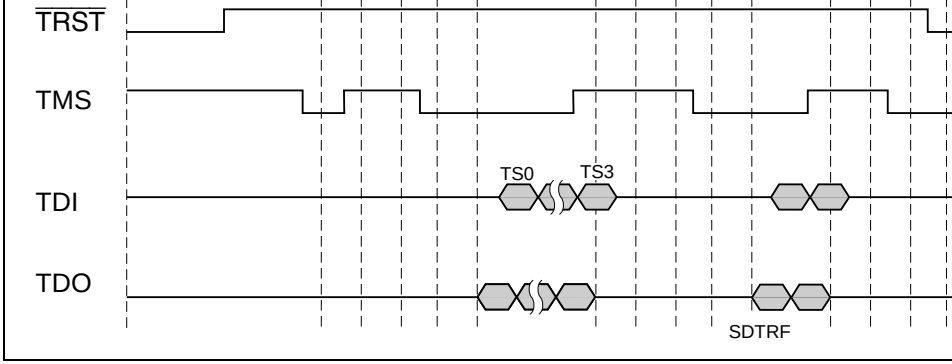


Figure 17.4 Data Input/Output Timing Chart (2)

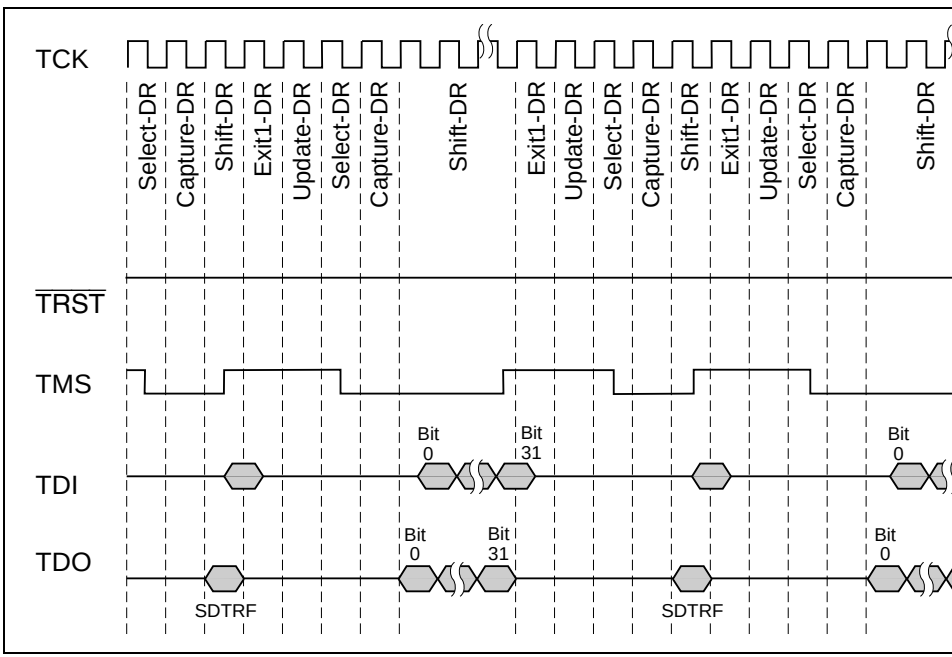


Figure 17.5 Data Input/Output Timing Chart (3)

The H-UDI pins can be placed in the boundary scan mode stipulated by IEEE1149.1 b
command in SDIR.

17.5.1 Supported Instructions

The SH7615 supports the three essential instructions defined in IEEE1149.1 (BYPAS
SAMPLE/PRELOAD, and EXTEST) and optional instructions (CLAMP, HIGHZ, an

BYPASS: The BYPASS instruction is an essential standard instruction that operates t
register. This instruction shortens the shift path to speed up serial data transfer involvi
chips on the printed circuit board. While this instruction is executing, the test circuit h
on the system circuits. The instruction code is 1111.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction inputs values from the
internal circuitry to the boundary scan register, outputs values from the scan path, and
onto the scan path. When this instruction is executing, the SH7615's input pin signals
transmitted directly to the internal circuitry, and internal circuit values are directly out
externally from the output pins. The SH7615's system circuits are not affected by exe
instruction. The instruction code is 0100.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to t
circuitry, or a value to be transferred from the internal circuitry to an output pin, is lat
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
normal operation of the SH7615.

In a PRELOAD operation, an initial value is set in the parallel output latch of the bou
register from the scan path prior to the EXTEST instruction. Without a PRELOAD op
when the EXTEST instruction was executed an undefined value would be output from
pin until completion of the initial scan sequence (transfer to the output latch) (with the
instruction, the parallel output latch value is constantly output to the output pin).

EXTEST: This instruction is provided to test external circuitry when the SH7615 is n
printed circuit board. When this instruction is executed, output pins are used to output

CLAMP: When the CLAMP instruction is enabled, the output pin outputs the value of boundary scan register that has been set by the SAMPLE/PRELOAD instruction. While CLAMP instruction is enabled, the state of the boundary scan register maintains the pre regardless of the state of the TAP controller.

A bypass register is connected between TDI and TDO. The related circuit operates in th way when the BYPASS instruction is enabled.

The instruction code is 0010.

HIGHZ: When the HIGHZ instruction is enabled, all output pins enter a high-impedan While the HIGHZ instruction is enabled, the state of the boundary scan register maintai previous state regardless of the state of the TAP controller.

A bypass register is connected between TDI and TDO. The related circuit operates in th way when the BYPASS instruction is enabled.

The instruction code is 0011.

IDCODE: When the IDCODE instruction is enabled, the value of the ID code register from TDO with LSB first when the TAP controller is in the Shift-DR state. While this i is being executed, the test circuit does not affect the system circuit.

When the TAP controller is in the Test-Logic-Reset state, the instruction register is init the IDCODE instruction.

The instruction code is 1110.

17.5.2 Notes on Use

1. Boundary scan mode does not cover clock-related signals (EXTAL, XTAL, CKIO, CAP2).
2. Boundary scan mode does not cover reset-related signals ($\overline{\text{RES}}$, ASEMODE).
3. Boundary scan mode does not cover H-UDI-related signals (TCK, TDI, TDO, TMS
4. Fix the ASEMODE pin high.

- In data transfer, data input/output starts with the LSB. Figure 17.6 shows serial dat
input/output.
- When data that exceeds the number of bits of the register connected between TDI
serially transferred, the serial data that exceeds the number of register bits and outp
TDO is the same as that input from TDI.
- If the H-UDI serial transfer sequence is disrupted, a $\overline{\text{TRST}}$ reset must be executed.
should then be retried, regardless of the transfer operation.
- TDO is output at the falling edge of TCK when one of six instructions defined in I
is selected. Otherwise, it is output at the rising edge of TCK.

- SDIR and SDSR serial data input/output

In Capture-IR, SDIR and SDSR are captured into the shift register, and in Shift-IR bits 0 to 15 o
bits 0 to 15 of SDIR are output in that order from TDO.
In Update-IR, data input from TDI is written to SDIR, but not to SDSR.

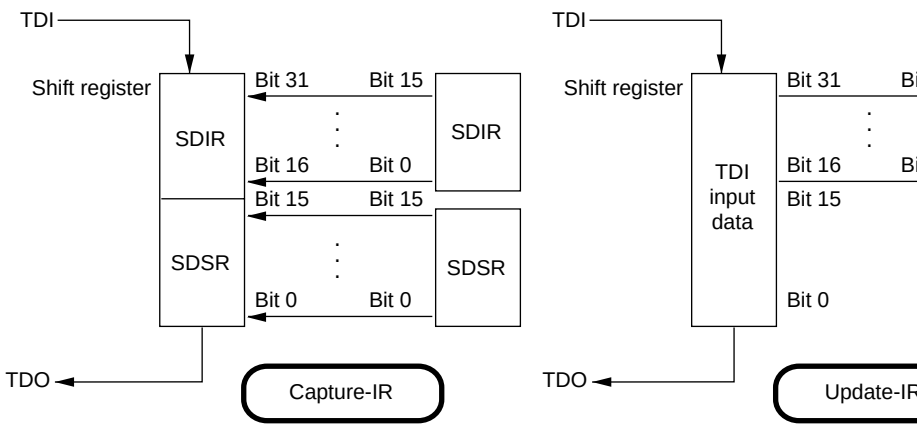
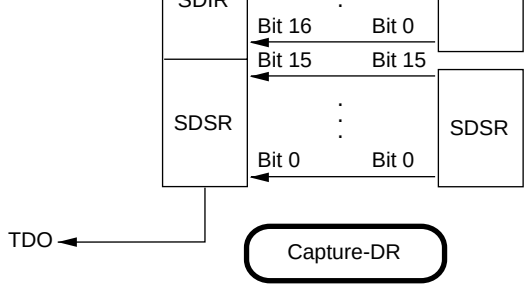


Figure 17.6 Serial Data Input/Output (1)



(2) In H-UDI interrupt mode, after SDTRF = 1 is read from TDO when an H-UDI interrupt is gen
SDDRH and SDDRL are captured into the shift register in Capture-DR, and in Shift-DR bits
SDDRL and bits 0 to 15 of SDDRH are output in that order from TDO.
Data input from TDI is written to SDDRH and SDDRL in Update-DR.

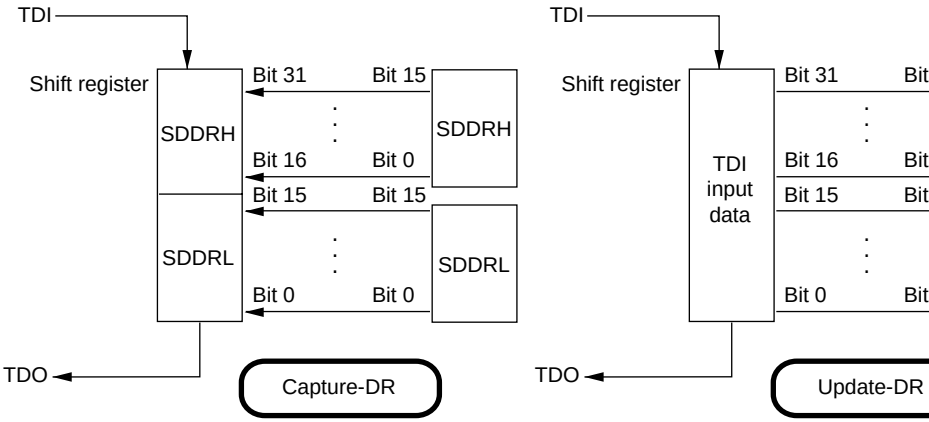


Figure 17.6 Serial Data Input/Output (2)

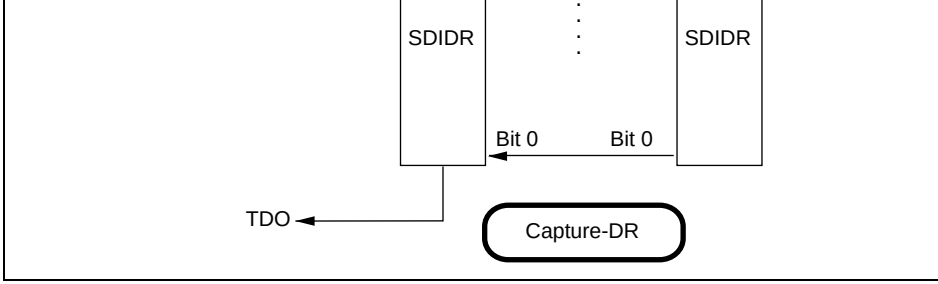


Figure 17.6 Serial Data Input/Output (3)

A	PA9	I/O	Port	STS0	I/O	SIO0	—	—	—	—
A	PA8	I/O	Port	STXD0	O	SIO0	—	—	—	—
A	$\overline{\text{WDTOVF}}$	O	WDT	PA7	I/O	Port	—	—	—	—
A	PA6	I/O	Port	FTCI	I	FRT	—	—	—	—
A	PA5	I/O	Port	FTI	I	FRT	—	—	—	—
A	PA4	I/O	Port	FTOA	O	FRT	—	—	—	—
A	CKPO	O	Port	FTOB	O	FRT	—	—	—	—
A	PA2	I/O	Port	LNKSTA	I	EtherC	—	—	—	—
A	PA1	I/O	Port	EXOUT	O	EtherC	—	—	—	—
A	PA0	I/O	Port	—	—	—	—	—	—	—
B	PB15	I/O	Port	—	—	—	SCK1	I/O	SCIF1	—
B	PB14	I/O	Port	—	—	—	RXD1	I	SCIF1	—
B	PB13	I/O	Port	—	—	—	TXD1	O	SCIF1	—
B	PB12	I/O	Port	SRCK2	I	SIO2	$\overline{\text{RTS}}$	O	SCIF1	STATS1
B	PB11	I/O	Port	SRS2	I	SIO2	$\overline{\text{CTS}}$	I	SCIF1	STATS0
B	PB10	I/O	Port	SRXD2	I	SIO2	TIOCA1	I/O	TPU1	—
B	PB9	I/O	Port	STCK2	I	SIO2	TIOCB1	I/O	TPU1	—
B	PB8	I/O	Port	STS2	I/O	SIO2	TIOCA2	I/O	TPU2	—
B	PB7	I/O	Port	STXD2	O	SIO2	TIOCB2	I/O	TPU2	—
B	PB6	I/O	Port	SRCK1	I	SIO1	SCK2	I/O	SCIF2	—
B	PB5	I/O	Port	SRS1	I	SIO1	RXD2	I	SCIF2	—
B	PB4	I/O	Port	SRXD1	I	SIO1	TXD2	O	SCIF2	—
B	PB3	I/O	Port	STCK1	I	SIO1	TIOCA0	I/O	TPU0	—
B	PB2	I/O	Port	STS1	I/O	SIO1	TIOCB0	I/O	TPU0	—
B	PB1	I/O	Port	STXD1	O	SIO1	TIOCC0	I/O	TPU0	—
B	PB0	I/O	Port	—	—	—	TIOCD0	I/O	TPU0	WOL

Notes: In the initial state, function 1 is selected.

* The initial value is "input."

The figures in brackets indicate the settings of the mode bits (MD1, MD0) in
select multiplexed functions in port A[0:13] and port B[0:15].

Port B control register	PBCR	R/W	H'0000	H'FFFFFC88	8
Port B I/O register	PBIOR	R/W	H'0000	H'FFFFFC8A	8
Port B control register 2	PBCR2	R/W	H'0000	H'FFFFFC8E	8

18.3 Register Descriptions

18.3.1 Port A Control Register (PACR)

Bit:	15	14	13	12	11	10	9
	—	—	PA13MD	PA12MD	PA11MD	PA10MD	PA9M
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PA7MD	PA6MD	PA5MD	PA4MD	PA3MD	PA2MD	PA1M
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port A control register (PACR) is a 16-bit read/write register that selects the funct
14 multiplex pins in port A.

PACR is initialized to H'0000 by a power-on reset. It is not initialized by a manual res
standby mode or sleep mode.

Bits 15 and 14—Reserved: These bits are always read as 0. The write value should alw

0	General input/output (PA12)	(Ir
1	SIO0 serial receive synchronous input (SRS0)	

Bit 11—PA11 Mode Bit (PA11MD): Selects the function of pin PA11/SRXD0.

Bit 11: PA11MD	Description	
0	General input/output (PA11)	(Ir
1	SIO0 serial receive data (SRXD0)	

Bit 10—PA10 Mode Bit (PA10MD): Selects the function of pin PA10/STCK0.

Bit 10: PA10MD	Description	
0	General input/output (PA10)	(Ir
1	SIO0 serial transmit clock (STCK0)	

Bit 9—PA9 Mode Bit (PA9MD): Selects the function of pin PA9/STS0.

Bit 9: PA9MD	Description	
0	General input/output (PA9)	(Ir
1	SIO0 serial transmit synchronous input/output (STS0)	

Bit 8—PA8 Mode Bit (PA8MD): Selects the function of pin PA8/STXD0.

Bit 8: PA8MD	Description	
0	General input/output (PA8)	(Ir
1	SIO0 serial transmit data output (STXD0)	

Bit 6—PA6 Mode Bit (PA6MD): Selects the function of pin PA6/FTCI.

Bit 6: PA6MD	Description	
0	General input/output (PA6)	(
1	FRT clock input (FTCI)	

Bit 5—PA5 Mode Bit (PA5MD): Selects the function of pin PA5/FTI.

Bit 5: PA5MD	Description	
0	General input/output (PA5)	(
1	FRT input capture input (FTI)	

Bit 4—PA4 Mode Bit (PA4MD): Selects the function of pin PA4/FTO4.

Bit 4: PA4MD	Description	
0	General input/output (PA4)	(
1	FRT output compare output (FTOA)	

Bit 3—PA3 Mode Bit (PA3MD): Selects the function of pin CKPO/FTOB.

Bit 3: PA3MD	Description	
0	Peripheral module clock output (CKPO)	(
1	FRT output compare output (FTOB)	

Bit 2—PA2 Mode Bit (PA2MD): Selects the function of pin PA2/LNKSTA.

Bit 2: PA2MD	Description	
0	General input/output (PA2)	(
1	EtherC rink status input (LNKSTA)	

0	General input/output (PA0)	(In
1	Reserved	

18.3.2 Port A I/O Register (PAIOR)

Bit:	15	14	13	12	11	10	9
	—	—	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PA7IOR	PA6IOR	PA5IOR	PA4IOR	—	PA2IOR	PA1IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W

The port A I/O register (PAIOR) is a 16-bit read/write register that selects the input/out direction of the 14 multiplex pins in port A. Bits PA13IOR to PA4IOR and PA2IOR to correspond to individual pins in port A. PAIOR is enabled when port A pins function as input pins (PA13 to PA4 and PA2 to PA0), and disabled otherwise. When port A pins f PA13 to PA0, a pin becomes an output when the corresponding bit in PAIOR is set to 1 input when the bit is cleared to 0.

PAIOR is initialized to H'0000 by a power-on reset. It is not initialized by a manual res standby mode or sleep mode.

Port B Control Register (PBCR)

Bit:	15	14	13	12	11	10	9
	PB15 MD1	PB15 MD0	PB14 MD1	PB14 MD0	PB13 MD1	PB13 MD0	PB1 MD1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PB11 MD1	PB11 MD0	PB10 MD1	PB10 MD0	PB9 MD1	PB9 MD0	PB8 MD1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 14—PB15 Mode Bits 1 and 0 (PB15MD1, PB15MD0): These bits select t
of pin PB15/SCK1.

Bit 15: PB15MD1	Bit 14: PB15MD0	Description
0	0	General input/output (PB15) (
	1	Reserved
1	0	SCIF1 serial clock input/output (SCK1)
	1	Reserved

Bits 13 and 12—PB14 Mode Bits 1 and 0 (PB14MD1, PB14MD0): These bits select t
of pin PB14/RXD1.

Bit 13: PB14MD1	Bit 12: PB14MD0	Description
0	0	General input/output (PB14) (
	1	Reserved
1	0	SCIF1 serial data input (RXD1)
	1	Reserved

Bits 9 and 8—PB12 Mode Bits 1 and 0 (PB12MD1, PB12MD0): These bits select the f
pin PB12/SRCK2/$\overline{\text{RTS}}$/STATS1.

Bit 9: PB12MD1	Bit 8: PB12MD0	Description
0	0	General input/output (PB12) (In
	1	SIO2 serial receive clock input (SRCK2)
1	0	SCIF1 transmit request ($\overline{\text{RTS}}$)
	1	BSC status 1 output (STATS1)

Bits 7 and 6—PB11 Mode Bits 1 and 0 (PB11MD1, PB11MD0): These bits select the f
pin PB11/SRS2/$\overline{\text{CTS}}$/STATS0.

Bit 7: PB11MD1	Bit 6: PB11MD0	Description
0	0	General input/output (PB11) (In
	1	SIO2 serial receive synchronous input (SR
1	0	SCIF1 transmit permission ($\overline{\text{CTS}}$)
	1	BSC status 0 output (STATS0)

Bits 5 and 4—PB10 Mode Bits 1 and 0 (PB10MD1, PB10MD0): These bits select the f
pin PB10/SRXD2/TIOCA1.

Bit 5: PB10MD1	Bit 4: PB10MD0	Description
0	0	General input/output (PB10) (In
	1	SIO2 serial receive data input (SRXD2)
1	0	TPU1 input capture input/output compare (TIOCA1)
	1	Reserved

Note: * Timer clock input C (TCLKC) is selected when the TPU phase counting mo
according to the setting of bits TPSC2 to TPSC0 in TCR.

Bits 1 and 0—PB8 Mode Bits 1 and 0 (PB8MD1, PB8MD0): These bits select the fun
PB8/STS2/TIOCA2.

Bit 1: PB8MD1	Bit 0: PB8MD0	Description
0	0	General input/output (PB8) (
	1	SIO2 serial transmit synchronous input/o
1	0	TPU2 input capture input/output compare (TIOCA2)
	1	Reserved

Port B Control Register 2 (PBCR2)

Bit:	15	14	13	12	11	10	9
	PB7 MD1	PB7 MD0	PB6 MD1	PB6 MD0	PB5 MD1	PB5 MD0	PB4 MD1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PB3 MD1	PB3 MD0	PB2 MD1	PB2 MD0	PB1 MD1	PB1 MD0	PB0 MD1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Timer clock input D (TCLKD) is selected when the TPU phase counting mod
according to the setting of bits TPSC2 to TPSC0 in TCR.

Bits 13 and 12—PB6 Mode Bits 1 and 0 (PB6MD1, PB6MD0): These bits select the fu
pin PB6/SRCK1/SCK2.

Bit 13: PB6MD1	Bit 12: PB6MD0	Description
0	0	General input/output (PB6) (Ir
	1	SIO1 serial receive clock input (SRCK1)
1	0	SCIF2 serial clock input/output (SCK2)
	1	Reserved

Bits 11 and 10—PB5 Mode Bits 1 and 0 (PB5MD1, PB5MD0): These bits select the fu
pin PB5/SRS1/RXD2.

Bit 11: PB5MD1	Bit 10: PB5MD0	Description
0	0	General input/output (PB5) (Ir
	1	SIO1 serial receive synchronous input (SR
1	0	SCIF2 serial data input (RXD2)
	1	Reserved

Bits 9 and 8—PB4 Mode Bits 1 and 0 (PB4MD1, PB4MD0): These bits select the func
PB4/SRXD1/TXD2.

Bit 9: PB4MD1	Bit 8: PB4MD0	Description
0	0	General input/output (PB4) (Ir
	1	SIO1 serial receive data input (SRXD1)
1	0	SCIF2 serial data output (TXD2)
	1	Reserved

Bits 5 and 4—PB2 Mode Bits 1 and 0 (PB2MD1, PB2MD0): These bits select the fun
PB2/STS1/TIOCB0.

Bit 5: PB2MD1	Bit 4: PB2MD0	Description
0	0	General input/output (PB2) (
	1	SIO1 serial transmit synchronous input/o
1	0	TPU0 input capture input/output compare (TIOCB0)
	1	Reserved

Bits 3 and 2—PB1 Mode Bits 1 and 0 (PB1MD1, PB1MD0): These bits select the fun
PB1/STXD1/TIOCC0/TCLKA.

Bit 3: PB1MD1	Bit 2: PB1MD0	Description
0	0	General input/output (PB1) (
	1	SIO1 serial transmit data output (STXD1)
1	0	TPU0 input capture input/output compare (TIOCC0)*
	1	Reserved

Note: * Timer clock input A (TCLKA) is selected when the TPU phase counting mo
according to the setting of bits TPSC2 to TPSC0 in TCR.

Note: * Timer clock input B (TCLKB) is selected when the TPU phase counting mod
according to the setting of bits TPSC2 to TPSC0 in TCR.

18.3.4 Port B I/O Register (PBIOR)

Bit:	15	14	13	12	11	10	9
	PB15 IOR	PB14 IOR	PB13 IOR	PB12 IOR	PB11 IOR	PB10 IOR	PB9 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

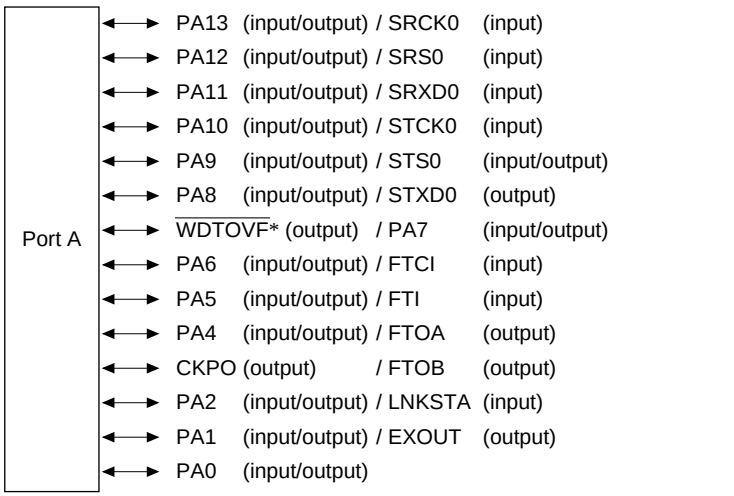
Bit:	7	6	5	4	3	2	1
	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port B I/O register (PBIOR) is a 16-bit read/write register that selects the input/out
direction of the 16 multiplex pins in port B. Bits PB15IOR to PB0IOR correspond to in
pins in port B. PBIOR is enabled when port B pins function as general input pins (PB15
and disabled otherwise. When port B pins function as PB15 to PB0, a pin becomes an o
the corresponding bit in PBIOR is set to 1, and an input when the bit is cleared to 0.

PBIOR is initialized to H'0000 by a power-on reset. It is not initialized by a manual res
standby mode or sleep mode.

19.2 Port A

Port A is an input/output port with the 14 pins shown in figure 19.1. Of the 14 pins, th
has no port data register bit, and is multiplexed as an internal clock pin.



Note: * The fact that the $\overline{\text{WDTOVF}}$ pin is set to output mode after a reset must be noted when it is to be used as a general I/O port (PA7).

Figure 19.1 Port A

19.2.2 Port A Data Register (PADR)

Bit:	15	14	13	12	11	10	9
	—	—	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PA7DR	PA6DR	PA5DR	PA4DR	—	PA2DR	PA1DR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W

The port A data register (PADR) is a 16-bit read/write register that stores port A data. B
and 3 are reserved: they always read 0, and the write value should always be 0. Bits PA
PA0DR correspond to pins PA13 to PA0. When a pin functions as a general output, if a
written to PADR, that value is output directly from the pin, and if PADR is read, the reg
is returned directly regardless of the pin state. When a pin functions as a general input,
read the pin state, not the register value, is returned directly. If a value is written to PAD
although that value is written into PADR it does not affect the pin state. Table 19.2 sum
port A data register read/write operations.

PADR is initialized to H'0000 by a power-on reset. It is not initialized by a manual rese
standby mode or sleep mode.

19.3 Port B

Port B is an input/output port with the 16 pins shown in figure 19.2.

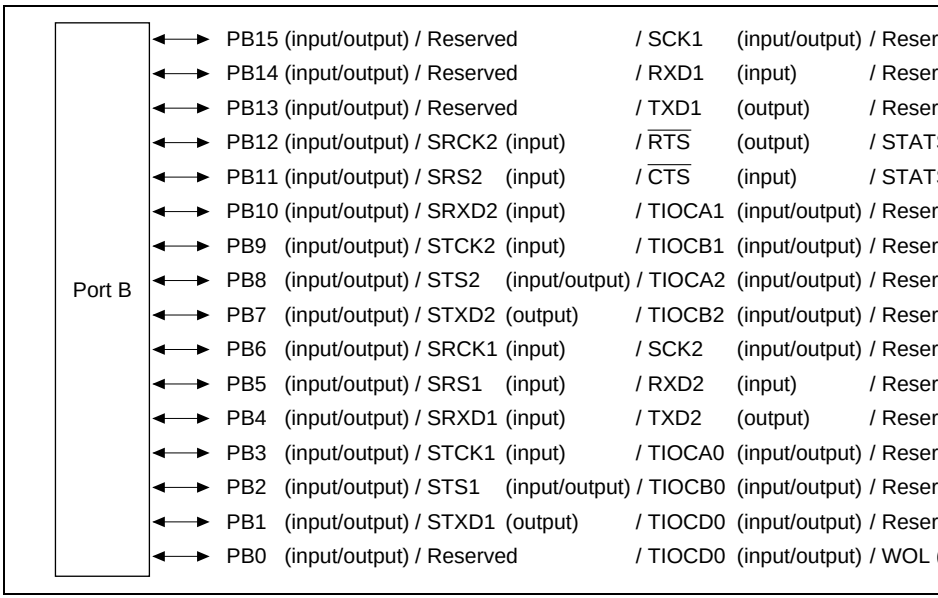


Figure 19.2 Port B

19.3.2 Port B Data Register (PBDR)

Bit:	15	14	13	12	11	10	9
	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port B data register (PBDR) is a 16-bit read/write register that stores port B data. B
PB15DR to PB0DR correspond to pins PB15 to PB0. When a pin functions as a genera
a value is written to PBDR, that value is output directly from the pin, and if PBDR is re
register value is returned directly regardless of the pin state. When a pin functions as a
input, if PBDR is read the pin state, not the register value, is returned directly. If a valu
to PBDR, although that value is written into PBDR it does not affect the pin state. Tabl
shows port B data register read/write operations.

PBDR is initialized to H'0000 by a power-on reset. It is not initialized by a manual rese
standby mode or sleep mode.

affect pin state

20.1.1 Power-Down Modes

The following modes and function are provided as power-down modes:

1. Sleep mode
2. Standby mode
3. Module standby function
   (UBC, DMAC, DSP, FRT, SCIF1 to SCIF2, TPU, SIO0 to SIO2)

Table 20.1 shows the transition conditions for entering the modes from the program ex
state, as well as the CPU and peripheral module states in each mode and the procedure
canceling each mode.

mode	instruction executed with SBY bit set to 0 in SBYCR1							2 3 4
Standby mode	SLEEP instruction executed with SBY bit set to 1 in SBYCR1	Halted	Halted	Halted	Halted, and register values held	UBC: Halted, and register values held Other than UBC: Halted	Held or high impedance	1 2 3
Module standby function	MSTP bit for relevant module is set to 1	Runs	Runs	When MSTP is 1, the clock supply is halted	Runs	When an MSTP bit is 1, the clock supply to the relevant module is halted	FRT, and SCIF1, 2 pins are initialized, and others operate	1 2 3

20.1.2 Register

Table 20.2 shows the register configuration.

Table 20.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Acc
Standby control register 1	SBYCR1	R/W	H'00	H'FFFFFE91	8
Standby control register 2	SBYCR2	R/W	H'00	H'FFFFFE93	8

R/W:	R/W	R/W	R/W	R/W	R/W	R	R/W

Standby control register 1 (SBYCR1) is an 8-bit read/write register that sets the power
mode. SBYCR is initialized to H'00 by a reset.

Bit 7—Standby (SBY): Specifies transition to standby mode. To enter the standby mo
WDT (set the TME bit in WTCSR to 0) and set the SBY bit.

Bit 7: SBY	Description	
0	Executing a SLEEP instruction puts the chip into sleep mode	(In
1	Executing a SLEEP instruction puts the chip into standby mode	

Bit 6—Port High Impedance (HIZ): Selects whether output pins are set to high imped
retain the output state in standby mode. When HIZ = 0 (initial state), the specified pin
output state. When HIZ = 1, the pin goes to the high-impedance state. See appendix B
States during Resets, Power-Down States and Bus Release State, for which pins are co

Bit 6: HIZ	Description	
0	Pin state retained in standby mode	(In
1	Pin goes to high impedance in standby mode	

Bit 5—Module Stop 5 (MSTP5): Specifies halting the clock supply to the user break c
(UBC). When the MSTP5 bit is set to 1, the supply of the clock to the UBC is halted.
clock halts, the UBC registers retain their pre-halt state. Do not set this bit while the U
running.

Bit 5: MSTP5	Description	
0	UBC running	(In
1	Clock supply to UBC halted	

Bit 3—Module Stop 3 (MSTP3): Specifies halting the clock supply to the DSP unit. W
MSTP3 bit is set to 1, the supply of the clock to the DSP unit is halted. When the clock
operation result prior to the halt is retained. This bit should be set when the DSP unit is
When the DSP unit is halted, no instructions with a DSP register, MACH, or MACL as
can be used.

Bit 3: MSTP3	Description	
0	DSP running	(Init
1	Clock supply to DSP halted	

Bit 2—Reserved: This bit is always read as 0. The write value should always be 0.

Bit 1—Module Stop 1 (MSTP1): Specifies halting the clock supply to the 16-bit free-ru
timer (FRT). When the MSTP1 bit is set to 1, the supply of the clock to the FRT is halt
the clock halts, all FRT registers are initialized except the FRT interrupt vector register
which holds its previous value. When MSTP1 is cleared to 0 and the FRT begins runni
starts operating from its initial state.

Bit 1: MSTP1	Description	
0	FRT running	(Init
1	Clock supply to FRT halted	

Bit 0—Reserved: This bit is always read as 0. The write value should always be 0.

Standby control register 2 (SBYCR2) is an 8-bit read/write register that sets the power
mode state. SBYCR2 is initialized to H'00 by a reset.

Bits 7 and 6—Reserved: These bits are always read as 0. The write value should alwa

Bit 5—Module Stop 11 (MSTP11): Specifies halting the clock supply to the 16-bit tin
unit (TPU). When the MSTP11 bit is set to 1, the supply of the clock to the TPU is ha
the clock halts, the TPU retains its pre-halt state, and the TPU interrupt vector register
retains its pre-halt value. Therefore, when MSTP11 is cleared to 0 and the clock suppl
is resumed, the TPU starts operating again.

Bit 5: MSTP11	Description	
0	TPU running	(In
1	Clock supply to TPU halted	

Bit 4—Module Stop 10 (MSTP10): Specifies halting the clock supply to SIO channel
MSTP10 bit is set to 1, the supply of the clock to SIO channel 2 is halted. When the c
SIO channel 2 retains its pre-halt state, and the SIO channel 2 interrupt vector register
retains its pre-halt value. Therefore, when MSTP10 is cleared to 0 and the clock suppl
channel 2 is restarted, operation starts again.

Bit 4: MSTP10	Description	
0	SIO channel 2 running	(In
1	Clock supply to SIO channel 2 halted	

Bit 2—Module Stop 8 (MSTP8): Specifies halting the clock supply to SIO channel 0. W
MSTP8 bit is set to 1, the supply of the clock to SIO channel 0 is halted. When the cloc
SIO channel 0 retains its pre-halt state, and the SIO channel 0 interrupt vector register i
retains its pre-halt value. Therefore, when MSTP8 is cleared to 0 and the clock supply t
channel 0 is restarted, operation starts again.

Bit 2: MSTP8	Description	
0	SIO channel 0 running	(Init
1	Clock supply to SIO channel 0 halted	

Bit 1—Module Stop 7 (MSTP7): Specifies halting the clock supply to SCIF2. When th
bit is set to 1, the supply of the clock to SCIF2 is halted. When the clock halts, the SCI
are initialized, but the SCIF2 interrupt vector register in the INTC retains its pre-halt va
Therefore, when MSTP7 is cleared to 0 and SCIF2 begins running again, it starts opera
its initial state.

Bit 1: MSTP7	Description	
0	SCIF2 running	(Init
1	Clock supply to SCIF2 halted	

Bit 0—Module Stop 6 (MSTP6): Specifies halting the clock supply to SCIF1. When th
bit is set to 1, the supply of the clock to SCIF1 is halted. When the clock halts, the SCI
are initialized, but the SCIF1 interrupt vector register in the INTC retains its pre-halt va
Therefore, when MSTP6 is cleared to 0 and SCIF1 begins running again, it starts opera
its initial state.

Bit 0: MSTP6	Description	
0	SCIF1 running	(Init
1	Clock supply to SCIF1 halted	

20.3.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt, DMA address error, power-on reset, or manua

Cancellation by an Interrupt: When an interrupt occurs, sleep mode is canceled and exception handling is executed. Sleep mode is not canceled if the interrupt cannot be a because its priority level is equal to or less than the mask level set in the CPU's status (SR) or if an interrupt by an on-chip peripheral module is disabled at the peripheral m

Cancellation by a DMA Address Error: If a DMA address error occurs, sleep mode and DMA address error exception handling is executed.

Cancellation by a Power-On Reset: A power-on reset cancels sleep mode.

Cancellation by a Manual Reset: A manual reset cancels sleep mode.

20.4 Standby Mode

20.4.1 Transition to Standby Mode

To enter standby mode, set the SBY bit to 1 in SBYCR1, then execute the SLEEP inst chip switches from the program execution state to standby mode. The NMI interrupt c accepted when the SLEEP instruction is executed, or for the following five cycles. In mode, the clock supply to all on-chip peripheral modules is halted as well as the CPU. register contents are held, and some on-chip peripheral modules are initialized.

controller (DMAC)	register 0, 1 DMA operation register	registers 0 and 1 • DMA destination address registers 0 and 1 • DMA transfer count registers 0 and 1 • DMA request/ response selection control registers 0 and 1 • Vector number setting registers DMA0 and DMA1	
Watchdog timer (WDT)	Bits 7 to 5 of the timer control/status register Reset control/status register	Bits 2 to 0 of the timer control/status register Timer counter	—
16-bit free-running timer (FRT)	All registers	—	—
Serial communication interface with FIFO (SCIF1 to SCIF2)	All registers	—	—
Serial I/O (SIO0 to SIO2)	—	All registers	—
High-performance user debugging interface (H-UDI)	—	All registers	—
16-bit timer pulse unit (TPU)	—	All registers	—
Pin function controller (PFC)	—	All registers	—
Ethernet controller direct memory access controller (E-DMAC)	All registers	—	—
Ethernet controller (EtherC)	All registers	—	—
Others	—	Standby control registers 1 and 2 Frequency modification register	—

standby mode is entered (when the clock is halted), and goes low on recovering from s
mode (when the clock starts after oscillation has stabilized). The low level at the NMI
be held for at least 3 cycles after the start of clock signal output from the CKIO pin. W
mode is canceled by a rising edge in the NMI signal, insure that the NMI pin goes low
standby mode is entered (when the clock is halted), and goes high on recovering from
mode (when the clock starts after oscillation has stabilized). The high level at the NMI
be held for at least 3 cycles after the start of clock signal output from the CKIO pin.

Cancellation by a Power-On Reset: A power-on reset cancels standby mode.

Cancellation by a Manual Reset: A manual reset cancels standby mode.

20.4.3 Standby Mode Cancellation by NMI Interrupt

The following example describes moving to the standby mode upon the fall of the NM
clearing the standby mode when the NMI signal rises. Figure 20.1 shows the timing.

When the NMI pin level changes from high to low after the NMI edge select bit (NMI
interrupt control register (ICR) has been set to 0 (detect falling edge), an NMI interrup
accepted. When the NMIE bit is set to 1 (detect rising edge) by the NMI exception ser
the standby bit (SBY) of the standby control register 1 (SBYCR1) is set to 1 and a SL
instruction is executed, the standby mode is entered. The standby mode is cleared the
NMI pin level changes from low level to high level. The high level at the NMI pin sho
for at least 3 cycles after the start of clock signal output from the CKIO pin.

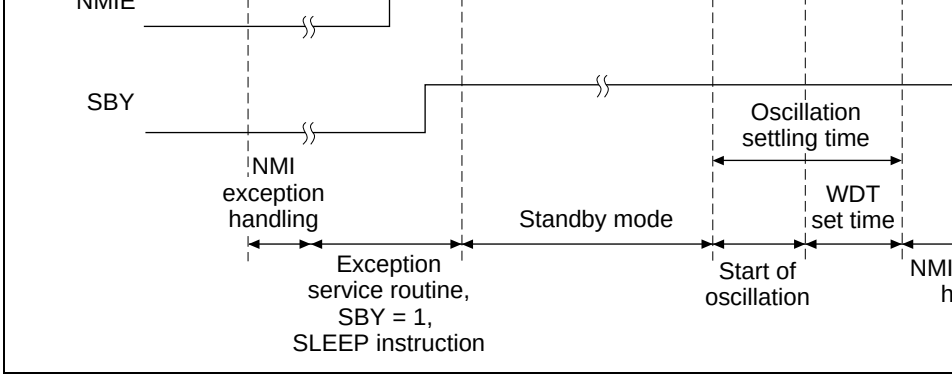


Figure 20.1 Standby Mode Cancellation by NMI Interrupt

20.4.4 Clock Pause Function

When the clock is input from the CKIO pin, the clock frequency can be modified or the
stopped. The $\overline{\text{CKPREQ}}$/CKM pin is provided for this purpose. Note that clock pauses a
accepted while the watchdog timer (WDT) is operating (i.e. when the timer enable bit (
the WDT's timer control/status register (WTCSR) is 1). When the clock pause request
used, the standby bit (SBY) in the standby control register 1 (SBYCR1) must be set to
inputting the request signal. The clock pause function is used as described below.

1. Set the TME bit in the watchdog timer's WTCSR register to 0, and set the SBY bit
   SBYCR1 to 1.
2. Apply a low level to the $\overline{\text{CKPREQ}}$/CKM pin.
3. When the chip enters the standby state internally, a low level is output from the $\overline{\text{CK}}$
4. After confirming that the $\overline{\text{CKPACK}}$ pin has gone low, perform clock halting or freq
   modification.
5. To cancel the clock pause state (standby state), apply a high level to the $\overline{\text{CKPREQ}}$/
   (Inside the chip , the standby state is canceled by detecting a rising edge at the $\overline{\text{CKP}}$
   pin.)
6. When PLL circuit 1 is operational, the WDT starts counting up inside the chip. Whe
   circuit 1 is halted, the WDT is not activated.

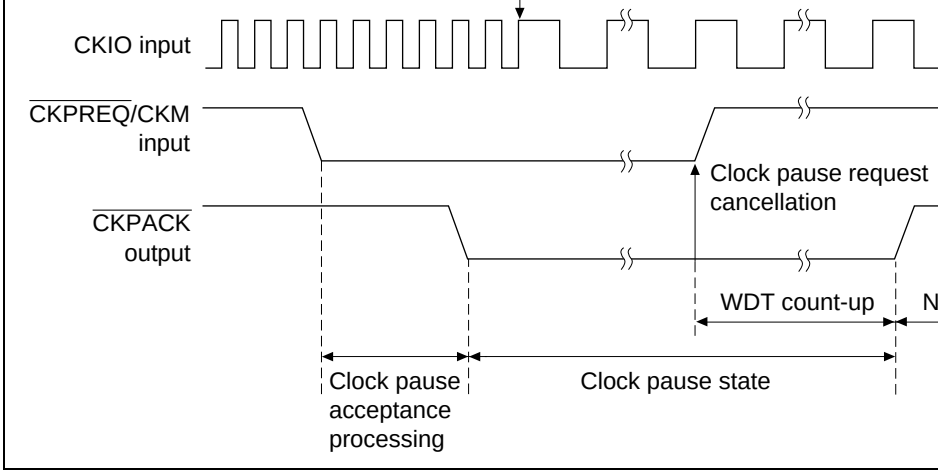


Figure 20.2 Clock Pause Function Timing Chart (PLL Circuit 1 Operat

Figure 20.3 shows the clock pause function timing chart when the PLL circuit is halte

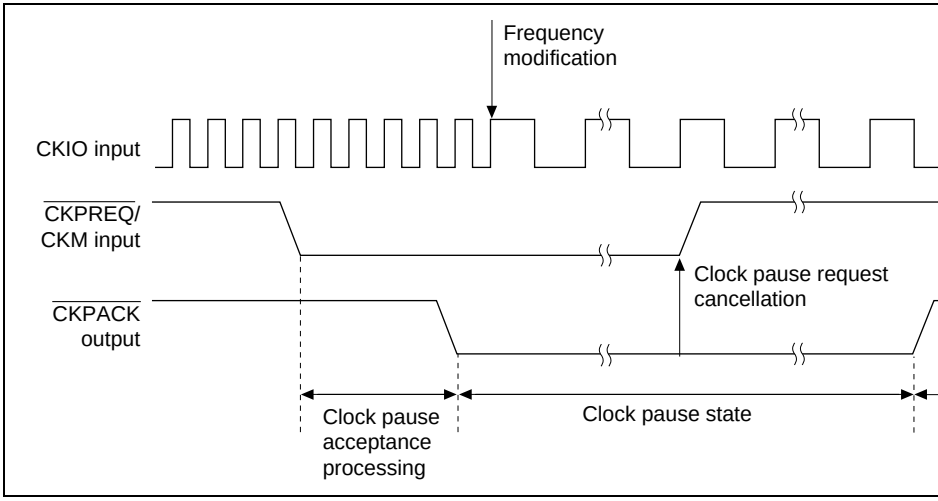


Figure 20.3 Clock Pause Function Timing Chart (PLL Circuit 1 Halte

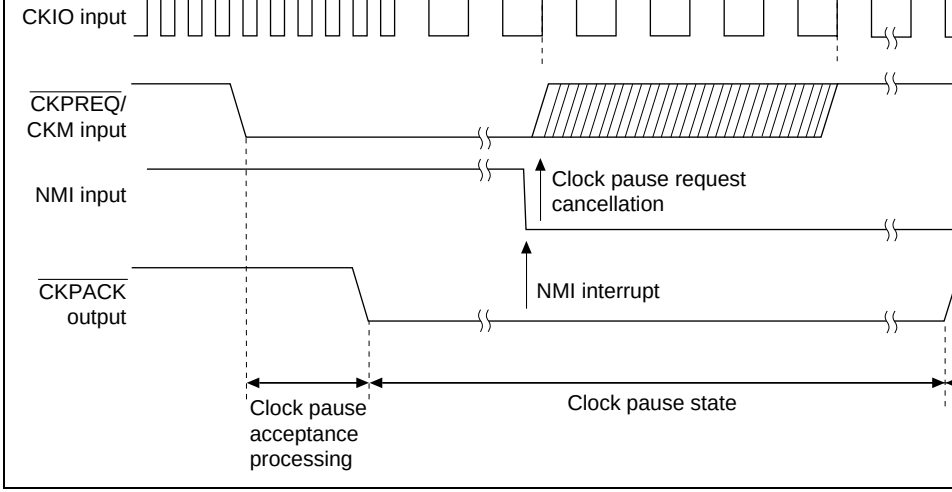


Figure 20.4 Clock Pause Function Timing Chart (Cancellation by NMI In

20.4.5 Notes on Standby Mode

1. When the chip enters standby mode during use of the cache, disable the cache befor the mode transition. Initialize the cache beforehand when the cache is used after ret standby mode. The contents of the on-chip RAM are not retained in standby mode v is used as on-chip RAM.
2. If an on-chip peripheral register is written in the 10 clock cycles before the chip tran standby mode, read the register before executing the SLEEP instruction.
3. When using clock mode 0, 1, or 2, the CKIO pin is the clock output pin. Note the fo when standby mode is used in these clock modes. When standby mode is canceled b unstable clock is output from the CKIO pin during the oscillation settling time after input. This also applies to clock output in the case of cancellation by a power-on res manual reset. Power-on reset and manual reset input should be continued for a perio equal to for the oscillation settling time.
4. Before entering the standby mode, stop operation of the internal DMAC (E-DMAC DMAC).

With the module standby function, the external pins of the DMAC and SIO0 to SIO2 peripheral modules retain their states prior to halting, as do DMAC, DSP, and SIO0 to registers. The external pins of the FRT, SCIF1 to SCIF2, and TPU are reset and all the are initialized.

An on-chip peripheral module corresponding to a module standby bit must not be swit module standby state while it is running. Also, interrupts from a module placed in the state should be disabled.

20.5.2 Clearing the Module Standby Function

Clear the module standby function by clearing the MSTP11 to MSTP3, MSTP1 bits, c power-on reset or manual reset.

Power supply voltage (internal)	V_{CC}	–0.3 to +4.2	V
Power supply voltage (5 V I/O)	PV_{CC}	–0.3 to +7.0	V
Input voltage (excluding 5 V I/O)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (5 V I/O)	V_{in}	–0.3 to PV_{CC} +0.3	V
Operating temperature	T_{opr}	–20 to +75	°
Storage temperature	T_{stg}	–55 to +125	°

Notes: 1. Permanent damage to the chip may result if the maximum ratings are exce

2. When powering on, turn on the 5 V I/O power supply (PV_{CC}) after, or at the as, the internal power supply (V_{CC}). When powering off, cut V_{CC} after, or at time as, PV_{CC}.

Input high voltage	$\overline{RES}$, NMI, MD4 to MD0, $\overline{TRST}$, $\overline{CKPREQ}$/CKM	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	Both 3.3 V and 5 V		2.6	—	PV_{CC} + 0.3	V	
	EXTAL, CKIO		$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	Other input pins		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, NMI, MD4 to MD0, $\overline{TRST}$, $\overline{CKPREQ}$/CKM	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	Other input pins		–0.3	—	0.8	V	
Schmitt trigger input voltage	PB14/RXD1, PB5/SRS1/RXD2	V_T^-	—	—	0.8	V	
		V_T^+	4.0	—	—	V	PV_{CC} =
		V_T^+	2.6	—	—	V	Other t
		$V_T^+ - V_T^-$	0.3	—	—	V	
Input leakage current	All input pins	$\lvert I_{in} \rvert$	—	—	1.0	μA	V_{in} = 0. V_{CC} – 0 V_{in} = 0. PV_{CC} –
Three-state leakage current	All I/O and output pins (off status)	$\lvert I_{TSI} \rvert$	—	—	1.0	μA	V_{in} = 0. V_{CC} – 0 V_{in} = 0. PV_{CC} –
Output high voltage	Both 3.3 V and 5 V	V_{OH}	PV_{CC} – 0.7	—	—	V	I_{OH} = –2
	Other output pins		V_{CC} – 0.5	—	—	V	I_{OH} = –2
			V_{CC} – 1.0	—	—	V	I_{OH} = –1
Output low voltage	Both 3.3 V and 5 V	V_{OL}	—	—	0.6	V	I_{OL} = 1.
	Other output pins		—	—	0.4	V	I_{OL} = 1.

					62.5 M not us
	Sleep mode	—	—	250	mA	3.6 V, opera 62.5 M periph modul
	Standby mode	—	—	990	µA	

Note: Do not leave the $PLLV_{CC}$ and $PLLV_{SS}$ pins open when the PLL circuit is not use
the $PLLV_{CC}$ pin to V_{CC} and the $PLLV_{SS}$ pin to V_{SS}.

Table 21.3 Permissible Output Currents

Conditions: $V_{CC} = PLLV_{CC} = 3.3$ V ±0.3 V, $PV_{CC} = 5.0$ V ± 0.5 V/3.3 V ±0.3 V, PV
$V_{SS} = PV_{SS} = PLLV_{SS} = 0$ V, Ta = –20 to +75°C

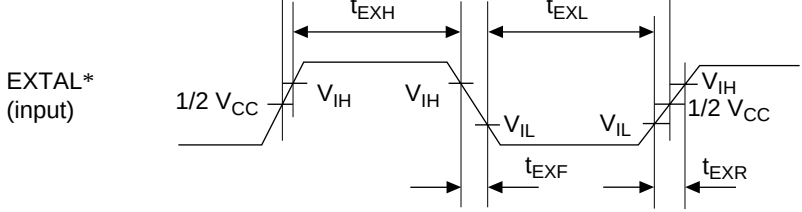
Item	Symbol	Min	Typ	Max
Permissible output low current (per pin)	I_{OL}	—	—	2.0
Permissible output low current (total)	ΣI_{OL}	—	—	80
Permissible output high current (per pin)	$-I_{OH}$	—	—	2.0
Permissible output high current (total)	$\Sigma(-I_{OH})$	—	—	25

Note: To protect chip reliability, do not exceed the output current values in table 21.3.

Item		Symbol	Min	Typ	Max	Unit	N
Operating frequency	CPU, DSP	f	1	—	62.5	MHz	t_{I}
	External bus (SDRAM not used)		1	—	31.25		t_{E}
	External bus (SDRAM used)		1	—	62.5		t_{E}
	Peripheral modules		1	—	31.25		t_{P}

EXTAL clock input cycle time	t_{EXcyc}	32	1000	ns
EXTAL clock input low-level pulse width	t_{EXL}	8*1, 12*2	—	ns
EXTAL clock input high-level pulse width	t_{EXH}	8*1, 12*2	—	ns
EXTAL clock input rise time	t_{EXR}	—	4	ns
EXTAL clock input fall time	t_{EXF}	—	4	ns
CKIO clock input frequency	f_{CKI}	1	31.25	MHz
CKIO clock input cycle time	t_{CKIcyc}	32	1000	ns
CKIO clock input low-level pulse width	t_{CKIL}	8*3, 12*4	—	ns
CKIO clock input high-level pulse width	t_{CKIH}	8*3, 12*4	—	ns
CKIO clock input rise time	t_{CKIR}	—	4	ns
CKIO clock input fall time	t_{CKIF}	—	4	ns
CKIO clock output frequency	f_{OP}	1*5, 8*6	62.5	MHz
CKIO clock output cycle time	t_{cyc}	16	1000*5, 125*6	ns
CKIO clock output low-level pulse width	t_{CKOL}	3	—	ns
CKIO clock output high-level pulse width	t_{CKOH}	3	—	ns
CKIO clock rise time	t_{CKOR}	—	5	ns
CKIO clock fall time	t_{CKOF}	—	5	ns
CKPO clock output cycle time	t_{CKPCYC}	32	1000	ns
CKPO clock output low-level pulse width	t_{CKPOL}	11	—	ns
CKPO clock output high-level pulse width	t_{CKPOH}	11	—	ns
CKPO clock rise time	t_{CKPOr}	—	5	ns
CKPO clock fall time	t_{CKPOf}	—	5	ns
Power-on oscillation stabilization time	t_{OSC1}	10	—	ms
Standby recovery oscillation stabilization time 1	t_{OSC2}	10	—	ms
Standby recovery oscillation stabilization time 2	t_{OSC3}	10	—	ms
PLL synchronization stabilization time	t_{PLL}	1	—	ms

Notes: 1. When PLL circuit 2 is operating



Note: * When clock is input from EXTAL pin

Figure 21.1 EXTAL Clock Input Timing

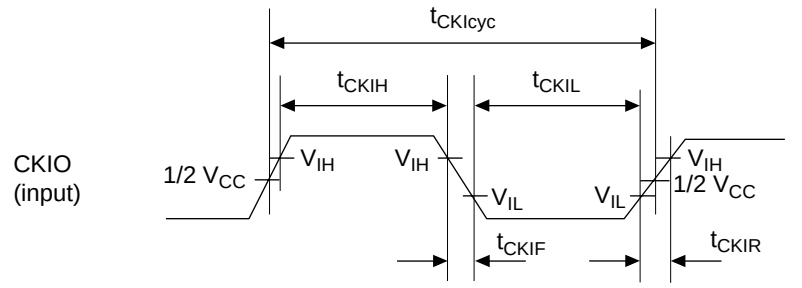


Figure 21.2 CKIO Clock Input Timing

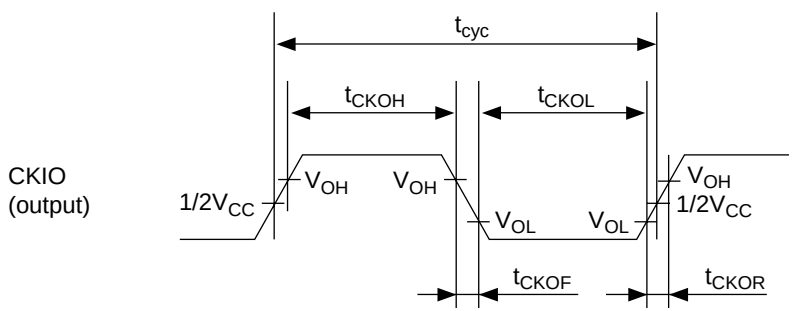


Figure 21.3 CKIO Clock Output Timing

Figure 21.4 CKPO Clock Output Timing

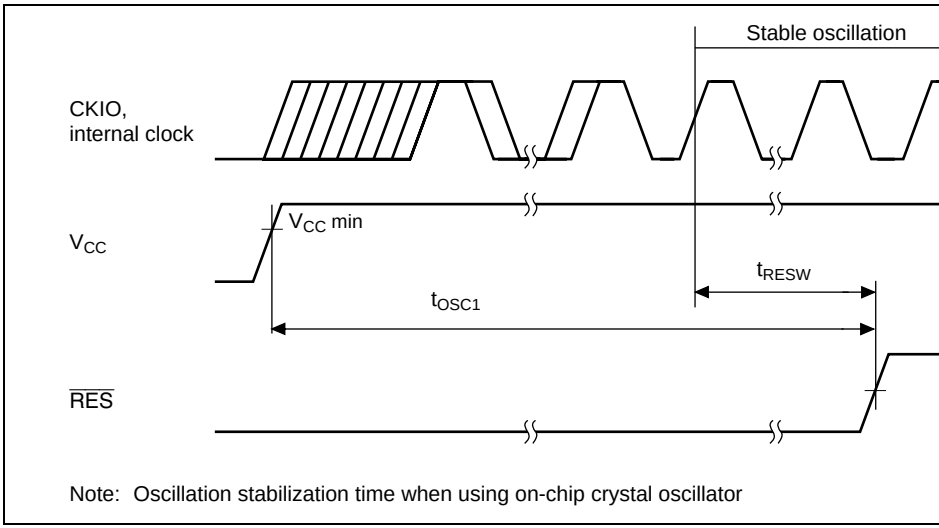


Figure 21.5 Power-On Oscillation Stabilization Time at Power-On

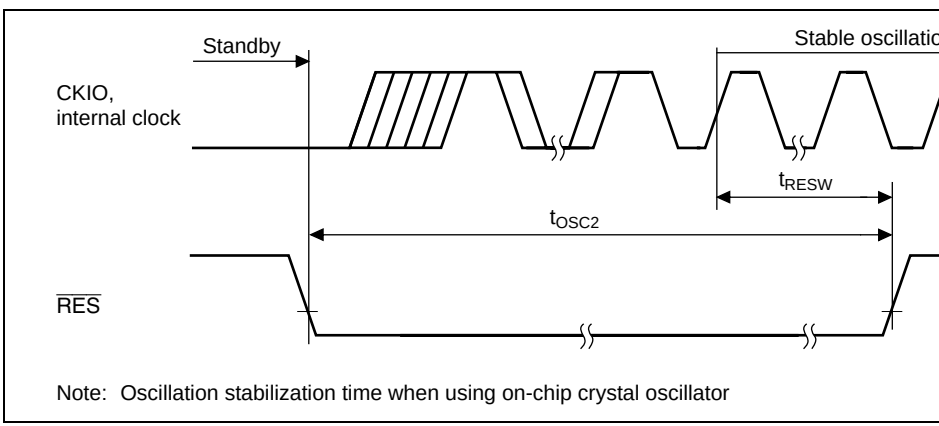


Figure 21.6 Oscillation Stabilization Time after Standby Recovery (Recovery

Note: Oscillation stabilization time when using on-chip crystal oscillator

Figure 21.7 Oscillation Stabilization Time after Standby Recovery (Recovery f

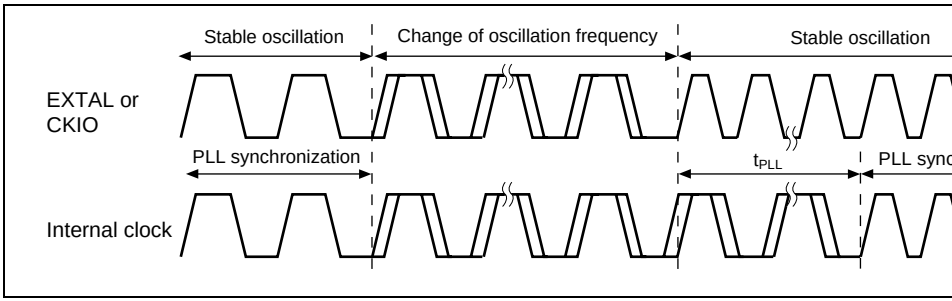


Figure 21.8 PLL Synchronization Stabilization Time

$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{Pcyc}	
NMI reset setup time	t_{NMIRS}	t_{Pcyc} + 10	—	ns	
NMI reset hold time	t_{NMIRH}	t_{Pcyc} + 10	—	ns	
NMI rise and fall time	t_{NMIr}, t_{NMIf}	—	200	ns	
$\overline{RES}$ setup time*	t_{RESS}	$3t_{Ecyc}$ + 40	—	ns	2
NMI setup time*	t_{NMIS}	40	—	ns	
$\overline{IRL3}$ to $\overline{IRL0}$ setup time*	t_{IRLS}	30	—	ns	
NMI hold time	t_{NMIH}	20	—	ns	
$\overline{IRL3}$ to $\overline{IRL0}$ hold time*	t_{IRLH}	20	—	ns	
$\overline{BRLS}$ setup time	t_{BLSS}	10	—	ns	2
$\overline{BRLS}$ hold time	t_{BLSH}	5	—	ns	
$\overline{BGR}$ delay time	t_{BGRD}	—	15	ns	
Bus tri-state delay time	t_{BOFF}	0	35	ns	
Bus buffer on time	t_{BON}	0	35	ns	

Note: * The $\overline{RES}$, NMI, and $\overline{IRL3}$ to $\overline{IRL0}$ signals are asynchronous inputs. If the se
shown here are observed, a transition is judged to have occurred at the fall
if the setup times cannot be observed, recognition may be delayed until the
the clock.

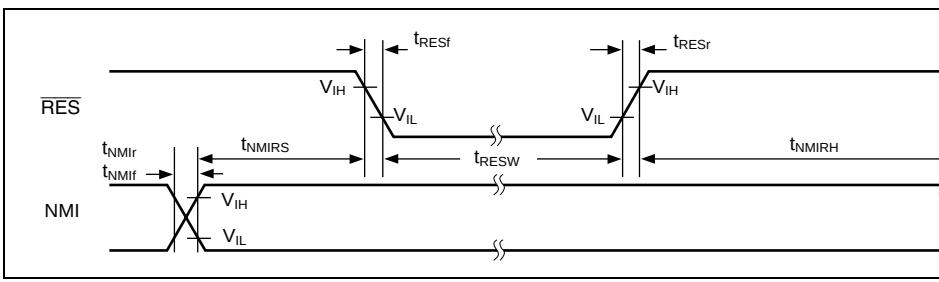


Figure 21.9 Reset Input Timing

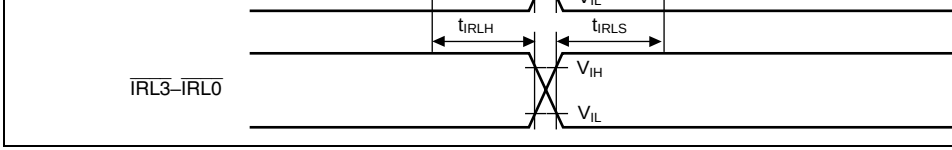


Figure 21.10 Interrupt Signal Input Timing

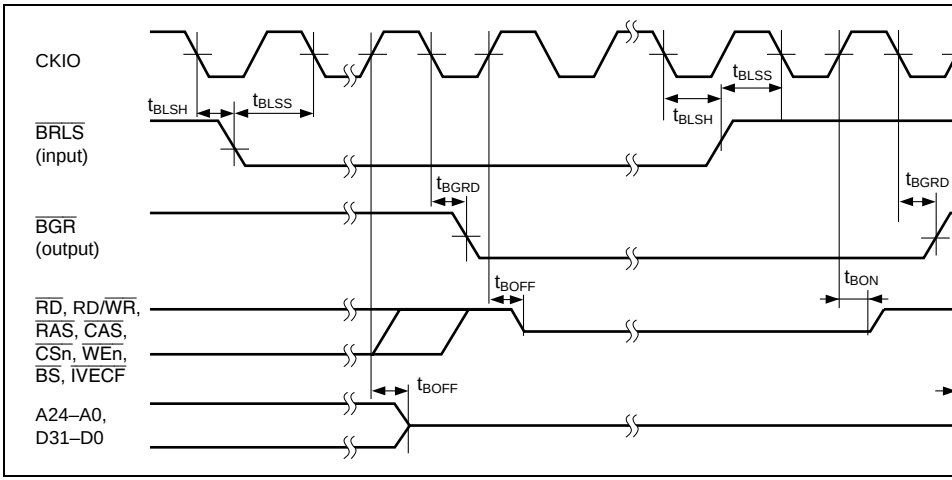
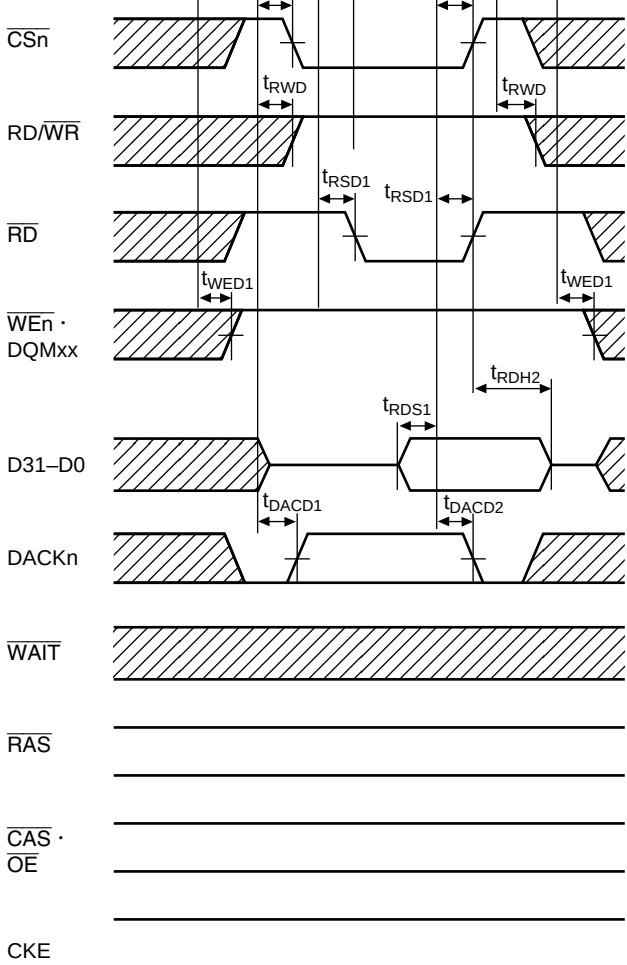


Figure 21.11 Bus Release Timing

$\overline{BS}$ delay time	t_{BSD}	—	15	ns	21.12, 13, 16, 17, 19, 26, 29, 31, 32, 34, 35,
$\overline{CS}$ delay time 1	t_{CSD1}	1	14	ns	21.12, 13, 16, 17, 19, 29, 31 to 35, 40, 42, 4
$\overline{CS}$ delay time 2	t_{CSD2}	—	14	ns	21.12, 13, 34, 35, 40,
Read/write delay time	t_{RWD}	1	14	ns	21.12, 13, 16, 17, 19, 26, 29 to 35, 40, 43 to
Read strobe delay time 1	t_{RSD1}	—	14	ns	21.12, 13, 16, 17, 23, 38, 40, 41, 43 to 45
Read data setup time 1	t_{RDS1}	8	—	ns	21.12, 34, 38, 43 to 45
Read data setup time 2 (EDO)	t_{RDS2}	8	—	ns	21.40, 41
Read data setup time 3 (SDRAM)	t_{RDS3}	6.5	—	ns	21.16, 17
Read data hold time 2	t_{RDH2}	0	—	ns	21.12, 43
Read data hold time 4 (SDRAM)	t_{RDH4}	2	—	ns	21.16, 17
Read data hold time 5 (DRAM)	t_{RDH5}	0	—	ns	21.34, 38
Read data hold time 6 (EDO)	t_{RDH6}	3	—	ns	21.40, 41
Read data hold time 7 (EDO)	t_{RDH7}	1	—	ns	21.40
Read data hold time 8 (interrupt vector)	t_{RDH8}	2	—	ns	21.44, 45
Write enable delay time 1	t_{WED1}	—	14	ns	21.12, 13
Write data delay time 1 (except t_{Ecyc}:t_{Pcyc} = 1:1)	t_{WDD1}	—	22	ns	21.13, 23, 25, 27, 35,
Write data delay time 2 (t_{Ecyc}:t_{Pcyc} = 1:1)	t_{WDD2}	—	12	ns	21.26, 28
Write data hold time 1	t_{WDH1}	2	—	ns	21.13, 23, 25 to 28, 35
Data buffer on time	t_{DON}	—	15	ns	21.13, 23, 25, 26, 35
Data buffer off time	t_{DOF}	—	15	ns	21.13, 23, 25, 26, 35

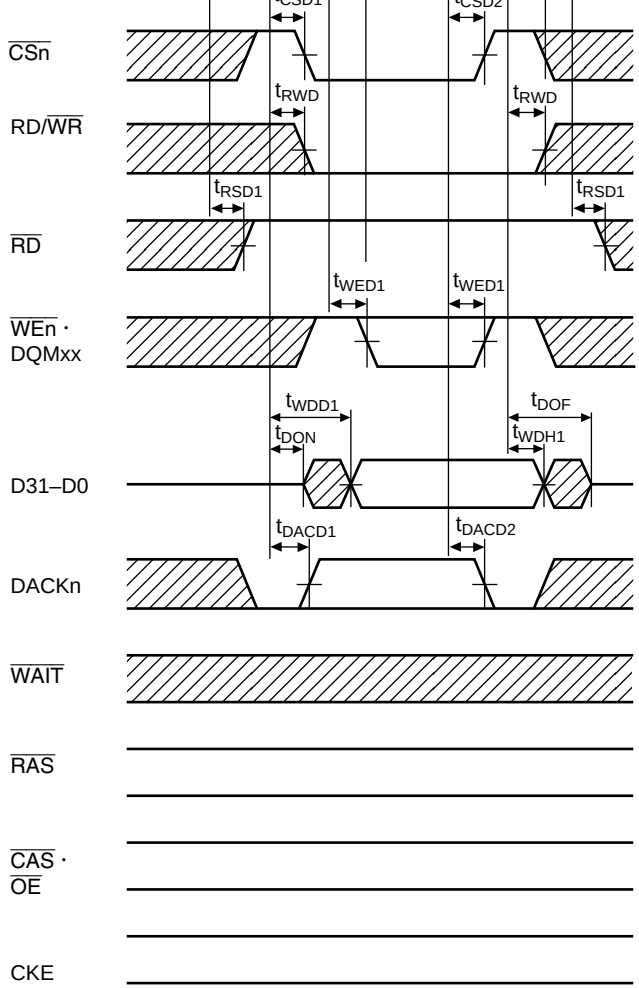
$\overline{RAS}$ delay time 3 (EDO)	t_{RASD3}	—	14	ns	21.40
$\overline{CAS}$ delay time 1 (SDRAM)	t_{CASD1}	1	14	ns	21.16, 17, 18, 19, 2 to 33, 43
$\overline{CAS}$ delay time 2 (DRAM)	t_{CASD2}	—	14	ns	21.34, 35, 38 to 42
DQM delay time	t_{DQMD}	1	14	ns	21.16, 17, 19 to 21 30
CKE delay time	t_{CKED}	1	14	ns	21.33
$\overline{OE}$ delay time 1	t_{OED1}	—	14	ns	21.40
$\overline{OE}$ delay time 2	t_{OED2}	—	14	ns	21.40
$\overline{IVECF}$ delay time	t_{IVD}	—	15	ns	21.44, 45
Row address setup time	t_{ASR}	0	—	ns	21.34, 35, 40
Column address setup time	t_{ASC}	0	—	ns	21.34, 35, 38, 39, 4
Data input setup time	t_{DS}	0	—	ns	21.35, 39
Read/write address setup time	t_{AS}	0	—	ns	21.12, 13
REFOUT delay time	t_{REFOD}	—	15	ns	21.47

(SDRAM)					
Write data delay time 2 (Iϕ:Eϕ = 1:1)	t_{WDD2}	—	9.5	ns	21.26, 28
Write data hold time 1	t_{WDH1}	2	—	ns	21.26, 28
Address delay time	t_{AD}	4	11	ns	21.16, 17, 19, 21, 23 28, 29, 31, 32, 33
$\overline{CS}$ delay time 1	t_{CSD1}	2.5	9.5	ns	21.16, 17, 19, 21, 23 29, 31, 32, 33
Read/write delay time	t_{RWD}	2.5	9.5	ns	21.16, 17, 19, 21, 22 29, 30, 31, 32, 33
DQM delay time	t_{DQMD}	2.5	9.5	ns	21.16, 17, 19, 20, 21 27, 28, 29, 30
$\overline{RAS}$ delay time 1 (SDRAM)	t_{RASD1}	2.5	9.5	ns	21.16, 17, 18, 21, 22 26, 29, 30, 31, 32, 33
$\overline{CAS}$ delay time 1 (SDRAM)	t_{CASD1}	2.5	9.5	ns	21.16, 17, 18, 19, 23 27, 28, 29, 31, 32, 33
CKE delay time	t_{CKED}	2.5	9.5	ns	21.33



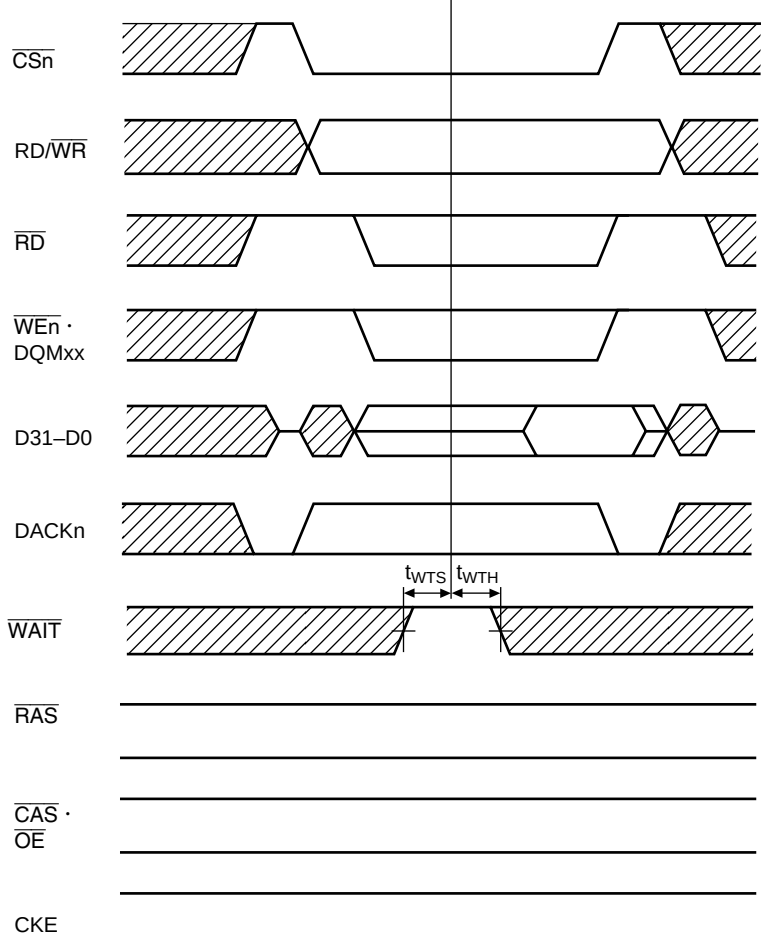
Notes: 1. t_{RDH2} is measured from the rise of $\overline{CSn}$ or $\overline{RD}$, whichever comes first.
2. DACKn waveform when active-high is specified

Figure 21.12 Basic Read Cycle (No Wait)



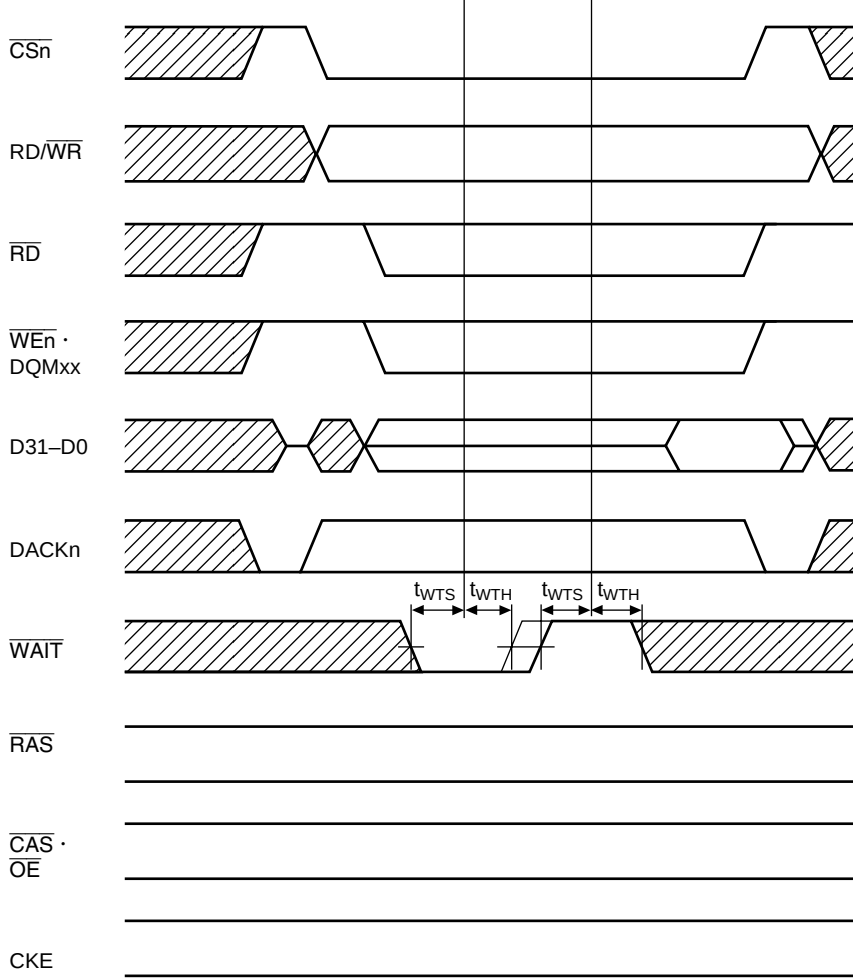
Note: DACKn waveform when active-high is specified

Figure 21.13 Basic Write Cycle (No Wait)



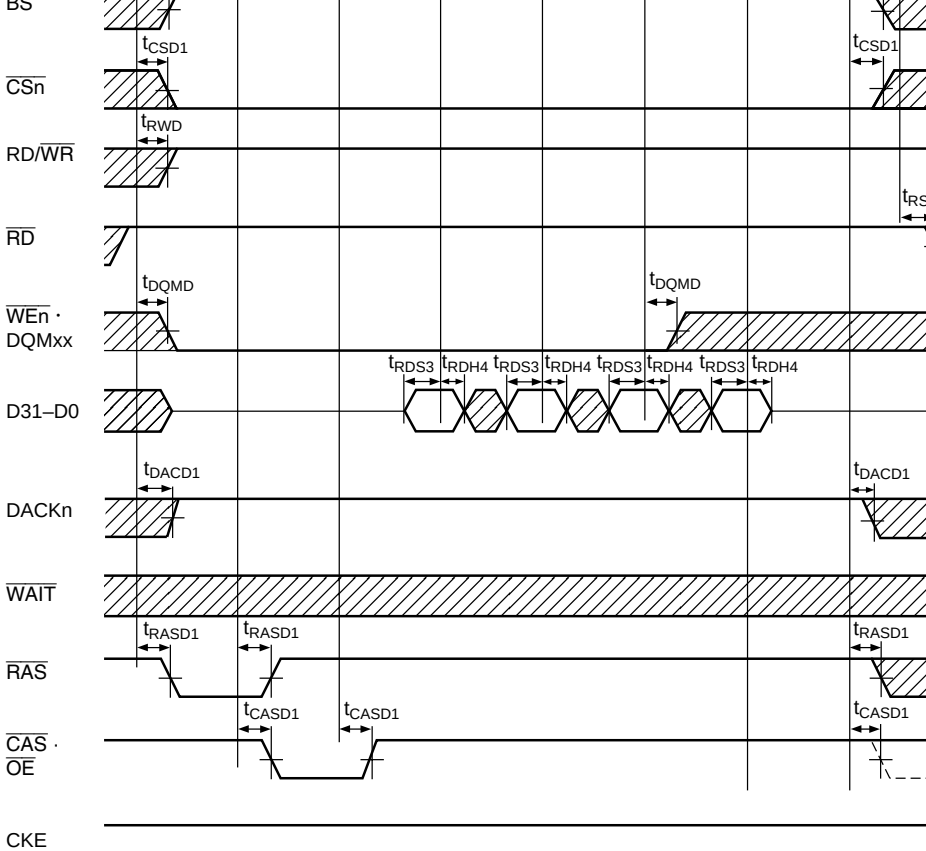
Note: DACKn waveform when active-high is specified

Figure 21.14 Basic Bus Cycle (1 Wait Cycle)



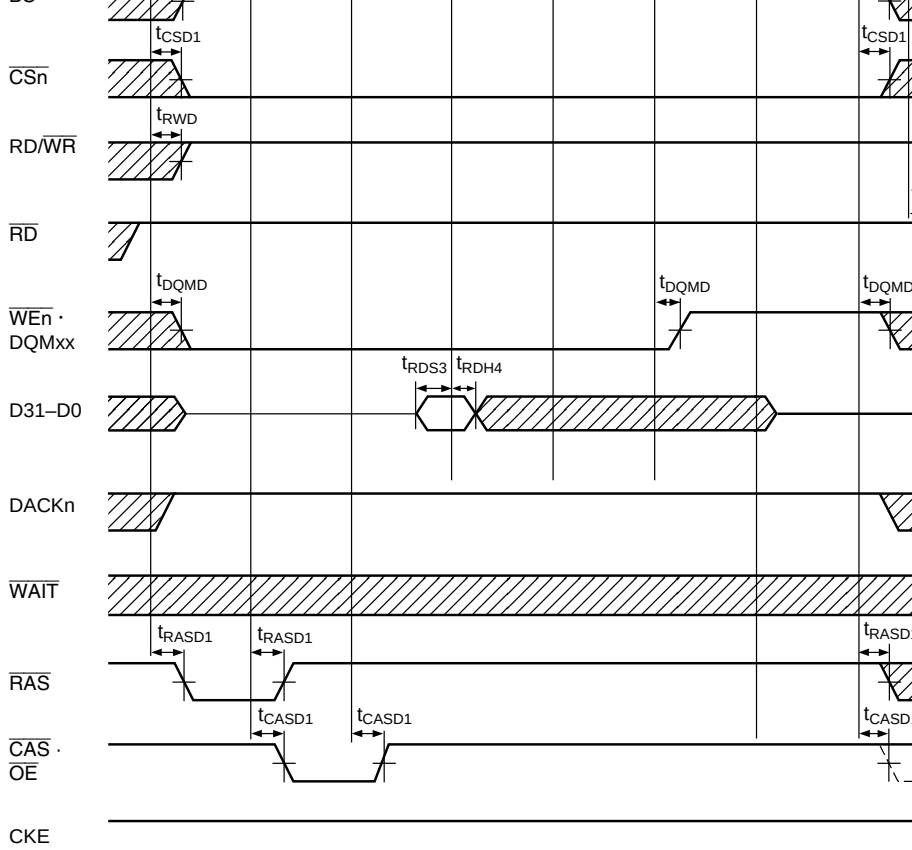
Note: DACKn waveform when active-high is specified

Figure 21.15 Basic Bus Cycle (External Wait Input)



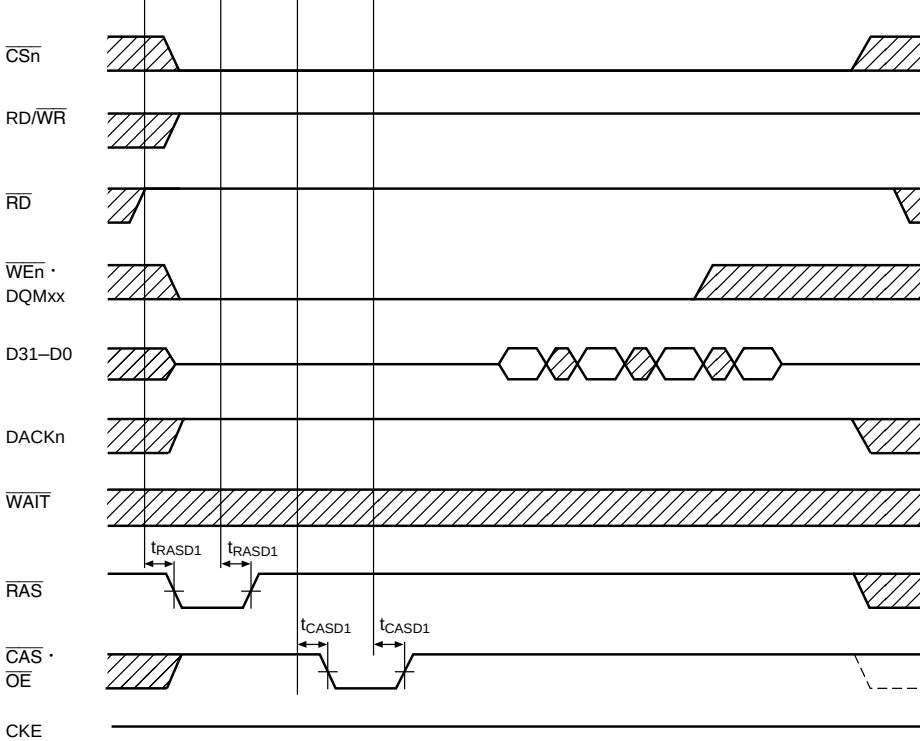
Notes: 1. Dotted line shows the case where synchronous DRAM in a different CS space is accessed.
2. DACKn waveform when active-high is specified

Figure 21.16 Synchronous DRAM Read Bus Cycle (RCD = 1 Cycle, CAS Latency = 1 Cycle, Burst = 4)



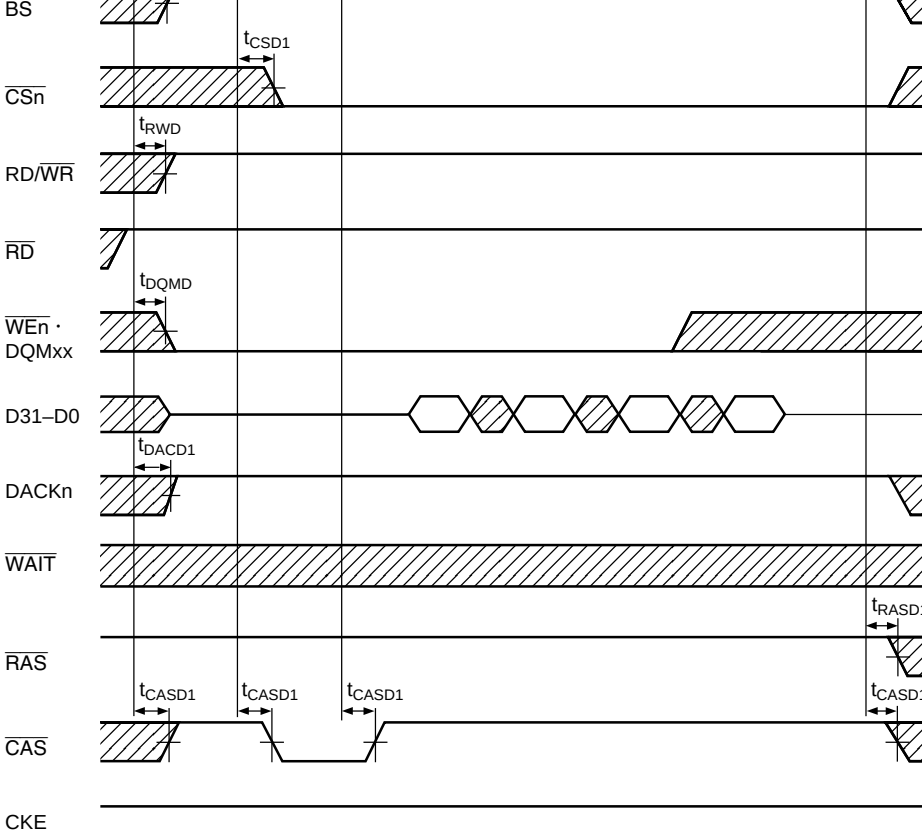
Notes: 1. Dotted line shows the case where synchronous DRAM in a different CS space is accessed.
2. DACKn waveform when active-high is specified

Figure 21.17 Synchronous DRAM Single Read Bus Cycle (RCD = 1 Cycle, CAS Latency = 1 Cycle, Burst = 4)



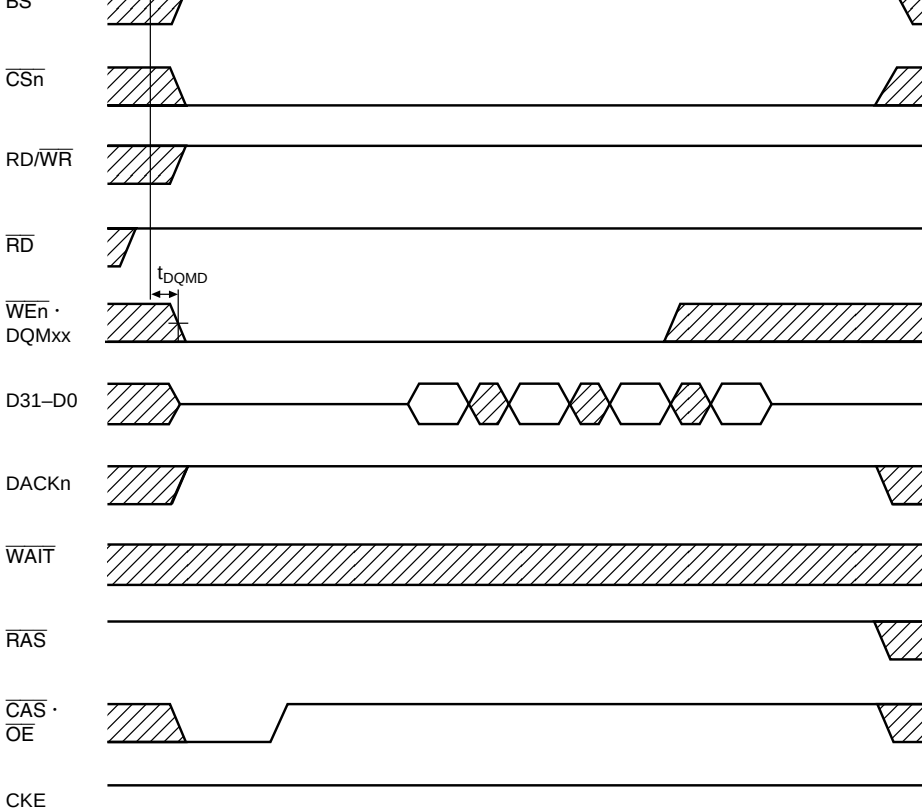
Notes: 1. Dotted line shows the case where synchronous DRAM in a different CS space is accessed.
2. DACKn waveform when active-high is specified

Figure 21.18 Synchronous DRAM Read Bus Cycle (RCD = 2 Cycles, CAS Latency = 2 Cycles, Burst = 4)



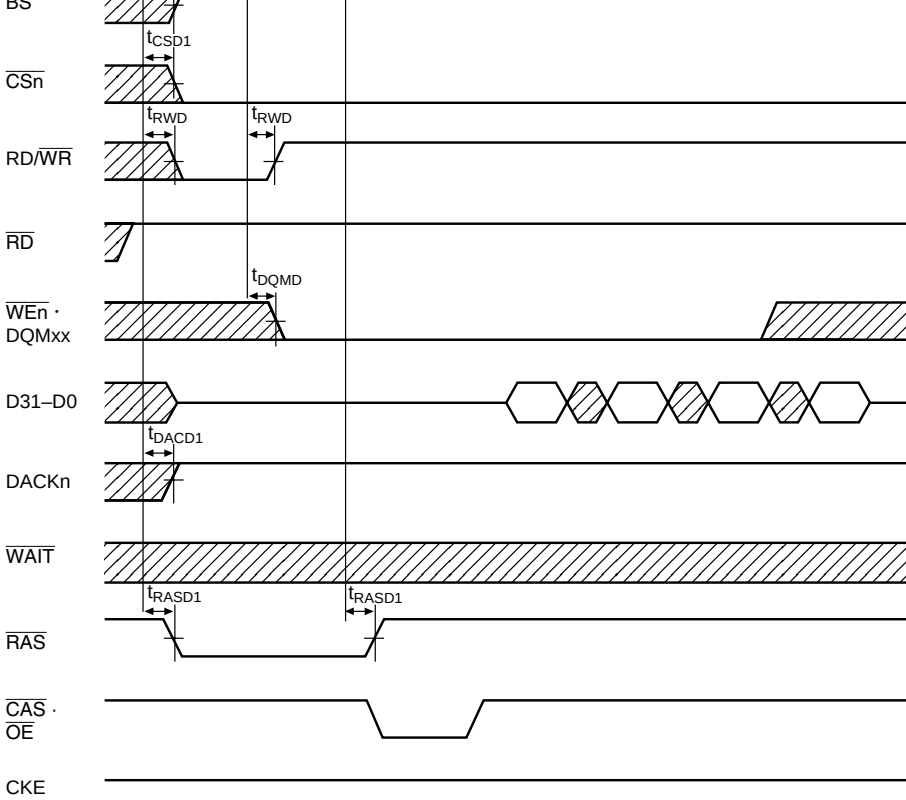
Note: DACKn waveform when active-high is specified

Figure 21.19 Synchronous DRAM Read Bus Cycle (Bank Active, Same Row Access, CAS Latency = 1 Cycle)



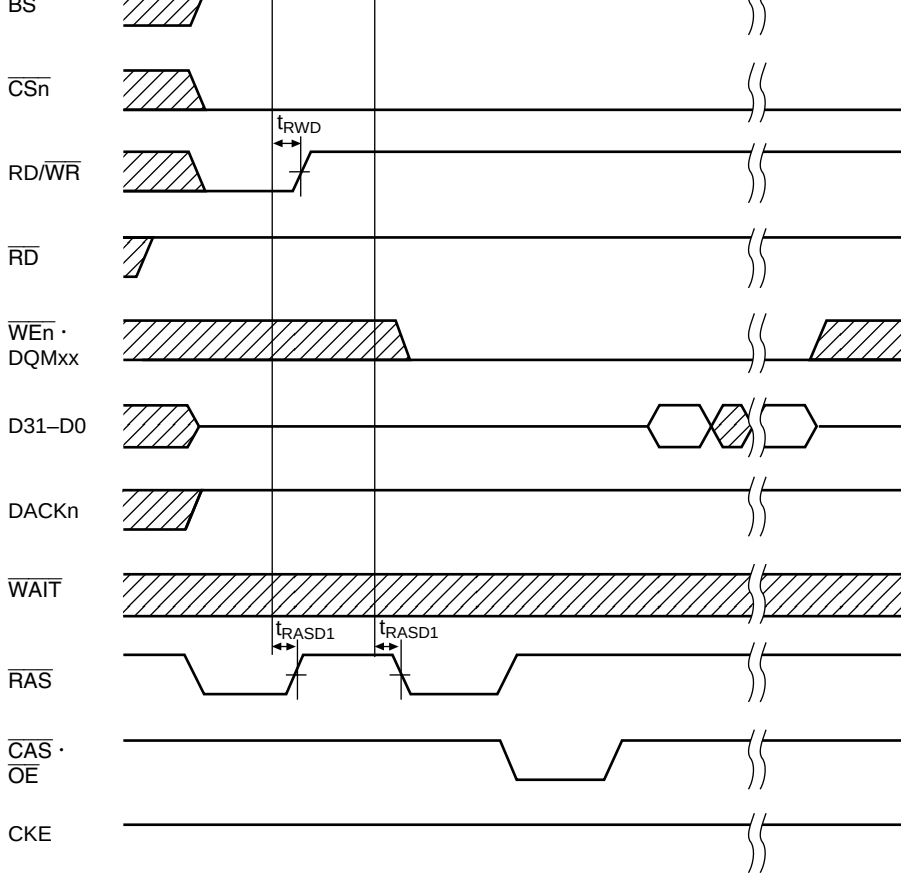
Note: DACKn waveform when active-high is specified

Figure 21.20 Synchronous DRAM Read Bus Cycle (Bank Active, Same Row Access, CAS Latency = 2 Cycles)



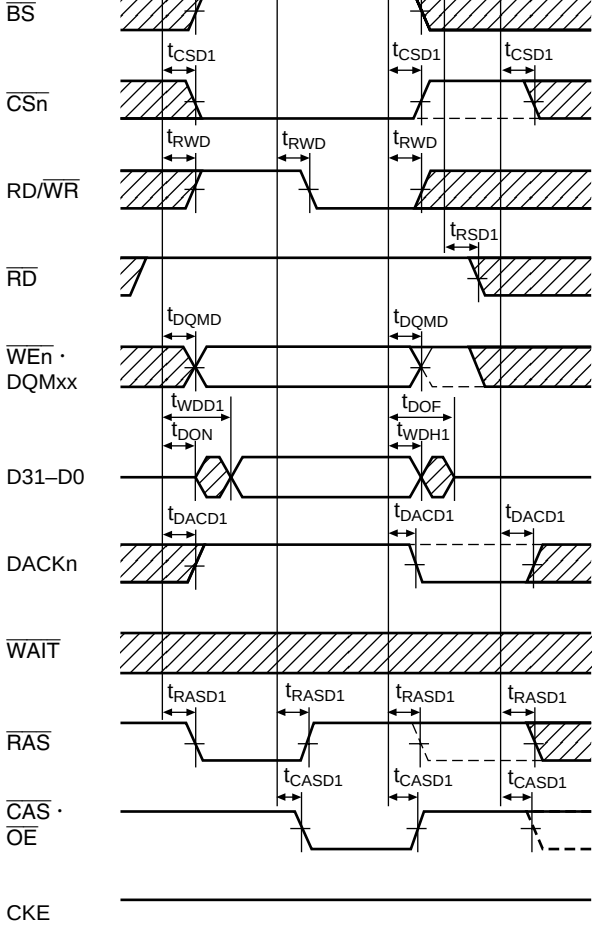
Note: DACKn waveform when active-high is specified

Figure 21.21 Synchronous DRAM Read Bus Cycle
(Bank Active, Different Row Access, TRP = 1 Cycle, RCD = 1 Cycle, CAS Latenc



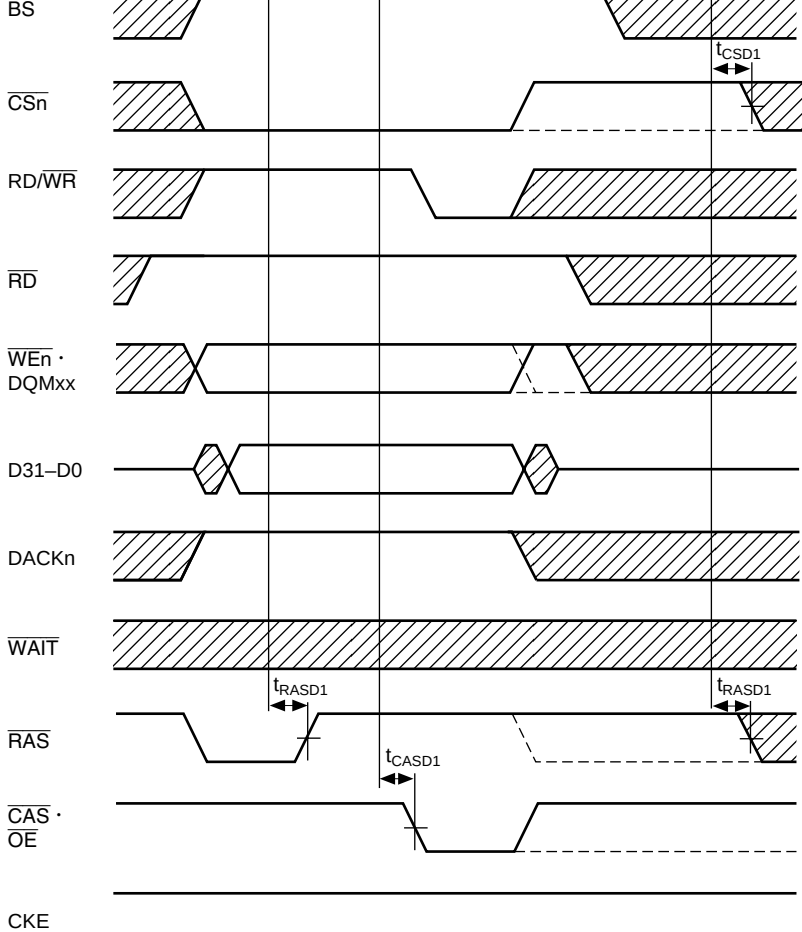
Note: DACKn waveform when active-high is specified

Figure 21.22 Synchronous DRAM Read Bus Cycle (Bank Active, Different Row, TRP = 2 Cycles, RCD = 1 Cycle, CAS Latency = 1 Cycle)



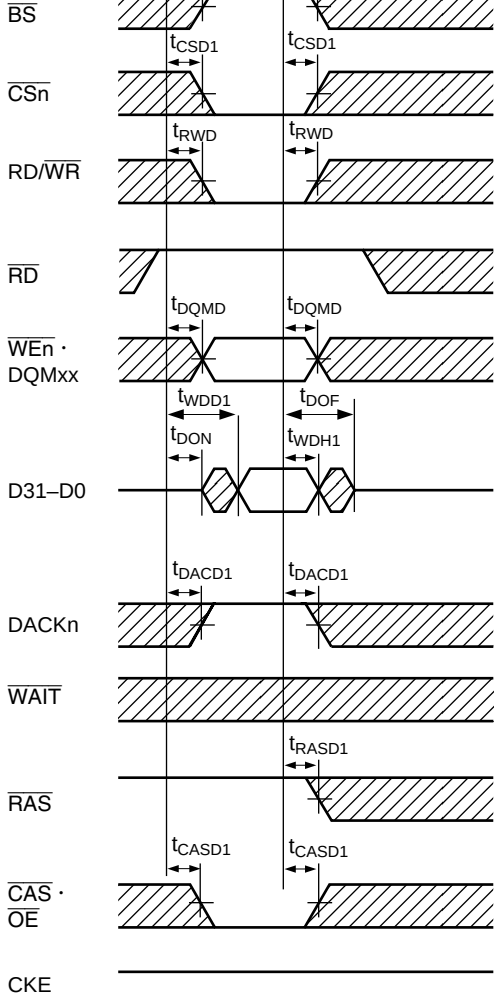
Notes: 1. Dotted line shows the case where synchronous DRAM in a different CS sp
accessed.
2. DACKn waveform when active-high is specified

Figure 21.23 Synchronous DRAM Write Bus Cycle (RASD = 0, RCD = 1 Cycle, TRWL = 1 Cycle)



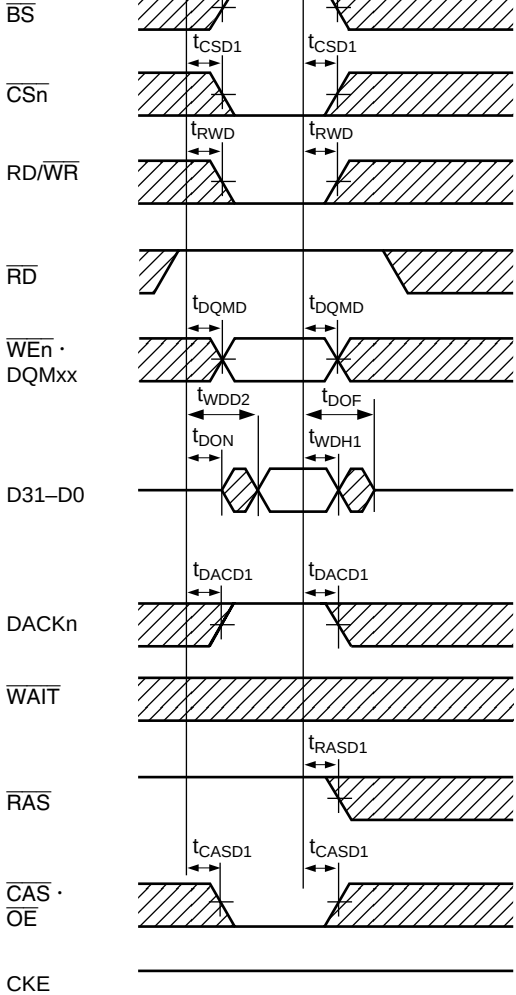
Notes: 1. Dotted line shows the case where synchronous DRAM in a different CS spac
accessed.
2. DACKn waveform when active-high is specified

Figure 21.24 Synchronous DRAM Write Bus Cycle (RASD = 0, RCD = 2 Cycles, TRWL = 2 Cycles)



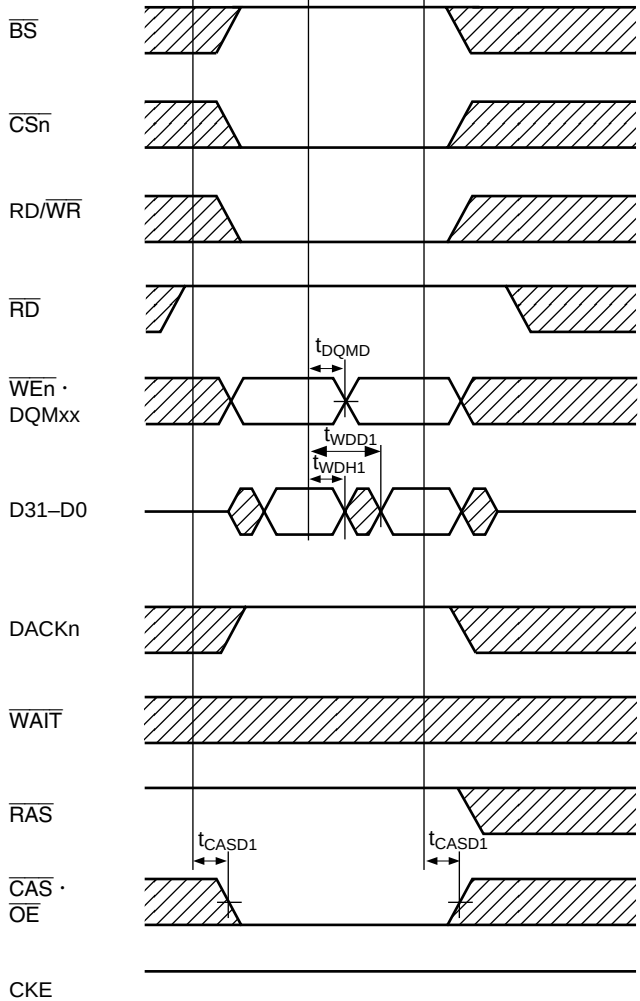
Note: DACKn waveform when active-high is specified

Figure 21.25 Synchronous DRAM Write Bus Cycle (Bank Active, Same Row Access, Except t_{Ecyc}:t_{Pcyc} 1:1)



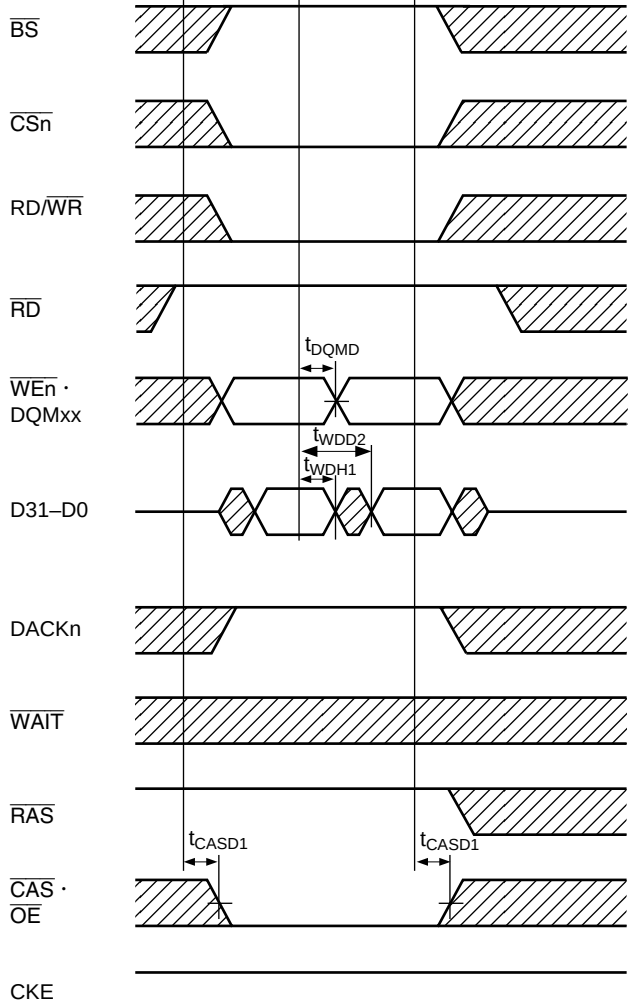
Note: DACKn waveform when active-high is specified

Figure 21.26 Synchronous DRAM Write Cycle (Bank Active, Same Row Access, Iϕ:Eϕ = 1:1)



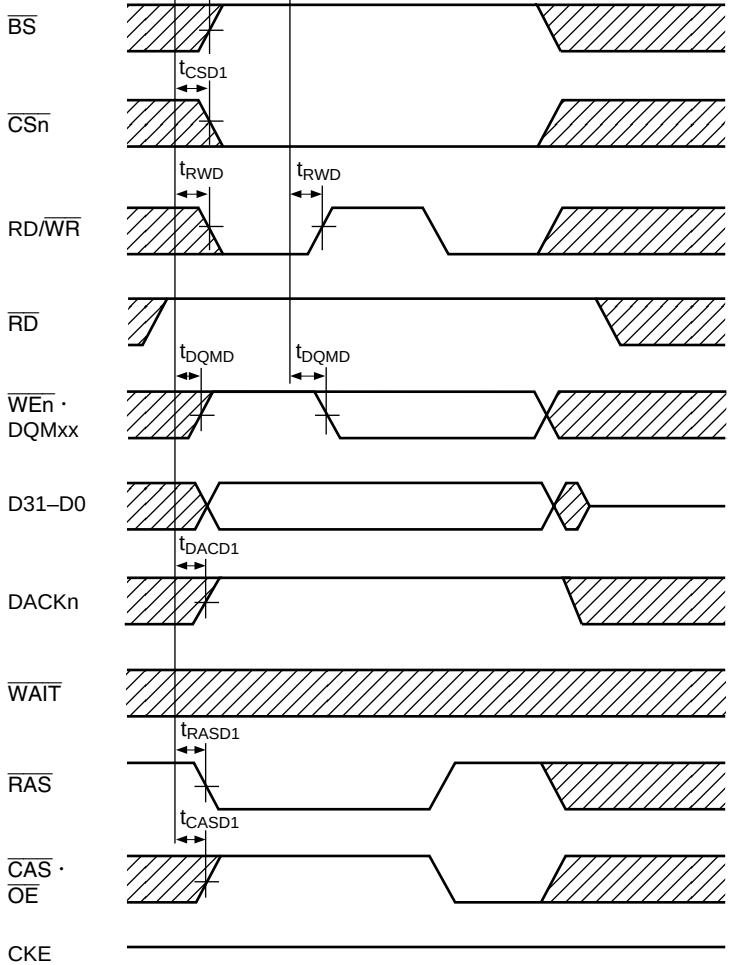
Note: DACKn waveform when active-high is specified

Figure 21.27 Synchronous DRAM Continuous Write Cycle (Bank Active, Same Row Access, Except t_{Ecyc}:t_{Pcyc} 1:1)



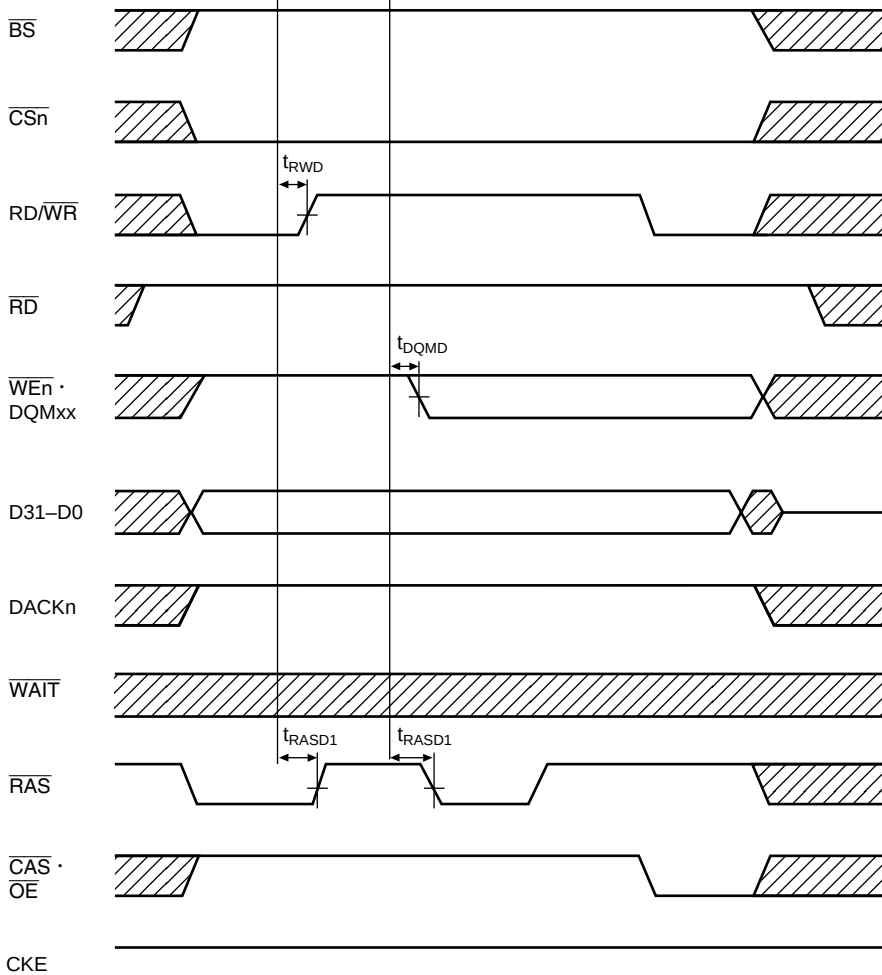
Note: DACKn waveform when active-high is specified

Figure 21.28 Synchronous DRAM Continuous Write Cycle (Bank Active, Same Row Access, Iϕ:Eϕ = 1:1)



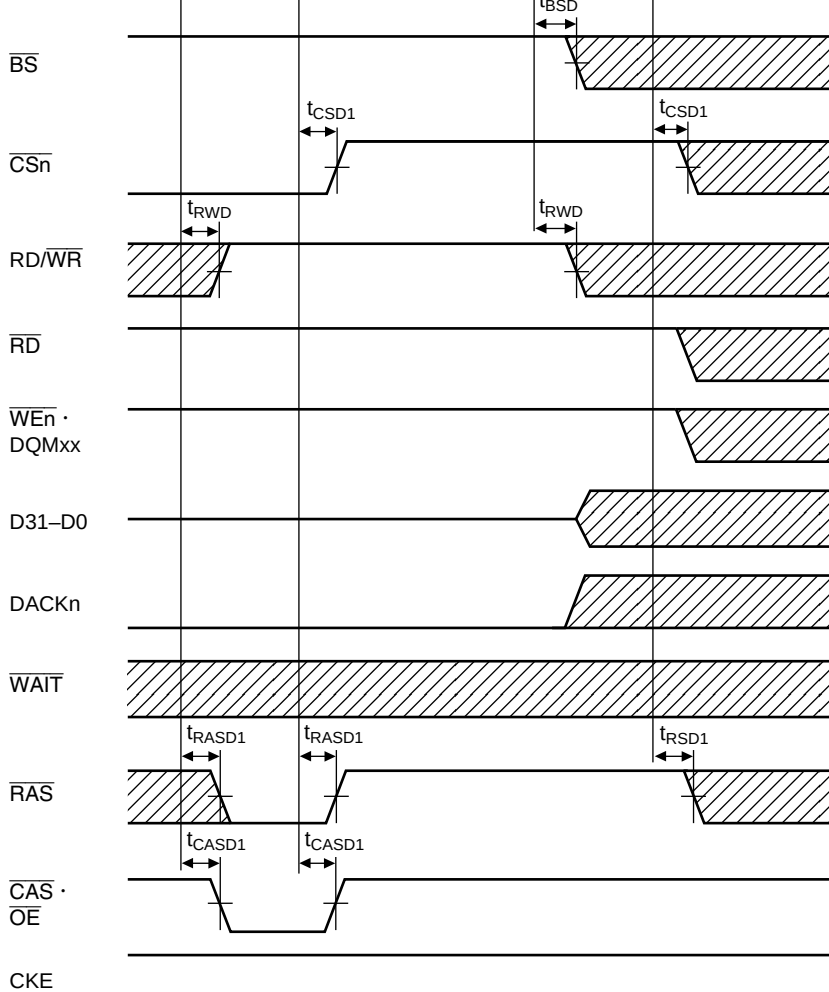
Note: DACKn waveform when active-high is specified

Figure 21.29 Synchronous DRAM Write Bus Cycle (Bank Active, Different Row Access, TRP = 1 Cycle, RCD = 1 Cycle)



Note: DACKn waveform when active-high is specified

Figure 21.30 Synchronous DRAM Write Bus Cycle
(Bank Active, Different Row Access, TRP = 2 Cycles, RCD = 2 Cycles)



Note: An auto-refresh cycle is always preceded by a precharge cycle. The number of
between the two is determined by the number of cycles specified by TRP.

Figure 21.31 Synchronous DRAM Auto-Refresh Cycle (TRAS = 4 Cycles)

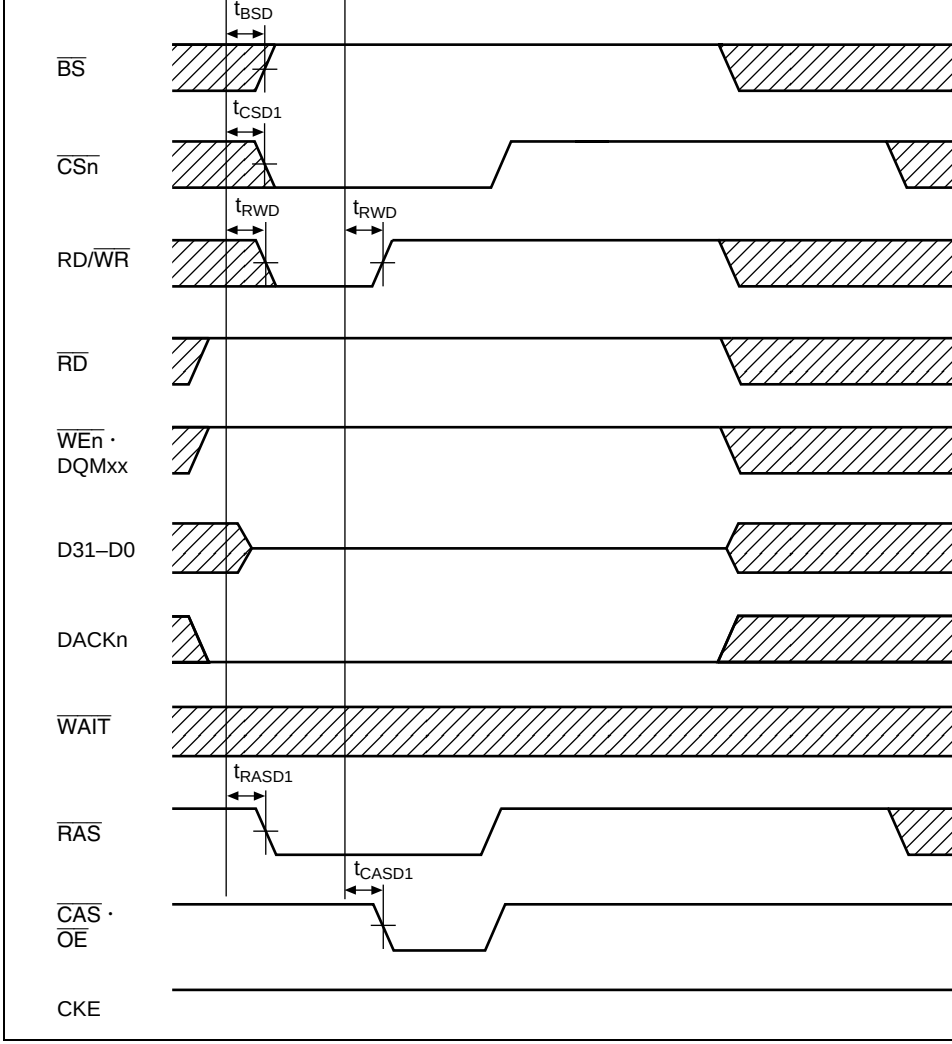
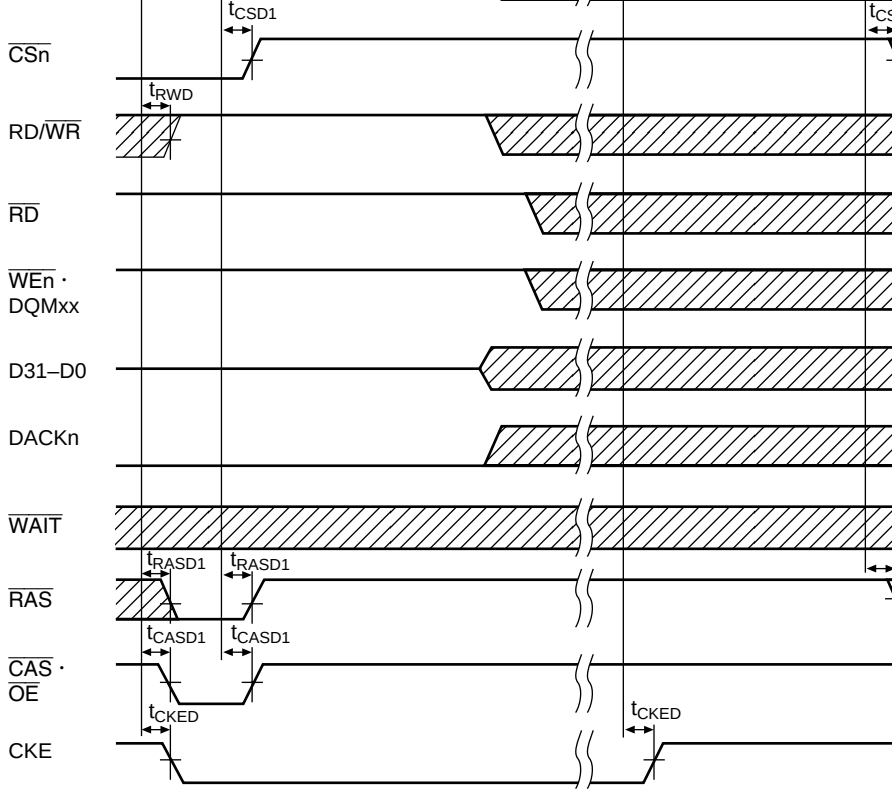
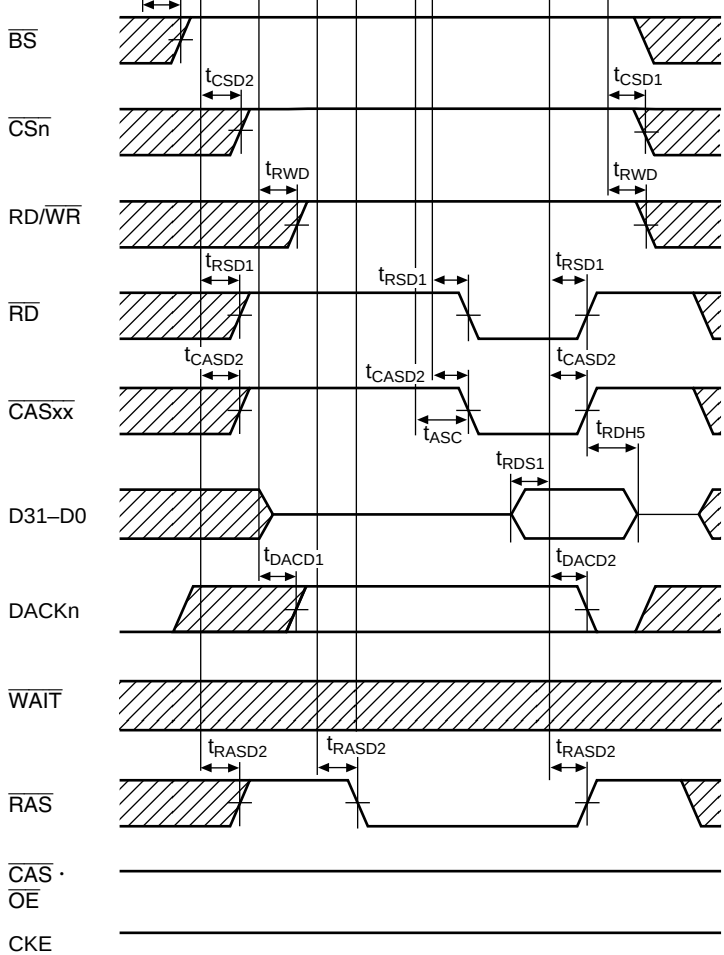


Figure 21.32 Synchronous DRAM Auto-Refresh Cycle
(Shown from Precharge Cycle, TRP = 1 Cycle, TRAS = 4 Cycles)



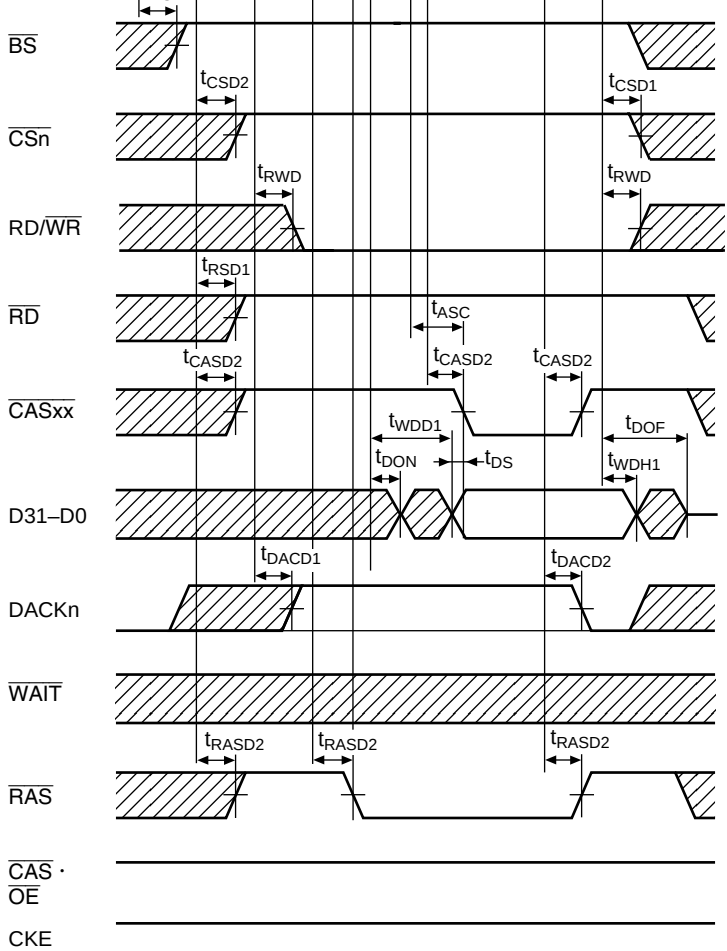
Note: A self-refresh cycle is always preceded by a precharge cycle. The number o
between the two is determined by the number of cycles specified by TRP.

Figure 21.33 Synchronous DRAM Self-Refresh Cycle (TRAS = 3)



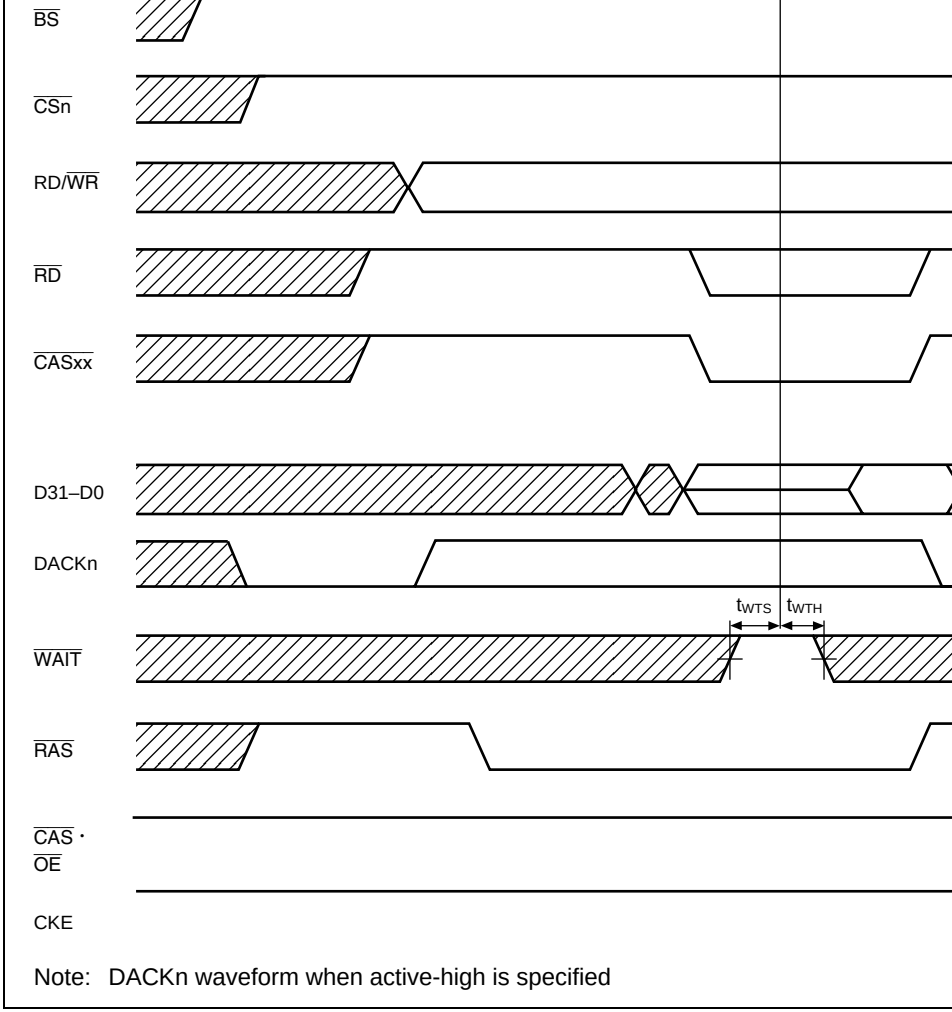
Notes: 1. t_{RDH5} is measured from the rise of $\overline{RD}$ or $\overline{CASxx}$, whichever comes first.
2. DACKn waveform when active-high is specified

Figure 21.34 DRAM Read Cycle (TRP = 1 Cycle, RCD = 1 Cycle, No Wait)



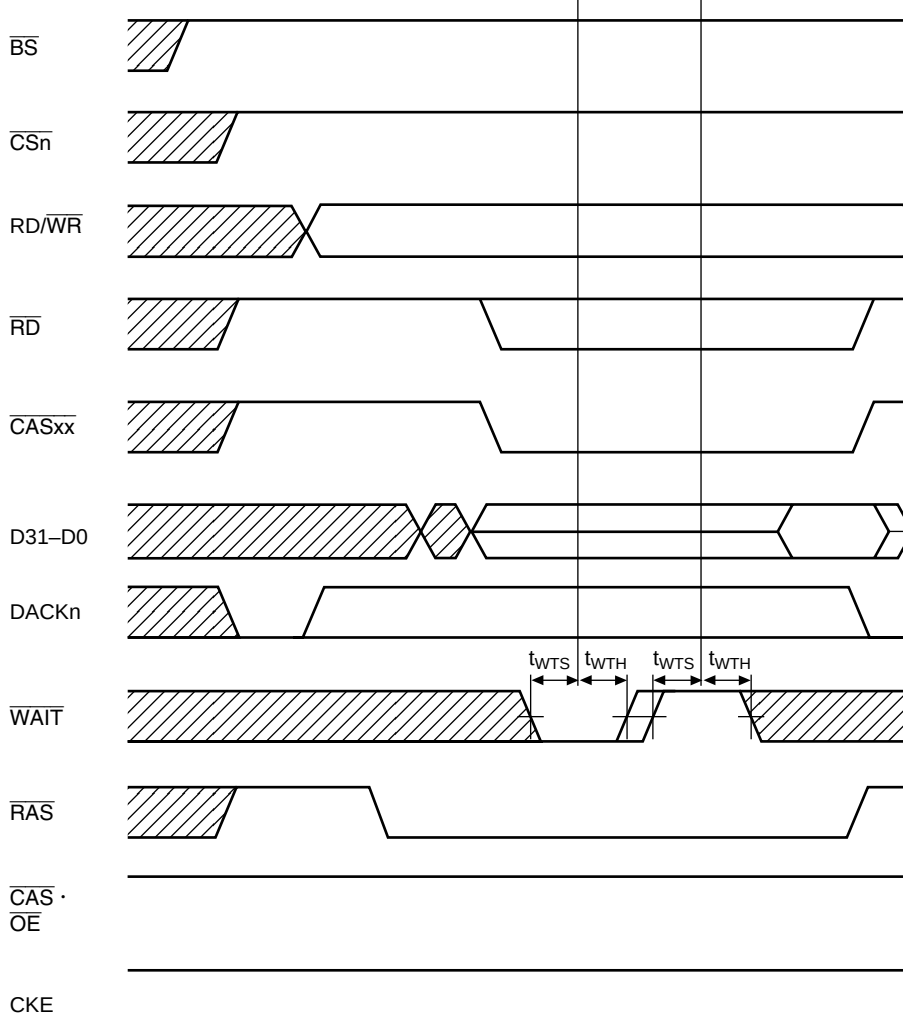
Note: DACKn waveform when active-high is specified

Figure 21.35 DRAM Write Cycle
(TRP = 1 Cycle, RCD = 1 Cycle, No Wait)



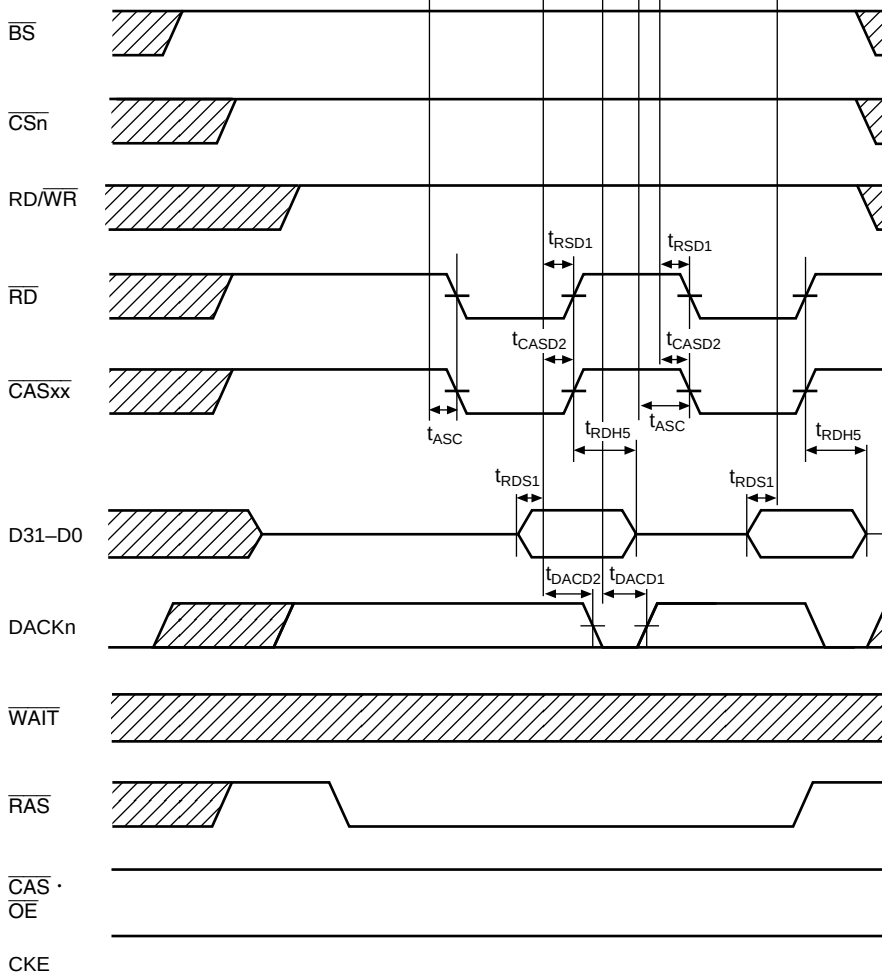
Note: DACKn waveform when active-high is specified

Figure 21.36 DRAM Bus Cycle
(TRP = 2 Cycles, RCD = 2 Cycles, 1 Wait)



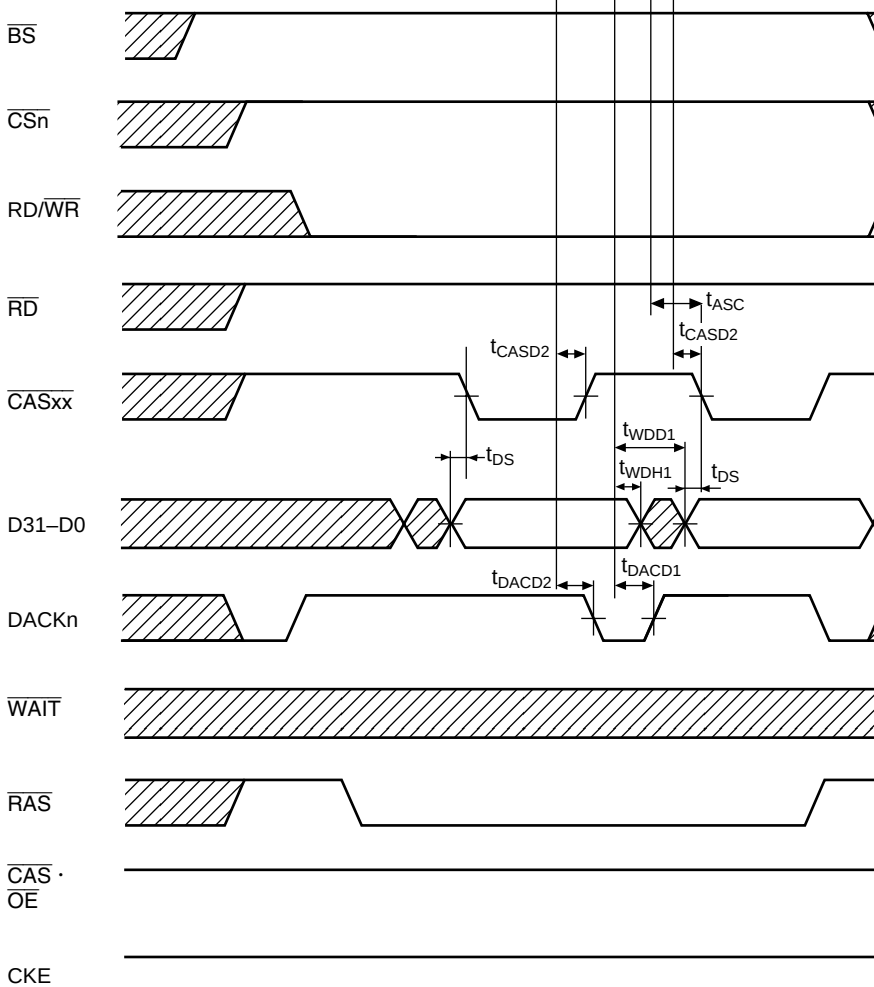
Note: DACKn waveform when active-high is specified

Figure 21.37 DRAM Bus Cycle
(TRP = 1 Cycle, RCD = 1 Cycle, External Wait Input)



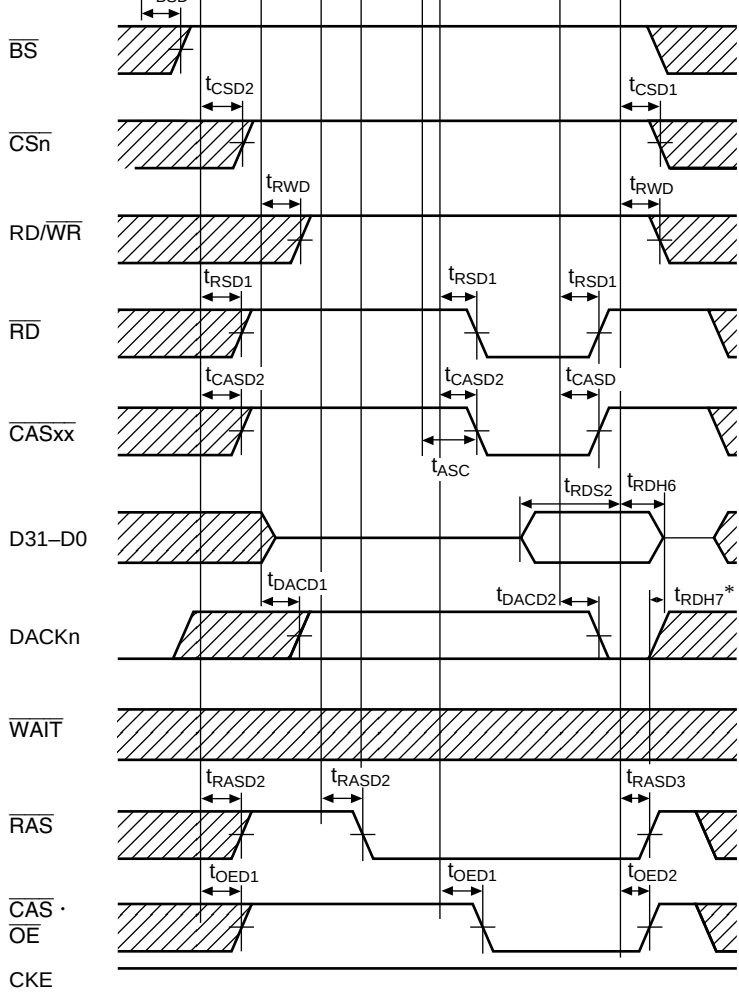
Notes: 1. t_{RDH5} is measured from the rise of $\overline{RD}$ or $\overline{CASxx}$, whichever comes first.
2. DACKn waveform when active-high is specified

Figure 21.38 DRAM Burst Read Cycle (TRP = 1 Cycle, RCD = 1 Cycle, No Wait)



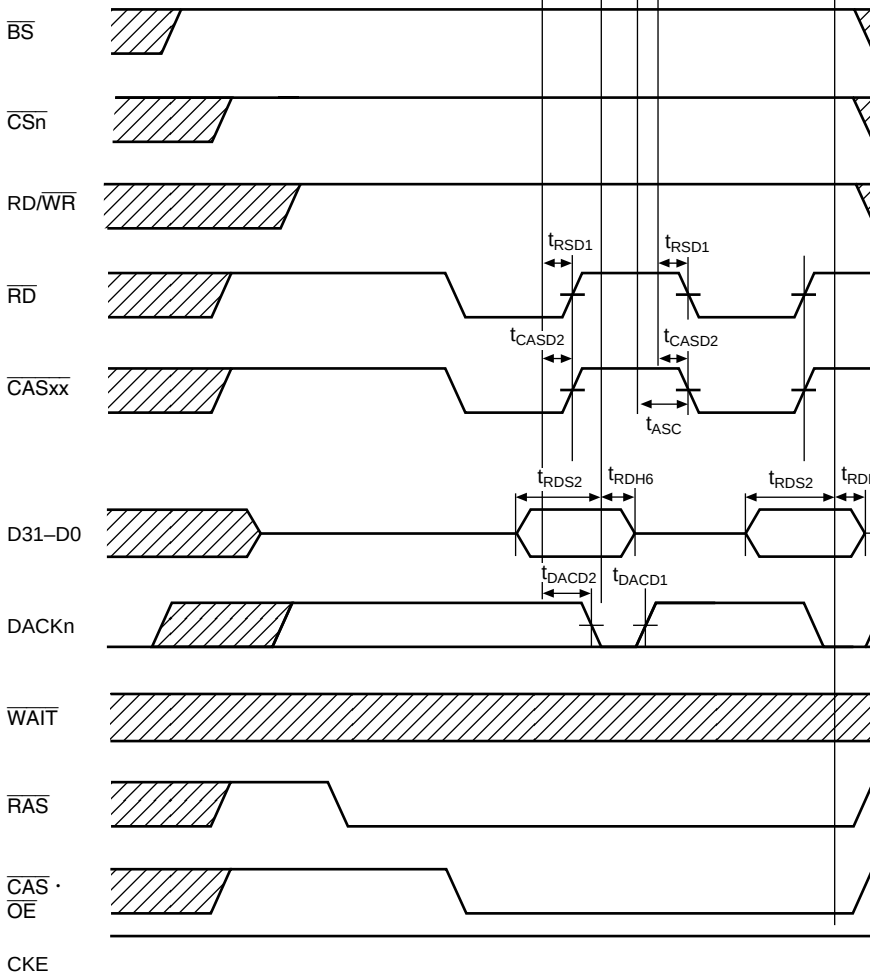
Note: DACKn waveform when active-high is specified

Figure 21.39 DRAM Burst Write Cycle
(TRP = 1 Cycle, RCD = 1 Cycle, No Wait)



Notes: DACKn waveform when active-high is specified
* t_{RDH7} is measured from the rise of $\overline{RAS}$ or $\overline{CAS} \cdot \overline{OE}$, whichever comes first.

Figure 21.40 EDO Read Cycle
(TRP = 1 Cycle, RCD = 1 Cycle, No Wait)



Note: DACKn waveform when active-high is specified

Figure 21.41 EDO Burst Read Cycle
(TRP = 1 Cycle, RCD = 1 Cycle, No Wait)

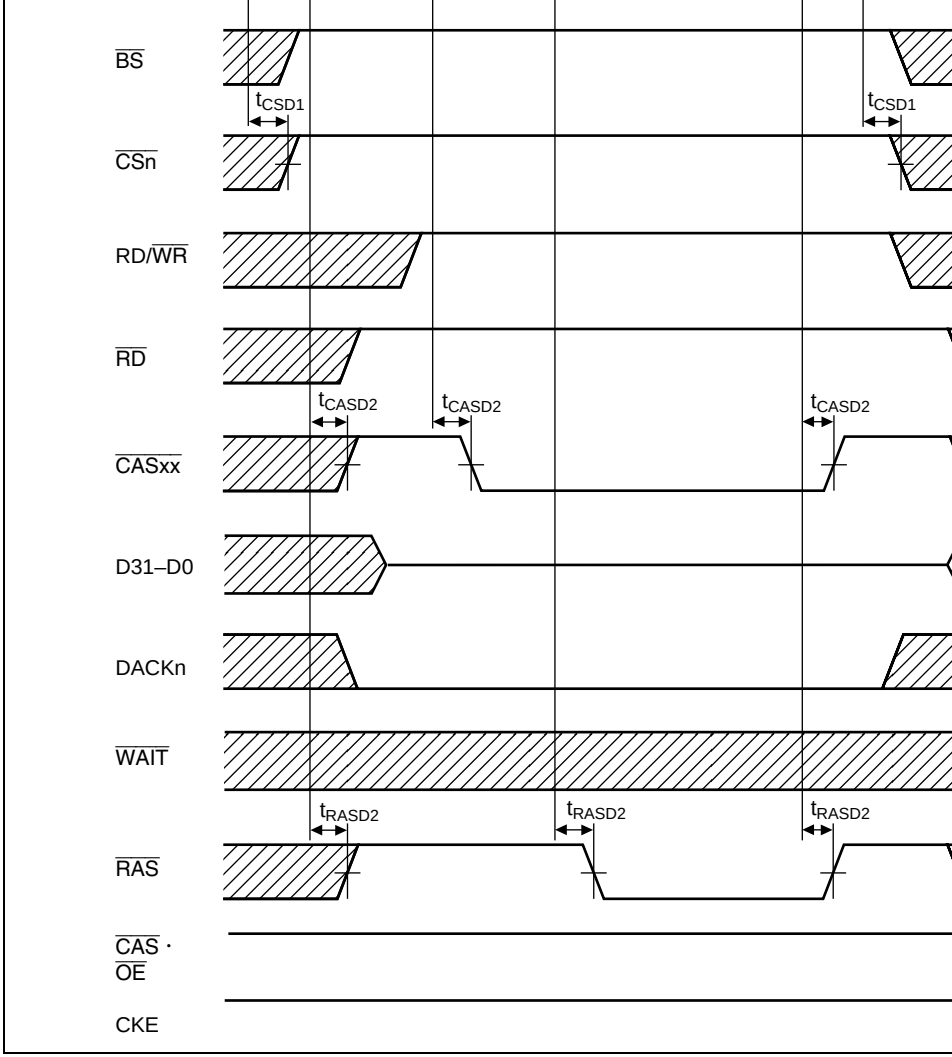
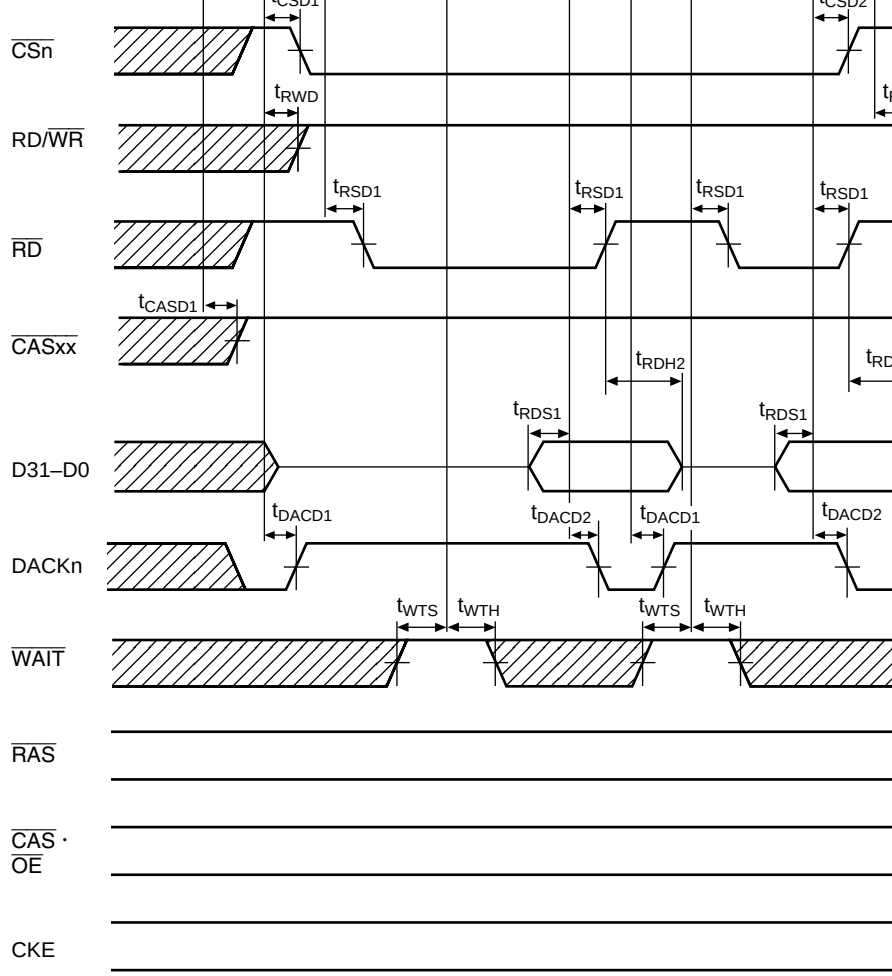


Figure 21.42 DRAM $\overline{\text{CAS}}$-Before-$\overline{\text{RAS}}$ Refresh Cycle (TRP = 1 Cycle, TRAS = 2 Cycles)



Note: DACKn waveform when active-high is specified

Figure 21.43 Burst ROM Read Cycle (Wait = 1)

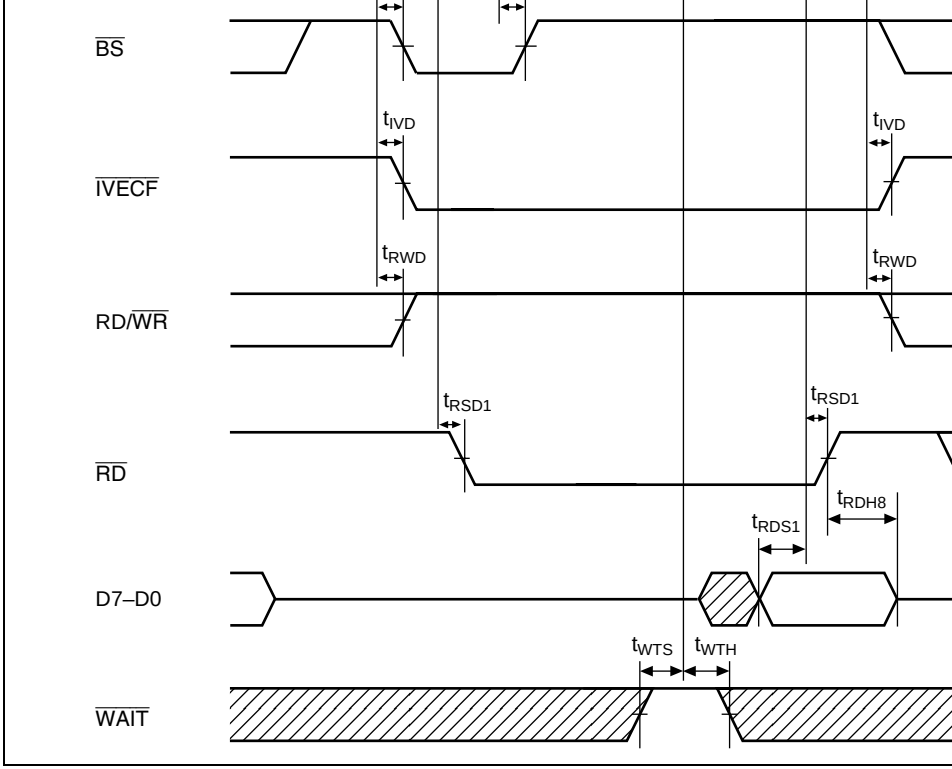


Figure 21.44 Interrupt Vector Fetch Cycle
(No Wait, Iϕ:Eϕ = 1:1)

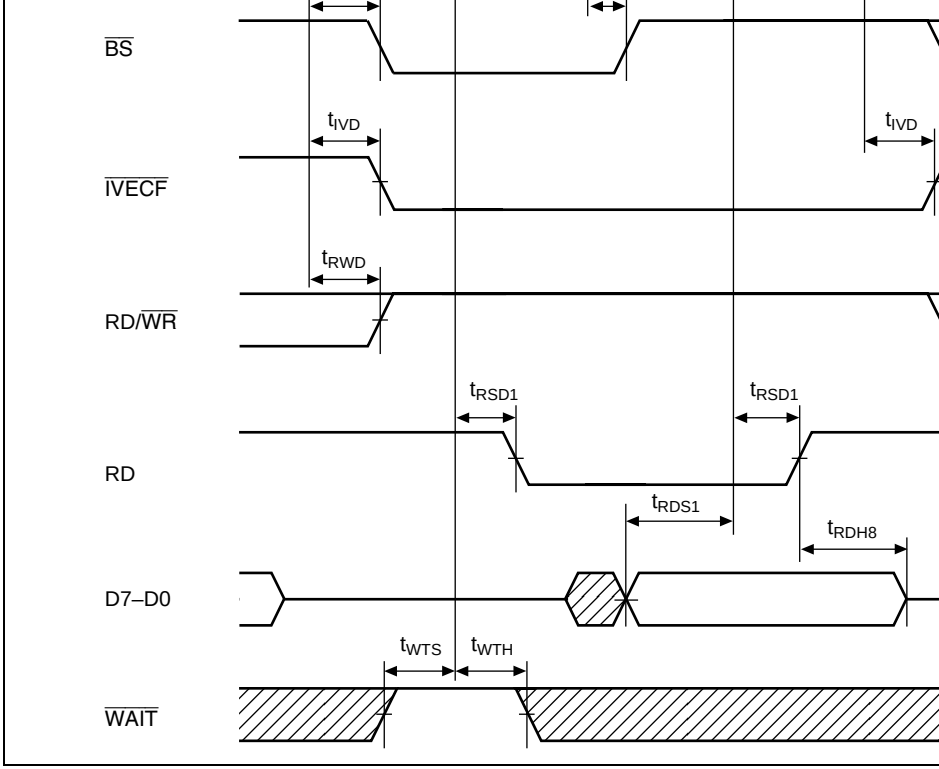


Figure 21.45 Interrupt Vector Fetch Cycle
(No Wait, Except t_{Ecyc}:t_{Pcyc} 1:1)

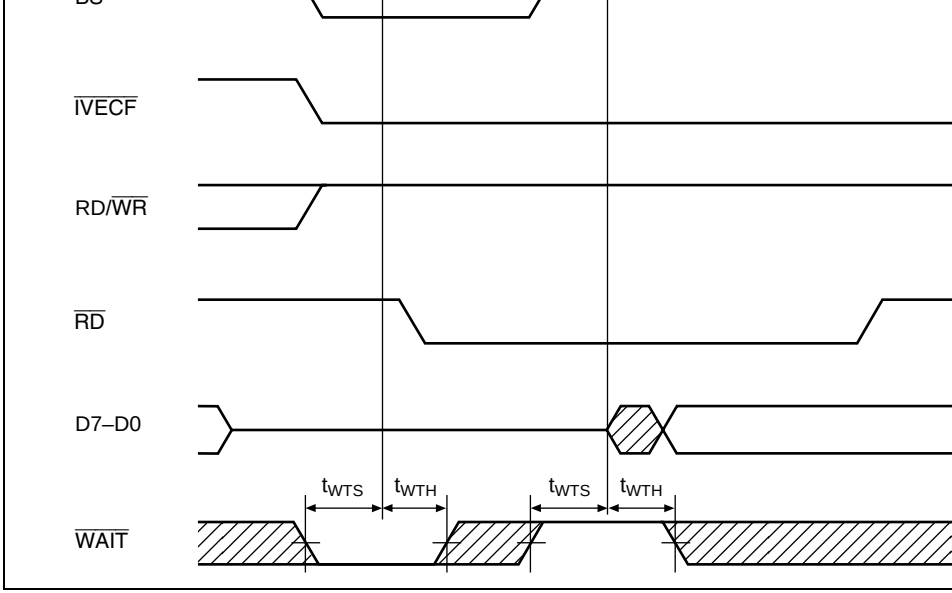


Figure 21.46 Interrupt Vector Fetch Cycle (External Wait Input, Except t_{Ecyc}:t_{Pcyc} 1:1)

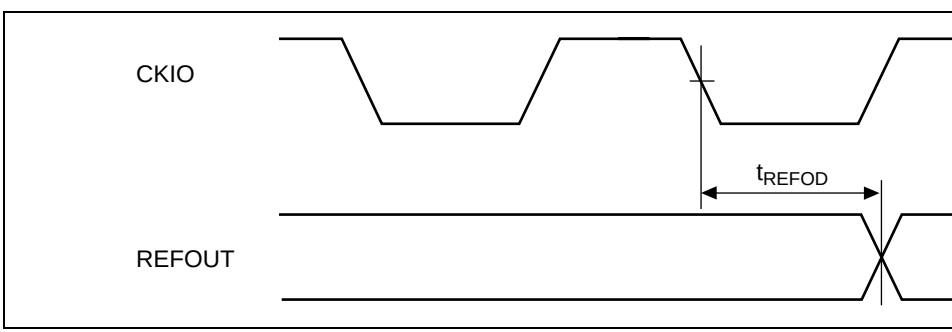


Figure 21.47 REFOUT Delay Time

DREQ0, DREQ1 hold time	t_{DRQH}	5	—	ns

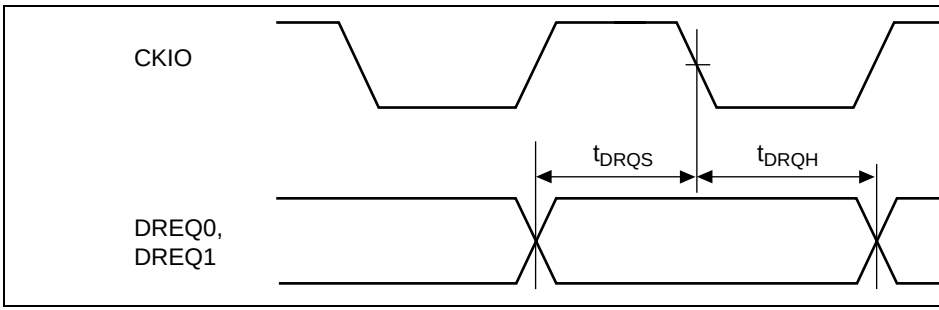


Figure 21.48 DREQ0, DREQ1 Input Timing

Input capture input setup time (t_{Ecyc}:t_{Pcyc} = 1:1)	t_{FICS}	50	—	ns	21
Input capture input setup time (t_{Ecyc}:t_{Pcyc} = 1:2)	t_{FICS}	t_{cyc} + 50	—	ns	21
Input capture input setup time (t_{Ecyc}:t_{Pcyc} = 1:4)	t_{FICS}	$3t_{cyc}$ + 50	—	ns	21
Input capture input hold time	t_{FICH}	50	—	ns	21
Timer clock input setup time (t_{Ecyc}:t_{Pcyc} = 1:1)	t_{FCKS}	50	—	ns	21
Timer clock input setup time (t_{Ecyc}:t_{Pcyc} = 1:2)	t_{FCKS}	t_{cyc} + 50	—	ns	21
Timer clock input setup time (t_{Ecyc}:t_{Pcyc} = 1:4)	t_{FCKS}	$3t_{cyc}$ + 50	—	ns	21
Timer clock pulse width (single edge specified)	t_{FCKWH}	4.5	—	t_{Pcyc}	21
Timer clock pulse width (both edges specified)	t_{FCKWL}	8.5	—	t_{Pcyc}	

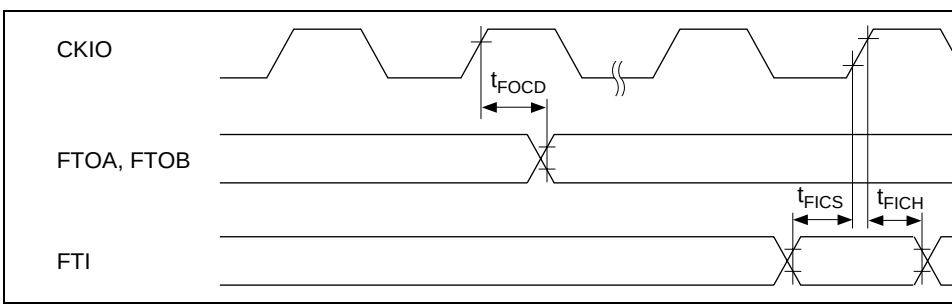


Figure 21.49 FRT Input/Output Timing (t_{Ecyc}:t_{Pcyc} = 1:1)

Figure 21.50 FRT Input/Output Timing (Except t_{Ecyc}:t_{Pcyc} 1:1)

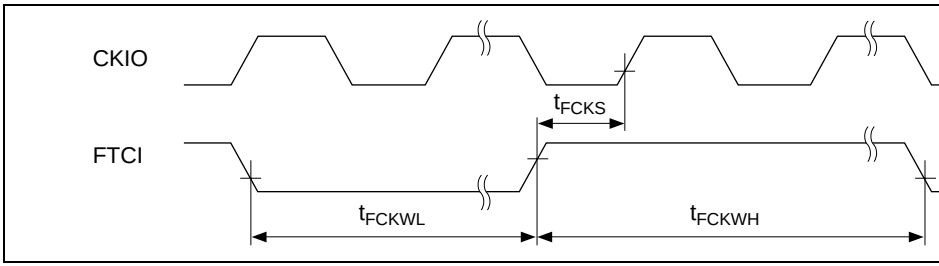


Figure 21.51 FRT Clock Input Timing (t_{Ecyc}:t_{Pcyc} = 1:1)

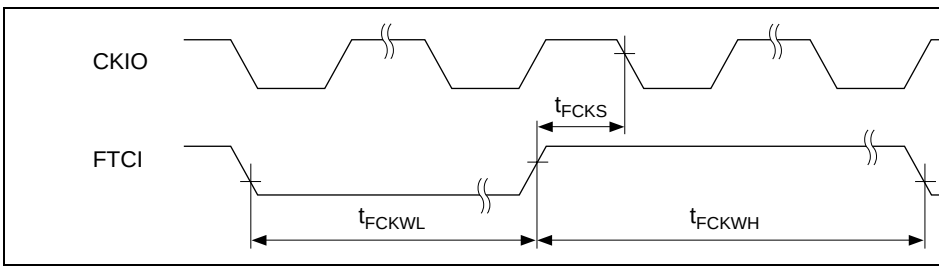


Figure 21.52 FRT Clock Input Timing (Except t_{Ecyc}:t_{Pcyc} 1:1)

Input clock cycle (synchronous mode)	t_{Scyc}	6	—	t_{Pcyc}	2
Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	2
Transmit data delay time (synchronous mode)	t_{TXD}	—	100	ns	2
Receive data setup time (synchronous mode)	t_{RXS}	100	—	ns	
Receive data hold time (synchronous mode)	t_{RXH}	100	—	ns	
$\overline{RTS}$ delay time	t_{RTSD}	—	100	ns	2
$\overline{CTS}$ setup time (synchronous mode)	t_{CTSS}	100	—	ns	
$\overline{CTS}$ hold time (synchronous mode)	t_{CTSH}	100	—	ns	

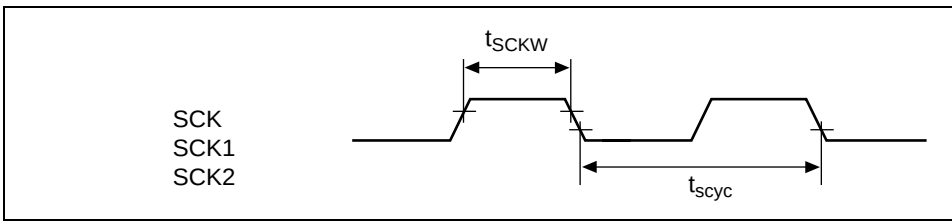


Figure 21.53 Input Clock Input/Output Timing

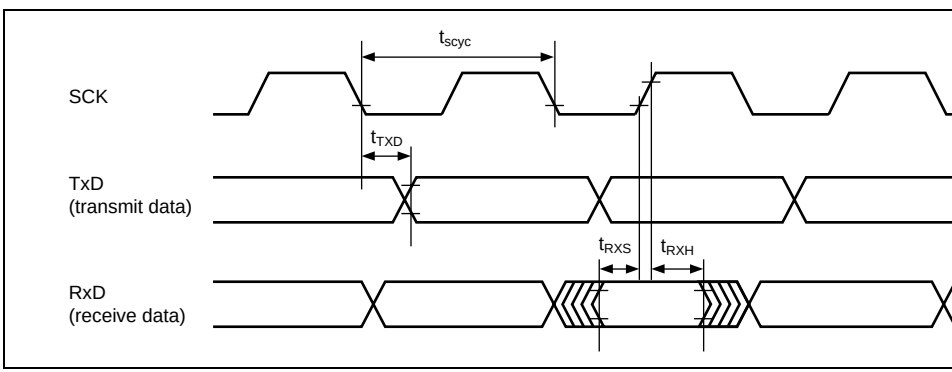


Figure 21.54 SCI Input/Output Timing (Synchronous Mode)

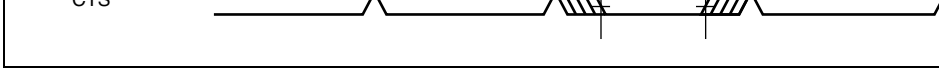


Figure 21.55 $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ Input/Output Timing

Table 21.11 16-Bit Timer-Pulse Unit

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, PV_{CC} = 5.0 V ± 0.5 V/3.3 V ±0.3 V, PV_{C}
V_{SS} = PV_{SS} = $PLLV_{SS}$ = 0 V, Ta = –20 to +75°C

Item		Symbol	Min	Max	Unit
Timer output delay time		t_{TOCD}	—	100	ns
Timer input setup time (t_{Ecyc}:t_{Pcyc} = 1:1)		t_{TICS}	50	—	ns
Timer input setup time (t_{Ecyc}:t_{Pcyc} = 1:2)		t_{TICS}	t_{cyc} + 50	—	ns
Timer input setup time (t_{Ecyc}:t_{Pcyc} = 1:4)		t_{TICS}	$3t_{cyc}$ + 50	—	ns
Timer clock input setup time (t_{Ecyc}:t_{Pcyc} = 1:1)		t_{TCKS}	50	—	ns
Timer clock input setup time (t_{Ecyc}:t_{Pcyc} = 1:2)		t_{TCKS}	t_{cyc} + 50	—	ns
Timer clock input setup time (t_{Ecyc}:t_{Pcyc} = 1:4)		t_{TCKS}	$3t_{cyc}$ + 50	—	ns
Timer clock pulse width	Single edge specified	t_{TCKWH}	1.5	—	t_{cyc}
	Both edges specified	t_{TCKWL}	2.5	—	

Note: * TIOCA0–TIOCA2, TIOCB0–TIOCB2, TIOCC0, TIOCD0

Figure 21.56 TPU Input/Output Timing (t_{Ecyc}:t_{Pcyc} = 1:1)

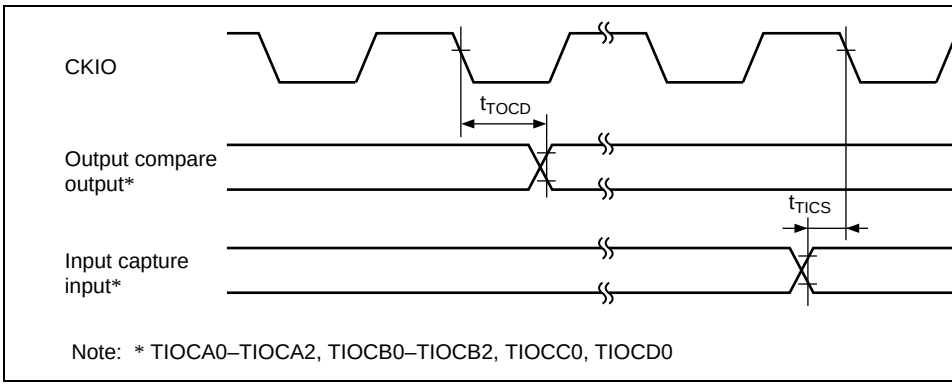


Figure 21.57 TPU Input/Output Timing (Except t_{Ecyc}:t_{Pcyc} 1:1)

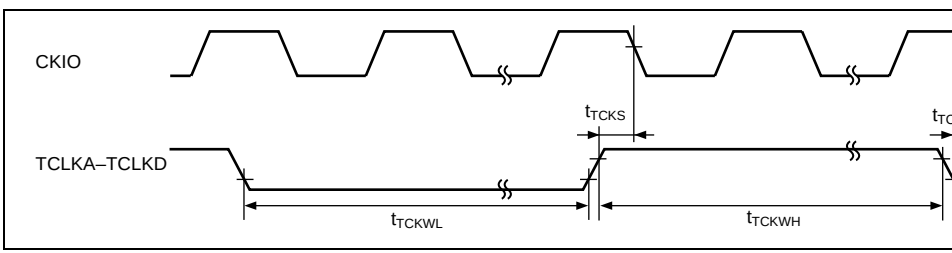


Figure 21.58 TPU Clock Input Timing

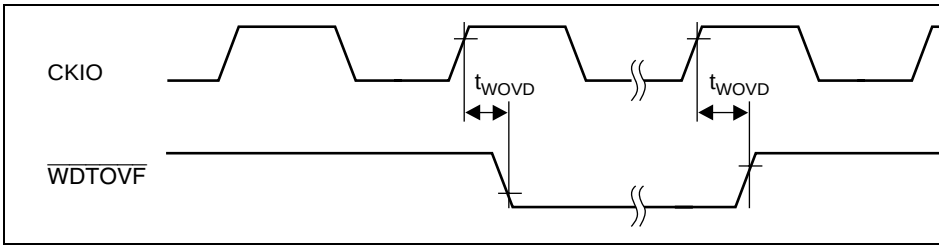


Figure 21.59 Watchdog Timer Output Timing (t_{Ecyc}:t_{Pcyc} = 1:1)

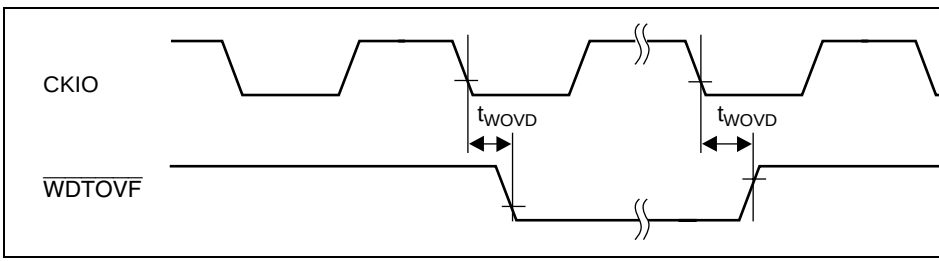


Figure 21.60 Watchdog Timer Output Timing (Except t_{Ecyc}:t_{Pcyc} 1:1)

SRCK, STCK clock input low-level width	t_{WL}	$0.4 \times t_{slcyc}$	—	ns
SRCK, STCK clock input high-level width	t_{WH}	$0.4 \times t_{slcyc}$	—	ns
SRS input setup time	t_{RSS}	15	—	ns
SRS input hold time	t_{RSH}	10	—	ns
SRXD input setup time	t_{SRDS}	15	—	ns
SRXD input hold time	t_{SRDH}	10	—	ns
STS input setup time	t_{TSS}	15	—	ns
STS input hold time	t_{TSH}	10	—	ns
STS output delay time	t_{TSD}	0	20	ns
STXD output delay time	t_{TDD}	0	20	ns

Note: * Specified as tPcyc or 66.7, whichever is greater.

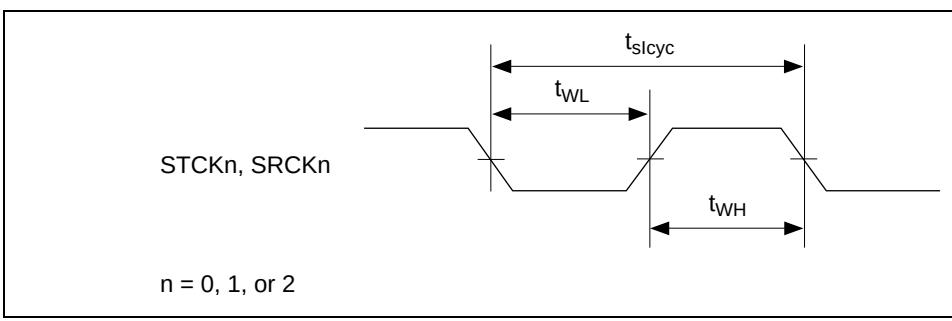


Figure 21.61 SIO Input Clock Timing

n = 0, 1, or 2

Figure 21.62 SIO Receive Timing

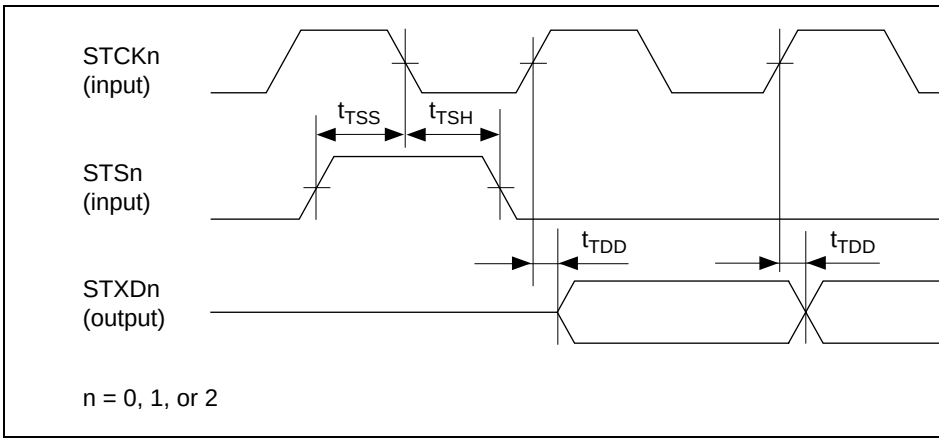


Figure 21.63 SIO Transmit Timing (TMn = 0 Mode)

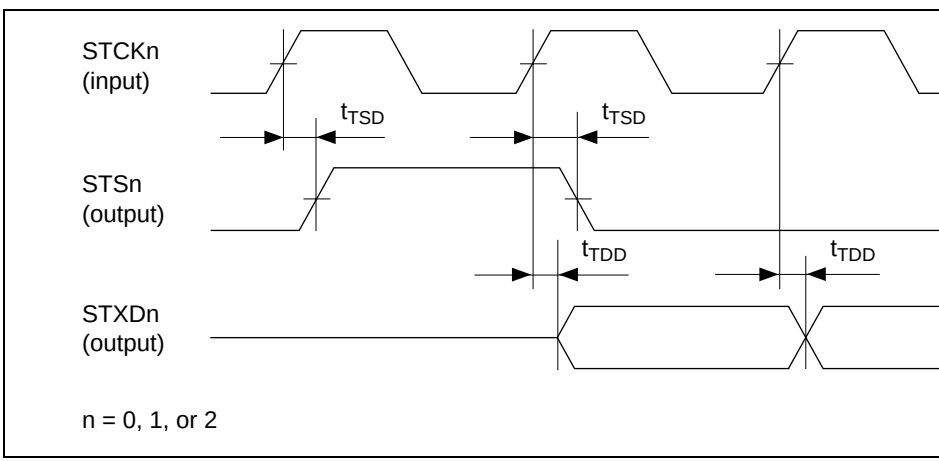


Figure 21.64 SIO Transmit Timing (TMn = 1 Mode)

TCK clock input high-level width	t_{TCKH}	0.4	0.6	t_{tcyc}	
TCK clock input low-level width	t_{TCKL}	0.4	0.6	t_{tcyc}	
$\overline{TRST}$ pulse width	t_{TRSW}	20	—	t_{tcyc}	21
$\overline{TRST}$ setup time	t_{TRSS}	40	—	ns	
TMS setup time	t_{TMSS}	30	—	ns	21
TMS hold time	t_{TMSH}	10	—	ns	
TDI setup time	t_{TDIS}	30	—	ns	
TDI hold time	t_{TDIH}	10	—	ns	
TDO delay time	t_{TDOD}	0	30	ns	

Note: * Specified as tPcyc or 66.7, whichever is greater.

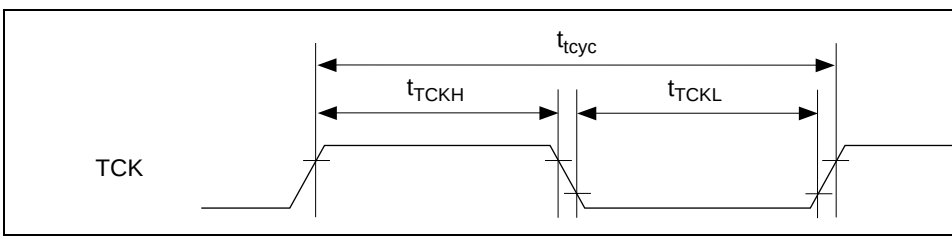


Figure 21.65 H-UDI Clock Timing

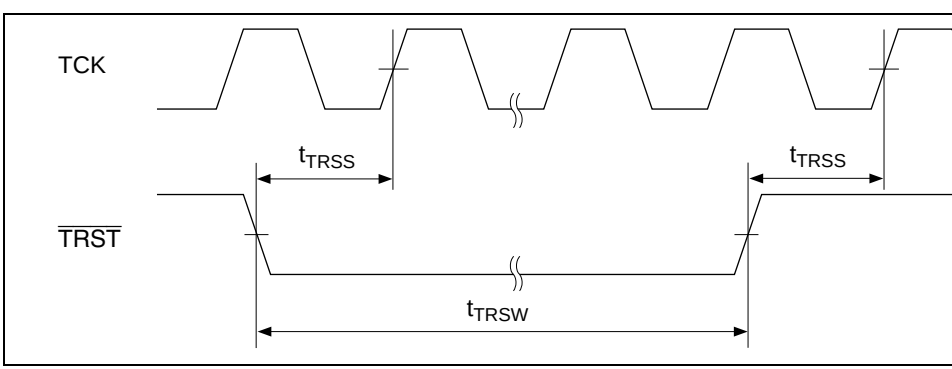


Figure 21.66 H-UDI $\overline{TRST}$ Timing

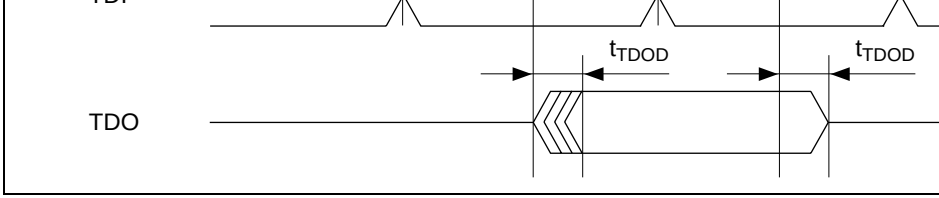


Figure 21.67 H-UDI Input/Output Timing

21.3.10 I/O Port Timing

Table 21.15 I/O Port Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, PV_{CC} = 5.0 V ± 0.5 V/3.3 V ±0.3 V, PV_{C}
V_{SS} = PV_{SS} = $PLLV_{SS}$ = 0 V, Ta = –20 to +75°C

Item	Symbol	Min	Max	Unit	F
Port output data delay time	t_{PWD}	—	50	ns	2
Port input data setup time (t_{Ecyc}:t_{Pcyc} = 1:1)	t_{PRS}	50	—	ns	2
Port input data setup time (t_{Ecyc}:t_{Pcyc} = 1:2)	t_{PRS}	t_{cyc} + 50	—	ns	2
Port input data setup time (t_{Ecyc}:t_{Pcyc} = 1:4)	t_{PRS}	$3t_{cyc}$ + 50	—	ns	
Port input data hold time	t_{PRH}	50	—	ns	2

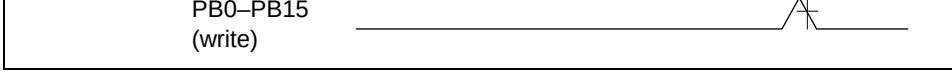


Figure 21.68 I/O Port Input/Output Timing (t_{Ecyc}:t_{Pcyc} = 1:1)

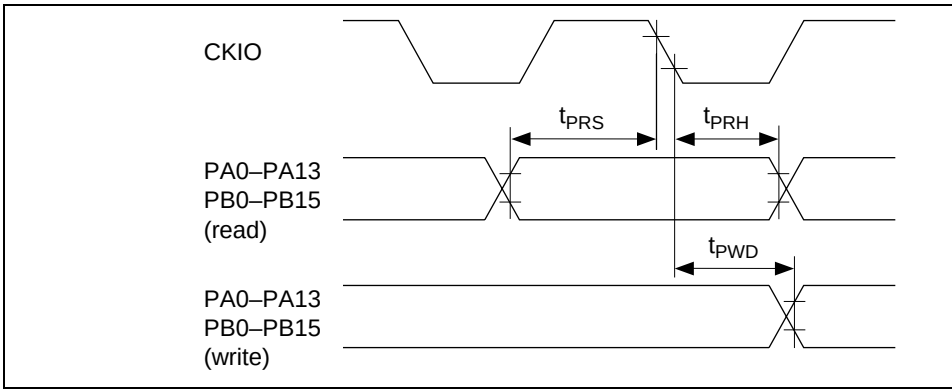


Figure 21.69 I/O Port Input/Output Timing (Except t_{Ecyc}:t_{Pcyc} 1:1)

TX-EN output delay time	t_{TENd}	3	—	20	ns
ETXD[3:0] output delay time	t_{ETDd}	3	—	20	ns
CRS setup time	t_{CRSs}	10	—	—	ns
CRS hold time	t_{CRSh}	10	—	—	ns
COL setup time	t_{COLs}	10	—	—	ns
COL hold time	t_{COLh}	10	—	—	ns
RX-CLK cycle time	t_{Rcyc}	40	—	—	ns
RX-DV setup time	t_{RDVs}	10	—	—	ns
RX-DV hold time	t_{RDVh}	3	—	—	ns
ERXD[3:0] setup time	t_{ERDs}	10	—	—	ns
ERXD[3:0] hold time	t_{ERDh}	3	—	—	ns
RX-ER setup time	t_{RERs}	10	—	—	ns
RX-ER hold time	t_{RERh}	3	—	—	ns
MDIO setup time	t_{MDIOs}	10	—	—	ns
MDIO hold time	t_{MDIOh}	10	—	—	ns
MDIO output data hold time*	t_{MDIOdh}	5	—	18	ns
WOL output delay time	t_{WOLd}	1	—	14	ns
EXOUT output delay time	t_{EXOUTd}	1	—	25	ns

Note: * The user must ensure that the code satisfies this condition.

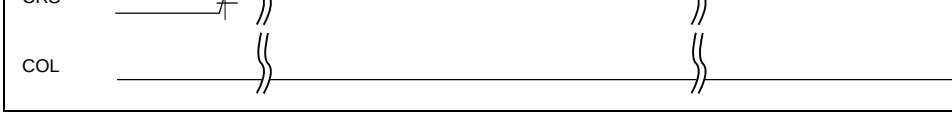


Figure 21.70 MII Transmit Timing (Normal Operation)

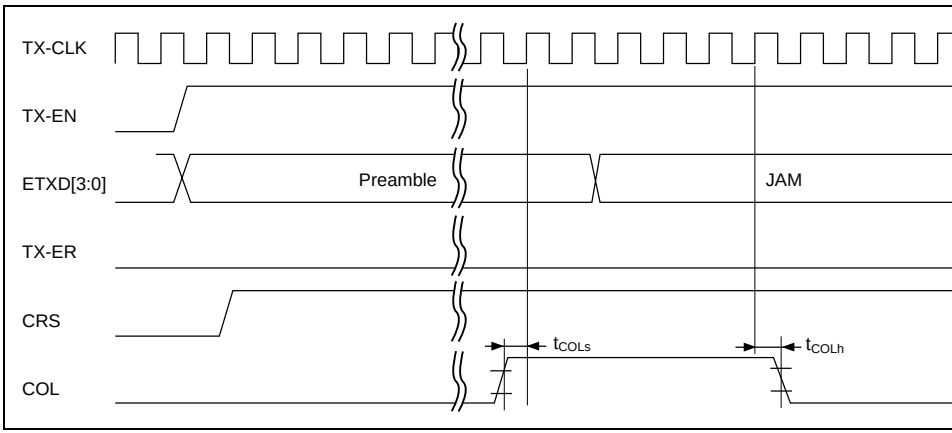


Figure 21.71 MII Transmit Timing (Case of Conflict)

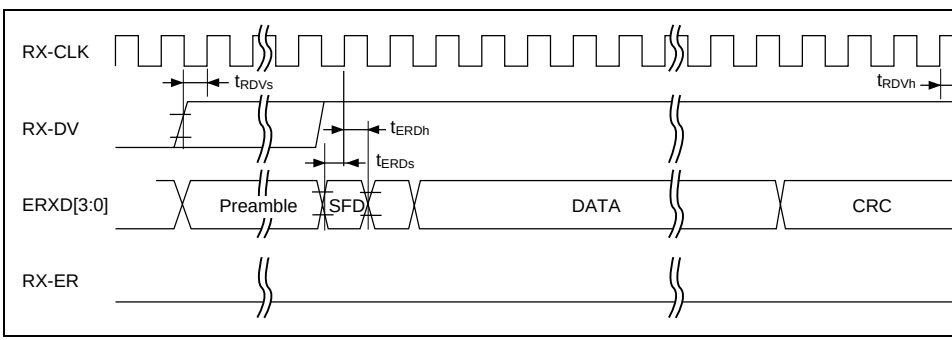


Figure 21.72 MII Receive Timing (Normal Operation)

Figure 21.73 MII Receive Timing (Case of Error)

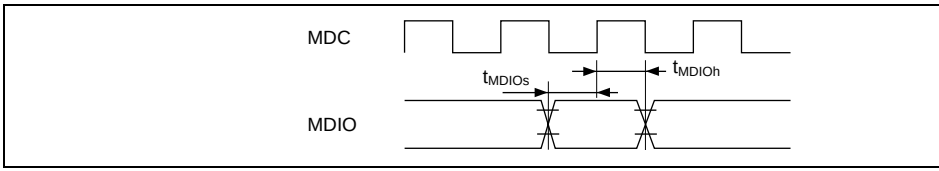


Figure 21.74 MDIO Input Timing

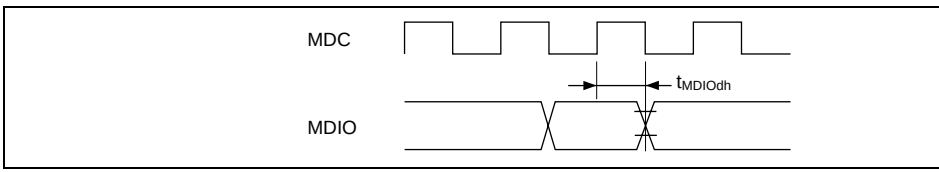


Figure 21.75 MDIO Output Timing

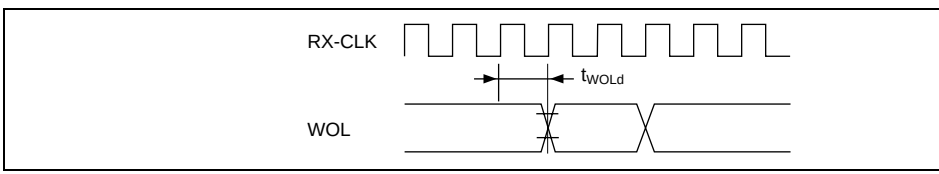


Figure 21.76 WOL Output Timing

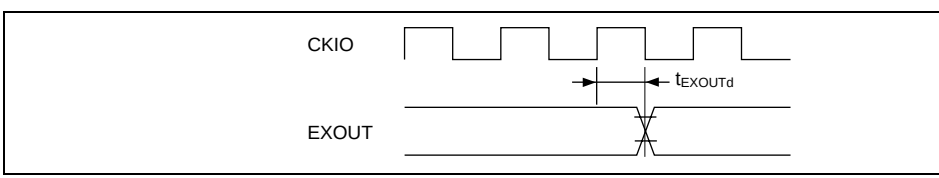


Figure 21.77 EXOUT Output Timing

$\overline{BH}$ output rising edge delay time	t_{BHNrd}	—	—	16	ns
$\overline{BH}$ output falling edge delay time	t_{BHNfd}	—	—	16	ns
$\overline{BUSHiZ}$ setup time	t_{BHIZs}	7	—	—	ns
$\overline{BUSHiZ}$ hold time	t_{BHIZh}	8	—	—	ns
Output delay time of target pins	t_{BHIZd}	—	—	16	ns

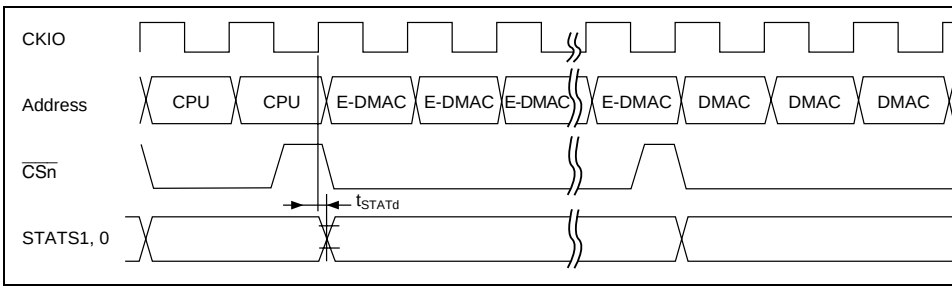


Figure 21.78 STATS Output Timing

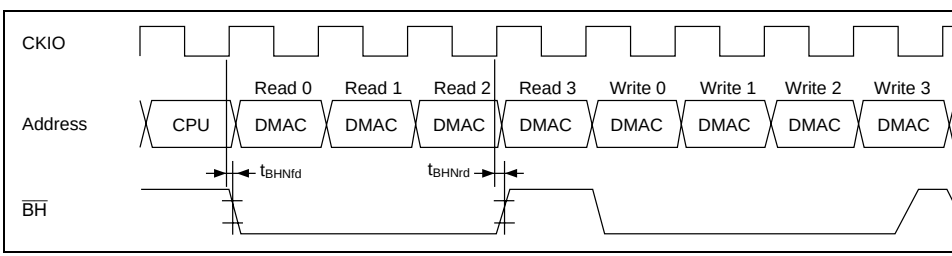
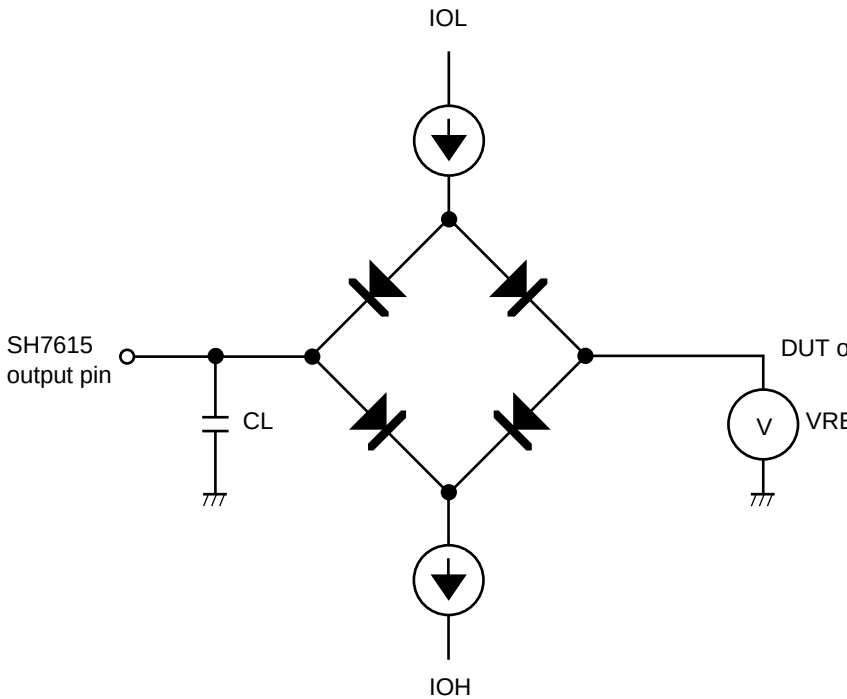


Figure 21.79 $\overline{BH}$ Output Timing

Figure 21.80 BUSHiZ **Bus Timing**

The output load circuit is shown in figure 21.80.



CL is the total value, including the capacitance of the test jig, etc. The capacitance of e
as follows:

30 pF: CKIO, A24–A0, D31–D0, $\overline{\text{BS}}$, $\overline{\text{RD}}$, $\overline{\text{CS4}}$–$\overline{\text{CS0}}$, DQMUU/$\overline{\text{WE3}}$–DQMLL/$\overline{\text{WE0}}$, $\overline{\text{CA}}$
$\overline{\text{RAS}}$, $\overline{\text{CAS}}$/$\overline{\text{OE}}$, DACK1, DACK0

50 pF: All other pins

IOL and IOH values are as shown in table 21.3, Permissible Output Currents.

Figure 21.81 Output Load Circuit

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
H'FFFFFC00	SIRDR0								
H'FFFFFC01									
H'FFFFFC02	SITDR0								
H'FFFFFC03									
H'FFFFFC04	SICTR0	—	—	—	—	—	—	—	—
H'FFFFFC05		—	TM	SE	DL	TIE	RIE	TE	RE
H'FFFFFC06	SISTR0	—	—	—	—	—	—	—	—
H'FFFFFC07		—	—	—	—	TERR	RERR	TDRE	RDRF
H'FFFFFC08 to H'FFFFFC0F	—	—	—	—	—	—	—	—	—
H'FFFFFC10	SIRDR1								
H'FFFFFC11									
H'FFFFFC12	SITDR1								
H'FFFFFC13									
H'FFFFFC14	SICTR1	—	—	—	—	—	—	—	—
H'FFFFFC15		—	TM	SE	DL	TIE	RIE	TE	RE
H'FFFFFC16	SISTR1	—	—	—	—	—	—	—	—
H'FFFFFC17		—	—	—	—	TERR	RERR	TDRE	RDRF
H'FFFFFC18 to H'FFFFFC1F	—	—	—	—	—	—	—	—	—
H'FFFFFC20	SIRDR2								
H'FFFFFC21									
H'FFFFFC22	SITDR2								
H'FFFFFC23									
H'FFFFFC24	SICTR2	—	—	—	—	—	—	—	—
H'FFFFFC25		—	TM	SE	DL	TIE	RIE	TE	RE
H'FFFFFC26	SISTR2	—	—	—	—	—	—	—	—
H'FFFFFC27		—	—	—	—	TERR	RERR	TDRE	RDRF

H'FFFFFC50	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'FFFFFC51	TMDR0	—	—	BFB	BFA	MD3	MD2	MD1	MD0
H'FFFFFC52	TIOR0H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
H'FFFFFC53	TIOR0L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
H'FFFFFC54	TIER0	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
H'FFFFFC55	TSR0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
H'FFFFFC56	TCNT0								
H'FFFFFC57									
H'FFFFFC58	TGR0A								
H'FFFFFC59									
H'FFFFFC5A	TGR0B								
H'FFFFFC5B									
H'FFFFFC5C	TGR0C								
H'FFFFFC5D									
H'FFFFFC5E	TGR0D								
H'FFFFFC5F									
H'FFFFFC60	TCR1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'FFFFFC61	TMDR1	—	—	—	—	MD3	MD2	MD1	MD0
H'FFFFFC62	TIOR1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
H'FFFFFC63	—	—	—	—	—	—	—	—	—
H'FFFFFC64	TIER1	—	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
H'FFFFFC65	TSR1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
H'FFFFFC66	TCNT1								
H'FFFFFC67									
H'FFFFFC68	TGR1A								
H'FFFFFC69									
H'FFFFFC6A	TGR1B								
H'FFFFFC6B									
H'FFFFFC6C to H'FFFFFC6F	—	—	—	—	—	—	—	—	—

H'FFFFFC77									
H'FFFFFC78	TGR2A								
H'FFFFFC79									
H'FFFFFC7A	TGR2B								
H'FFFFFC7B									
H'FFFFFC7C to H'FFFFFC7F	—	—	—	—	—	—	—	—	—
H'FFFFFC80	PACR	—	—	PA13MD	PA12MD	PA11MD	PA10MD	PA9MD	PA8M
H'FFFFFC81		PA7MD	PA6MD	PA5MD	PA4MD	PA3MD	PA2MD	PA1MD	PA0M
H'FFFFFC82	PAIOR	—	—	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IO
H'FFFFFC83		PA7IOR	PA6IOR	PA5IOR	PA4IOR	—	PA2IOR	PA1IOR	PA0IO
H'FFFFFC84	PADR	—	—	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8D
H'FFFFFC85		PA7DR	PA6DR	PA5DR	PA4DR	—	PA2DR	PA1DR	PA0D
H'FFFFFC86 to H'FFFFFC87	—	—	—	—	—	—	—	—	—
H'FFFFFC88	PBCR	PB15MD1	PB15MD0	PB14MD1	PB14MD0	PB13MD1	PB13MD0	PB12MD1	PB12
H'FFFFFC89		PB11MD1	PB11MD0	PB10MD1	PB10MD0	PB9MD1	PB9MD0	PB8MD1	PB8M
H'FFFFFC8A	PBIOR	PB15IOR	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IO
H'FFFFFC8B		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IO
H'FFFFFC8C	PBDR	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8D
H'FFFFFC8D		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0D
H'FFFFFC8E	PBCR2	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4M
H'FFFFFC8F		PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0M
H'FFFFFC90 to H'FFFFFCAF	—	—	—	—	—	—	—	—	—
H'FFFFFCB0	SDIR	TS3	TS2	TS1	TS0	—	—	—	—
H'FFFFFCB1		—	—	—	—	—	—	—	—
H'FFFFFCB2	SDSR	—	—	—	—	—	—	—	—
H'FFFFFCB3		—	—	—	—	—	—	—	SDTR

H'FFFFFCC0	SCSMR1	C/A	CHR/ICK3	PE/ICK2	O/E/ICK1	STOP/ ICK0	MP	CKS1	CKS0
H'FFFFFCC1	—	—	—	—	—	—	—	—	—
H'FFFFFCC2	SCBRR1								
H'FFFFFCC3	—	—	—	—	—	—	—	—	—
H'FFFFFCC4	SCSCR1	TIE	RIE	TE	RE	MPIE	—	CKE1	CKE0
H'FFFFFCC5	—	—	—	—	—	—	—	—	—
H'FFFFFCC6	SCFTDR1								
H'FFFFFCC7	—	—	—	—	—	—	—	—	—
H'FFFFFCC8	SC1SSR1	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0
H'FFFFFCC9		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
H'FFFFFCCA	SC2SSR1	TLM	RLM	N1	N0	MPB	MPBT	EI	ORER
H'FFFFFCCB	—	—	—	—	—	—	—	—	—
H'FFFFFCCC	SCFRDR1								
H'FFFFFCCD	—	—	—	—	—	—	—	—	—
H'FFFFFCCE	SCFCR1	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP
H'FFFFFCCF	—	—	—	—	—	—	—	—	—
H'FFFFFCD0	SCFDR1	—	—	—	T4	T3	T2	T1	T0
H'FFFFFCD1		—	—	—	R4	R3	R2	R1	R0
H'FFFFFCD2	SCFER1	ED15	ED14	ED13	ED12	ED11	ED10	ED9	ED8
H'FFFFFCD3		ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
H'FFFFFCD4	SCIMR1	IRMOD	PSEL	RIVS	—	—	—	—	—
H'FFFFFCD5 to H'FFFFFCDF	—	—	—	—	—	—	—	—	—
H'FFFFFCE0	SCSMR2	C/$\overline{A}$	CHR/ICK3	PE/ICK2	O/$\overline{E}$/ICK1	STOP/ ICK0	MP	CKS1	CKS0
H'FFFFFCE1	—	—	—	—	—	—	—	—	—
H'FFFFFCE2	SCBRR2								
H'FFFFFCE3	—	—	—	—	—	—	—	—	—

H'FFFFFCEA	SC2SSR2	TLM	RLM	N1	N0	MPB	MPBT	EI	ORER
H'FFFFFCEB	—	—	—	—	—	—	—	—	—
H'FFFFFCEC	SCFRDR2								
H'FFFFFCED	—	—	—	—	—	—	—	—	—
H'FFFFFCEE	SCFCR2	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP
H'FFFFFCEF	—	—	—	—	—	—	—	—	—
H'FFFFFCF0	SCFDR2	—	—	—	T4	T3	T2	T1	T0
H'FFFFFCF1		—	—	—	R4	R3	R2	R1	R0
H'FFFFFCF2	SCFER2	ED15	ED14	ED13	ED12	ED11	ED10	ED9	ED8
H'FFFFFCF3		ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
H'FFFFFCF4	SCIMR2	IRMOD	PSEL	RIVS	—	—	—	—	—
H'FFFFFCF5 to H'FFFFFCFF	—	—	—	—	—	—	—	—	—
H'FFFFFD00	EDMR	—	—	—	—	—	—	—	—
H'FFFFFD01		—	—	—	—	—	—	—	—
H'FFFFFD02		—	—	—	—	—	—	—	—
H'FFFFFD03		—	—	DL1	DL0	—	—	—	SWR
H'FFFFFD04	EDTRR	—	—	—	—	—	—	—	—
H'FFFFFD05		—	—	—	—	—	—	—	—
H'FFFFFD06		—	—	—	—	—	—	—	—
H'FFFFFD07		—	—	—	—	—	—	—	TR
H'FFFFFD08	EDRRR	—	—	—	—	—	—	—	—
H'FFFFFD09		—	—	—	—	—	—	—	—
H'FFFFFD0A		—	—	—	—	—	—	—	—
H'FFFFFD0B		—	—	—	—	—	—	—	RR

H'FFFFFD12		RDLA15	RDLA14	RDLA13	RDLA12	RDLA11	RDLA10	RDLA9	RDLA8
H'FFFFFD13		RDLA7	RDLA6	RDLA5	RDLA4	RDLA3	RDLA2	RDLA1	RDLA0
H'FFFFFD14	EESR	—	—	—	—	—	TABT	RABT	RFCOF
H'FFFFFD15		—	ECI	TC	TDE	TFUF	FR	RDE	RFOF
H'FFFFFD16		—	—	—	ITF	CND	DLC	CD	TRO
H'FFFFFD17		RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
H'FFFFFD18	EESIPR	—	—	—	—	—	—	—	RFCOF
H'FFFFFD19		—	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
H'FFFFFD1A		—	—	—	ITFIP	CNDIP	DLCIP	CDIP	TROIP
H'FFFFFD1B		RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
H'FFFFFD1C	TRSCER	—	—	—	—	—	—	—	—
H'FFFFFD1D		—	—	—	—	—	—	—	—
H'FFFFFD1E		—	—	—	—	—	—	—	—
H'FFFFFD1F		RMAFCE	—	—	—	—	—	—	—
H'FFFFFD20	RMFCR	—	—	—	—	—	—	—	—
H'FFFFFD21		—	—	—	—	—	—	—	—
H'FFFFFD22		MFC15	MFC14	MFC13	MFC12	MFC11	MFC10	MFC9	MFC8
H'FFFFFD23		MFC7	MFC6	MFC5	MFC4	MFC3	MFC2	MFC1	MFC0
H'FFFFFD24	TFTR	—	—	—	—	—	—	—	—
H'FFFFFD25		—	—	—	—	—	—	—	—
H'FFFFFD26		—	—	—	—	—	TFT10	TFT9	TFT8
H'FFFFFD27		TFT7	TFT6	TFT5	TFT4	TFT3	TFT2	TFT1	TFT0
H'FFFFFD28	FDR	—	—	—	—	—	—	—	—
H'FFFFFD29		—	—	—	—	—	—	—	—
H'FFFFFD2A		—	—	—	—	—	—	—	TFD
H'FFFFFD2B		—	—	—	—	—	—	—	RFD

H'FFFFFD32		—	—	—	—	—	—	—	—
H'FFFFFD33		—	—	—	—	FEC	AEC	EDH	—
H'FFFFFD34 to H'FFFFFD3F	—	—	—	—	—	—	—	—	—
H'FFFFFD40	RBWAR	RBWA31	RBWA30	RBWA29	RBWA28	RBWA27	RBWA26	RBWA25	RBW
H'FFFFFD41		RBWA23	RBWA22	RBWA21	RBWA20	RBWA19	RBWA18	RBWA17	RBW
H'FFFFFD42		RBWA15	RBWA14	RBWA13	RBWA12	RBWA11	RBWA10	RBWA9	RBW
H'FFFFFD43		RBWA7	RBWA6	RBWA5	RBWA4	RBWA3	RBWA2	RBWA1	RBW
H'FFFFFD44	RDFAR	RDFA31	RDFA30	RDFA29	RDFA28	RDFA27	RDFA26	RDFA25	RDFA
H'FFFFFD45		RDFA23	RDFA22	RDFA21	RDFA20	RDFA19	RDFA18	RDFA17	RDFA
H'FFFFFD46		RDFA15	RDFA14	RDFA13	RDFA12	RDFA11	RDFA10	RDFA9	RDFA
H'FFFFFD47		RDFA7	RDFA6	RDFA5	RDFA4	RDFA3	RDFA2	RDFA1	RDFA
H'FFFFFD48 to H'FFFFFD4B	—	—	—	—	—	—	—	—	—
H'FFFFFD4C	TBRAR	TBRA31	TBRA30	TBRA29	TBRA28	TBRA27	TBRA26	TBRA25	TBRA
H'FFFFFD4D		TBRA23	TBRA22	TBRA21	TBRA20	TBRA19	TBRA18	TBRA17	TBRA
H'FFFFFD4E		TBRA15	TBRA14	TBRA13	TBRA12	TBRA11	TBRA10	TBRA9	TBRA
H'FFFFFD4F		TBRA7	TBRA6	TBRA5	TBRA4	TBRA3	TBRA2	TBRA1	TBRA
H'FFFFFD50	TDFAR	TDFA31	TDFA30	TDFA29	TDFA28	TDFA27	TDFA26	TDFA25	TDFA
H'FFFFFD51		TDFA23	TDFA22	TDFA21	TDFA20	TDFA19	TDFA18	TDFA17	TDFA
H'FFFFFD52		TDFA15	TDFA14	TDFA13	TDFA12	TDFA11	TDFA10	TDFA9	TDFA
H'FFFFFD53		TDFA7	TDFA6	TDFA5	TDFA4	TDFA3	TDFA2	TDFA1	TDFA
H'FFFFFD54 to H'FFFFFD5F	—	—	—	—	—	—	—	—	—

H'FFFFFD66		—	—	—	—	—	—	—	—
H'FFFFFD67		—	—	—	—	—	LCHNG	MPD	ICD
H'FFFFFD68	ECSIPR	—	—	—	—	—	—	—	—
H'FFFFFD69		—	—	—	—	—	—	—	—
H'FFFFFD6A		—	—	—	—	—	—	—	—
H'FFFFFD6B		—	—	—	—	—	LCHNGIP	MPDIP	ICDIP
H'FFFFFD6C	PIR	—	—	—	—	—	—	—	—
H'FFFFFD6D		—	—	—	—	—	—	—	—
H'FFFFFD6E		—	—	—	—	—	—	—	—
H'FFFFFD6F		—	—	—	—	MDI	MDO	MMD	MDC
H'FFFFFD70	MAHR	MA47	MA46	MA45	MA44	MA43	MA42	MA41	MA40
H'FFFFFD71		MA39	MA38	MA37	MA36	MA35	MA34	MA33	MA32
H'FFFFFD72		MA31	MA30	MA29	MA28	MA27	MA26	MA25	MA24
H'FFFFFD73		MA23	MA22	MA21	MA20	MA19	MA18	MA17	MA16
H'FFFFFD74	MALR	—	—	—	—	—	—	—	—
H'FFFFFD75		—	—	—	—	—	—	—	—
H'FFFFFD76		MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8
H'FFFFFD77		MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
H'FFFFFD78	RFLR	—	—	—	—	—	—	—	—
H'FFFFFD79		—	—	—	—	—	—	—	—
H'FFFFFD7A		—	—	—	—	RFL11	RFL10	RFL9	RFL8
H'FFFFFD7B		RFL7	RFL6	RFL5	RFL4	RFL3	RFL2	RFL1	RFL0
H'FFFFFD7C	PSR	—	—	—	—	—	—	—	—
H'FFFFFD7D		—	—	—	—	—	—	—	—
H'FFFFFD7E		—	—	—	—	—	—	—	—
H'FFFFFD7F		—	—	—	—	—	—	—	LMON

H'FFFFFD86		COLDC15	COLDC14	COLDC13	COLDC12	COLDC11	COLDC10	COLDC9	COLD
H'FFFFFD87		COLDC7	COLDC6	COLDC5	COLDC4	COLDC3	COLDC2	COLDC1	COLD
H'FFFFFD88	LCCR	—	—	—	—	—	—	—	—
H'FFFFFD89		—	—	—	—	—	—	—	—
H'FFFFFD8A		LCC15	LCC14	LCC13	LCC12	LCC11	LCC10	LCC9	LCC8
H'FFFFFD8B		LCC7	LCC6	LCC5	LCC4	LCC3	LCC2	LCC1	LCC0
H'FFFFFD8C	CNDCR	—	—	—	—	—	—	—	—
H'FFFFFD8D		—	—	—	—	—	—	—	—
H'FFFFFD8E		CNDC15	CNDC14	CNDC13	CNDC12	CNDC11	CNDC10	CNDC9	CNDC
H'FFFFFD8F		CNDC7	CNDC6	CNDC5	CNDC4	CNDC3	CNDC2	CNDC1	CNDC
H'FFFFFD90	IFLCR	—	—	—	—	—	—	—	—
H'FFFFFD91		—	—	—	—	—	—	—	—
H'FFFFFD92		IFLC15	IFLC14	IFLC13	IFLC12	IFLC11	IFLC10	IFLC9	IFLC8
H'FFFFFD93		IFLC7	IFLC6	IFLC5	IFLC4	IFLC3	IFLC2	IFLC1	IFLC0
H'FFFFFD94	CEFCR	—	—	—	—	—	—	—	—
H'FFFFFD95		—	—	—	—	—	—	—	—
H'FFFFFD96		CEFC15	CEFC14	CEFC13	CEFC12	CEFC11	CEFC10	CEFC9	CEFC
H'FFFFFD97		CEFC7	CEFC6	CEFC5	CEFC4	CEFC3	CEFC2	CEFC1	CEFC
H'FFFFFD98	FRECR	—	—	—	—	—	—	—	—
H'FFFFFD99		—	—	—	—	—	—	—	—
H'FFFFFD9A		FREC15	FREC14	FREC13	FREC12	FREC11	FREC10	FREC9	FREC
H'FFFFFD9B		FREC7	FREC6	FREC5	FREC4	FREC3	FREC2	FREC1	FREC
H'FFFFFD9C	TSFRCR	—	—	—	—	—	—	—	—
H'FFFFFD9D		—	—	—	—	—	—	—	—
H'FFFFFD9E		TSFC15	TSFC14	TSFC13	TSFC12	TSFC11	TSFC10	TSFC9	TSFC
H'FFFFFD9F		TSFC7	TSFC6	TSFC5	TSFC4	TSFC3	TSFC2	TSFC1	TSFC

H'FFFFFDA6		RFC15	RFC14	RFC13	RFC12	RFC11	RFC10	RFC9	RFC8
H'FFFFFDA7		RFC7	RFC6	RFC5	RFC4	RFC3	RFC2	RFC1	RFC0
H'FFFFFDA8	MAFCR	—	—	—	—	—	—	—	—
H'FFFFFDA9		—	—	—	—	—	—	—	—
H'FFFFFDAA		MAFC15	MAFC14	MAFC13	MAFC12	MAFC11	MAFC10	MAFC9	MAFC8
H'FFFFFDAB		MAFC7	MAFC6	MAFC5	MAFC4	MAFC3	MAFC2	MAFC1	MAFC0
H'FFFFFDAC to H'FFFFFE0F	—	—	—	—	—	—	—	—	—
H'FFFFFE10	TIER	ICIE	—	—	—	OCIAE	OCIBE	OVIE	—
H'FFFFFE11	FTCSR	ICF	—	—	—	OCFA	OCFB	OVF	CCLRA
H'FFFFFE12	FRCH								
H'FFFFFE13	FRCL								
H'FFFFFE14	OCRAH								
	OCRBH								
H'FFFFFE15	OCRAL								
	OCRBL								
H'FFFFFE16	TCR	IEDG	—	—	—	—	—	CKS1	CKS0
H'FFFFFE17	TOCR	—	—	—	OCRS	—	—	OLVLA	OLVLB
H'FFFFFE18	FICRH								
H'FFFFFE19	FICRL								
H'FFFFFE1A to H'FFFFFE3F	—	—	—	—	—	—	—	—	—
H'FFFFFE40	IPRD	TPU0IP3	TPU0IP2	TPU0IP1	TPU0IP0	TPU1IP3	TPU1IP2	TPU1IP1	TPU1IP
H'FFFFFE41		TPU2IP3	TPU2IP2	TPU2IP1	TPU2IP0	SCF1IP3	SCF1IP2	SCF1IP1	SCF1IP
H'FFFFFE42	VCRE	—	TG0AV6	TG0AV5	TG0AV4	TG0AV3	TG0AV2	TG0AV1	TG0AV
H'FFFFFE43		—	TG0BV6	TG0BV5	TG0BV4	TG0BV3	TG0BV2	TG0BV1	TG0BV
H'FFFFFE44	VCRF	—	TG0CV6	TG0CV5	TG0CV4	TG0CV3	TG0CV2	TG0CV1	TG0CV
H'FFFFFE45		—	TG0DV6	TG0DV5	TG0DV4	TG0DV3	TG0DV2	TG0DV1	TG0DV

H'FFFFFE4D		—	TG2BV6	TG2BV5	TG2BV4	TG2BV3	TG2BV2	TG2BV1	TG2B
H'FFFFFE4E	VCRK	—	TC2VV6	TC2VV5	TC2VV4	TC2VV3	TC2VV2	TC2VV1	TC2V
H'FFFFFE4F		—	TC2UV6	TC2UV 5	TC2UV4	TC2UV3	TC2UV2	TC2UV1	TC2U
H'FFFFFE50	VCRL	—	SER1V6	SER1V5	SER1V4	SER1V3	SER1V2	SER1V1	SER1
H'FFFFFE51		—	SRX1V6	SRX1V5	SRX1V4	SRX1V3	SRX1V2	SRX1V1	SER1
H'FFFFFE52	VCRM	—	SBR1V6	SBR1V5	SBR1V4	SBR1V3	SBR1V2	SBR1V1	SBR1
H'FFFFFE53		—	STX1V6	STX1V5	STX1V4	STX1V3	STX1V2	STX1V1	STX1
H'FFFFFE54	VCRN	—	SER2V6	SER2V5	SER2V4	SER2V3	SER2V2	SER2V1	SER2
H'FFFFFE55		—	SRX2V6	SRX2V5	SRX2V4	SRX2V3	SRX2V2	SRX2V1	SRX2
H'FFFFFE56	VCRO	—	SBR2V6	SBR2V5	SBR2V4	SBR2V3	SBR2V2	SBR2V1	SBR2
H'FFFFFE57		—	STX2V6	STX2V5	STX2V4	STX2V3	STX2V2	STX2V1	STX2
H'FFFFFE58 to H'FFFFFE5F	—	—	—	—	—	—	—	—	—
H'FFFFFE60	IPRB	E-DMACIP3	E-DMACIP2	E-DMACIP1	E-DMACIP0	FRTIP3	FRTIP2	FRTIP1	FRTIP
H'FFFFFE61		—	—	—	—	—	—	—	—
H'FFFFFE62	VCRA	—	EINV6	EINV5	EINV4	EINV3	EINV2	EINV1	EINV0
H'FFFFFE63		—	—	—	—	—	—	—	—
H'FFFFFE64	VCRB	—	—	—	—	—	—	—	—
H'FFFFFE65		—	—	—	—	—	—	—	—
H'FFFFFE66	VCRC	—	FICV6	FICV5	FICV4	FICV3	FICV2	FICV1	FICV0
H'FFFFFE67		—	FOCV6	FOCV5	FOCV4	FOCV3	FOCV2	FOCV1	FOCV
H'FFFFFE68	VCRD	—	FOVV6	FOVV5	FOVV4	FOVV3	FOVV2	FOVV1	FOVV
H'FFFFFE69		—	—	—	—	—	—	—	—
H'FFFFFE6A to H'FFFFFE70	—	—	—	—	—	—	—	—	—
H'FFFFFE71	DRCR0	—	—	—	RS4	RS3	RS2	RS1	RS0
H'FFFFFE72	DRCR1	—	—	—	RS4	RS3	RS2	RS1	RS0

H'FFFFFE84 to H'FFFFFE8F	—	—	—	—	—	—	—	—	—
H'FFFFFE90	FMR	PLL2ST	PLL1ST	CKIOST	—	FR3	FR2	FR1	FR0
H'FFFFFE91	SBYCR1	SBY	HIZ	MSTP5	MSTP4	MSTP3	—	MSTP1	—
H'FFFFFE92	CCR	W1	W0	WB	CP	TW	OD	ID	CE
H'FFFFFE93	SBYCR2	—	—	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6
H'FFFFFE94 to H'FFFFFEBF	—	—	—	—	—	—	—	—	—
H'FFFFFEC0	IPRE	SCF2IP3	SCF2IP2	SCF2IP1	SCF2IP0	SIO0IP3	SIO0IP2	SIO0IP1	SIO0IP
H'FFFFFEC1		SIO1IP3	SIO1P2	SIO2P1	SIO1IP0	SIO2IP3	SIO2IP2	SIO2IP1	SIO2IP
H'FFFFFEC2	VCRP	—	RER0V6	RER0V5	RER0V4	RER0V3	RER0V2	RER0V1	RER0V
H'FFFFFEC3		—	TER0V6	TER0V5	TER0V4	TER0V3	TER0V2	TER0V1	TER0V
H'FFFFFEC4	VCRQ	—	RDF0V6	RDF0V5	RDF0V4	RDF0V3	RDF0V2	RDF0V1	RDF0V
H'FFFFFEC5		—	TDE0V6	TDE0V5	TDE0V4	TDE0V3	TDE0V2	TDE0V1	TDE0V
H'FFFFFEC6	VCRR	—	RER1V6	RER1V5	RER1V4	RER1V3	RER1V2	RER1V1	RER1V
H'FFFFFEC7		—	TER1V6	TER1V5	TER1V4	TER1V3	TER1V2	TER1V1	TER1V
H'FFFFFEC8	VCRS	—	RDF1V6	RDF1V5	RDF1V4	RDF1V3	RDF1V2	RDF1V1	RDF1V
H'FFFFFEC9		—	TDE1V6	TDE1V5	TDE1V4	TDE1V3	TDE1V2	TDE1V1	TDE1V
H'FFFFFECA	VCRT	—	RER2V6	RER2V5	RER2V4	RER2V3	RER2V2	RER2V1	RER2V
H'FFFFFECB		—	TER2V6	TER2V5	TER2V4	TER2V3	TER2V2	TER2V1	TER2V
H'FFFFFECC	VCRU	—	RDF2V6	RDF2V5	RDF2V4	RDF2V3	RDF2V2	RDF2V1	RDF2V
H'FFFFFECD		—	TDE2V6	TDE2V5	TDE2V4	TDE2V3	TDE2V2	TDE2V1	TDE2V
H'FFFFFECE to H'FFFFFEDF	—	—	—	—	—	—	—	—	—
H'FFFFFEE0	ICR	NMIL	—	—	—	—	—	—	NMIE
H'FFFFFEE1		—	—	—	—	—	—	EXIMD	VECMD
H'FFFFFEE2	IPRA	—	—	—	—	DMACIP3	DMACIP2	DMACIP1	DMACI
H'FFFFFEE3		WDTIP3	WDTIP2	WDTIP1	WDTIP0	—	—	—	—

to H'FFFFFEFF									
H'FFFFFF00	BARAH	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA2
H'FFFFFF01		BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA1
H'FFFFFF02	BARAL	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8
H'FFFFFF03		BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0
H'FFFFFF04	BAMRAH	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA
H'FFFFFF05		BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA
H'FFFFFF06	BAMRAL	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA
H'FFFFFF07		BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA
H'FFFFFF08	BBRA	—	—	—	—	—	—	—	—
H'FFFFFF09		CPA1	CPA0	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0
H'FFFFFF0A to H'FFFFFF0F	—	—	—	—	—	—	—	—	—
H'FFFFFF10	BRFR	SVF	PID2	PID1	PID0	—	—	—	—
H'FFFFFF11		DVF	—	—	—	—	—	—	—
H'FFFFFF12 to H'FFFFFF13	—	—	—	—	—	—	—	—	—
H'FFFFFF14	BRSRH	BSA31	BSA30	BSA29	BSA28	BSA27	BSA26	BSA25	BSA2
H'FFFFFF15		BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA1
H'FFFFFF16	BRSRL	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8
H'FFFFFF17		BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0
H'FFFFFF18	BRDRH	BDA31	BDA30	BDA29	DA28	BDA27	BDA26	BDA25	BDA2
H'FFFFFF19		BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA1
H'FFFFFF1A	BRDRL	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8
H'FFFFFF1B		BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0

H'FFFFFF23		BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0
H'FFFFFF24	BAMRBH	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB2
H'FFFFFF25		BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB1
H'FFFFFF26	BAMRBL	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8
H'FFFFFF27		BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0
H'FFFFFF28	BBRB	—	—	—	—	—	—	—	—
H'FFFFFF29		CPB1	CPB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0
H'FFFFFF2A to H'FFFFFF2F	—	—	—	—	—	—	—	—	—
H'FFFFFF30	BRCRH	CMFCA	CMFPA	—	—	PCTE	PCBA	—	—
H'FFFFFF31		CMFCB	CMFPB	—	SEQ1	SEQ0	PCBB	—	—
H'FFFFFF32	BRCRL	CMFCC	CMFPC	ETBEC	—	DBEC	PCBC	—	—
H'FFFFFF33		CMFCD	CMFPD	ETBED	—	DBED	PCBD	—	—
H'FFFFFF34 to H'FFFFFF3F	—	—	—	—	—	—	—	—	—
H'FFFFFF40	BARCH	BAC31	BAC30	BAC29	BAC28	BAC27	BAC26	BAC25	BAC24
H'FFFFFF41		BAC23	BAC22	BAC21	BAC20	BAC19	BAC18	BAC17	BAC16
H'FFFFFF42	BARCL	BAC15	BAC14	BAC13	BAC12	BAC11	BAC10	BAC9	BAC8
H'FFFFFF43		BAC7	BAC6	BAC5	BAC4	BAC3	BAC2	BAC1	BAC0
H'FFFFFF44	BAMRCH	BAMC31	BAMC30	BAMC29	BAMC28	BAMC27	BAMC26	BAMC25	BAMC2
H'FFFFFF45		BAMC23	BAMC22	BAMC21	BAMC20	BAMC19	BAMC18	BAMC17	BAMC1
H'FFFFFF46	BAMRCL	BAMC15	BAMC14	BAMC13	BAMC12	BAMC11	BAMC10	BAMC9	BAMC8
H'FFFFFF47		BAMC7	BAMC6	BAMC5	BAMC4	BAMC3	BAMC2	BAMC1	BAMC0
H'FFFFFF48	BBRC	—	—	—	—	—	—	XYEC	XYSC
H'FFFFFF49		CPC1	CPC0	IDC1	IDC0	RWC1	RWC0	SZC1	SZC0
H'FFFFFF4A to H'FFFFFF4F	—	—	—	—	—	—	—	—	—

H'FFFFFF56	BDMRCL	BDMC15	BDMC14	BDMC13	BDMC12	BDMC11	BDMC10	BDMC9	BDMC
H'FFFFFF57		BDMC7	BDMC6	BDMC5	BDMC4	BDMC3	BDMC2	BDMC1	BDMC
H'FFFFFF58	BETRC	—	—	—	—	ETRC11	ETRC10	ETRC9	ETRC
H'FFFFFF59		ETRC7	ETRC6	ETRC5	ETRC4	ETRC3	ETRC2	ETRC1	ETRC
H'FFFFFF5A to H'FFFFFF5F	—	—	—	—	—	—	—	—	—
H'FFFFFF60	BARDH	BAD31	BAD30	BAD29	BAD28	BAD27	BAD26	BAD25	BAD2
H'FFFFFF61		BAD23	BAD22	BAD21	BAD20	BAD19	BAD18	BAD17	BAD1
H'FFFFFF62	BARDL	BAD15	BAD14	BAD13	BAD12	BAD11	BAD10	BAD9	BAD8
H'FFFFFF63		BAD7	BAD6	BAD5	BAD4	BAD3	BAD2	BAD1	BAD0
H'FFFFFF64	BAMRDH	BAMD31	BAMD30	BAMD29	BAMD28	BAMD27	BAMD26	BAMD25	BAMD
H'FFFFFF65		BAMD23	BAMD22	BAMD21	BAMD20	BAMD19	BAMD18	BAMD17	BAMD
H'FFFFFF66	BAMRDL	BAMD15	BAMD14	BAMD13	BAMD12	BAMD11	BAMD10	BAMD9	BAMD
H'FFFFFF67		BAMD7	BAMD6	BAMD5	BAMD4	BAMD3	BAMD2	BAMD1	BAMD
H'FFFFFF68	BBRD	—	—	—	—	—	—	XYED	XYSD
H'FFFFFF69		CPD1	CPD0	IDD1	IDD0	RWD1	RWD0	SZD1	SZD0
H'FFFFFF6A to H'FFFFFF6F	—	—	—	—	—	—	—	—	—
H'FFFFFF70	BDRDH	BDD31	BDD30	BDD29	BDD28	BDD27	BDD26	BDD25	BDD2
H'FFFFFF71		BDD23	BDD22	BDD21	BDD20	BDD19	BDD18	BDD17	BDD1
H'FFFFFF72	BDRDL	BDD15	BDD14	BDD13	BDD12	BDD11	BDD10	BDD9	BDD8
H'FFFFFF73		BDD7	BDD6	BDD5	BDD4	BDD3	BDD2	BDD1	BDD0
H'FFFFFF74	BDMRDH	BDMD31	BDMD30	BDMD29	BDMD28	BDMD27	BDMD26	BDMD25	BDMD
H'FFFFFF75		BDMD23	BDMD22	BDMD21	BDMD20	BDMD19	BDMD18	BDMD17	BDMD

H'FFFFFF7F									
H'FFFFFF80	SAR0								
H'FFFFFF81									
H'FFFFFF82									
H'FFFFFF83									
H'FFFFFF84	DAR0								
H'FFFFFF85									
H'FFFFFF86									
H'FFFFFF87									
H'FFFFFF88	TCR0	—	—	—	—	—	—	—	—
H'FFFFFF89									
H'FFFFFF8A									
H'FFFFFF8B									
H'FFFFFF8C	CHCR0	—	—	—	—	—	—	—	—
H'FFFFFF8D		—	—	—	—	—	—	—	—
H'FFFFFF8E		DM1	DM0	SM1	SM0	TS1	TS0	AR	AM
H'FFFFFF8F		AL	DS	DL	TB	TA	IE	TE	DE
H'FFFFFF90	SAR1								
H'FFFFFF91									
H'FFFFFF92									
H'FFFFFF93									
H'FFFFFF94	DAR1								
H'FFFFFF95									
H'FFFFFF96									
H'FFFFFF97									
H'FFFFFF98	TCR1	—	—	—	—	—	—	—	—
H'FFFFFF99									
H'FFFFFF9A									
H'FFFFFF9B									

H'FFFFFFA2									
H'FFFFFFA3		VC7	VC6	VC5	VC4	VC3	VC2	VC1	VC0
H'FFFFFFA4 to H'FFFFFFA7	—	—	—	—	—	—	—	—	—
H'FFFFFFA8	VCRDMA1	—	—	—	—	—	—	—	—
H'FFFFFFA9		—	—	—	—	—	—	—	—
H'FFFFFFAA		—	—	—	—	—	—	—	—
H'FFFFFFAB		VC7	VC6	VC5	VC4	VC3	VC2	VC1	VC0
H'FFFFFFAC to H'FFFFFFAF	—	—	—	—	—	—	—	—	—
H'FFFFFFB0	DMAOR	—	—	—	—	—	—	—	—
H'FFFFFFB1		—	—	—	—	—	—	—	—
H'FFFFFFB2		—	—	—	—	—	—	—	—
H'FFFFFFB3		—	—	—	—	PR	AE	NMIF	DME
H'FFFFFFB4 to H'FFFFFFBF	—	—	—	—	—	—	—	—	—
H'FFFFFFC0	WCR2	A4WD1	A4WD0	—	A4WM	A3WM	A2WM	A1WM	A0W
H'FFFFFFC1		—	—	—	—	IW41	IW40	W41	W40
H'FFFFFFC2 to H'FFFFFFC3	—	—	—	—	—	—	—	—	—
H'FFFFFFC4	WCR3	—	—	A4SW2	A4SW1	A4SW0	—	A4HW1	A4HW
H'FFFFFFC5		A3SHW1	A3SHW0	A2SHW1	A2SHW0	A1SHW1	A1SHW0	A0SHW1	A0SH
H'FFFFFFC6 to H'FFFFFFDF	—	—	—	—	—	—	—	—	—
H'FFFFFFE0	BCR1	—	A4LW1	A4LW0	A2ENDIAN	BSTROM	—	AHLW1	AHLW
H'FFFFFFE1		A1LW1	A1LW0	A0LW1	A0LW0	A4ENDIAN	DRAM2	DRAM1	DRAI
H'FFFFFFE2 to H'FFFFFFE3	—	—	—	—	—	—	—	—	—

H'FFFFFFEA to H'FFFFFFEB	—	—	—	—	—	—	—	—	—
H'FFFFFFEC	MCR	TRP0	RCD0	TRWL0	TRAS1	TRAS0	BE	RASD	TRWL1
H'FFFFFFED		AMX2	SZ	AMX1	AMX0	RFSH	RMODE	TRP1	RCD1
H'FFFFFFEE to H'FFFFFFEF	—	—	—	—	—	—	—	—	—
H'FFFFFFF0	RTCSR	—	—	—	—	—	—	—	—
H'FFFFFFF1		CMF	CMIE	CKS2	CKS1	CKS0	RRC2	RRC1	RRC0
H'FFFFFFF2 to H'FFFFFFF3	—	—	—	—	—	—	—	—	—
H'FFFFFFF4	RTCNT	—	—	—	—	—	—	—	—
H'FFFFFFF5									
H'FFFFFFF6 to H'FFFFFFF7	—	—	—	—	—	—	—	—	—
H'FFFFFFF8	RTCOR	—	—	—	—	—	—	—	—
H'FFFFFFF9									
H'FFFFFFFA to H'FFFFFFFB	—	—	—	—	—	—	—	—	—
H'FFFFFFFC	BCR3	—	—	—	—	A4LW2	AHLW2	A1LW2	A0LW2
H'FFFFFFFD		DSWW1	DSWW0	—	—	—	BASEL	EDO	BWE
H'FFFFFFFE to H'FFFFFFFF	—	—	—	—	—	—	—	—	—

Pin Type	Pin Name	Reset	Acquired	Released	(HIZ = 0)	(HIZ = 1)	Mode
Bus control	A24 to A0	O	O	Z	Z	Z	O
	D31 to D0	Z	IO	Z	Z	Z	IO
	$\overline{CS4}$ to $\overline{CS0}$	H	O	Z	H	H	H
	RD/$\overline{WR}$	H	O	Z	H	H	H
	$\overline{RAS}$	H	O	Z	H	H	H
	$\overline{CAS}$/$\overline{OE}$	H	O	Z	H	H	H
	$\overline{WAIT}$	Z	I	Z	Z	Z	I
	$\overline{BS}$	H	O	Z	H	H	H
	$\overline{RD}$	H	O	Z	H	H	H
	$\overline{BGR}$	H	O	O	H	H	O
	$\overline{BRLS}$	Z	I	I	Z	Z	I
	CKE	H	O	H	O	O	O
	DQMUU/$\overline{WE3}$	H	O	Z	H	H	H
	DQMUL/$\overline{WE2}$	H	O	Z	H	H	H
	DQMLU/$\overline{WE1}$	H	O	Z	H	H	H
	DQMLL/$\overline{WE0}$	H	O	Z	H	H	H
	REFOUT	L	O	O	L	Z	O
	$\overline{CAS3}$ to $\overline{CAS0}$	H	O	Z	H	H	H
	$\overline{BH}$	H	O	Z	H	H	H
	$\overline{BUSHiZ}$	Z	I	Z	Z	Z	I
Interrupt	NMI	I	I	I	I	I	I
	$\overline{IRL3}$ to $\overline{IRL0}$	Z	Z	Z	I	I	I
	$\overline{IVECF}$	H	H	H	H	Z	H

	CKPACK	H	H	H	H	H	H
	$\overline{\text{CKPREQ}}$/CKM	I	I	I	I	I	I
	PLLCAP2, PLLCAP1	IO	IO	IO	IO	IO	IO
DMAC	DREQ1, DREQ0	Z	Z	Z	Z	Z	I
	DACK1, DACK0	H	H	H	K	Z	O
System control	$\overline{\text{RES}}$	I	I	I	I	I	I
	MD4 to MD0	I	I	I	I	I	I
Port, Internal peripheral module	PB15/SCK1	Z	IO/Z	IO/Z	K	Z	IO
	PB14/RXD1	Z	IO/Z	IO/Z	K	Z	IO/I
	PB13/TXD1	Z	IO/Z	IO/Z	K	Z	IO/O
	PB12/SRCK2/$\overline{\text{RTS}}$/STATS1	Z	IO/Z/Z/O	IO/Z/Z/O	K/K/K/O	Z	IO/I/O/O
	PB11/SRS2/$\overline{\text{CTS}}$/STATS0	Z	IO/Z/Z/O	IO/Z/Z/O	K/K/K/O	Z	IO/I/I/O
	PB10/SRXD2/TIOCA1	Z	IO/Z/Z	IO/Z/Z	K/K/K	Z	IO/I/IO
	PB9/STCK2/TIOCB1, TCLKC	Z	IO/Z/Z	IO/Z/Z	K/K/K	Z	IO/I/IO
	PB8/STS2/TIOCA2	Z	IO/Z/Z	IO/Z/Z	K/K/K	Z	IO/IO/IO
	PB7/STXD2/TIOCB2, TCLKD	Z	IO/Z/Z	IO/Z/Z	K/K/K	Z	IO/O/IO
	PB6/SRCK1/SCK2	Z	IO/Z/Z	IO/Z/Z	K/K/K	Z	IO/I/IO
	PB5/SRS1/RXD2	Z	IO/Z/Z	IO/Z/Z	K/K/K	Z	IO/I/I
	PB4/SRXD1/TXD2	Z	IO/Z/Z	IO/Z/Z	K/K/K	Z	IO/I/O
	PB3/STCK1/TIOCA0	Z	IO/Z/Z	IO/Z/Z	K/K/K	Z	IO/I/IO
	PB2/STS1/TIOCB0	Z	IO/Z/Z	IO/Z/Z	K/K/K	Z	IO/IO/IO
	PB1/STXD1/TIOCC0, TCLKA	Z	IO/Z/Z	IO/Z/Z	K/K/K	Z	IO/O/IO

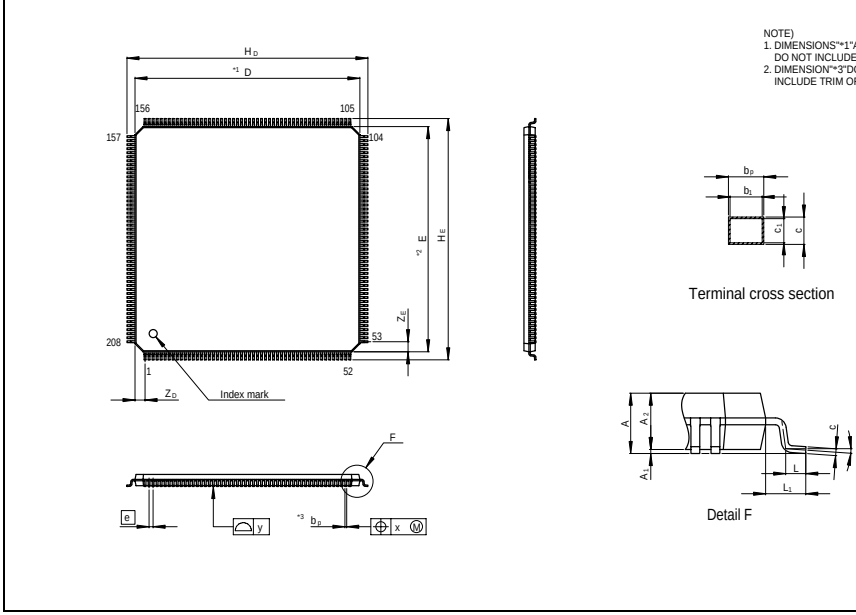
	PA12/SRS0	Z	IO/Z	IO/Z	K/K	Z	IO/I
	PA11/SRXD0	Z	IO/Z	IO/Z	K/K	Z	IO/I
	PA10/STCK0	Z	IO/Z	IO/Z	K/K	Z	IO/I
	PA9/STS0	Z	IO/Z	IO/Z	K/K	Z	IO/IO
	PA8/STXD0	Z	IO/Z	IO/Z	K/K	Z	IO/O
	$\overline{\text{WDTOVF}}$/PA7	H	H/IO	H/IO	O/K	O/Z	O/IO
	PA6/FTCI	Z	IO/Z	IO/Z	K	Z	IO/I
	PA5/FTI	Z	IO/Z	IO/Z	K	Z	IO/I
	PA4/FTOA	Z	IO/L	IO/L	K	Z	IO/O
	CKPO/FTOB	H	H/L	H/L	K	Z	O/O
	PA2/LNKSTA	Z	IO/I	IO/I	K	Z	IO/I
	PA1/EXOUT	Z	IO/O	IO/O	K	Z	IO/O
	PA0	Z	IO	IO	K	Z	IO
H-UDI	$\overline{\text{TRST}}$	I	I	I	I	I	I
	TCK	I	I	I	I	I	I
	TMS	I	I	I	I	I	I
	TDI	I	I	I	I	I	I
	TDO	O	O	O	O	O	O
	ASEMODE	I	I	I	I	I	I

ETXD3 to ETXD0	O	O	O	O	O	O
CRS	I	I	I	I	I	I
COL	I	I	I	I	I	I
MDC	O	O	O	O	O	O
MDIO	IO	IO	IO	IO	IO	IO
RX-CLK	I	I	I	I	I	I
RX-DV	I	I	I	I	I	I
RX-ER	I	I	I	I	I	I
ERXD3 to ERXD0	I	I	I	I	I	I

I: Input
O: Output
H: High-level output
L: Low-level output
Z: High-impedance state
K: Input pins are in the high-impedance state; output pins maintain their previous state.

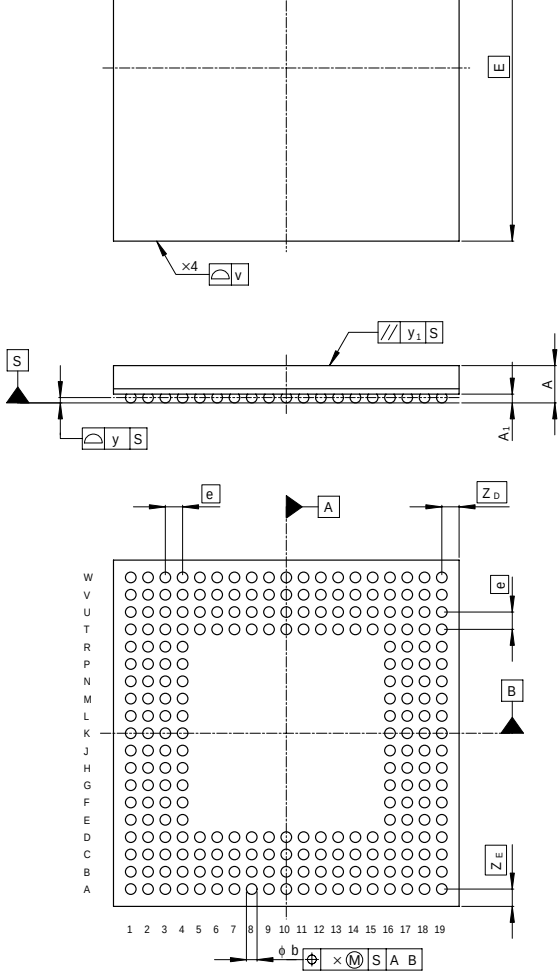
Notes: In sleep mode, if the DMAC is operating the address/data bus and bus control s
according to the operation of the DMAC. (The same applies when refreshing is p
* Depends on the clock mode ($\overline{\text{CKPREQN}}$, MD2 to MD0 setting).

HD6417615ARBP	BP
HD6417615ARBPV	BP



Reference Symbol	
	M
D	—
E	—
A 2	—
H D	29
H E	29
A	—
A 1	0.
b p	0.
b 1	—
c	0.
c 1	—
	(
e	—
x	—
y	—
Z D	—
Z E	—
L	0
L 1	—

Figure D.1 Package Dimensions (FP-208C, FP-208CV)



Reference Symbol	Dime	
	Min	
D	—	
E	—	
v	—	
w	—	
A	—	
A_1	0.28	
e	—	
b	0.35	
x	—	
y	—	
y_1	—	
S_D	—	
S_E	—	
Z_D	—	
Z_E	—	

Figure D.2 Package Dimensions (BP-240A, BP-240AV)

Renesas 32-Bit RISC Microcomputer

Hardware Manual

SH7615 Group

Publication Date: 1st Edition, September 2000
Rev.2.00, March 17, 2005
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Technical Documentation & Information Department
Renesas Kodaira Semiconductor Co., Ltd.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-000

RENESAS SALES OFFICES

http://www.r

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 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 (Shanghai) Co., Ltd.
Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

SH7615 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0157-0200O