To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

32

SH7705 Group

Hardware Manual

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

Renesas Electronics
www.renesas.com

Rev.2.00 2003.09

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

SH7705 Group

Hardware Manual

RENESAS

REJ09B00

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection o
   Renesas Technology Corp. product best suited to the customer's application; they do
   convey any license under any intellectual property rights, or any other rights, belong
   Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringeme
   third-party's rights, originating in the use of any product data, diagrams, charts, prog
   algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, char
   programs and algorithms represents information on products at the time of publicati
   materials, and are subject to change by Renesas Technology Corp. without notice du
   product improvements or other reasons. It is therefore recommended that customer
   Renesas Technology Corp. or an authorized Renesas Technology Corp. product dist
   the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or o
   rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by
   means, including the Renesas Technology Corp. Semiconductor home page
   (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including pro
   diagrams, charts, programs, and algorithms, please be sure to evaluate all informatio
   system before making a final decision on the applicability of the information and pr
   Renesas Technology Corp. assumes no responsibility for any damage, liability or ot
   resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for us
   device or system that is used under circumstances in which human life is potentially
   Please contact Renesas Technology Corp. or an authorized Renesas Technology Co
   distributor when considering the use of a product contained herein for any specific p
   such as apparatus or systems for transportation, vehicular, medical, aerospace, nucle
   undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or r
   whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictio
   must be exported under a license from the Japanese government and cannot be impo
   country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Jap
   the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or th
   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. E
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. Index

to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the SH7705 MCU to the above users.
Refer to the SH-3/SH-3E/SH3-DSP Programming Manual for a detailed of the instruction set.

Notes on reading this manual:

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

Product Classifications and Abbreviations

Basic Classification	Product Code
SH7705	HD6417705

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized on the CPU, system control functions, peripheral functions and electrical characteri
- In order to understand the details of the CPU's functions
  Read the SH-3/SH-3E/SH3-DSP Programming Manual.

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

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

SH7705 manuals:

Manual Title	ADE No
SH7705 Hardware Manual	This ma
SH-3/SH-3E/SH3-DSP Programming Manual	ADE-60

Users manuals for development tools:

Manual Title	ADE No
SH Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	ADE-70
SH Series Simulator/Debugger (for Windows) User's Manual	ADE-70
SH Series Simulator/Debugger (for UNIX) User's Manual	ADE-70
Embedded Workshop User's Manual	ADE-70
SH Series Embedded Workshop, Debugging Interface Tutorial	ADE-70

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
UDI	User Debugging Interface
INTC	Interrupt Controller
IrDA	Infrared Data Association
JTAG	Joint Test Action Group
LQFP	Low Profile QFP
LRU	Least Recently Used
LSB	Least Significant Bit
MMU	Memory Management Unit
MPX	Multiplex
MSB	Most Significant Bit
PC	Program Counter
PFC	Pin Function Controller
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
RTC	Realtime Clock
SCIF	Serial Communication Interface with FIFO

USB	Universal Serial Bus
WDT	Watchdog Timer

FP-208C	TBP-208A	Pin Name	I/O	Description
139	G15	TDI*7/PTG0	I / I/O	Test data inpu input/output p
140	G14	TCK*7/PTG1	I / I/O	Test clock (UD port G
141	F17	TMS*7/PTG2	I / I/O	Test mode se input/output p
142	F16	$\overline{\text{TRST}}$*1 *7/PTG3	I / I/O	Test reset (UD port G
143	F15	TDO/PTF5	O / I/O	Test data outp input/output p
144	F14	$\overline{\text{ASEBRKAK}}$/ PTF6	O / I/O	ASE break ac (UDI) / input/o
145	E17	$\overline{\text{ASEMD0}}$*2 *7/ PTF7	I / I/O	ASE mode (U port F
195	C6	$\overline{\text{RESETP}}$*6	I	Power-on rese

Notes: 6. Pull-up MOS connected.

7. The pull-up MOS turns on if the pin function (PFC) is used to select other functions (UDI).

4.4.1 Address Array	105	Description amended
Address-Array Write (Associative Operation)		····· This operation is used to invalidate the addr specification for a cache.
4.4.3 Usage Examples	107	Description largely revised
Invalidating a Specific Entry		
Invalidating an Address Specification	108	Description added
5.2.5 Exception Source Acceptance Timing and Priority	117	Note *3 amended
		Note: 3. If an interrupt is accepted, the exceptio register (EXPEVT) is not changed. ·····
Table 5.1 Exception Event Vectors		

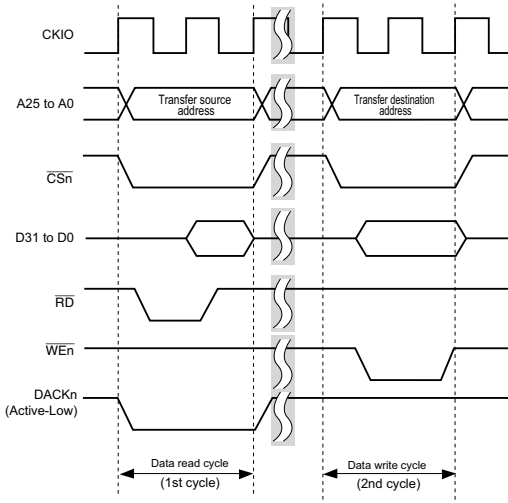
Legend:
DMAC : Direct memory access controller
SCIF : Serial communication interface (with FIFO)
ADC : A/D converter
USB : USB interface
TMU : Timer pulse unit
TPU : 16-bit timer pulse unit

6.4.6 Interrupt Exception Handling and Priority Table 6.4 Interrupt Exception Handling Sources and Priority (IRQ Mode)	140	IPR (bit numbers) amended for interrupt source T IPRA (7 to 4)
7.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B)	160	Bits 14 to 12 description added Note: SDRAM can be specified only in area 2 ar SDRAM is connected to only one area, SDRAM specified for area 3. In this case area 2 should be as normal space.
	161	Note 5 added Note: 5. The SDRAM bank active mode can only for the CS3 space. (Refer to the explanation of th bit in the SDRAM control register.)
7.4.5 Refresh Timer Control/Status Register (RTCSR)	177	Bits 31 to 18 description amended

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	Reserved

7.4.6 Refresh Timer Counter (RTCNT)	179	Bits 31 to 18 description amended

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	Reserved

8.3.4 DMA Channel Control Registers (CHCR)	244	Bits 15, 14 description amended 00: Fixed destination address (setting prohibited in 16-byte transfer)
	245	Bits 13, 12 description amended 00: Fixed source address (setting prohibited in 16-byte transfer)
8.4.3 Channel Priority Round-Robin Mode	258	····· The priority of round-robin mode is CH0 > C CH3 immediately after a reset. When the round-robin mode is specified, cycle-s and burst mode should not be mixed among the for multiple channels.
8.4.4 DMA Transfer Types Address Modes Figure 8.6 Example of DMA Transfer Timing in Dual Mode (Source: Ordinary Memory, Destination: Ordinary Memory)	262	Figure amended 
Bus Mode and channel Priority Order	266	Description largely revised
8.5 Precautions	270	Newly added

		or 1/4,096 is selected using bits CKS2 to CKS0, watchdog timer counter overflow occurs, resultin manual reset, the LSI will generate two manual r succession. This will not affect its operation but w change in the state of the STATUS pin.
11.6.1 Transition to Module Standby Function	301	Description amended This function can be used to reduce the power co in the normal mode and sleep mode.
16.5 SCIF Interrupt Sources and DMAC Table 16.4 SCIF Interrupt Sources	427	Table amended

Interrupt Source	Description	DMA
ERI	Interrupt initiated by receive error flag (ER) or break flag (BRK)	Not p
RXI	Interrupt initiated by receive FIFO data full flag (RDF) or receive data ready (DR)	Poss
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE) or transmit data stop flag (TSF)	Poss

18.1 Features	437	Description amended • The UDC (USB device controller) conforming to and transceiver process USB protocol automati
19.2.7 Port F Control Register (PFCR)	489	Note *2 added to Bits 15 and 14 Note 2. Pull-up MOS on.
19.2.9 Port G Control Register (PGCR)	491	Note *2 added to Bits 7 to 0 Note 2. Pull-up MOS on.

AND.B #imm,@(R0,GBR)	3
OR.B #imm,@(R0,GBR)	3
TAS.B @Rn	3
TST.B #imm,@(R0,GBR)	3
XOR.B #imm,@(R0,GBR)	3
LDC Rm,SR	4
LDC Rm,GBR	4
LDC Rm,VBR	4
LDC Rm,SSR	4
LDC Rm,SPC	4
LDC Rm,R0_BANK	4
LDC Rm,R1_BANK	4
LDC Rm,R2_BANK	4
LDC Rm,R3_BANK	4
LDC Rm,R4_BANK	4
LDC Rm,R5_BANK	4
LDC Rm,R6_BANK	4
LDC Rm,R7_BANK	4

LDC.L @Rm+,R2_BANK	
LDC.L @Rm+,R3_BANK	
LDC.L @Rm+,R4_BANK	
LDC.L @Rm+,R5_BANK	
LDC.L @Rm+,R6_BANK	
LDC.L @Rm+,R7_BANK	
LDC.L @Rn+,MOD	
LDC.L @Rn+,RS	
LDC.L @Rn+,RE	
LDC Rn,MOD	
LDC Rn,RS	
LDC Rn,RE	
BSR label	
BSRF Rm	
JSR @Rm	

23.2 Input/Output Pins	569	Note * added Note: * The pull-up MOS turns on if the pin func controller (PFC) is used to select other functions
23.3.3 Boundary Scan Register (SDBSR)	570	Description amended SDBSR is a 385-bit shift register, located on the controlling the input/output pins of this LSI.
23.5.2 Points for Attention	582	Item 7 added under “23.5.2 Points for Attention” 7. The CKIO cock should operate during bound The MD[2:0] pin should be set to the clock mod during normal operation, and EXTAL and CKIO set within the frequency range specified in the C Generator (CPG) section. As during normal operation, the boundary scan be performed after allowing sufficient settling tim crystal oscillator, PLL1, and PLL2.
24.1 Register Addresses (by functional module, in order of the corresponding section numbers)	592	Access size of EP1 data register and EP2 data amended to 8/32

Timing

Table 25.6 Control Signal Timing

(Conditions: $V_{CC}Q$ = V_{CC}-RTC = V_{CC}-USB = 3.0 t
V_{CC} = V_{CC}-PLL1 = V_{CC}-PLL2 = 1.4 to 1.6 V, AV_{CC}
3.6 V, $V_{SS}Q$ = V_{SS} = V_{SS}-RTC = V_{SS}-USB = V_{SS}-
PLL2 = AV_{SS} = 0 V, T_a = –20 to 75°C, Clock mod
0/1/2/4/5/6/7)

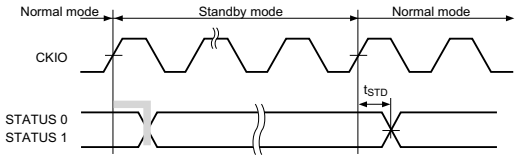
Note *1 amended

Note: 1. $\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI, and IRQ5 to I
asynchronous. ·····

Figure 25.15 Pin Drive Timing at Standby

638

Figure amended

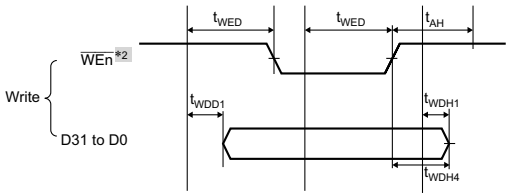


25.3.4 Basic Timing

640

Note *2 added

Figure 25.16 Basic Bus Cycle (No Wait)

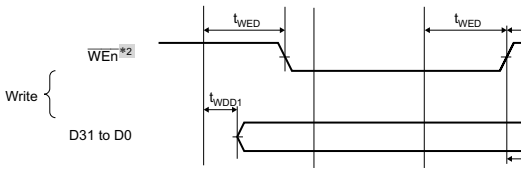


Notes: 1. DACKn is a waveform when active-low is specified.
2. Output timing is the same when reading byte-selection SRAM.

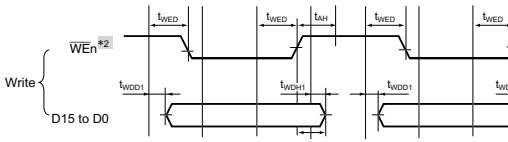
Figure 25.17 Basic Bus Cycle (One Software Wait)

641

Note *2 added



Notes: 1. DACKn is a waveform when active-low is specified.
2. Output timing is the same when reading byte-selection SRAM

Figure 25.19 Basic Bus Cycle (One Software Wait, External Wait Enabled (WM Bit = 0), No Idle Cycle Setting)	643	Note *2 added  Notes: 1. DACKn is a waveform when active-low is specified. 2. Output timing is the same when reading byte-selection SRAM.
25.3.11 SCIF Module Signal Timing Table 25.13 SCIF Module Signal Timing	671	Item amended Transmission data delay time (clock synchroniz $\overline{RTS}$ delay time (clock synchronization)
A. I/O Port States in Each Processing State Table A.1 I/O Port States in Each Processing State	679	Note *11 added

		Reset		Power-Down States		
Category	Pin	Power-on Reset	Manual Reset	Software Standby	Sleep	Bus Mastersh Released
System control	$\overline{RESETP}$	I*11	I*11	I*11	I*11	I*11
	$\overline{RESETM}$	I	I	I	I	I

	NF/PJ[6:0]	H						

Note: 13. The values of PTJ6, PTJ1, and PTJ0 d
power-on reset and after the power-on reset stat
released. They conform to the port J data registe
after being switched to port status by the pin func
controller (PFC).

	During Power-On Reset	After Power-On Re	
		PTD5/NF = 1	PT
PTJ6/NF	1	0	1
PTJ1/NF	1	1	0
PTJ0/NF	1	0	1

High-speed data transfers can be formed by an on-chip direct memory access controll
and an external memory access support function enables direct connection to different
memory. This LSI also includes powerful peripheral functions that are essential to sys
configuration, such as USB (Function) functionality and a serial interface with a large

A powerful built-in power-management function keeps power consumption low, even
high-speed operation. This LSI is ideal for use in electronic devices such as those for a
that require both high speeds and low power consumption.

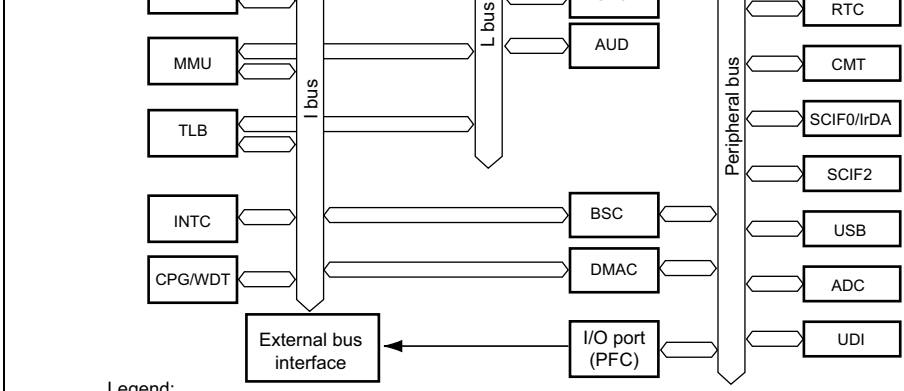
The features of this LSI are listed in table 1.1.

	Five 32-bit control registers Four 32-bit system registers • RISC-type instruction set Instruction length: 16-bit fixed length and improved code efficienc Load/store architecture Delayed branch instructions Instruction set based on C language • Instruction execution time: one instruction/cycle for basic instructi • Logical address space: 4 Gbytes • Five-stage pipeline
Memory management unit (MMU)	• 4 Gbytes of address space, 256 address space identifiers (ASID: • Page unit sharing • Supports multiple page sizes: 1 kbyte or 4 kbytes • 128-entry, 4-way set associative TLB • Supports software selection of replacement method and random-replacement algorithms • Contents of TLB are directly accessible by address mapping
Cache memory	• 32-kbyte cache, mixture of instructions and data • 512 entries, 4-way set associative, 16-byte block length • Write-back, write-through, LRU replacement algorithm • 1-stage write-back buffer
Interrupt controller (INTC)	• Seven external interrupt pins (NMI, IRQ5 to IRQ0) • On-chip peripheral interrupt: Priority level is independently selecte module

	Setting of idle wait cycles (for the same area or different area) Specifying the memory type to be connected to each area enab connection to SRAM, byte selection SRAM, SDRAM, and burst areas support address/data multiplex I/O (MPX). Outputs chip select signal ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A}$/$\overline{B}$, $\overline{CS6A}$/$\overline{B}$ corresponding area (Programs are used to select the $\overline{CS}$ assert timing.) • SDRAM refresh function Supports auto-refresh and self-refresh modes • SDRAM burst access function Different SDRAM can be connected to area 2 or area 3 (size/lat • Usable as either big or little endian machine
Direct memory access controller (DMAC)	• Four channels. Two of these channels support external requests • Burst mode and cycle steal mode • Outputs transfer end signal in channel with DREQ (one channel • Supports intermittent mode (supports 16 or 64 cycles)
Clock pulse generator (CPG)	• Clock mode: Input clock can be selected from external input (EX CKIO) or crystal resonator • Three types of clocks generated CPU clock: max. 133.34 MHz/100 MHz Bus clock: max. 66.67 MHz Peripheral clock: max. 33.34 MHz • Seven types of clock mode (selection of multiplication ratio of PL PLL2, and selection external clock or crystal resonator)
Watchdog timer (WDT)	• One-channel watchdog timer
Power-down mode	• Supports power-down mode Sleep mode Software standby mode and hardware standby mode Module standby mode

unit (TPU)	• Supports PWM function
Realtime clock (RTC)	• Clock and calendar functions (BCD format) • 30-second adjust function • Alarm/periodic/carry interrupt • Automatic leap year adjustment
Serial communication interface (SCIF_0, SCIF_2)	• Clock synchronous/asynchronous mode • 64-byte transmit/receive FIFOs • High-speed UART • UART supports FIFO stop and FIFO trigger • Supports $\overline{\text{RTS}}$/$\overline{\text{CTS}}$ • Supports IrDA 1.0 (only channel 0)
USB function module (USB)	• Conforms to USB 2.0 full-speed specification • Supports modes with an on-chip and external USB transceiver • Supports control transfer (endpoint 0), bulk transfer (endpoint 1, 2 interrupt transfer (endpoint 3) • The USB standard commands are supported, and class and bend commands are handled by firmware • On-chip FIFO buffer for endpoints (128 bytes/endpoint 1, 2) • Module input clock: 48 MHz
I/O port	• Bitwise selection of input/output for input/output port
A/D converter	• 10 bits ± 4 LSB, four channels • Input range: 0 to AVcc (max. 3.6 V)
User break controller (UBC)	• Address, data value, access type, and data size are available for break conditions • Supports the sequential break function • Two break channels

Name	I/O	Modules	Frequency	Product Code
SH7705	3.3 ± 0.3 V	1.5 ± 0.1 V	133 MHz	HD6417705F133
			100 MHz	HD6417705F100
			133 MHz	HD6417705BP133
			100 MHz	HD6417705BP100



Legend:

CACHE:	Cache memory	TMU:	Timer unit
CCN:	Cache memory controller	TPU:	16-bit timer pulse unit
MMU:	Memory management unit	RTC:	Realtime clock
TLB:	Translation look-aside buffer	CMT:	Compare match timer
INTC:	Interrupt controller	SCIF:	Serial communication interface with F
CPG/WDT:	Clock pulse generator/watchdog timer	IrDA:	Infrared data association module
CPU:	Central processing unit	USB:	Universal serial bus
UBC:	User break controller	ADC:	A/D converter
AUD:	Advanced user debugger	UDI:	User debugging interface
BSC:	Bus state controller	PFC:	Pin function controller
DMAC:	Direct memory access controller		

Figure 1.1 Block Diagram of SH7705

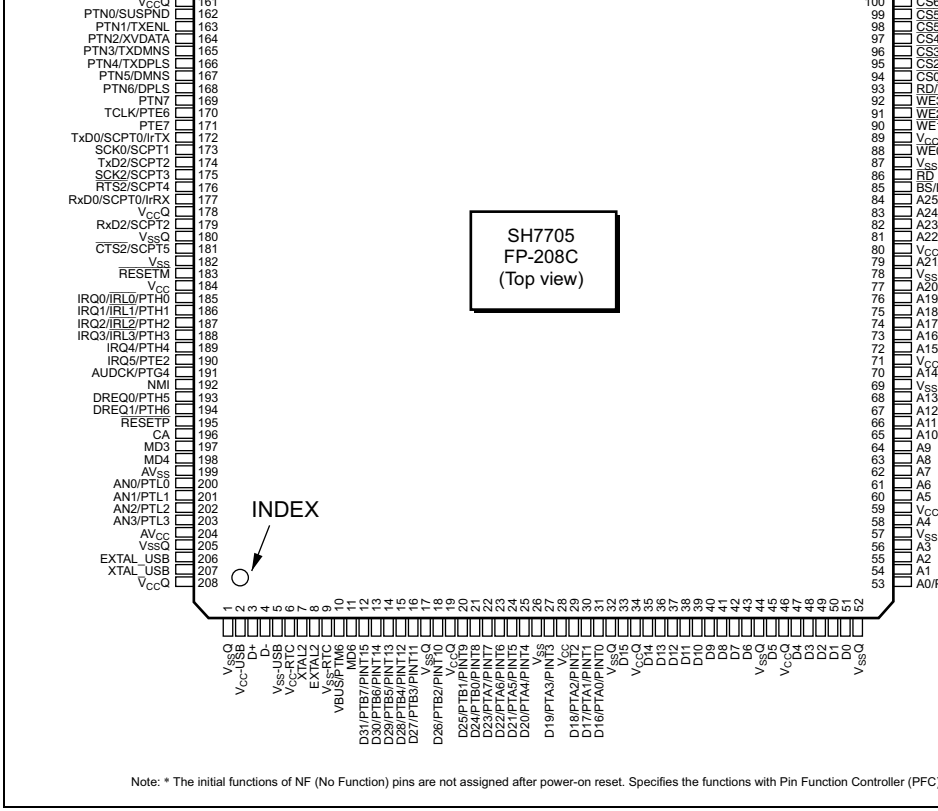
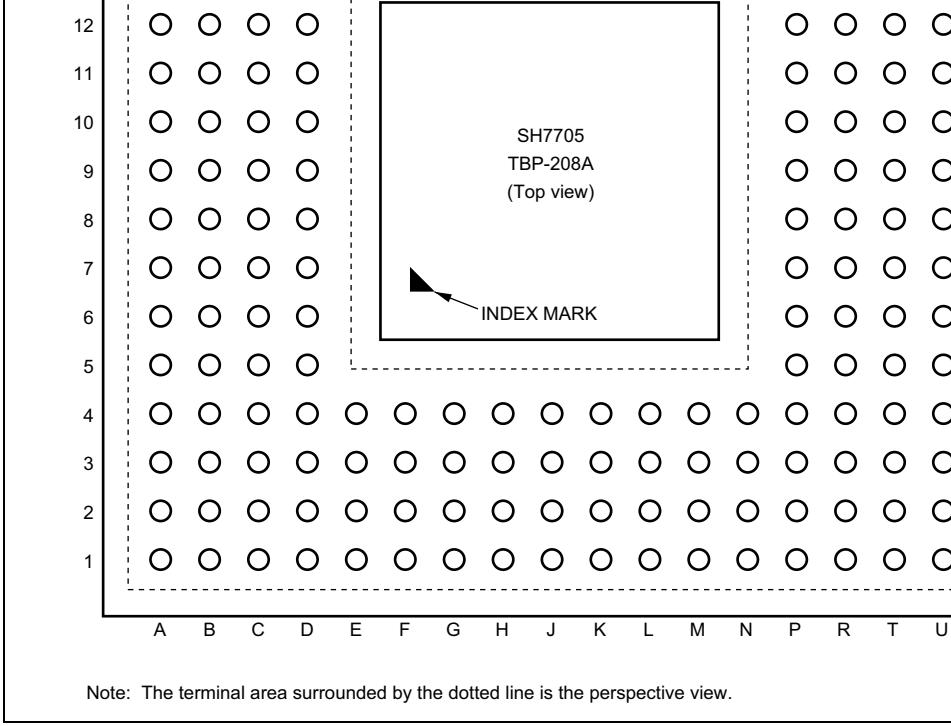


Figure 1.2 Pin Assignment (FP-208C)



Note: The terminal area surrounded by the dotted line is the perspective view.

Figure 1.3 Pin Assignment (TBP-208A)

4	C2	D-	I/O	USB data line
5	C1	Vss-USB	—	USB power supply (0 V)
6	D3	Vcc-RTC*5	—	RTC power supply (3.3 V)*5
7	D2	XTAL2	O	Crystal oscillator pin for on-chip RTC
8	D1	EXTAL2	I	Crystal oscillator pin for on-chip RTC
9	E4	Vss-RTC*5	—	RTC power supply (0 V)*5
10	E3	VBUS/PTM6	I / I/O	USB power supply detection / input/
11	E2	MD6	I	connect to I/O power supply (0V)
12	E1	D31/PTB7/PINT15	I/O / I/O / I	Data bus / input/output port B / PINT
13	F4	D30/PTB6/PINT14	I/O / I/O / I	Data bus / input/output port B / PINT
14	F3	D29/PTB5/PINT13	I/O / I/O / I	Data bus / input/output port B / PINT
15	F2	D28/PTB4/PINT12	I/O / I/O / I	Data bus / input/output port B / PINT
16	F1	D27/PTB3/PINT11	I/O / I/O / I	Data bus / input/output port B / PINT
17	G4	VssQ	—	I/O power supply (0 V)
18	G3	D26/PTB2/PINT10	I/O / I/O / I	Data bus / input/output port B / PINT
19	G2	VccQ	—	I/O power supply (3.3 V)
20	G1	D25/PTB1/PINT9	I/O / I/O / I	Data bus / input/output port B / PINT
21	H4	D24/PTB0/PINT8	I/O / I/O / I	Data bus / input/output port B / PINT
22	H3	D23/PTA7/PINT7	I/O / I/O / I	Data bus / input/output port A / PINT
23	H2	D22/PTA6/PINT6	I/O / I/O / I	Data bus / input/output port A / PINT
24	H1	D21/PTA5/PINT5	I/O / I/O / I	Data bus / input/output port A / PINT
25	J4	D20/PTA4/PINT4	I/O / I/O / I	Data bus / input/output port A / PINT
26	J2	Vss	—	Internal power supply (0 V)
27	J1	D19/PTA3/PINT3	I/O / I/O / I	Data bus / input/output port A / PINT
28	J3	Vcc	—	Internal power supply (1.5 V)
29	K1	D18/PTA2/PINT2	I/O / I/O / I	Data bus / input/output port A / PINT
30	K2	D17/PTA1/PINT1	I/O / I/O / I	Data bus / input/output port A / PINT
31	K3	D16/PTA0/PINT0	I/O / I/O / I	Data bus / input/output port A / PINT

37	M1	D12	I/O	Data bus
38	M2	D11	I/O	Data bus
39	M3	D10	I/O	Data bus
40	M4	D9	I/O	Data bus
41	N1	D8	I/O	Data bus
42	N2	D7	I/O	Data bus
43	N3	D6	I/O	Data bus
44	N4	VssQ	—	I/O power supply (0 V)
45	P1	D5	I/O	Data bus
46	P2	VccQ	—	I/O power supply (3.3 V)
47	P3	D4	I/O	Data bus
48	R1	D3	I/O	Data bus
49	R2	D2	I/O	Data bus
50	P4	D1	I/O	Data bus
51	T1	D0	I/O	Data bus
52	T2	VssQ	—	I/O power supply (0 V)
53	U1	A0/PTK0	O / I/O	Address bus / input/output port K
54	U2	A1	O	Address bus
55	R3	A2	O	Address bus
56	T3	A3	O	Address bus
57	U3	VssQ	—	I/O power supply (0 V)
58	R4	A4	O	Address bus
59	T4	VccQ	—	I/O power supply (3.3 V)
60	U4	A5	O	Address bus
61	P5	A6	O	Address bus
62	R5	A7	O	Address bus
63	T5	A8	O	Address bus

69	P7	VssQ	—	I/O power supply (0 V)
70	R7	A14	O	Address bus
71	T7	VccQ	—	I/O power supply (3.3 V)
72	U7	A15	O	Address bus
73	P8	A16	O	Address bus
74	R8	A17	O	Address bus
75	T8	A18	O	Address bus
76	U8	A19/PTK1	O / I/O	Address bus / input/output port K
77	P9	A20/PTK2	O / I/O	Address bus / input/output port K
78	T9	Vss	—	Internal power supply (0 V)
79	U9	A21/PTK3	O / I/O	Address bus / input/output port K
80	R9	Vcc	—	Internal power supply (1.5 V)
81	U10	A22/PTK4	O / I/O	Address bus / input/output port K
82	T10	A23/PTK5	O / I/O	Address bus / input/output port K
83	R10	A24/PTK6	O / I/O	Address bus / input/output port K
84	P10	A25/PTK7	O / I/O	Address bus / input/output port K
85	U11	$\overline{BS}$/PTC0	O / I/O	Bus cycle start signal / input/outpu
86	T11	$\overline{RD}$	O	Read strobe
87	R11	VssQ	—	I/O power supply (0 V)
88	P11	$\overline{WE0}$/DQMLL	O / O	D7 to D0 select signal / DQM (SDR
89	U12	VccQ	—	I/O power supply (3.3 V)
90	T12	$\overline{WE1}$/DQMLU	O / O	D15 to D8 select signal / DQM (SD
91	R12	$\overline{WE2}$/DQMUL/PTC1	O / O / I/O	D23 to D16 select signal / DQM (S input/output port C
92	P12	$\overline{WE3}$/DQMUU/$\overline{AH}$/ PTC2	O / O / O / I/O	D31 to D24 select signal / DQM (S address hold / input/output port C
93	U13	RD/$\overline{WR}$	O	Read/write
94	T13	$\overline{CS0}$	O	Chip select 0

100	U15	$\overline{CS6A}$*3/PTC7	O / I/O	Chip select 6A / input/output port
101	T15	$\overline{CS6B}$*3/PTD7	O / I/O	Chip select 6B / input/output port
102	P14	$\overline{RASL}$/PTD0	O / I/O	Lower 32 Mbytes address RAS (S input/output port D
103	U16	$\overline{RASU}$*3/PTD1	O / I/O	Upper 32 Mbytes address RAS (S input/output port D
104	T16	$\overline{CASL}$/PTD2	O / I/O	Lower 32 Mbytes address CAS (S input/output port D
105	U17	VssQ	—	I/O power supply (0 V)
106	T17	$\overline{CASU}$*3/PTD3	O / I/O	Upper 32 Mbytes address CAS (S input/output port D
107	R15	CKE/PTD4	O / I/O	CK enable (SDRAM) / input/outpu
108	R16	PTD5/NF*4	I	Input port D / NF*4
109	R17	$\overline{BACK}$/PTG5	O / I/O	Bus acknowledge / input/output po
110	P15	$\overline{BREQ}$/PTG6	I / I/O	Bus request / input/output port G
111	P16	VssQ	—	I/O power supply (0 V)
112	P17	$\overline{WAIT}$/PTG7	I / I/O	Hardware wait request / input/outp
113	N14	VccQ	—	I/O power supply (3.3 V)
114	N15	DACK0/PTE0	O / I/O	DMA acknowledge 0 / input/outpu
115	N16	DACK1/PTE1	O / I/O	DMA acknowledge 1 / input/outpu
116	N17	TEND0/PTE3	O / I/O	DMA transfer end notification / inp port E
117	M14	AUDSYNC/PTF4	O / I/O	AUD synchronous / input/output p
118	M15	AUDATA0/PTF0/TO0	O / I/O / O	AUD data output / input/output po output
119	M16	AUDATA1/PTF1/TO1	O / I/O / O	AUD data output / input/output po output
120	M17	AUDATA2/PTF2/TO2	O / I/O / O	AUD data output / input/output po output

125	K14	NF[*4]/PTJ3	O	NF[*4] /output port J
126	K15	NF[*4]/PTJ4	O	NF[*4] /output port J
127	K16	NF[*4]/PTJ5	O	NF[*4] /output port J
128	K17	NF[*4]/PTJ6	O	NF[*4] /output port J
129	J14	NF[*4]/PTJ7	O	NF[*4] /output port J
130	J16	Vss	—	Internal power supply (0 V)
131	J17	NF[*4]/PTM4	I	NF[*4] / input port M
132	J15	Vcc	—	Internal power supply (1.5 V)
133	H17	VssQ	—	I/O power supply (0 V)
134	H16	PTM0	I/O	Input/output port M
135	H15	PTM1	I/O	Input/output port M
136	H14	PTM2	I/O	Input/output port M
137	G17	PTM3	I/O	Input/output port M
138	G16	VccQ	—	I/O power supply (3.3 V)
139	G15	TDI[*7]/PTG0	I / I/O	Test data input (UDI) / input/outp
140	G14	TCK[*7]/PTG1	I / I/O	Test clock (UDI) / input/output po
141	F17	TMS[*7]/PTG2	I / I/O	Test mode select (UDI) / input/ou
142	F16	$\overline{\text{TRST}}$[*1 *7]/PTG3	I / I/O	Test reset (UDI) / input/output po
143	F15	TDO/PTF5	O / I/O	Test data output (UDI) / input/ou
144	F14	$\overline{\text{ASEBRKAK}}$/PTF6	O / I/O	ASE break acknowledge (UDI) / port F
145	E17	$\overline{\text{ASEMD0}}$[*2 *7]/PTF7	I / I/O	ASE mode (UDI) / input/output p
146	E16	MD0	I	Clock mode setting
147	E15	MD1	I	Clock mode setting
148	E14	MD2	I	Clock mode setting
149	D17	Vcc-PLL1	—	PLL1 power supply (1.5 V)
150	D16	Vss-PLL1	—	PLL1 power supply (0 V)

156	B16	VssQ	—	I/O power supply (0 V)
157	A17	STATUS0/PTE4/ $\overline{\text{RTS0}}$	O / I/O / O	Processor status / input/output port transmit request
158	A16	STATUS1/PTE5/ $\overline{\text{CTS0}}$	O / I/O / I	Processor status / input/output port transmit clear
159	C15	VssQ	—	I/O power supply (0 V)
160	B15	CKIO	I/O	System clock input/output
161	A15	VccQ	—	I/O power supply (3.3 V)
162	C14	PTN0/SUSPND	I/O / O	input/output port N / USB suspend
163	B14	PTN1/TXENL	I/O / O	input/output port N / USB output ena
164	A14	PTN2/XVDATA	I/O / I	input/output port N / USB differentia input
165	D13	PTN3/TXDMNS	I/O / O	input/output port N / USB D– transm
166	C13	PTN4/TXDPLS	I/O / O	input/output port N / USB D+ transm
167	B13	PTN5/DMNS	I/O / I	input/output port N / D– input from U receiver
168	A13	PTN6/DPLS	I/O / I	input/output port N / D+ input from U receiver
169	D12	PTN7	I/O	input/output port N
170	C12	TCLK/PTE6	I / I/O	TMU clock input / input/output port I
171	B12	PTE7	I/O	Input/output port E
172	A12	TxD0/SCPT0/IrTX	O / O / O	SCIF0 transmit data / SC port / IrDA
173	D11	SCK0/SCPT1	I/O / I/O	SCIF0 clock / SC port
174	C11	TxD2/SCPT2	O / O	SCIF2 transmit data / SC port
175	B11	SCK2/SCPT3	I/O / I/O	SCIF2 clock / SC port
176	A11	$\overline{\text{RTS2}}$/SCPT4	O / I/O	SCIF2 transmit request / SC port
177	D10	RxD0/SCPT0/IrRX	I / I / I	SCIF0 receive data / SC port / IrDA
178	C10	VccQ	—	I/O power supply (3.3 V)
179	B10	RxD2/SCPT2	I / I	SCIF2 receive data / SC port

185	A8	IRQ0/$\overline{\text{IRL0}}$/PTH0	I / I / I/O	External interrupt request / input/o
186	B8	IRQ1/$\overline{\text{IRL1}}$/PTH1	I / I / I/O	External interrupt request / input/o
187	C8	IRQ2/$\overline{\text{IRL2}}$/PTH2	I / I / I/O	External interrupt request / input/o
188	D8	IRQ3/$\overline{\text{IRL3}}$/PTH3	I / I / I/O	External interrupt request / input/o
189	A7	IRQ4/PTH4	I / I/O	External interrupt request / input/o
190	B7	IRQ5/PTE2	I / I/O	External interrupt request / input/o
191	C7	AUDCK/PTG4	O / I/O	AUD clock / input/output port G
192	D7	NMI	I	Nonmaskable interrupt request
193	A6	DREQ0/PTH5	I / I/O	DMA request / input/output port H
194	B6	DREQ1/PTH6	I / I/O	DMA request / input/output port H
195	C6	$\overline{\text{RESETP}}$*6	I	Power-on reset request
196	D6	CA	I	Hardware standby request
197	A5	MD3	I	Area 0 bus width setting
198	B5	MD4	I	Area 0 bus width setting
199	C5	AVss	—	Analog power supply (0 V)
200	D5	AN0/PTL0	I / I	A/D converter input / input port L
201	A4	AN1/PTL1	I / I	A/D converter input / input port L
202	B4	AN2/PTL2	I / I	A/D converter input / input port L
203	C4	AN3/PTL3	I / I	A/D converter input / input port L
204	A3	AVcc	—	Analog power supply (3.3 V)
205	B3	VssQ	—	I/O power supply (0 V)
206	D4	EXTAL_USB	I	USB clock
207	A2	XTAL_USB	O	USB clock
208	B2	VccQ	—	I/O power supply (3.3 V)

Notes: The unused pins should be handled according to table A.1, I/O Port States in E Processing State, in Appendix.

1. The $\overline{\text{TRST}}$ pin must be driven low for a specified period when power supply regardless of whether the UDI function is used or not. As the same as the $\overline{\text{R}}$ the $\overline{\text{TRST}}$ pin should be driven low at the power-on set state and driven hig power-on reset state is released.

7. The pull-up MOS turns on if the pin function controller (PFC) is used to selec
functions (UDI).

				power supply. There w operation if any pins a
	Vss	—	Ground	Ground pin. Connect a the system power sup There will be no opera pins are open.
	VccQ	—	Power supply	Power supply for I/O p all VccQ pins to the sy supply. There will be n if any pins are open.
	VssQ	—	Ground	Ground pin. Connect a to the system power s There will be no opera pins are open.
Clock	Vcc-PLL1	—	PLL1 power supply	Power supply for the o oscillator.
	Vss-PLL1	—	PLL1 ground	Ground pin for the on- oscillator.
	Vcc-PLL2	—	PLL2 power supply	Power supply for the o oscillator.
	Vss-PLL2	—	PLL2 ground	Ground pin for the on- oscillator.
	EXTAL	I	External clock	For connection to a cry resonator. An external may also be input to th pin.
	XTAL	O	Crystal	For connection to a cry resonator.
	CKIO	I/O	System clock	Supplies the system cl external devices.

System control	RESETP	I	Power-on reset	When low, the system e power-on reset state.
	$\overline{\text{RESETM}}$	I	Manual reset	When low, the system e manual reset state.
	STATUS1, STATUS0	O	Status output	Indicates the operating
	$\overline{\text{BREQ}}$	I	Bus request	Low when an external d requests the release of mastership.
	$\overline{\text{BACK}}$	O	Bus request acknowledge	Indicates that the bus m has been released to an device. Reception of the signal informs the devic has output the $\overline{\text{BREQ}}$ si has acquired the bus.
	CA	I	Chip active	High in normal operatio in hardware standby mo
Interrupts	NMI	I	Non-maskable interrupt	Non-maskable interrupt pin. Fix to high level wh use.
	IRQ5 to IRQ0	I	Interrupt requests 5 to 0	Maskable interrupt requ Selectable as level inpu input. The rising edge o edge is selectable as th edge. The low level or h selectable as the detect
	$\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$	I	Interrupt requests 3 to 0	Maskable interrupt requ Input a coded interrupt
	PINT15 to PINT0	I	Interrupt requests 15 to 0	PINT interrupt request p
Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D31 to D0	I/O	Data bus	32-bit bidirectional data

$\overline{WE3}$	O	Highest-byte write	Indicates that bits 31 t data in the external me device are being writte
$\overline{WE2}$	O	Second-highest-byte write	Indicates that bits 23 t data in the external me device are being writte
$\overline{WE1}$	O	Second-lowest-byte write	Indicates that bits 15 t data in the external me device are being writte
$\overline{WE0}$	O	Lowest-byte write	Indicates that bits 7 to in the external memory are being written.
CKE	O	CK enable	Clock enable. (SDRAM
DQMUU	O	DQ mask UU	Selects D31 to D24. (S
DQMUL	O	DQ mask UL	Selects D23 to D16. (S
DQMLU	O	DQ mask LU	Selects D15 to D8. (SI
DQMLL	O	DQ mask LL	Selects D7 to D0. (SD
$\overline{RASU}$	O	Row address U	Specifies a row addres
$\overline{RASL}$	O	Row address L	Specifies a row addres
$\overline{CASU}$	O	Column address U	Specifies a column ad (SDRAM)
$\overline{CASL}$	O	Column address L	Specifies a column ad (SDRAM)
$\overline{AH}$	O	Address hold	Address hold signal.
$\overline{WAIT}$	I	Wait	Inserts a wait cycle int cycles during access t external space.

				capture control input pin
16-bit timer pulse unit (TPU)	TO3 to TO0	O	Timer output	Output compare/PWM o
Serial communication interface with FIFO (SCIF0, SCIF2)	TxD0, TxD2	O	Transmit data	Transmit data pin.
	RxD0, RxD2	I	Receive data	Receive data pin.
	SCK0, SCK2	I/O	Serial clock	Clock input/output pin.
	$\overline{\text{RTS0}}$, $\overline{\text{RTS2}}$	O	Transmit request	Modem control pin.
	$\overline{\text{CTS0}}$, $\overline{\text{CTS2}}$	I	Transmit enable	Modem control pin.
IrDA	IrTX	O	IrDA TX port	IrDA transmit data outpu
	IrRX	I	IrDA RX port	IrDA receive data input.
Realtime clock (RTC)	EXTAL2	I	RTC clock	RTC crystal oscillator pi kHz)
	XTAL2	O	RTC clock	RTC crystal oscillator pi kHz)
	Vcc-RTC	—	RTC power supply	Power supply pin for the
	Vss-RTC	—	RTC ground	Ground pin for the RTC
A/D converter (ADC)	AN3 to AN0	I	Analog input pin	Analog input pin.
	AVcc	—	A/D analog power supply	Power supply for the A/ converter. When the A/ is not in use, connect th port power supply (Vcc(
	AVss	—	A/D analog ground	Ground pin for the A/D Connect this pin to the s power supply (Vss).

TXDPLS	O	D+ output	D+ transmit output pin driver.
TXDMNS	O	D– output	D– transmit output pin driver.
DPLS	I	D+ input	D+ signal input pin fro receiver to the driver.
DMNS	I	D– input	D– signal input pin fro receiver to the driver.
TXENL	O	Output enable	Output enable pin for t
SUSPND	O	Suspend	Suspend-state output transceiver.
Vcc-USB	—	USB analog power supply	USB power supply pin USB is not in use, con to the port power supp
Vss-USB	—	USB analog ground	USB ground pin. Conn to the system power s
D–	I/O	D– I/O	On-chip USB transceiv
D+	I/O	D+ I/O	On-chip USB transceiv

PTE7 to PTE0	I/O	General purpose port	8-bit general-purpose I/
PTF7 to PTF0	I/O	General purpose port	8-bit general-purpose I/
PTG7 to PTG0	I/O	General purpose port	8-bit general-purpose I/
PTH6 to PTH0	I/O	General purpose port	7-bit general-purpose I/
PTJ7 to PTJ0	O	General purpose port	8-bit general-purpose o pins.
PTK7 to PTK0	I/O	General purpose port	8-bit general-purpose I/
PTL3 to PTL0	I	General purpose port	4-bit general-purpose in pins.
PTM6, PTM4 to PTM0	I/O	General purpose port	6-bit general-purpose I/
PTN7 to PTN0	I/O	General purpose port	8-bit general-purpose I/
SCPT5 to SCPT0	I/O	Serial port	6-bit serial port pins.

	TRST	I	Test reset	Initial signal input pin.
Advanced user debugger (AUD)	AUDATA3 to AUDATA0	O	AUD data	Destination-address o branch-trace mode.
	AUDCK	O	AUD clock	Synchronous clock ou branch-trace mode.
	AUDSYNC	O	AUD synchronous signal	Data start-position ack signal output pin in bra mode.
E10A interface	$\overline{\text{ASEBRKAK}}$	O	Break mode acknowledge	Indicates that the E10A has entered its break m For the connection wit see the SH7705 E10A User's Manual (tentati
	$\overline{\text{ASEMD0}}$	I	ASE mode	Sets ASE mode.

program execution state, and a low-power consumption state, according to the CPU pr
states.

Reset State: In the reset state, the CPU is reset. The LSI supports two types of resets:
reset and manual reset. For details on resets, refer to section 5, Exception Handling.

In power-on reset, the registers and internal statuses of all LSI on-chip modules are ini
manual reset, the register contents of a part of the LSI on-chip modules, such as the bu
controller (BSC), are retained. For details, refer to section 24, List of Registers. The C
statuses and registers are initialized both in power-on reset and manual reset. After ini
the program branches to address H'A0000000 to pass control to the reset processing pr
executed.

Exception Handling State: In the exception handling state, the CPU processing flow
temporarily by a general exception or interrupt exception processing. The program cou
and status register (SR) are saved in the save program counter (SPC) and save status re
(SSR), respectively. The program branches to an address obtained by adding a vector
vector base register (VBR) and passes control to the exception processing program de
user to be executed. For details on reset, refer to section 5, Exception Handling.

Program Execution State: The CPU executes programs sequentially.

Low-Power Consumption State: The CPU stops operation to reduce power consump
low-power consumption state can be entered by executing the SLEEP instruction. For
the low-power consumption state, refer to section 11, Power-Down Modes.

Figure 2.1 shows a status transition diagram.

registers, including SR, and some of the address spaces cannot be accessed by the user
and system control instructions cannot be executed. This function effectively protects th
resources from the user program. To change the processing mode from user to privilege
transition to exception handling state is required.*

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

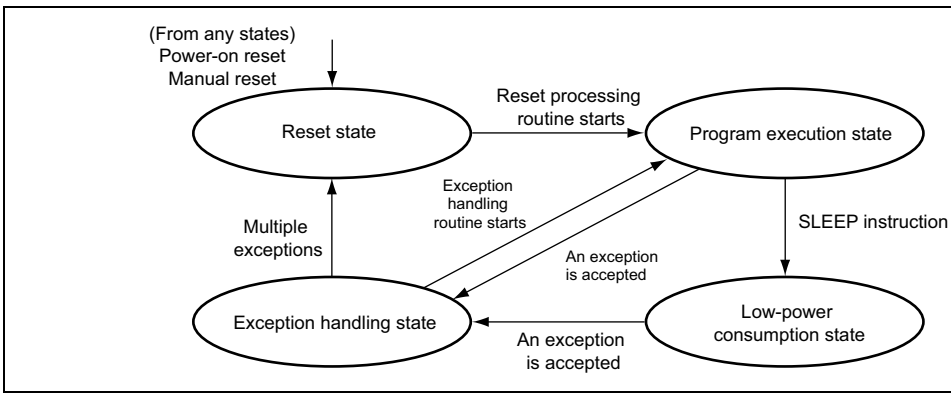


Figure 2.1 Processing State Transitions

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

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

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

P1 Area: The P1 area is defined as a cacheable but non-address translatable area. Nor
programs executed at high speed in privileged mode, such as exception processing har
are at the core of the operating system (S), are assigned to the P1 area.

P2 Area: The P2 area is defined as a non-cacheable but non-address translatable area.
processing program to be called from the reset state is described at the start address (H
of the P2 area. Normally, programs such as system initialization routines and OS initia
programs are assigned to the P2 area. To access a part of an on-chip module control re
corresponding program should be assigned to the P2 area.

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

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

H'A0000000 to H'BFFFFFFF	P2	Privileged mode	0.5-Gbyte physical space, non-ca
H'C0000000 to H'DFFFFFFF	P3	Privileged mode	0.5-Gbyte physical space, cachea address translatable
H'E0000000 to H'FFFFFFFF	P4	Privileged mode	0.5-Gbyte control space, non-cac

2.2.2 External Memory Space

The LSI uses 29 bits of the 32-bit logical address to access external memory. In this cas Gbyte of external memory space can be accessed. The external memory space is manag units. Different types of memory can be connected to each area, as shown in figure 2.2. details, please refer to section 7, Bus State Controller (BSC). In addition, area 1 in the e memory space is used as an on-chip I/O space where most of this LSI's on-chip module registers are mapped. *[1]

Normally, the upper three bits of the 32-bit logical address are masked and the lower 29 used for external memory addresses.*[2] For example, address H'00000100 in the P0 are H'80000100 in the P1 area, address H'A0000100 in the P2 area, and address H'C00001 area of the logical address space are mapped into address H'00000100 of area 0 in the e memory space. The P4 area in the logical address space is not mapped into the external address. If an address in the P4 area is accessed, an external memory cannot be accesse

Notes: 1. To access an on-chip module control register mapped into area 1 in the exte memory space, access the address from the P2 area which is not cached in th address space.

2. If the address translation unit is enabled, arbitrary mapping in page units can specified. For details, refer to section 3, Memory Management Unit (MMU)

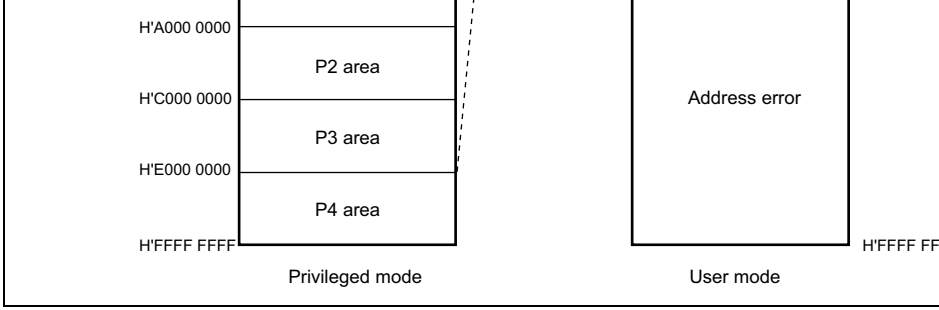


Figure 2.2 Logical Address to External Memory Space Mapping

2.3 Register Descriptions

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

General Registers: This LSI incorporates 24 general registers: R0_BANK0 to R7_BA R0_BANK1 to R7_BANK1 and R8 to R15. R0 to R7 are banked. The process mode a register bank (RB) bit in the status register (SR) define which set of banked registers (to R7_BANK0 or R0_BANK1 to R7_BANK1) are accessed as general registers.

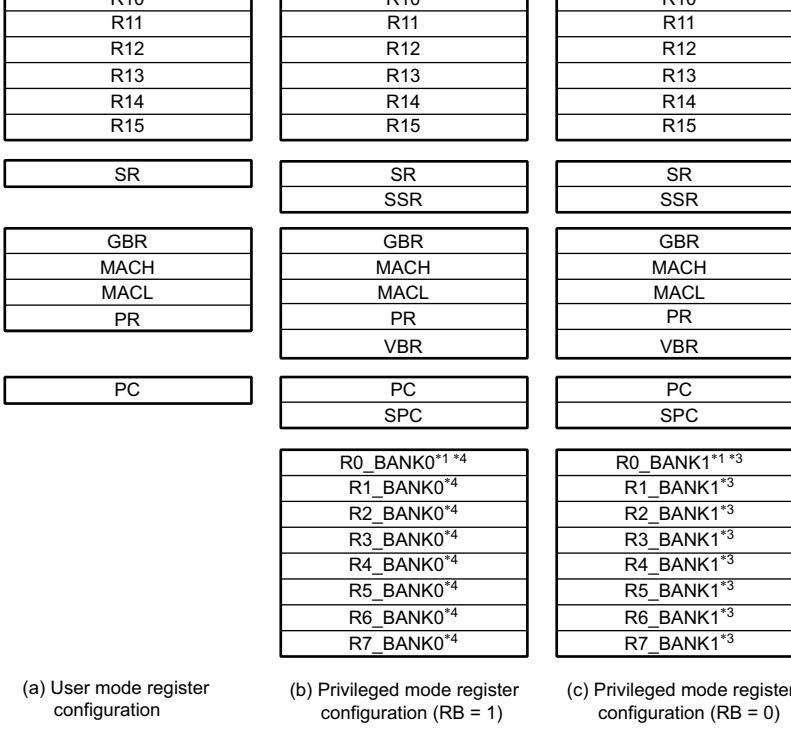
System Registers: This LSI incorporates the multiply and accumulate registers (MAC and procedure register (PR) as system registers. These registers can be accessed regard processing mode.

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

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

System registers	MACH, MACL, PR	Undefined
Program counter	PC	H'A0000000
Control registers	SR	MD bit = 1, RB bit = 1, BL bit = 1, bits = H'F (1111), reserved bits = bits = undefined
	GBR, SSR, SPC	Undefined
	VBR	H'00000000

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



(a) User mode register configuration

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

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

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

Figure 2.3 Register Configuration in Each Processing Mode

In user mode, bank 0 is selected regardless of he RB bit value. Sixteen registers: R0_BA
R7_BANK0 and R8 to R15 are accessed as general registers R0 to R15. The R0_BANK
R7_BANK1 registers in bank 1 cannot be accessed.

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

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

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

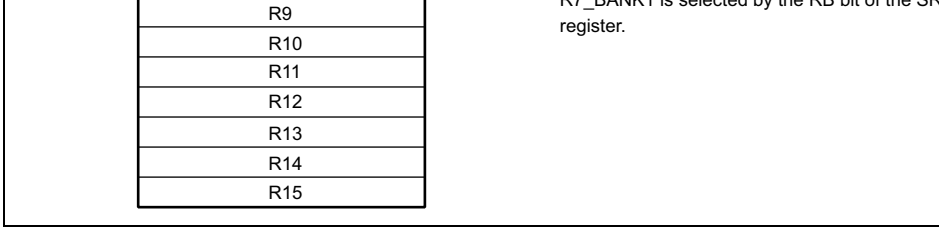


Figure 2.4 General Registers

2.3.2 System Registers

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

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

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

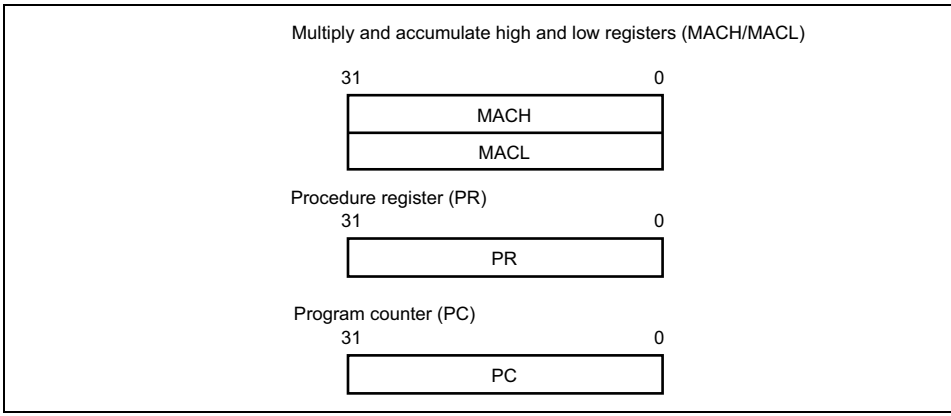


Figure 2.5 System Registers and Program Counter

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value shou 0.
30	MD	1	R/W	Processing Mode Indicates the CPU processing mode. 0: User mode 1: Privileged mode The MD bit is set to 1 in reset or exception handli
29	RB	1	R/W	Register Bank The general registers R0 to R7 are banked regist bit selects a bank used in the privileged mode. 0: Selects bank 0 registers. In this case, R0_BAN R7_BANK0 and R8 to R15 are used as genera R0_BANK1 to R7_BANK1 can be accessed by STR instruction. 1: Selects bank 1 registers. In this case, R0_BAN R7_BANK1 and R8 to R15 are used as genera R0_BANK0 to R7_BANK0 can be accessed by STR instruction. The RB bit is set to 1 in reset or exception handli
28	BL	1	R/W	Block Specifies whether an exception, interrupt, or user enabled or not. 0: Enables an exception, interrupt, or user break. 1: Disables an exception, interrupt, or user break The BL bit is set to 1 in reset or exception handlir
27 to 10	—	0	R	Reserved These bits are always read as 0. The write value always be 0.

7 to 4	I3 to I0	1	R/W	Interrupt Mask Indicates the interrupt mask level. These bits do n even if an interrupt occurs. At reset, these bits are to B'1111. These bits are not affected in an excep handling state.
3, 2	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.
1	S	—	R/W	Saturation Mode Specifies the saturation mode for multiply instructi multiply and accumulate instructions. This bit can by the SETS and CLRS instructions in user mode. At reset, this bit is undefined. This bit is not affecte exception handling state.
0	T	—	R/W	T Bit Indicates true or false for compare instructions or borrow occurrence for an operation instruction wit borrow. This bit can be specified by the SETT and instructions in user mode. At reset, this bit is undefined. This bit is not affecte exception handling state.

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

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

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

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

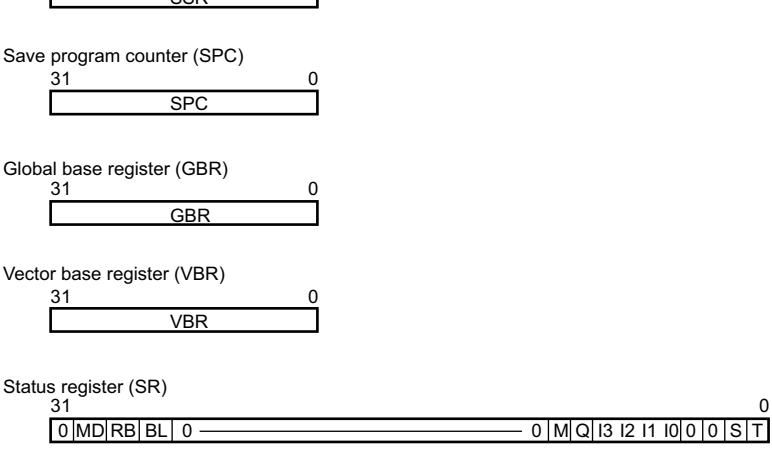
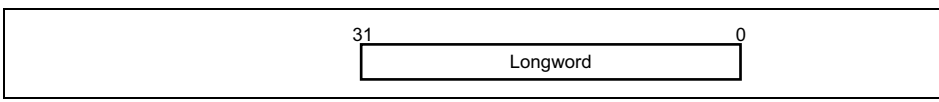


Figure 2.6 Control Register Configuration

2.4 Data Formats

2.4.1 Register Data Format

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



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

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

The on-chip device registers assigned to memory are accessed in big endian mode. Not
available access size (byte, word, or long word) differs in each register.

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

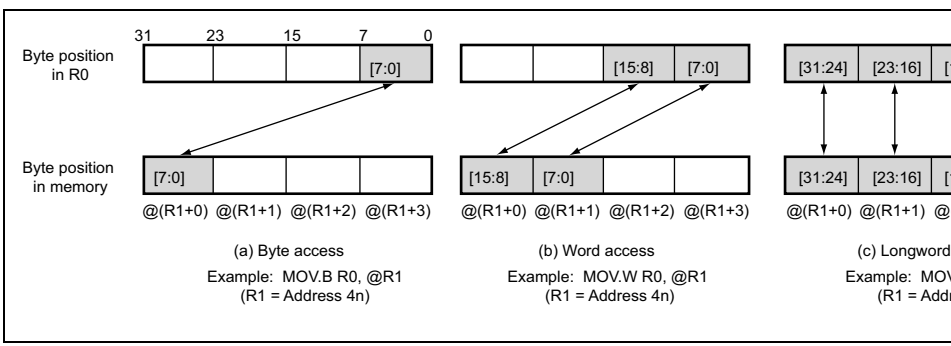


Figure 2.7 Data Format on Memory (Big Endian Mode)

and the LSB byte of the R1 register is stored at the address indicated by the R1.

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

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

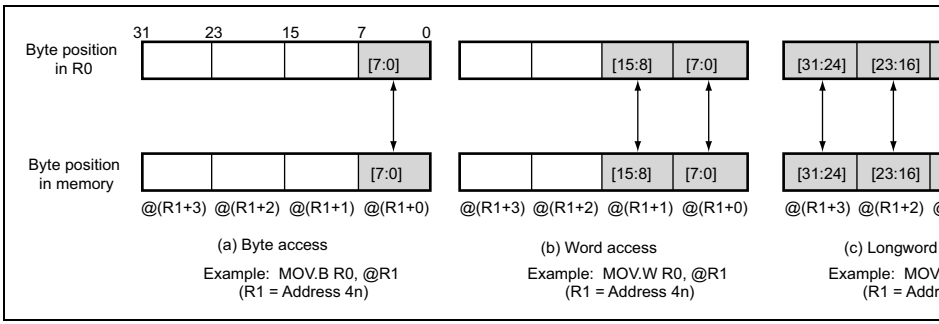


Figure 2.8 Data Format on Memory (Little Endian Mode)

data. Immediate data is sign-extended to longword size for arithmetic operations (MOV and CMP/EQ instructions) or zero-extended to longword size for logical operations (TS OR, and XOR instructions).

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

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

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

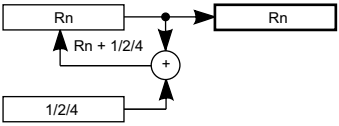
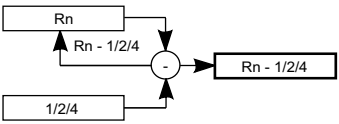
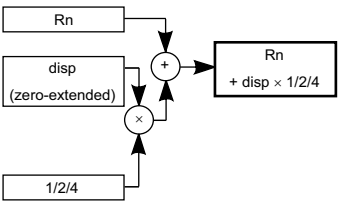
```
Example:  BRA     TARGET
          ADD     R1, R0      ; ADD is executed before branching to the TARGE
```

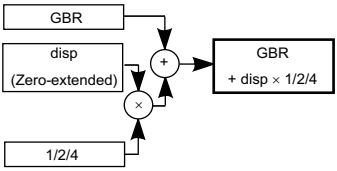
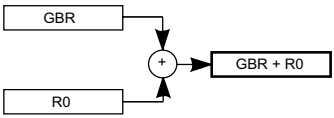
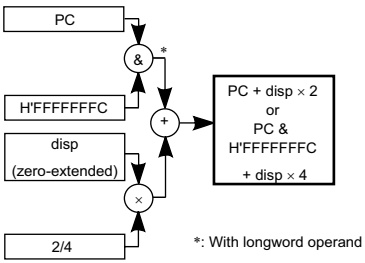
T Bit: The result of a comparison is indicated by the T bit in the status register (SR), an conditional branch is performed according to whether the result is True or False. Proces has been improved by keeping the number of instructions that modify the T bit to a min

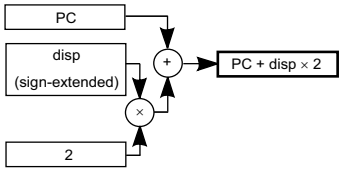
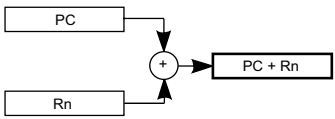
```
Example:  ADD     #1, R0      ; The T bit cannot be modified by the ADD instruc
          CMP/EQ  #0, R0      ; The T bit is set to 1 if R0 is 0.
          BT      Target      ; Branch to TARGET if the T bit is set to 1 (R0=0
```

method whereby immediate data is loaded when an instruction is executed, this value
transferred to a register and the data is referenced using register indirect addressing m

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

Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn
Register indirect with post-increment	@Rn+	Effective address is register Rn contents. A constant is added to Rn after instruction execution: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Rn After in executi Byte: Rn Word: R Longwor Rn
Register indirect with pre-decrement	@–Rn	Effective address is register Rn contents, decremented by a constant beforehand: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Byte: Rn Word: R Longwor Rn (Instructi with Rn calculati
Register indirect with displacement	@(disp:4, Rn)	Effective address is register Rn contents with 4-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Byte: R Word: F 2 Longwo disp × 4

displacement		zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Word: × 2 Longw disp ×
Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents. 	GBR +
PC-relative with displacement	@(disp:8, PC)	Effective address is PC with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word) or 4 (longword), according to the operand size. With a longword operand, the lower 2 bits of PC are masked. 	Word: Longw PC&H' + disp ×

		2	
	disp:12	Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2 	PC + di
	Rn	Effective address is sum of PC and Rn. 	PC + R
Immediate	#imm:8	8-bit immediate data imm of TST, AND, OR, or XOR instruction is zero-extended.	—
	#imm:8	8-bit immediate data imm of MOV, ADD, or CMP/EQ instruction is sign-extended.	—
	#imm:8	8-bit immediate data imm of TRAPA instruction is zero-extended and multiplied by 4.	—

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

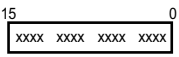
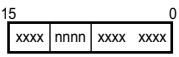
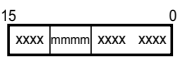
```
@ (disp:4, Rn)      ; Register indirect with displacement
@ (disp:8, Rn)      ; GBR indirect with displacement
@ (disp:8, PC)      ; PC relative with displacement
disp:8, disp        ; PC relative
```

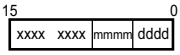
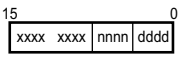
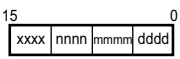
nnnn: Destination register

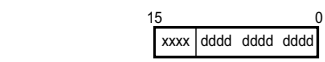
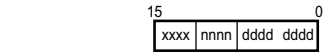
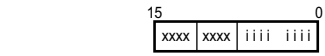
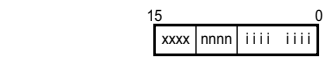
iiii: Immediate data

dddd: Displacement

Table 2.4 CPU Instruction Formats

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

	indirect (multiply-and-accumulate operation) nnnn: * post-increment register indirect (multiply-and-accumulate operation)		
	mmmm: post-increment register indirect	nnnn: register direct	MOV.L @Rm
	mmmm: register direct	nnnn: pre-decrement register indirect	MOV.L Rm,@
	mmmm: register direct	nnnn: indexed register indirect	MOV.L Rm,@
md type 	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp
nd4 type 	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(
nmd type 	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@
	mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L @(disp

	PC-relative with displacement			
	dddddddd: PC-relative	—	BF	label
d12 type 	ddddddddddd: PC-relative	—	BRA	label (label
nd8 type 	dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L	@(dis
i type 	iiiiiiii: immediate	Indexed GBR indirect	AND.B	#imm
	iiiiiiii: immediate	R0 (register direct)	AND	#imm
	iiiiiiii: immediate	—	TRAPA	#imm
ni type 	iiiiiiii: immediate	nnnn: register direct	ADD	#imm

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

Table 2.3 CPU Instruction Types

Type	Kinds of Instruction	Op Code	Function	N Ir
Data transfer instructions	5	MOV	Data transfer	3
			Immediate data transfer	
			Peripheral module data transfer	
			Structure data transfer	
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Upper/lower swap	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operation instructions	21	ADD	Binary addition	3
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Signed division initialization	
		DIV0U	Unsigned division initialization	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate, double-precision multiply-and-accumulate	
		MUL	Double-precision multiplication (32×32 bits)	

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

		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control instructions	15	CLRT	T bit clear	7
		CLRMAC	MAC register clear	
		CLRS	S bit clear	
		LDC	Load into control register	
		LDS	Load into system register	
		LDTLB	PTEH/PTEL load into TLB	
		NOP	No operation	
		PREF	Data prefetch to cache	
		RTE	Return from exception handling	
		SETS	S bit setting	
		SETT	T bit setting	
		SLEEP	Transition to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	68			1

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

Rm: Source register

Rn: Destination register

imm: Immediate data

disp: Displacement

mmmm: Source register

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

iiii: Immediate data

dddd: Displacement*2

→, ←: Transfer direction

(xx): Memory operand

M/Q/T: Flag bits in SR

&: Logical AND of each bit

|: Logical OR of each bit

^: Exclusive logical OR of each bit

~: Logical NOT of each bit

<<n: n-bit left shift

>>n: n-bit right shift

Notes: 1. The table shows the minimum number of execution states. In practice, the
instruction execution states will be increased in cases such as the following
a. When there is a conflict between an instruction fetch and a data a
b. When the destination register of a load instruction (memory → reg
used by the following instruction
2. Scaled (x1, x2, or x4) according to the instruction operand size, etc.

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

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

CMP/EQ	Rm,Rn	0011nnnnmmmm0000	If Rn = Rm, 1 → T	–	1
CMP/HS	Rm,Rn	0011nnnnmmmm0010	If Rn ≥ Rm with unsigned data, 1 → T	–	1
CMP/GE	Rm,Rn	0011nnnnmmmm0011	If Rn ≥ Rm with signed data, 1 → T	–	1
CMP/HI	Rm,Rn	0011nnnnmmmm0110	If Rn > Rm with unsigned data, 1 → T	–	1
CMP/GT	Rm,Rn	0011nnnnmmmm0111	If Rn > Rm with signed data, 1 → T	–	1
CMP/PL	Rn	0100nnnn00010101	If Rn ≥ 0, 1 → T	–	1
CMP/PZ	Rn	0100nnnn00010001	If Rn > 0, 1 → T	–	1
CMP/STR	Rm,Rn	0010nnnnmmmm1100	If Rn and Rm have an equivalent byte, 1 → T	–	1
DIV1	Rm,Rn	0011nnnnmmmm0100	Single-step division (Rn/Rm)	–	1
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	–	1
DIV0U		0000000000011001	0 → M/Q/T	–	1
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	–	2 (to 5)*
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	–	2 (to 5)*
DT	Rn	0100nnnn00010000	Rn – 1 → Rn, if Rn = 0, 1 → T, else 0 → T	–	1

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

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

OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	–	1
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	–	1
OR.B	#imm,@(R0, GBR)	11001111iiiiiiii	(R0+GBR) | imm → (R0+GBR)	–	3
TAS.B	@Rn	0100nnnn00011011	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)	–	4
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

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

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

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

LDC	Rm,GBR	0100mmmm00011110	Rm→GBR	–	4
LDC	Rm,VBR	0100mmmm00101110	Rm→VBR	√	4
LDC	Rm,SSR	0100mmmm00111110	Rm→SSR	√	4
LDC	Rm,SPC	0100mmmm01001110	Rm→SPC	√	4
LDC	Rm,R0_BANK	0100mmmm10001110	Rm→R0_BANK	√	4
LDC	Rm,R1_BANK	0100mmmm10011110	Rm→R1_BANK	√	4
LDC	Rm,R2_BANK	0100mmmm10101110	Rm→R2_BANK	√	4
LDC	Rm,R3_BANK	0100mmmm10111110	Rm→R3_BANK	√	4
LDC	Rm,R4_BANK	0100mmmm11001110	Rm→R4_BANK	√	4
LDC	Rm,R5_BANK	0100mmmm11011110	Rm→R5_BANK	√	4
LDC	Rm,R6_BANK	0100mmmm11101110	Rm→R6_BANK	√	4
LDC	Rm,R7_BANK	0100mmmm11111110	Rm→R7_BANK	√	4
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm)→SR, Rm+4→Rm	√	8
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm)→GBR, Rm+4→Rm	–	4
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm)→VBR, Rm+4→Rm	√	4
LDC.L	@Rm+,SSR	0100mmmm00110111	(Rm)→SSR, Rm+4→Rm	√	4
LDC.L	@Rm+,SPC	0100mmmm01000111	(Rm)→SPC, Rm+4→Rm	√	4
LDC.L	@Rm+, R0_BANK	0100mmmm10000111	(Rm)→R0_BANK, Rm+4→Rm	√	4
LDC.L	@Rm+, R1_BANK	0100mmmm10010111	(Rm)→R1_BANK, Rm+4→Rm	√	4
LDC.L	@Rm+, R2_BANK	0100mmmm10100111	(Rm)→R2_BANK, Rm+4→Rm	√	4
LDC.L	@Rm+, R3_BANK	0100mmmm10110111	(Rm)→R3_BANK, Rm+4→Rm	√	4
LDC.L	@Rm+, R4_BANK	0100mmmm11000111	(Rm)→R4_BANK, Rm+4→Rm	√	4
LDC.L	@Rm+, R5_BANK	0100mmmm11010111	(Rm)→R5_BANK, Rm+4→Rm	√	4

LDS	Rm,PR	0100mmmm00101010	Rm→PR	–	1
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm)→MACH, Rm+4→Rm	–	1
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm)→MACL, Rm+4→Rm	–	1
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm)→PR, Rm+4→Rm	–	1
LDTLB		0000000000111000	PTEH/PTEL→TLB	√	1
NOP		0000000000001001	No operation	–	1
PREF	@Rm	0000mmmm10000011	(Rm) → cache	–	1
RTE		0000000000101011	Delayed branch, SSR → SR, SPC → PC	√	5
SETS		0000000001011000	1→S	–	1
SETT		0000000000011000	1→T	–	1
SLEEP		0000000000011011	Sleep	√	4*1
STC	SR,Rn	0000nnnn00000010	SR→Rn	√	1
STC	GBR,Rn	0000nnnn00010010	GBR→Rn	–	1
STC	VBR,Rn	0000nnnn00100010	VBR→Rn	√	1
STC	SSR, Rn	0000nnnn00110010	SSR→Rn	√	1
STC	SPC,Rn	0000nnnn01000010	SPC→Rn	√	1
STC	R0_BANK,Rn	0000nnnn10000010	R0_BANK→Rn	√	1
STC	R1_BANK,Rn	0000nnnn10010010	R1_BANK→Rn	√	1
STC	R2_BANK,Rn	0000nnnn10100010	R2_BANK→Rn	√	1
STC	R3_BANK,Rn	0000nnnn10110010	R3_BANK→Rn	√	1
STC	R4_BANK,Rn	0000nnnn11000010	R4_BANK→Rn	√	1
STC	R5_BANK,Rn	0000nnnn11010010	R5_BANK→Rn	√	1
STC	R6_BANK,Rn	0000nnnn11100010	R6_BANK→Rn	√	1
STC	R7_BANK,Rn	0000nnnn11110010	R7_BANK→Rn	√	1
STC.L	SR,@–Rn	0100nnnn00000011	Rn–4→Rn, SR→(Rn)	√	1
STC.L	GBR,@–Rn	0100nnnn00010011	Rn–4→Rn, GBR→(Rn)	–	1
STC.L	VBR,@–Rn	0100nnnn00100011	Rn–4→Rn, VBR→(Rn)	√	1

STC.L	R4_BANK,@–Rn	0100nnnn11000011	Rn–4→Rn, R4_BANK→(Rn)	√	1
STC.L	R5_BANK,@–Rn	0100nnnn11010011	Rn–4→Rn, R5_BANK→(Rn)	√	1
STC.L	R6_BANK,@–Rn	0100nnnn11100011	Rn–4→Rn, R6_BANK→(Rn)	√	1
STC.L	R7_BANK,@–Rn	0100nnnn11110011	Rn–4→Rn, R7_BANK→(Rn)	√	1
STS	MACH,Rn	0000nnnn00001010	MACH→Rn	–	1
STS	MACL,Rn	0000nnnn00011010	MACL→Rn	–	1
STS	PR,Rn	0000nnnn00101010	PR→Rn	–	1
STS.L	MACH,@–Rn	0100nnnn00000010	Rn–4→Rn, MACH→(Rn)	–	1
STS.L	MACL,@–Rn	0100nnnn00010010	Rn–4→Rn, MACL→(Rn)	–	1
STS.L	PR,@–Rn	0100nnnn00100010	Rn–4→Rn, PR→(Rn)	–	1
TRAPA	#imm	11000011iiiiiiii	Unconditional trap exception occurs*2	–	8

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

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

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

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

@ (disp:4, Rn) ; Register indirect with displacement

@ (disp:8, Rn) ; GBR indirect with displacement

@ (disp:8, PC) ; PC relative with displacement

disp:8, disp ; PC relative

1. Number of states before the chip enters the sleep state.
2. For details, refer to section 5, Exception Handling.

0000	Rn	Fx	0001							
0000	Rn	00MD	0010	STC	SR, Rn	STC	GBR, Rn	STC	VBR, Rn	STC
0000	Rn	01MD	0010	STC	SPC, Rn					
0000	Rn	10MD	0010	STC	R0_BANK, Rn	STC	R1_BANK, Rn	STC	R2_BANK, Rn	STC
0000	Rn	11MD	0010	STC	R4_BANK, Rn	STC	R5_BANK, Rn	STC	R6_BANK, Rn	STC
0000	Rm	00MD	0011	BSRF	Rm			BRAF	Rm	
0000	Rm	10MD	0011	PREF	@Rm					
0000	Rn	Rm	01MD	MOV.B	Rm, @(R0, Rn)	MOV.W	Rm, @(R0, Rn)	MOV.L	Rm,@(R0, Rn)	MUL.L
0000	0000	00MD	1000	CLRT		SETT		CLRMAC		LDTLB
0000	0000	01MD	1000	CLRS		SETS				
0000	0000	Fx	1001	NOP		DIV0U				
0000	0000	Fx	1010							
0000	0000	Fx	1011	RTS		SLEEP		RTE		
0000	Rn	Fx	1000							
0000	Rn	Fx	1001					MOVT	Rn	
0000	Rn	Fx	1010	STS	MACH, Rn	STS	MACL, Rn	STS	PR, Rn	
0000	Rn	Fx	1011							
0000	Rn	Rm	11MD	MOV. B Rn	@(R0, Rm),	MOV.W Rn	@(R0, Rm),	MOV.L	@(R0, Rm), Rn	MAC.L
0001	Rn	Rm	disp	MOV.L	Rm, @(disp:4, Rn)					
0010	Rn	Rm	00MD	MOV.B	Rm, @Rn	MOV.W	Rm, @Rn	MOV.L	Rm, @Rn	
0010	Rn	Rm	01MD	MOV.B	Rm, @–Rn	MOV.W	Rm, @–Rn	MOV.L	Rm, @–Rn	DIV0S
0010	Rn	Rm	10MD	TST	Rm, Rn	AND	Rm, Rn	XOR	Rm, Rn	OR
0010	Rn	Rm	11MD	CMP/STR	Rm, Rn	XTRCT	Rm, Rn	MULU.W	Rm, Rn	MULSW
0011	Rn	Rm	00MD	CMP/EQ	Rm, Rn			CMP/HS	Rm, Rn	CMP/GE

								Rn				Rn			
0100	Rn	00MD	0011	STC.L	SR, @–Rn			STC.L	GBR, @–Rn			STC.L	VBR, @–Rn		STC.L
0100	Rn	01MD	0011	STC.L	SPC, @–Rn										
0100	Rn	10MD	0011	STC.L				STC.L				STC.L			STC.L
				R0_BANK, @–Rn				R1_BANK, @–Rn				R2_BANK, @–Rn			R3_BA
0100	Rn	11MD	0011	STC.L				STC.L				STC.L			STC.L
				R4_BANK, @–Rn				R5_BANK, @–Rn				R6_BANK, @–Rn			R7_BA
0100	Rn	Fx	0100	ROTL	Rn							ROTCL	Rn		
0100	Rn	Fx	0101	ROTR	Rn			CMP/PL	Rn			ROTCR	Rn		
0100	Rm	Fx	0110	LDS.L				LDS.L	@Rm+, MACL			LDS.L	@Rm+, PR		
				@Rm+, MACH											
0100	Rm	00MD	0111	LDC.L	@Rm+, SR			LDC.L	@Rm+, GBR			LDC.L	@Rm+, VBR		LDC.L
0100	Rm	01MD	0111	LDC.L SPC	@Rm+,										
0100	Rm	10MD	0111	LDC.L				LDC.L				LDC.L			LDC.L
				@Rm+, R0_BANK				@Rm+, R1_BANK				@Rm+, R2_BANK			@Rm+
0100	Rm	11MD	0111	LDC.L				LDC.L				LDC.L			LDC.L
				@Rm+, R4_BANK				@Rm+, R5_BANK				@Rm+, R6_BANK			@Rm+
0100	Rn	Fx	1000	SHLL2	Rn			SHLL8	Rn			SHLL16	Rn		
0100	Rn	Fx	1001	SHLR2	Rn			SHLR8	Rn			SHLR16	Rn		
0100	Rm	Fx	1010	LDS MACH	Rm,			LDS MACL	Rm,			LDS	Rm, PR		
0100	Rm/ Rn	Fx	1011	JSR	@Rm			TAS.B	@Rn			JMP	@Rm		

0100	Rn	Rm	1111	MAC.W	Rm+, Rn+								
0101	Rn	Rm	disp	MOV.L	(disp:4, Rm), Rn								
0110	Rn	Rm	00MD	MOV.B	@Rm, Rn	MOV.W	@Rm, Rn	MOV.L	@Rm, Rn	MOV			
0110	Rn	Rm	01MD	MOV.B	@Rm+, Rn	MOV.W	@Rm+, Rn	MOV.L	@Rm+, Rn	NOT			
0110	Rn	Rm	10MD	SWAP.B	Rm, Rn	SWAP.W	Rm, Rn	NEGC	Rm, Rn	NEG			
0110	Rn	Rm	11MD	EXTU.B	Rm, Rn	EXTU.W	Rm, Rn	EXTS.B	Rm, Rn	EXTS.W			
0111	Rn	imm		ADD	# imm : 8, Rn								
1000	00MD	Rn	disp	MOV. B		MOV. W							
				R0, @(disp: 4, Rn)		R0, @(disp: 4, Rn)							
1000	01MD	Rm	disp	MOV.B		MOV.W							
				@(disp:4, Rm), R0		@(disp: 4, Rm), R0							
1000	10MD	imm/disp		CMP/EQ	#imm:8, R0	BT	disp: 8			BF			
1000	11MD	imm/disp				BT/S	disp: 8			BF/S			
1001	Rn	disp		MOV.W	(disp : 8, PC), Rn								
1010	disp			BRA	disp: 12								
1011	disp			BSR	disp: 12								
1100	00MD	imm/disp		MOV.B		MOV.W		MOV.L		TRAPA			
				R0, @(disp: 8, GBR)		R0, @(disp: 8, GBR)		R0, @(disp: 8, GBR)					
1100	01MD	disp		MOV.B		MOV.W		MOV.L		MOVA			
				@(disp: 8, GBR), R0		@(disp: 8, GBR), R0		@(disp: 8, GBR), R0		@(disp:			
1100	10MD	imm		TST	#imm: 8, R0	AND	#imm: 8, R0	XOR	#imm: 8, R0	OR			
1100	11MD	imm		TST.B		AND.B		XOR.B		OR.B			
				#imm: 8, @(R0, GBR)		#imm: 8, @(R0, GBR)		#imm: 8, @(R0, GBR)		#imm: 8			
1101	Rn	disp		MOV.L	@(disp: 8, PC), Rn								
1110	Rn	imm		MOV	#imm:8, Rn								
1111	**********												

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

3.1 Role of MMU

The MMU is a feature designed to make efficient use of physical memory. As shown
if a process is smaller in size than the physical memory, the entire process can be map
physical memory. However, if the process increases in size to the extent that it no long
physical memory, it becomes necessary to partition the process and to map those parts
execution onto memory as occasion demands (figure 3.1 (1)). Having the process itsel
this mapping onto physical memory would impose a large burden on the process. To l
burden, the idea of virtual memory was born as a means of performing en bloc mappi
physical memory (figure 3.1 (2)). In a virtual memory system, substantially more virtu
than physical memory is provided, and the process is mapped onto this virtual memor
process only has to consider operation in virtual memory. Mapping from virtual memo
physical memory is handled by the MMU. The MMU is normally controlled by the op
system, switching physical memory to allow the virtual memory required by a process
mapped onto physical memory in a smooth fashion. Switching of physical memory is
via secondary storage, etc.

The virtual memory system that came into being in this way is particularly effective in
sharing system (TSS) in which a number of processes are running simultaneously (fig
If processes running in a TSS had to take mapping onto virtual memory into considera
running, it would not be possible to increase efficiency. Virtual memory is thus used t
load on the individual processes and so improve efficiency (figure 3.1 (4)). In the virtu
system, virtual memory is allocated to each process. The task of the MMU is to perfor
mapping of these virtual memory areas onto physical memory. It also has a memory p
feature that prevents one process from inadvertently accessing another process's physi

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

Although the functions of the MMU could also be implemented by software alone, the
translation to be performed by software each time a process accesses physical memory

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

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

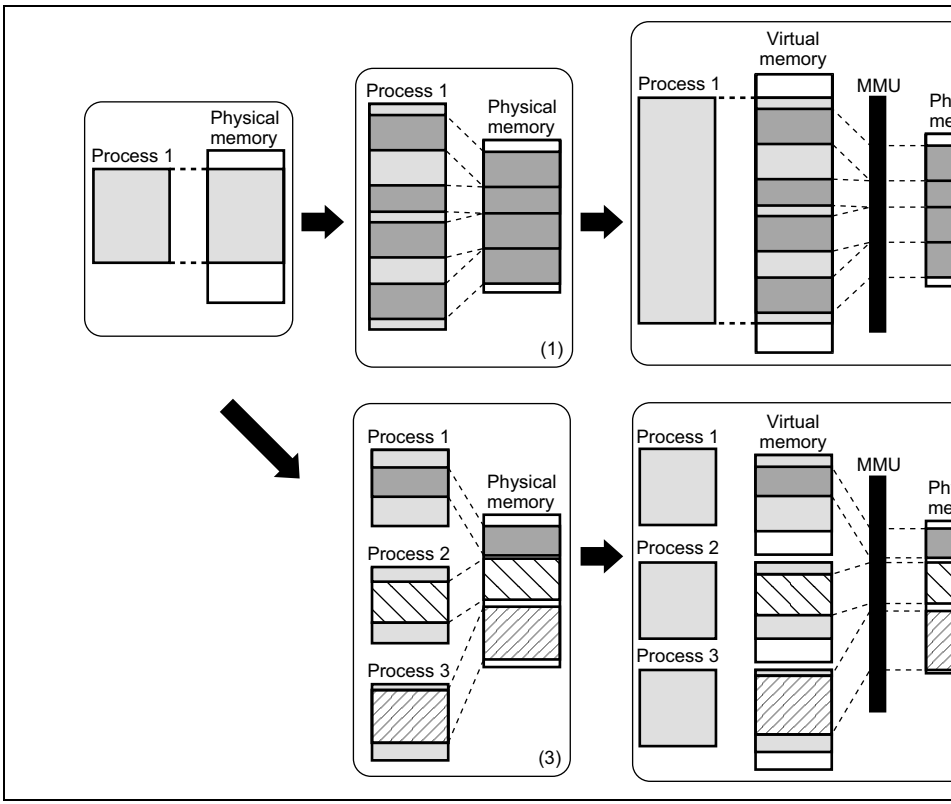


Figure 3.1 MMU Functions

can be used as any physical address area in 1- or 4-kbyte page units. By using an 8-bit space identifier, P0, P2, and U0 areas can be increased to up to 256 areas. Mapping fr address to 29-bit physical address can be achieved by the TLB.

1. P0, P3, and U0 Areas

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

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

2. P1 Area

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

3. P2 Area

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

4. P4 Area

   The P4 area is mapped to the on-chip module control register of this LSI. This area accessed via the cache and cannot be address-translated by the TLB. Figure 3.4 sh configuration of the P4 area.

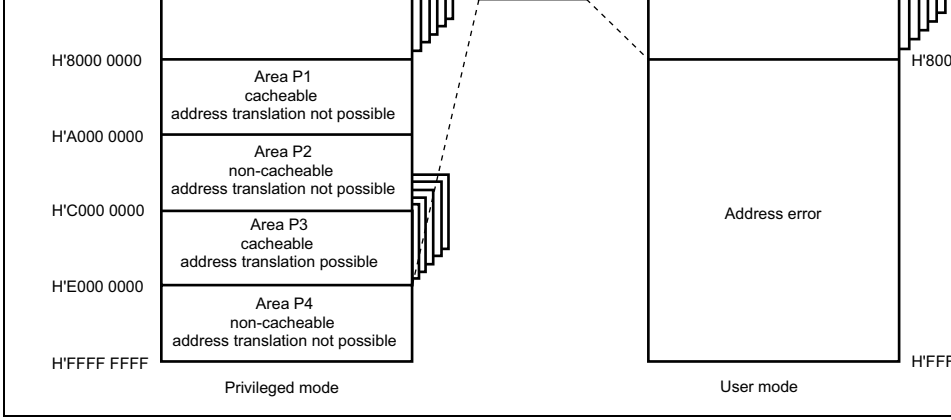


Figure 3.2 Virtual Address Space (MMUCR.AT = 1)

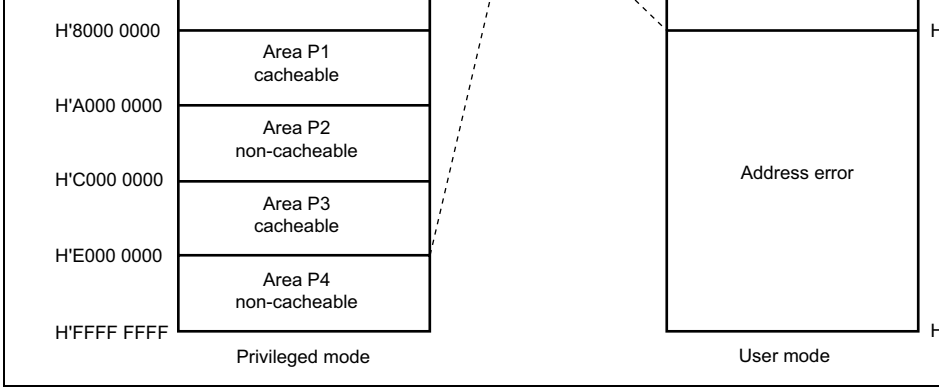


Figure 3.3 Virtual Address Space (MMUCR.AT = 0)

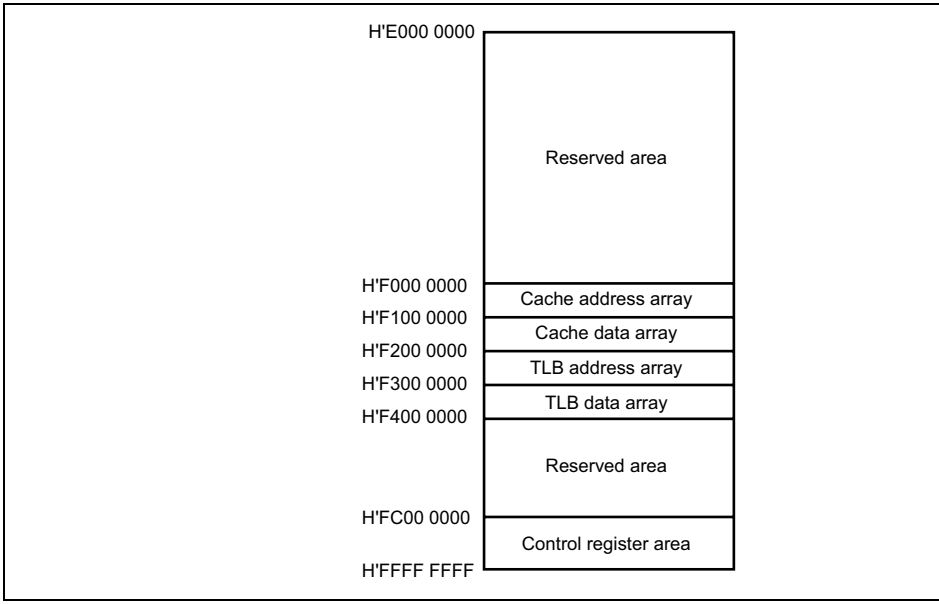


Figure 3.4 P4 Area

information, see section 3.6, Memory-Mapped TLB.

The area from H'FC00 0000 to H'FFFF FFFF is reserved for the on-chip module contro
For more information, see section 24, List of Registers.

Physical Address Space: This LSI supports a 29-bit physical address space. As shown
3.5, the physical address space is divided into eight areas. Area 1 is mapped to the on-c
control register area. Area 7 is reserved.

For details on physical address space, refer to section 7, Bus State Controller (BSC).

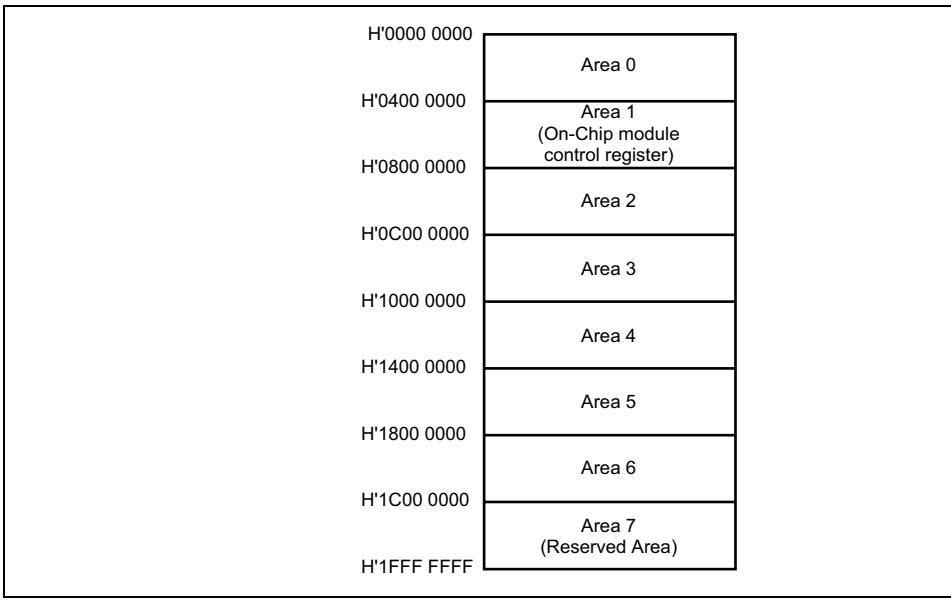


Figure 3.5 External Memory Space

If the virtual address is not registered in the TLB, a TLB miss exception occurs and pr
will shift to the TLB miss handler. In the TLB miss handler, the TLB address translati
external memory is searched and the corresponding physical address and the page con
information are registered in the TLB. After returning from the handler, the instructio
the TLB miss is re-executed. When the MMU is enabled, address translation informat
results in a physical address space of H'20000000 to H'FFFFFFFF should not be regis
TLB.

When the MMU is disabled, masking the upper three bits of the virtual address to 0s c
address in the corresponding physical address space. Since this LSI supports 29-bit ad
as physical address space, the upper three bits of the virtual address are ignored as sha
For details, refer to section 7, Bus State Controller (BSC). For example, address H'000
the P0 area, address H'80001000 in the P1 area, address H'A0001000 in the P2 area, a
H'C0001000 in the P3 area are all mapped to the same physical memory. If these addr
accessed while the cache is enabled, the upper three bits are always cleared to 0 to gua
continuity of addresses stored in the address array of the cache.

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

In terms of operation, the only difference between single virtual memory mode and m
virtual memory mode is in the TLB address comparison method (see section 3.3.3, TL
Comparison).

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

The MMU has the following registers. Refer to section 24, List of Registers, for the add
access size for these registers.

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

3.2.1 Page Table Entry Register High (PTEH)

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

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

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

31 to 29	—	0	R	Reserved These bits are always read as 0. The wri should always be 0.
28 to 10	PPN	—	R/W	Number of Physical Page
9	—	0	R	Page Management Information
8	V	—	R/W	For more details, see section 3.3, TLB F
7	—	0	R	
6, 5	PR	—	R/W	
4	SZ	—	R/W	
3	C	—	R/W	
2	D	—	R/W	
1	SH	—	R/W	
0	—	0	R	

3.2.3 Translation Table Base Register (TTB)

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

3.2.4 MMU Control Register (MMUCR)

The MMU control register (MMUCR) residing at address H'FFFFFFE0, which makes settings described in figure 3.3. Any program that modifies MMUCR should reside in area.

1, 0		0	R	Reserved These bits are always read as 0. The writ should always be 0.
5, 4	RC	0	R/W	Random Counter A 2-bit random counter that is automatical by hardware according to the following ru event of an MMU exception. When a TLB miss exception occurs, all of ways corresponding to the virtual address the exception occurred are checked. If all valid, 1 is added to RC; if there is one or r way, they are set by priority from way 0, i way 0, way 1, way 2, and way 3. In the ev MMU exception other than a TLB miss ex way which caused the exception is set in
3	—	0	R	Reserved These bits are always read as 0. The writ should always be 0.
2	TF	0	R/W	TLB Flush Write 1 to flush the TLB (clear all valid bit to 0). When they are read, 0 is always ret
1	IX	0	R/W	Index Mode 0: VPN bits 16 to 12 are used as the TLB number. 1: The value obtained by EX-ORing ASID in PTEH and VPN bits 16 to 12 is used index number.
0	AT	0	R/W	Address Translation Enables/disables the MMU. 0: MMU disabled 1: MMU enabled

addresses and TLB entries.

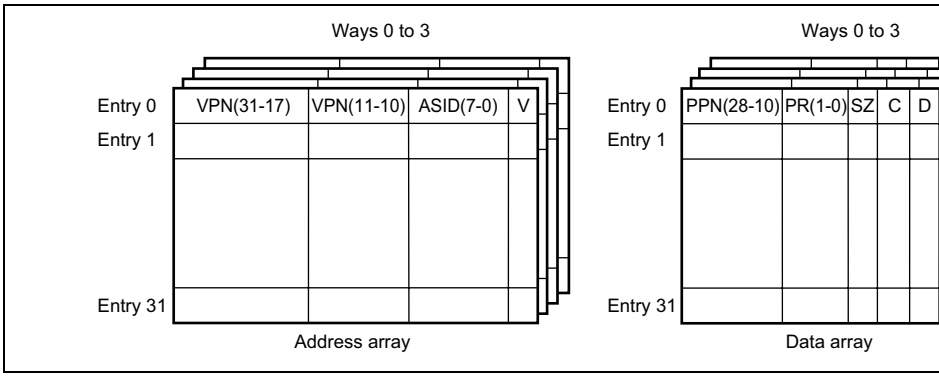
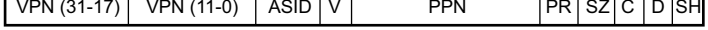


Figure 3.6 Overall Configuration of the TLB



TLB entry

Legend:

VPN: Virtual page number
Upper 22 bits of virtual address for a 1-kbyte page, or upper 20 bits of virtual address for a 4
page. Since VPN bits 16 to 12 are used as the index number, they are not stored in the TLB
Attention must be paid to the synonym problem (see section 3.4.4, Avoiding Synonym Probl

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

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

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

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

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

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

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

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

Figure 3.7 Virtual Address and TLB Structure

The first method is used to prevent lowered TLB efficiency that results when multiple
run simultaneously in the same virtual address space (multiple virtual memory) and a
entry is selected by indexing of each process. In single virtual memory mode (MMUC
IX bit should be set to 0. Figures 3.8 and 3.9 show the indexing schemes.

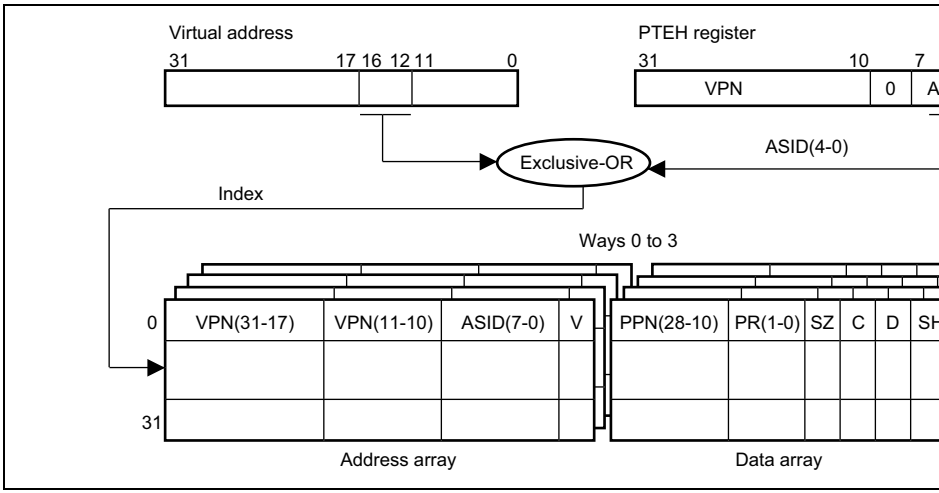


Figure 3.8 TLB Indexing (IX = 1)

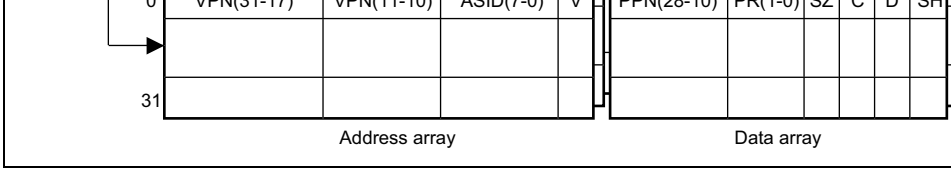


Figure 3.9 TLB Indexing (IX = 0)

3.3.3 TLB Address Comparison

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

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

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

not which there is sharing (SH = 1).

When single virtual memory is supported (MMUCR.SV = 1) and privileged mode is e
(SR.MD = 1), all process resources can be accessed. This means that ASIDs are not co
when single virtual memory is supported and privileged mode is engaged. The objects
comparison are shown in figure 3.10.

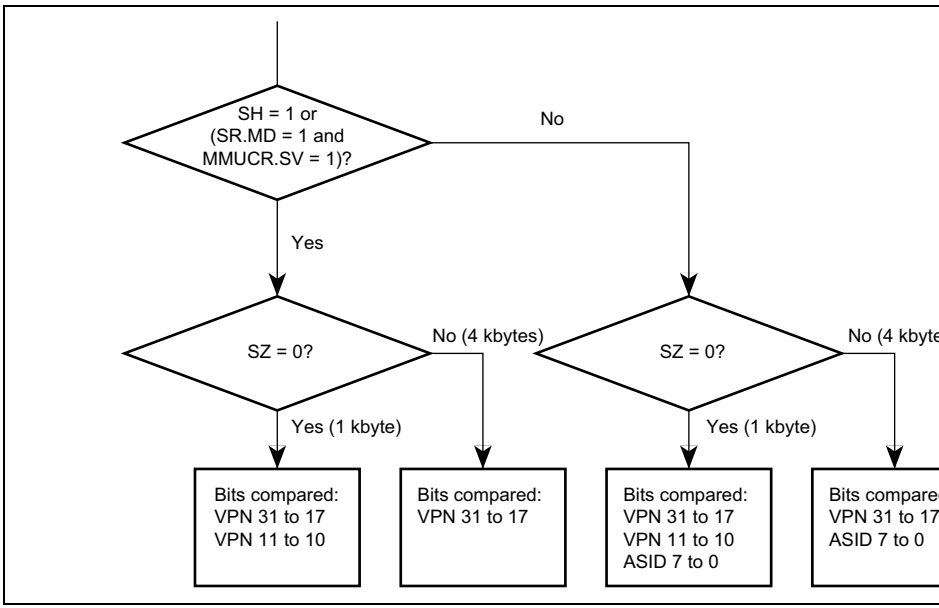


Figure 3.10 Objects of Address Comparison

memory. To record that there has been a write to a given page in the address translation memory, an initial page write exception is used.

The C bit in the entry indicates whether the referenced page resides in a cacheable or no cacheable area of memory. When the on-chip module control registers in area 1 are mapp the C bit to 0. The PR field specifies the access rights for the page in privileged and user and is used to protect memory. Attempts at non-permitted accesses result in TLB protec violation exceptions.

Access states designated by the D, C, and PR bits are shown in table 3.1.

Table 3.1 Access States Designated by D, C, and PR Bits

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

determines the MMU exception and whether the cache is to be accessed (using the
details of the determination method and the hardware processing, see section 3.5, I
Exceptions.

3.4.2 MMU Software Management

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

1. MMU register setting

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

   TLB entry recording can be done in two ways by using the LDTLB instruction, or
   directly to the memory-mapped TLB. For TLB entry deletion and reading, the mem
   allocation TLB can be accessed. See section 3.4.3, MMU Instruction (LDTLB), fo
   the LDTLB instruction, and section 3.6, Memory-Mapped TLB, for details of the m
   mapped TLB.
3. MMU exception processing

   When an MMU exception is generated, it is handled on the basis of information se
   hardware side. See section 3.5, MMU Exceptions, for details.

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

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

When an MMU exception occurs, the virtual page number of the virtual address that ca exception is set in PTEH by hardware. The way is set in the RC bit of MMUCR for eac according to the rules (see section 3.2.4, MMU Control Registers). Consequently, if the instruction is issued after setting only PTEL in the MMU exception processing routine, recording is possible. Any TLB entry can be updated by software rewriting of PTEH ar bits in MMUCR.

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

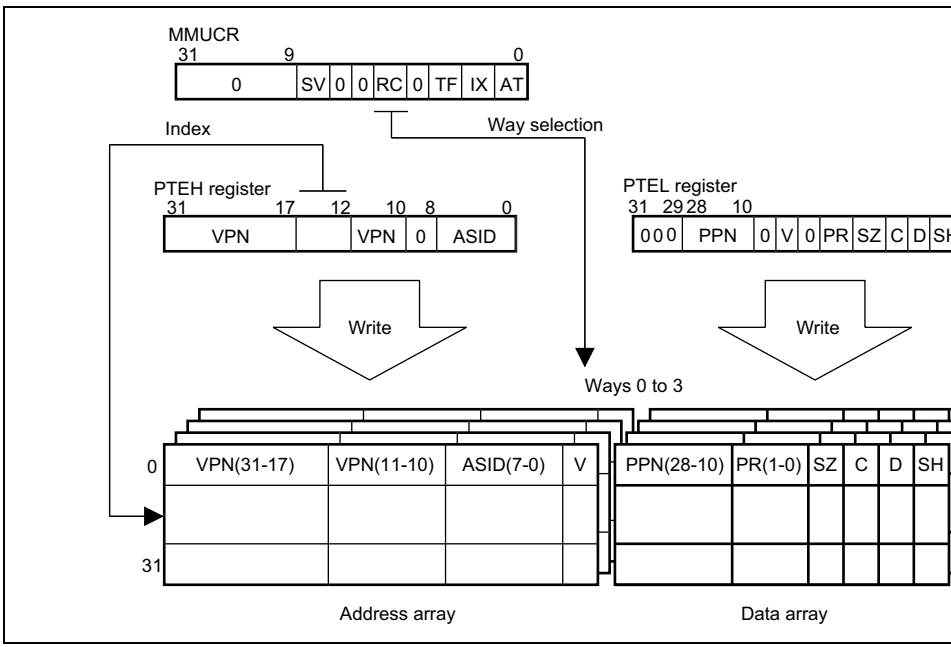


Figure 3.11 Operation of LDTLB Instruction

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

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

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

Virtual address 1 H'00000000 → physical address H'00000C00

Virtual address 2 H'00000C00 → physical address H'00000C00

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

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

1. When address translation information whereby a number of 1-kbyte page TLB ent translated into the same physical address is recorded in the TLB, ensure that the V 10 are the same.
2. When address translation information whereby a number of 4-kbyte page TLB ent translated into the same physical address is recorded in the TLB, ensure that the V the same.
3. Do not use the same physical addresses for address translation information of diffe sizes.

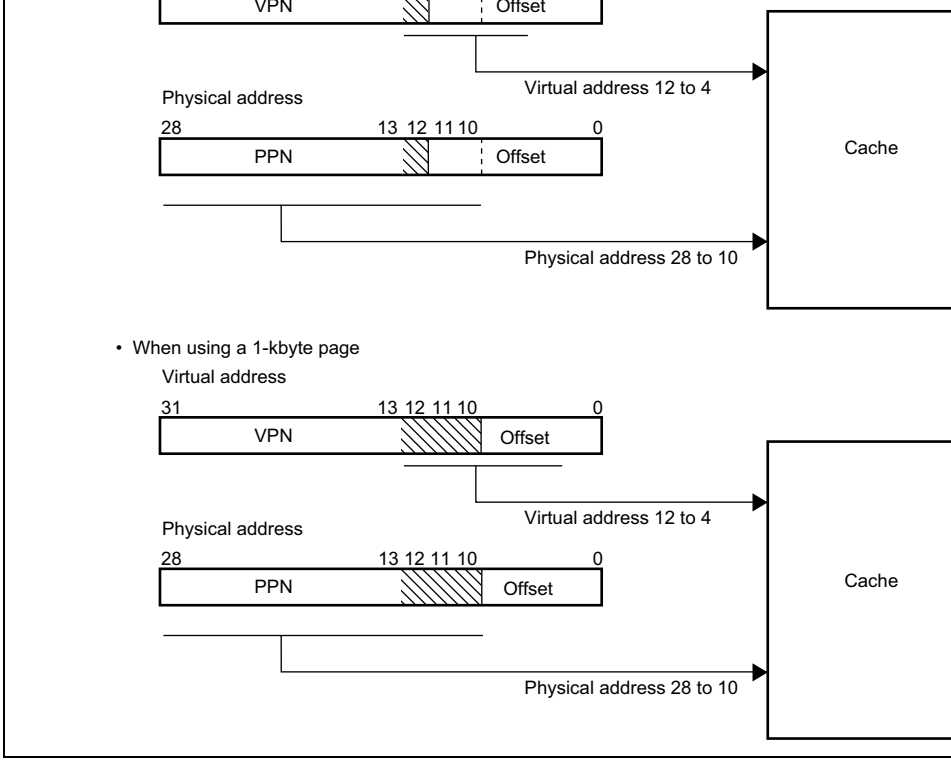


Figure 3.12 Synonym Problem (32-kbyte Cache)

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

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

1. The VPN field of the virtual address causing the exception is written to the PTEH
2. The virtual address causing the exception is written to the TEA register.
3. Either exception code H'040 for a load access, or H'060 for a store access, is writte EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occu written to the save program counter (SPC). If the exception occurred in a delay slo value indicating the address of the related delayed branch instruction is written to t
5. The contents of the status register (SR) at the time of the exception are written to t status register (SSR).
6. The mode (MD) bit in SR is set to 1 to place the privileged mode.
7. The block (BL) bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The RC field in the MMU control register (MMUCR) is incremented by 1 when al indexed are valid. When some entries indexed are invalid, the smallest way numbe set in RC.
10. Execution branches to the address obtained by adding the value of the VBR conten H'00000400 to invoke the user-written TLB miss exception handler.

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

1. Write the value of the physical page number (PPN) field and the protection key (PR (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page tab recorded in the address translation table in the external memory into the PTEL reg

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

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

1. The VPN field of the virtual address causing the exception is written to the PTEH r
2. The virtual address causing the exception is written to the TEA register.
3. Either exception code H'0A0 for a load access, or H'0C0 for a store access, is writte
EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occur
written into SPC (if the exception occurred in a delay slot, the PC value indicating t
of the related delayed branch instruction is written into SPC).
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1 to place the privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The RB bit in SR is set to 1.
9. The way that generated the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR content
H'00000100 to invoke the TLB protection violation exception handler.

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

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

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

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

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

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

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

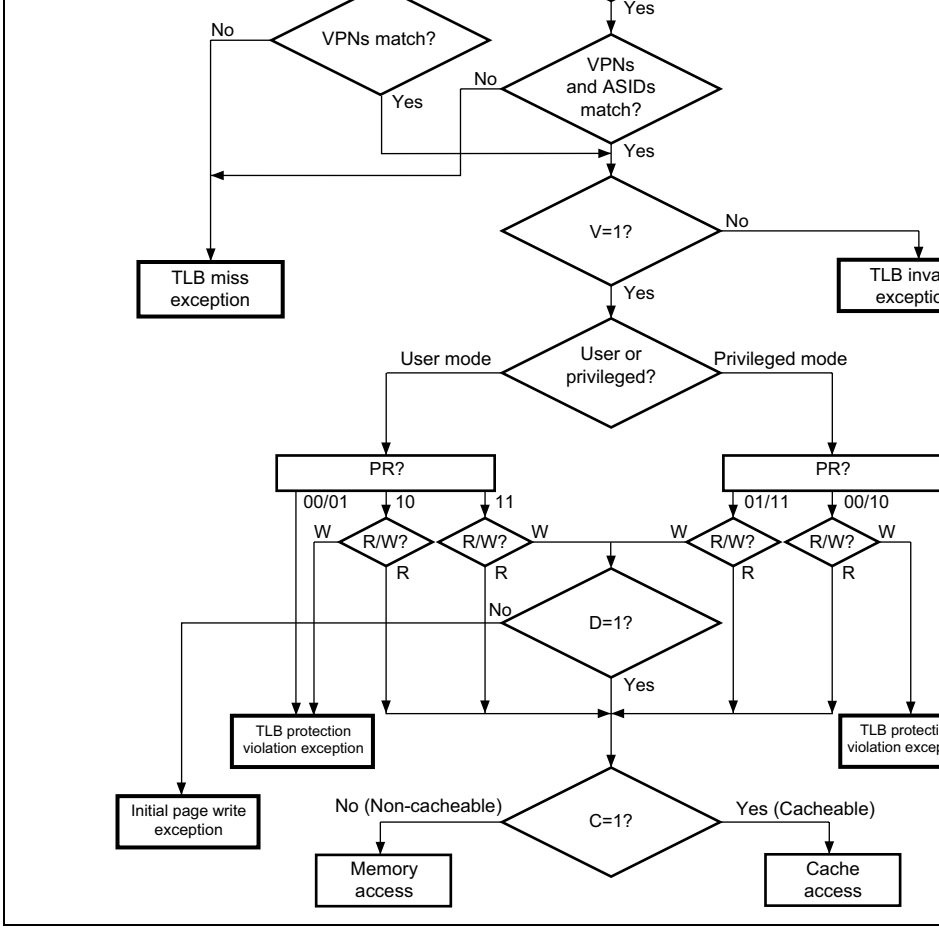


Figure 3.13 MMU Exception Generation Flowchart

data cannot be retained from both arrays.

3.6.1 Address Array

The address array is assigned to H'F2000000 to H'F2FFFFFF. To access an address arra
bit address field (for read/write operations) and 32-bit data field (for write operations) r
specified. The address field specifies information for selecting the entry to be accessed;
field specifies the VPN, V bit and ASID to be written to the address array (figure 3.14 (

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

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

1. Address array read

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

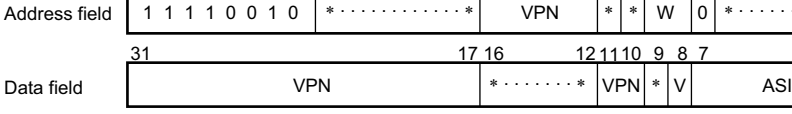
   The data specified in the data field are written to the TLB entry corresponding to th
   address and way set in the address field.

3.6.2 Data Array

The data array is assigned to H'F3000000 to H'F3FFFFFF. To access a data array, the 3
address field (for read/write operations), and 32-bit data field (for write operations) mu
specified. The address section specifies information for selecting the entry to be accesse
section specifies the longword data to be written to the data array (figure 3.14 (2)).

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

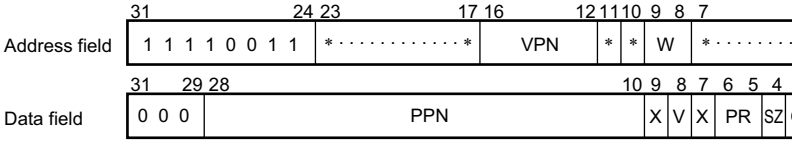
Both reading and writing use the longword of the data array specified by the entry addr
way number. The access size of the data array is fixed at longword.



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

(2) TLB data array access

• Read/write access



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

Figure 3.14 Specifying Address and Data for Memory-Mapped TLB Ac

```
; Index is cleared too,achieving invalidation.
MOV.L  R0,@R1
```

Reading the Data of a Specific Entry: This example reads the data section of a specif entry. The bit order indicated in the data field in figure 3.14 (2) is read. R0 specifies the and the data section of a selected entry is read to R1.

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

3.7 Usage Note

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

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

- Line size: 16 bytes
- Number of entries: 256 entries/way in 16-kbyte mode or 512 entries/way in 32-kby
- Write system: Write-back/write-through is selectable for spaces P0, P1, P3, and U0
  Group 1 (P0, P3, and U0 areas)
  Group 2 (P1 area)
- Replacement method: Least-recently used (LRU) algorithm

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

4.1.1 Cache Structure

The cache mixes instructions and data and uses a 4-way set associative system. It is co four ways (banks), and each of which is divided into an address section and a data sec that the following sections will be described for the 32-kbyte mode as an example. For size modes, change the number of entries and size/way according to table 4.1. Each of and data sections is divided into 512 entries. The entry data is called a line. Each line c 16 bytes (4 bytes × 4). The data capacity per way is 8 kbytes (16 bytes × 512 entries) i as a whole (4 ways). The cache capacity is 32 kbytes as a whole. Figure 4.1 shows the structure.

Table 4.1 Number of Entries and Size/Way in Each Cache Size

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

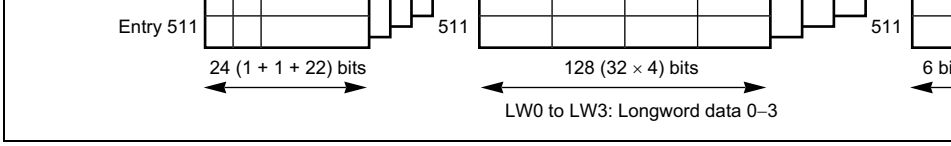


Figure 4.1 Cache Structure (32-kbyte Mode)

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

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

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

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

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

Six LRU bits indicate the way to be replaced, when a cache miss occurs. Table 4.2 show relationship between the LRU bits and the way to be replaced when the cache locking m is disabled. (For the relationship when the cache locking mechanism is enabled, refer to 4.2.2, Cache Control Register 2.) If a bit pattern other than those listed in table 4.2 is se LRU bits by software, the cache will not function correctly. When modifying the LRU software, set one of the patterns listed in table 4.2.

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

4.2 Register Descriptions

The cache has the following registers. For details on register addresses and register sta
each process, refer to section 24, List of Registers.

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

31 to 4	—	0	R	Reserved These bits are always read as 0. The writ should always be 0.
3	CF	0	R/W	Cache Flush Writing 1 flushes all cache entries (clears and LRU bits of all cache entries to 0). Th always read as 0. Write-back to external r not performed when the cache is flushed.
2	CB	0	R/W	Write-Back Indicates the cache's operating mode for 0: Write-through mode 1: Write-back mode
1	WT	0	R/W	Write-Through Indicates the cache's operating mode for U0, and P3. 0: Write-back mode 1: Write-through mode
0	CE	0	R/W	Cache Enable Indicates whether the cache function is us 0: The cache function is not used. 1: The cache function is used.

Table 4.3 shows the relationship between the settings of bits and the way that is to be
when the cache is missed by a prefetch instruction.

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

In cache lock mode, bits W3LOCK and W2LOCK restrict the way that is to be replac
instructions other than the prefetch instruction are issued. Table 4.4 shows the relation
between the settings of bits in CCR2 and the way that is to be replaced when the cach
by instructions other than the prefetch instruction.

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

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

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

1	0	*	1	1	Way 2
1	1	1	0	*	Way 3

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

Table 4.4 Way Replacement when Instructions other than the PREF Instructi Cache

DSP Bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replac
0	*	*	*	*	Determined by LR
1	*	0	*	0	Determined by LR
1	*	0	*	1	Determined by LR
1	*	1	*	0	Determined by LR
1	*	1	*	1	Determined by LR

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

Table 4.5 LRU and Way Replacement (when W2LOCK = 1 and W3LOCK = 0

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

Table 4.6 LRU and Way Replacement (when W2LOCK = 0 and W3LOCK = 1

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

4.2.3 Cache Control Register 3 (CCR3)

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

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

32-kbyte mode as an example.

Entries are selected using bits 12 to 4 of the address (virtual) of the access to memory address of that entry is read. The virtual address (bits 31 to 10) of the access to memo physical address (tag address) read from the address array are compared. The address uses all four ways. When the comparison shows a match and the selected entry is vali cache hit occurs. When the comparison does not show a match or the selected entry is = 0), a cache miss occurs. Figure 4.2 shows a hit on way 1.

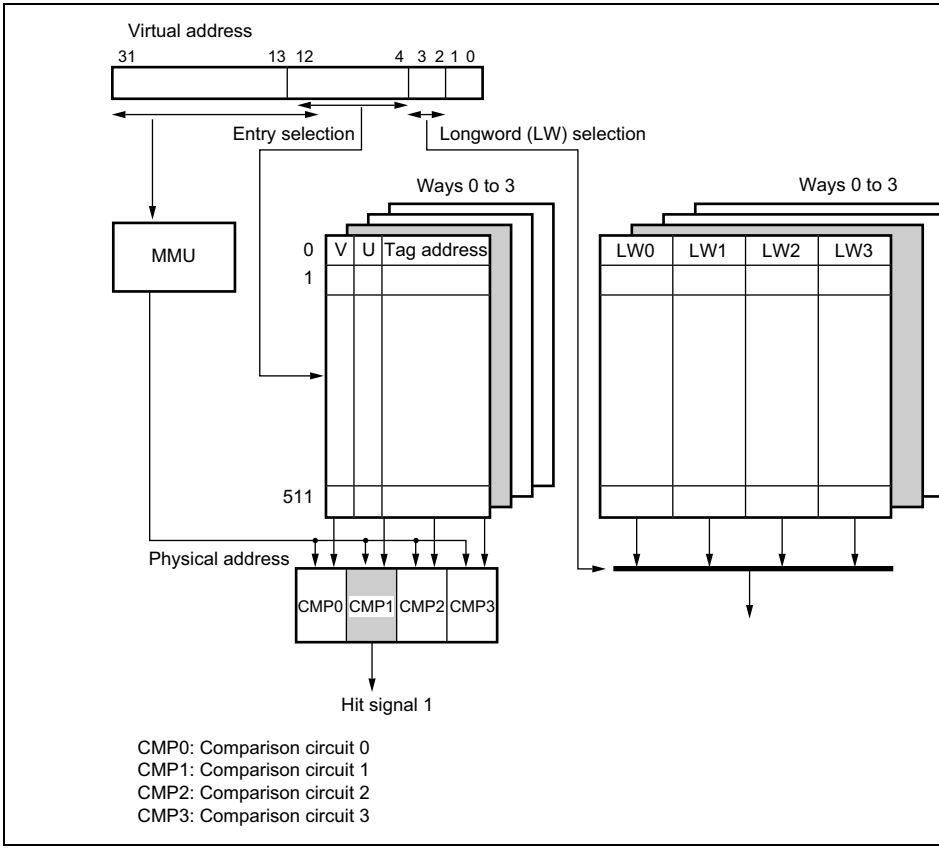


Figure 4.2 Cache Search Scheme

to 0 and the V bit is set to 1 to indicate that the hit way is the most recently hit way. Wr
bit for the entry which is to be replaced by entry updating in write-back mode is 1, the c
update cycle starts after the entry is transferred to the write-back buffer. After the cache
its update cycle, the write-back buffer writes the entry back to the memory. Transfer is
units.

4.3.3 Prefetch Operation

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

Prefetch Miss: Instructions and data are not transferred from the cache to the CPU. Th
is to be replaced is shown in table 4.2. The other operations are the same as those for a

4.3.4 Write Access

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

Write Miss: In write-back mode, an external write cycle starts when a write miss occur
entry is updated. The way to be replaced is shown in table 4.3. When the U bit of the en
is to be replaced by entry updating is 1, the cache-update cycle starts after the entry has
transferred to the write-back buffer. Data is written to the cache and the U bit and the V
to 1. The LRU is updated to indicate that the replaced way is the most recently updated
the cache has completed its update cycle, the write-back buffer writes the entry back to
memory. Transfer is in 16-byte units. In write-through mode, no write to cache occurs i
miss; the write is only to the external memory.

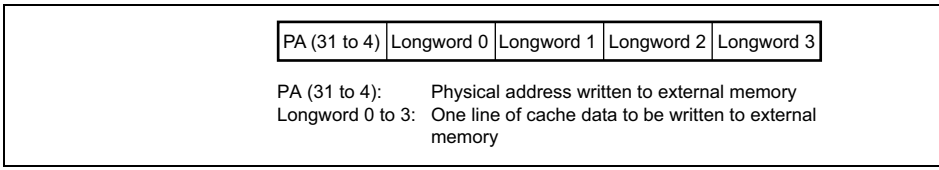


Figure 4.3 Write-Back Buffer Configuration

4.3.6 Coherency of Cache and External Memory

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

4.4.1 Address Array

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

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

In the data field, specify the tag address, LRU bits, U bit, and V bit. Figure 4.4 shows th
and data formats in 32-kbyte mode. The following three operations are available in the
array.

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

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

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

address field (for read/write accesses) and 32-bit data field (for write accesses) must b
The address field specifies information for selecting the entry to be accessed; the data
specifies the longword data to be written to the data array.

In the address field, specify the entry address for selecting the entry, L for indicating t
position within the (16-byte) line, W for selecting the way, and H'F1 for indicating da
access. As for L, 00 indicates longword 0, 01 indicates longword 1, 10 indicates longw
11 indicates longword 3. As for W, 00 indicates way 0, 01 indicates way 1, 10 indicat
and 11 indicates way 3).

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

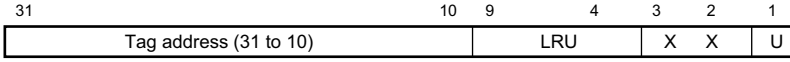
Figure 4.4 shows the address and data formats in 32-kbyte mode. For other cache size
change the entry address and w as shown in table 4.8.

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

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

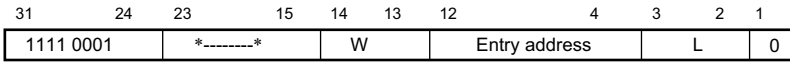
Data-Array Write: Write the longword data specified by the data filed, to the positio
by L of the address field, in the entry that corresponds to the entry address and the wa
by the address field.

(b) Data specification (both read and write accesses)

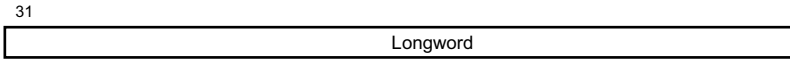


(2) Data array access (both read and write accesses)

(a) Address specification



(b) Data specification



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

Figure 4.4 Specifying Address and Data for Memory-Mapped Cache Acc (32-kbyte Mode)

Table 4.8 Address Format Based on Size of Cache to be Assigned to Memory

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

kbyte mode).

```
; R0 = H'0000 0000 LRU = H'000, U = 0, V = 0
; R1 = H'F000 2080; Way = 1, Entry = B'000001000, A = 0
;
MOV.L R0, @R1
```

To invalidate all entries and ways, write 0 to the following addresses.

32-kbyte mode (2,048 writes)
Addresses

```
F000 0000
F000 0010
F000 0020
    :
F000 7FF0
```

16-kbyte mode (1,024 writes)
Addresses

```
F000 0000
F000 0010
F000 0020
    :
F000 3FF0
```

The above operation should be performed using a non-cacheable area.

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

In the following example, an address (32-bit) to be purged is specified in R0.

```
MOV.L  #H'00001FF0, R1 ; 32-kbyte mode, H'00000FF0 in the 16-kby
AND    R0, R1          ; The entry address is fetched.
MOV.L  #H'00000008, R2 ;
OR     R1, R2          ; The start is set to H'F0 and the A bit
MOV.L  #H'1FFFFC00, R3 ;
AND    R0, R3          ; The tag address is fetched. U = V = 0.
MOV.L  R3, @R2         ; Associative purge.
```

The above operation should be performed using a non-cacheable area.

Reading the Data of a Specific Entry: To read the data field of a specific entry is enal memory-mapped cache access. The longword indicated in the data field of the data arra 4.4 is read into the register. In the example shown below, R0 specifies the address and I what is read (32-kbyte mode).

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

4.5 Usage Note

Do note execute the PREF instruction for the area that cannot be accessed using the cac P4 areas).

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

A reset input can terminate the normal program execution and pass control to the reset
register initialization. This reset operation can also be regarded as an exception handli
section describes an overview of the exception handling operation. Here, general exce
interrupts are referred to as exception handling. For interrupts, this section describes o
process executed for interrupt requests. For details on how to generate an interrupt req
section 6, Interrupt Controller (INTC).

5.1 Register Descriptions

There are five registers for exception handling. A register with an undefined initial val
initialized by the software. Refer to section 24, List of Registers, for the addresses and
of these registers.

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

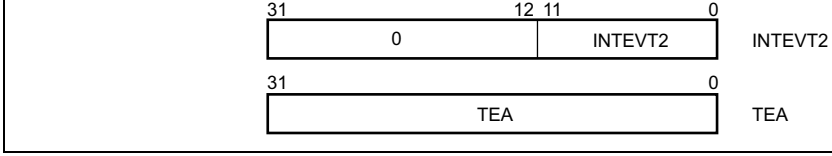


Figure 5.1 Register Bit Configuration

5.1.1 TRAPA Exception Register (TRA)

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

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

				These bits are always read as 0. The should always be 0.
11 to 0	EXPEVT	*	R/W	12-bit Exception Code

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

5.1.3 Interrupt Event Register (INTEVT)

INTEVT is assigned to address H'FFFFFFD8 and consists of a 12-bit exception code. codes to be specified in INTEVT are those for interrupt requests. These exception cod automatically specified by the hardware when an exception occurs. Only bits 11 to 0 c can be re-written using the software.

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

				These bits are always read as 0. The should always be 0.
11 to 0	INTEVT2	—	R	12-bit Exception Code

5.1.5 Exception Address Register (TEA)

TEA is assigned to address H'FFFFFFFC and stores the logical address for an exceptio occurrence when an exception related to memory accesses occurs. TEA can be modifie software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TEA	—	R/W	Logical address for exception occurre

contents of PC and SR to return to the processor state at the point of interruption and t
where the exception occurred.

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

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

The above operations from 1 to 8 are executed in sequence. During these operations, r
exceptions may be accepted unless multiple exception acceptance is enabled.

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

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

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

5.2.3 Exception Codes

The exception codes are written to bits 11 to 0 of the EXPEVT register (for reset or ger exceptions) or the INTEVT2 register (for interrupt requests) to identify each specific ex event. See section 6, Interrupt Controller (INTC), for details of the exception codes for requests. Table 5.1 lists exception codes for resets and general exceptions.

5.2.4 Exception Request and BL Bit (Multiple Exception Prevention)

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

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

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

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

asynchronously. The user cannot expect on which instruction an exception is requeste
general exceptions other than a DMA address error and a user break under a specific c
each general exception corresponds to a specific instruction.

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

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

Acceptance Priority and Test Priority: Acceptance priorities are determined for all
requests. The priority of resets, general exceptions, and interrupts are determined in th
reset is always accepted regardless of the CPU status. Interrupts are accepted only wh
general exceptions are not requested.

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

1. A processing-completion type exception generated at the previous instruction*
2. A user break before instruction execution (re-execution type)
3. An exception related to an instruction fetch (CPU address error and MMU related
   re-execution type)

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

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

Table 5.1 Exception Event Vectors

Exception Type	Current Instruction	Exception Event	Priority*1	Exception Order	Process at BL=1	Vector Code
Reset	Aborted	Power-on reset	1	1	Reset	H'A00
		Manual reset	1	2	Reset	H'020
General exception events	Re-executed	User break(before instruction execution)	2	0	Ignored	H'1E0
		CPU address error (instruction access)	2	1	Reset	H'0E0
		TLB miss *4 (instruction access)	2	1-1	Reset	H'040
		TLB invalid *4 (instruction access)	2	1-2	Reset	H'040
		TLB protection violation *4 (instruction access)	2	1-3	Reset	H'0A0
		Illegal general instruction exception	2	2	Reset	H'180
		Illegal slot instruction exception	2	2	Reset	H'1A0
		CPU address error (data access)	2	3	Reset	H'0E0/ H'100
		TLB miss *4 (data access)	2	3-1	Reset	H'040/ H'060

		instruction)				
		User breakpoint (After instruction execution, address)	2	5	Ignored	H'1E0
General interrupt requests	Completed	User breakpoint (Data break, I-BUS break)	2	5	Ignored	H'1E0
		DMA address error	2	6	Retained	H'5C0
Interrupt requests	Completed	Interrupt requests	3	—*2	Retained	—*3

Notes: 1. Priorities are indicated from high to low, 1 being the highest and 3 the lowe
A reset has the highest priority. An interrupt is accepted only when general
are not requested.

2. For details on priorities in multiple interrupt sources, refer to section 6, Inter
Controller (INTC).

3. If an interrupt is accepted, the exception event register (EXPEVT) is not ch
interrupt source code is specified in interrupt source register 2 (EXPEVT2).
refer to section 6, Interrupt Controller (INTC).

4. These exception codes are valid when the MMU is used.

- Conditions

  Power-on reset is request
- Operations

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

Manual Reset:

- Conditions

  Manual reset is request
- Operations

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

5.3.2 General Exceptions

CPU Address Error:

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

  Instruction synchronous, re-execution type
- Save address

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

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

— When undefined code not in a delay slot is decoded

Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE,

Note: For details on undefined code, refer to section 2.6.2, Operation Code Map. Wh undefined code other than H'F000 to H'FFFF is decoded, operation cannot be

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

Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that ac with LDC/STC are not privileged instructions.

- Types

  Instruction synchronous, re-execution type
- Save address

  An instruction address where an exception occurs
- Exception code

  H'180
- Remarks

  None

Illegal Slot Instruction:

- Conditions
  - — When undefined code in a delay slot is decoded

    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE,
  - — When a privileged instruction in a delay slot is decoded in user mode

    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that ac with LDC/STC are not privileged instructions.
  - — When an instruction that rewrites PC in a delay slot is decoded

    Instructions that rewrite PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S, TRAPA, LDC Rm, SR, LDC.L @Rm+, SR
- Types

  Instruction synchronous, re-execution type

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

User Break Point Trap:

- Conditions
  When a break condition set in the user break controller is satisfied
- Types
  Break (L bus) before instruction execution: Instruction synchronous, re-execution ty
  Operand break (L bus): Instruction synchronous, processing-completion type
  Data break (L bus): Instruction asynchronous, processing-completion type
  I bus break: Instruction asynchronous, processing-completion type
- Save address
  Re-execution type: An address of the instruction where a break occurs (a delayed br
  instruction address if an instruction is assigned to a delay slot)
  Processing-completion type: An address of the instruction following the instruction
  break occurs (a delayed branch instruction address if an instruction is assigned to a
- Exception code
  H'1E0

- Types
  Instruction asynchronous, processing-completion type
- Save address
  An address of the instruction following the instruction where a break occurs (a del
  instruction destination address if an instruction is assigned to a delay slot)
- Exception code
  H'5C0
- Remarks
  An exception occurs when a DMA transfer is executed while an illegal instruction
  described above is specified in the DMAC. Since the DMA transfer is performed
  asynchronously with the CPU instruction operation, an exception is also requested
  asynchronously with the instruction execution. For details on DMAC, refer to sect
  Memory Access Controller (DMAC).

5.3.3 General Exceptions (MMU Exceptions)

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

A vector offset for a TLB error exception is defined as H'00000400 to simplify except
determination. For details on MMU exception operations, refer to section 3, Memory
Unit (MMU).

TLB Miss Exception:

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

The logical address (32 bits) that caused the exception is set in TEA and the MMU
are updated. The vector address of the TLB miss exception is VBR + H'0400. To sp
TLB miss processing, the offset differs from other exceptions.

TLB Invalid Exception:

- Conditions

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

  Instruction synchronous, re-execution type
- Save address

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

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

  An exception occurred during read: H'040

  An exception occurred during write: H'060
- Remarks

  The logical address (32 bits) that caused the exception is set in TEA and the MMU
  are updated.

TLB Protection Exception:

- Conditions

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

  Instruction synchronous, re-execution type
- Save address

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

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

- Conditions

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

  Instruction synchronous, re-execution type
- Save address

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

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

  H'080
- Remarks

  The logical address (32 bits) that caused the exception is set in TEA and the MMU are updated.

execution type exception occurs.

2. In an instruction assigned at a delay slot of the RTE instruction, a user break canno accepted.
3. If the MD and BL bits of the SR register are changed by the LDC instruction, an ex accepted according to the changed SR value from the next instruction.* A process completion type exception is accepted before the next instruction is executed. An i and DMA address error in re-execution type exceptions are accepted before the ne instruction is executed.

Note:* If an LDC instruction is executed for the SR, the following instructions are re-f an instruction fetch exception is accepted according to the modified SR value.

- 16 levels of interrupt priority can be set

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

  An NMI input-level bit indicates the NMI pin state. By reading this bit in the inter
  exception service routine, the pin state can be checked, enabling it to be used as a
  canceller.
- IRQ interrupts can be set

  Detection of low level, high level, rising edge, or falling edge

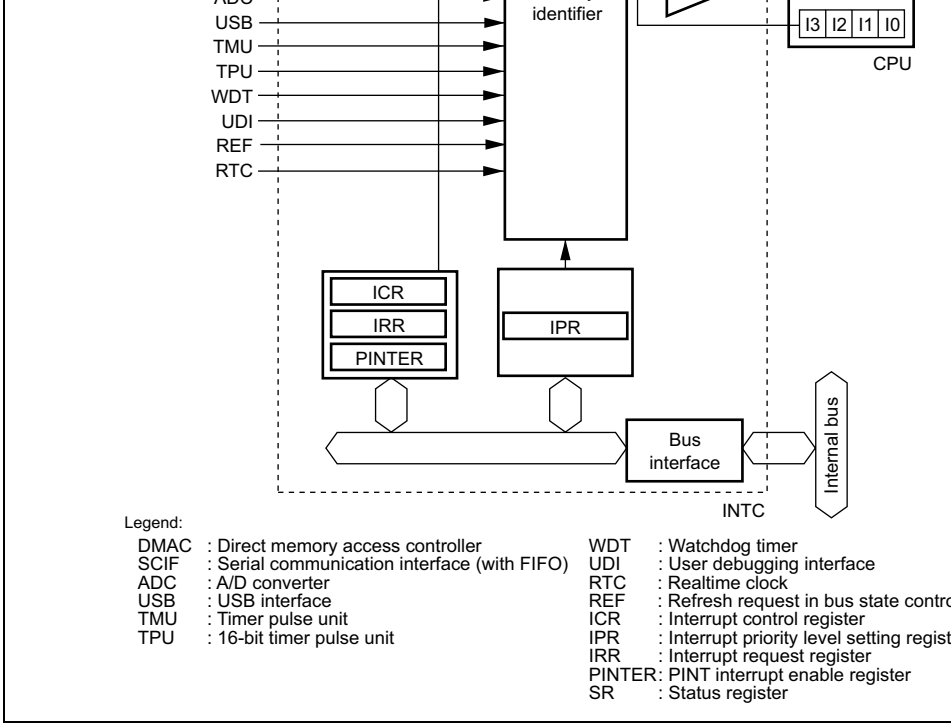


Figure 6.1 Block Diagram of INTC

Interrupt input pins	IRQ5 to IRQ0, $\overline{IRL3}$ to $\overline{IRL0}$	Input	Input of interrupt reques
Port interrupt input pins	PINT15 to PINT0	Input	Input of port interrupt sig

Note: $\overline{IRL3}$ to $\overline{IRL0}$ are multiplexed with IRQ3 to IRQ0; they cannot be used together IRQ0 at the same time.

6.3 Register Descriptions

The INTC has the following registers. For details on register addresses and register sta each processing, refer to section 24, List of Registers.

- Interrupt control register 0 (ICR0)
- Interrupt control register 1 (ICR1)
- Interrupt control register 2 (ICR2)
- PINT interrupt enable register (PINTER)
- Interrupt priority level setting register A (IPRA)
- Interrupt priority level setting register B (IPRB)
- Interrupt priority level setting register C (IPRC)
- Interrupt priority level setting register D (IPRD)
- Interrupt priority level setting register E (IPRE)
- Interrupt priority level setting register F (IPRF)
- Interrupt priority level setting register G (IPRG)
- Interrupt priority level setting register H (IPRH)
- Interrupt request register 0 (IRR0)
- Interrupt request register 1 (IRR1)
- Interrupt request register 2 (IRR2)

Table 6.2 Interrupt Sources and IPRA to IPRH

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 t
IPRA	TMU0	TMU1	TMU2	RTC
IPRB	WDT	REF	Reserved*	Reserve
IPRC	IRQ3	IRQ2	IRQ1	IRQ0
IPRD	PINT0 to PINT7	PINT8 to PINT15	IRQ5	IRQ4
IPRE	DMAC	SCIF0	SCIF2	ADC
IPRF	Reserved*	Reserved*	USB	Reserve
IPRG	TPU0	TPU1	Reserved*	Reserve
IPRH	TPU2	TPU3	Reserved*	Reserve

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

As shown in table 6.2, on-chip peripheral module, or IRQ or PINT interrupts are assign
4-bit groups in each register. These 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, a
to 0) are set with values from H'0 (0000) to H'F (1111). Setting H'0 means priority leve
(masking is requested); H'F means priority level 15 (the highest level).

				pin level. This bit cannot be modified 0: NMI input level is low 1: NMI input level is high
14 to 9	—	0	R	Reserved These bits are always read as 0. The should always be 0.
8	NMIE	0	R/W	NMI Edge Select Selects whether the falling or rising e interrupt request signal at the NMI pi detected. 0: Interrupt request is detected on fa NMI input 1: Interrupt request is detected on ris NMI input
7 to 0	—	0	R	Reserved These bits are always read as 0. The should always be 0.

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

				NMI interrupts in standby mode. 0: All interrupt requests are not maske low level is being input to the NMI p 1: All interrupt requests are masked w level is being input to the NMI pin
14	IRQLVL	1	R/W	Interrupt Request Level Detect Selects whether the IRQ3 to IRQ0 pin enabled or disabled to be used as fou independent interrupt pins. This bit do affect the IRQ4 and IRQ5 pins. 0: Used as four independent interrupt pins IRQ3 to IRQ0 1: Used as encoded 15-level interrupt $\overline{IRL3}$ to $\overline{IRL0}$
13	BLMSK	0	R/W	BL Bit Mask Specifies whether NMI interrupts are r when the BL bit of the SR register is 1 0: NMI interrupts are masked when th 1: NMI interrupts are accepted regardl BL bit setting
12	—	0	R	Reserved This bit is always read as 0. The write should always be 0.

		falling edge
0	1	An interrupt re detected at IR rising edge
1	0	An interrupt re detected at IR low level
1	1	An interrupt re detected at IR high level

[Legend] n = 0 to 5

				high levels. PINTnS 0: Interrupt requests are detected at lo input to the PINT pins 1: Interrupt requests are detected at h input to the PINT pins [Legend] n = 0 to 15

6.3.5 PINT Interrupt Enable Register (PINTER)

PINTER is a 16-bit register that enables interrupt requests input to external interrupt inp PINT0 to PINT15.

When all or some of these pins, PINT0 to PINT15 are not used as an interrupt input, a b corresponding to a pin unused as an interrupt request should be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PINT15E to PINT0E	0	R/W	PINT15 to PINT0 Interrupt Enable Select whether the interrupt requests i PINT15 to PINT0 pins is enabled. PINTnE 0: Disables PINT input interrupt reque 1: Enables PINT input interrupt reques [Legend] n = 0 to 15

				0: Interrupt requests are not input to PINT7 pins 1: Interrupt requests are input to PIN pins
6	PINT1R	0	R	PINT8 to PINT15 Interrupt Request Indicates whether interrupt requests a PINT8 to PINT15 pins. 0: Interrupt requests are not input to PINT15 pins 1: Interrupt requests are input to PIN PINT15 pins
5 to 0	IRQ5R to IRQ0R	0	R/W	IRQn Interrupt Request Indicates whether there is interrupt re to the IRQn pin. When edge-detectio set for IRQn, an interrupt request is c writing 0 to the IRQnR bit after readir 1. When level-detection mode is set for bits indicate whether an interrupt requ The interrupt request is set/cleared b input to the IRQn pin. IRQnR 0: No interrupt request input to IRQn 1: Interrupt request input to IRQn pin [Legend] n = 0 to 5

				1: A TXI0 interrupt request is generated
6	—	0	R	Reserved This bit is always read as 0.
5	RXI0R	0	R	RXI0 Interrupt Request Indicates whether an RXI0 (SCIF0) interrupt request i generated. 0: An RXI0 interrupt request is not generated 1: An RXI0 interrupt request is generated
4	ERI0R	0	R	ERI0 Interrupt Request Indicates whether an ERI0 (SCIF0) interrupt request i generated. 0: An ERI0 interrupt request is not generated 1: An ERI0 interrupt request is generated
3	DEI3R	0	R	DEI3 Interrupt Request Indicates whether a DEI3 (DMAC) interrupt request is 0: A DEI3 interrupt request is not generated 1: A DEI3 interrupt request is generated
2	DEI2R	0	R	DEI2 Interrupt Request Indicates whether a DEI2 (DMAC) interrupt request is 0: A DEI2 interrupt request is not generated 1: A DEI2 interrupt request is generated
1	DEI1R	0	R	DEI1 Interrupt Request Indicates whether a DEI1 (DMAC) interrupt request is 0: A DEI1 interrupt request is not generated 1: A DEI1 interrupt request is generated

6.5.6 Interrupt Request Register 2 (IRR2)

IRR2 is an 8-bit register that indicates whether SCIF2 or ADC interrupt requests are g

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	0	R	Reserved These bits are always read as 0.
4	ADIR	0	R	ADI Interrupt Request Indicates whether an ADI (ADC) interrupt request is 0: An ADI interrupt request is not generated 1: An ADI interrupt request is generated
3	TXI2R	0	R	TXI2 Interrupt Request Indicates whether a TXI2 (SCIF2) interrupt request i 0: A TXI2 interrupt request is not generated 1: A TXI2 interrupt request is generated
2	—	0	R	Reserved This bit is always read as 0.
1	RXI2R	0	R	RXI2 Interrupt Request Indicates whether an RXI2 (SCIF2) interrupt request generated. 0: An RXI2 interrupt request is not generated 1: An RXI2 interrupt request is generated
0	ERI2R	0	R	ERI2 Interrupt Request Indicates whether an ERI2 (SCIF2) interrupt request generated. 0: An ERI2 interrupt request is not generated 1: An ERI2 interrupt request is generated

The NMI interrupt has the highest priority level of 16. When the BLMSK bit in the inte
control register 1 (ICR1) is 1 or the BL bit in the status register (SR) is 0, NMI interrup
accepted. NMI interrupt is edge-detected. In sleep or standby mode, the interrupt is acc
regardless of the BL setting. The NMI edge select bit (NMIE) in the interrupt control re
(ICR0) is used to select either rising or falling edge detection.

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

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

6.4.2 IRQ Interrupts

IRQ interrupts are input by level or edge from pins IRQ0 to IRQ5. The priority level ca
interrupt priority registers C and D (IPRC and IPRD) in a range from 0 to 15.

When using edge-sensing for IRQ interrupts, clear the interrupt source by having softw
from the corresponding bit in IRR0, then write 0 to the bit.

When ICR1 is rewritten, IRQ interrupts may be mistakenly detected, depending on the
states. To prevent this, rewrite the register while interrupts are masked, then release the
clearing the illegal interrupt by writing 0 to interrupt request register 0 (IRR0).

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

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

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by IRQ inte
handling. IRQ interrupts can wake the chip up from the standby state when the relevant
level is higher than I3 to I0 in SR (but only when the RTC is used, the clock for the RT
wake the chip up from the standby state).

sampled at every peripheral clock remain unchanged for two consecutive cycles, so th
transient level on the $\overline{\text{IRL}}$ pin change is detected. In standby mode, as the peripheral c
stopped, noise cancellation is performed using the clock for the RTC instead. Therefo
RTC is not used, recovery from standby mode by means of IRL interrupts cannot be p
standby mode.

The priority level of the IRL interrupt must not be lowered unless the interrupt is acce
interrupt processing starts. However, the priority level can be changed to a higher one

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by $\overline{\text{IRL}}$ int
processing.

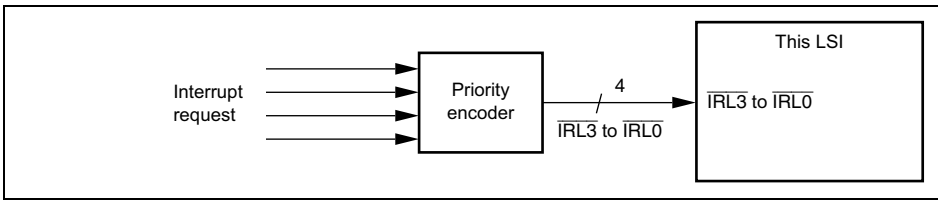


Figure 6.2 Example of IRL Interrupt Connection

0	1	0	1	10	Level 10 interrupt
0	1	1	0	9	Level 9 interrupt r
0	1	1	1	8	Level 8 interrupt r
1	0	0	0	7	Level 7 interrupt r
1	0	0	1	6	Level 6 interrupt r
1	0	1	0	5	Level 5 interrupt r
1	0	1	1	4	Level 4 interrupt r
1	1	0	0	3	Level 3 interrupt r
1	1	0	1	2	Level 2 interrupt r
1	1	1	0	1	Level 1 interrupt r
1	1	1	1	0	No interrupt reque

6.4.4 PINT Interrupt

PINT interrupts are input from pins PINT0 to PINT15 with a level. The priority level ca the interrupt priority level setting register D (IPRD) in a range from levels 0 to 15, in th PINT0 to PINT7 or PINT8 to PINT15. The PINT interrupt level should be held until th is accepted and interrupt handling is started.

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by PINT in processing. PINT interrupts can wake the chip up from the standby state when the relev interrupt level is higher than I3 to I0 in SR (but only when the RTC is used, the clock fo is used to wake the chip up from the standby state).

6.4.5 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following 10 modules:

- Direct memory access controller (DMAC)
- Serial communication interfaces (SCIF0 and SCIF2)
- A/D converter (ADC)

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

A priority level (from 0 to 15) can be set for each module except UDI by writing to th
priority level setting registers A to H (IPRA to IPRH). The priority level of the UDI in
(fixed).

The interrupt mask bits (I3 to I0) in the status register are not affected by on-chip peri
module interrupt handling.

6.4.6 Interrupt Exception Handling and Priority

There are five types of interrupt sources: NMI, IRQ, IRL, PINT, and on-chip peripher
The priority of each interrupt source is set within priority levels 0 to 16; level 16 is the
level 1 is the lowest. When the priority is set to level 0, that interrupt is masked and th
request is ignored.

Tables 6.4 and 6.5 list the codes for the interrupt source and the interrupt event registe
and INTEVT2) and the order of interrupt priority.

Each interrupt source is assigned a unique code by INTEVT and INTEVT2. The start
the interrupt service routine is common for each interrupt source. This is why, for inst
value of INTEVT2 is used as an offset at the start of the interrupt service routine and l
in order to identify the interrupt source.

IRQ and PINT interrupts, and on-chip peripheral module interrupt priorities can be set
between 0 and 15 for each module by setting the interrupt priority level setting registe
assigns priority level 0 to IRQ, PINT, and on-chip peripheral module interrupts.

If the same priority level is assigned to two or more interrupt sources and interrupts fr
sources occur simultaneously, their priority order is the default priority order indicated
in tables 6.4 and 6.5.

	IRQ2	H'640*3	0 to 15 (0)	IPRC (11 to 8)	—
	IRQ3	H'660*3	0 to 15 (0)	IPRC (15 to 12)	—
	IRQ4	H'680*3	0 to 15 (0)	IPRD (3 to 0)	—
	IRQ5	H'6A0*3	0 to 15 (0)	IPRD (7 to 4)	—
PINT	PINT0 to PINT7	H'700*3	0 to 15 (0)	IPRD (15 to 12)	—
	PINT8 to PINT15	H'720*3	0 to 15 (0)	IPRD (11 to 8)	—
DMAC	DEI0	H'800*3	0 to 15 (0)	IPRE (15 to 12)	High
	DEI1	H'820*3			
	DEI2	H'840*3			
	DEI3	H'860*3			Low
SCIF0	ERI0	H'880*3	0 to 15 (0)	IPRE (11 to 8)	High
	RXI0	H'8A0*3			
	TXI0	H'8E0*3			Low
SCIF2	ERI2	H'900*3	0 to 15 (0)	IPRE (7 to 4)	High
	RXI2	H'920*3			
	TXI2	H'960*3			Low
ADC	ADI	H'980*3	0 to 15 (0)	IPRE (3 to 0)	—
USB	USI0	H'A20*3	0 to 15 (0)	IPRF (7 to 4)	High
	USI1	H'A40*3			Low
TPU0	TPI0	H'C00*3	0 to 15 (0)	IPRG (15 to 12)	—
TPU1	TPI1	H'C20*3	0 to 15 (0)	IPRG (11 to 8)	—
TPU2	TPI2	H'C80*3	0 to 15 (0)	IPRH (15 to 12)	—
TPU3	TPI3	H'CA0*3	0 to 15 (0)	IPRH (11 to 8)	—

	PRI	H'4A0*2			
	CUI	H'4C0*2			Low
WDT	ITI	H'560*2	0 to 15 (0)	IPRB (15 to 12)	—
REF	RCMI	H'580*2	0 to 15 (0)	IPRB (11 to 8)	—

Notes: 1. The INTEVT2 code.
2. The same code as INTEVT2 is set in INTEVT.
3. The code indicating an interrupt level (H'200 to H'3C0 shown in table 6.6) is
INTEVT.

	$\overline{IRL}$(3:0) = 0010	H'240*3	13	—	—
	$\overline{IRL}$(3:0) = 0011	H'260*3	12	—	—
	$\overline{IRL}$(3:0) = 0100	H'280*3	11	—	—
	$\overline{IRL}$(3:0) = 0101	H'2A0*3	10	—	—
	$\overline{IRL}$(3:0) = 0110	H'2C0*3	9	—	—
	$\overline{IRL}$(3:0) = 0111	H'2E0*3	8	—	—
	$\overline{IRL}$(3:0) = 1000	H'300*3	7	—	—
	$\overline{IRL}$(3:0) = 1001	H'320*3	6	—	—
	$\overline{IRL}$(3:0) = 1010	H'340*3	5	—	—
	$\overline{IRL}$(3:0) = 1011	H'360*3	4	—	—
	$\overline{IRL}$(3:0) = 1100	H'380*3	3	—	—
	$\overline{IRL}$(3:0) = 1101	H'3A0*3	2	—	—
	$\overline{IRL}$(3:0) = 1110	H'3C0*3	1	—	—
IRQ	IRQ4	H'680*3	0 to 15 (0)	IPRD (3 to 0)	—
	IRQ5	H'6A0*3	0 to 15 (0)	IPRD (7 to 4)	—
PINT	PINT0 to PINT 7	H'700*3	0 to 15 (0)	IPRD (15 to 12)	—
	PINT8 to PINT 15	H'720*3	0 to 15 (0)	IPRD (11 to 8)	—
DMAC	DEI0	H'800*3	0 to 15 (0)	IPRE (15 to 12)	High
	DEI1	H'820*3			
	DEI2	H'840*3			
	DEI3	H'860*3			Low
SCIF0	ERI0	H'880*3	0 to 15 (0)	IPRE (11 to 8)	High
	RXI0	H'8A0*3			
	TXI0	H'8E0*3			Low

	USI1	H'A40*3			Low
TPU0	TPI0	H'C00*3	0 to 15 (0)	IPRG (15 to 12)	—
TPU1	TPI1	H'C20*3	0 to 15 (0)	IPRG (11 to 8)	—
TPU2	TPI2	H'C80*3	0 to 15 (0)	IPRH (15 to 12)	—
TPU3	TPI3	H'CA0*3	0 to 15 (0)	IPRH (11 to 8)	—
TMU0	TUNI0	H'400*2	0 to 15 (0)	IPRA (15 to 12)	—
TMU1	TUNI1	H'420*2	0 to 15 (0)	IPRA (11 to 8)	—
TMU2	TUNI2	H'440*2	0 to 15 (0)	IPRA (7 to 4)	High
	TICPI2	H'460*2			Low
RTC	ATI	H'480*2	0 to 15 (0)	IPRA (3 to 0)	High
	PRI	H'4A0*2			↕
	CUI	H'4C0*2			Low
WDT	ITI	H'560*2	0 to 15 (0)	IPRB (15 to 12)	—
REF	RCMI	H'580*2	0 to 15 (0)	IPRB (11 to 8)	—

Notes: 1. The INTEVT2 code.
2. The same code as INTEVT2 is set in INTEVT.
3. The code indicating an interrupt level (H'200 to H'3C0 shown in table 6.6) is INTEVT.

10	H'2A0
9	H'2C0
8	H'2E0
7	H'300
6	H'320
5	H'340
4	H'360
3	H'380
2	H'3A0
1	H'3C0

6.5 Operation

6.5.1 Interrupt Sequence

The sequence of interrupt operations is described below. Figure 6.3 is a flowchart of th operations.

1. The interrupt request sources send interrupt request signals to the interrupt controlle
2. The interrupt controller selects the highest-priority interrupt from the interrupt requ following the priority levels set in the interrupt priority level setting registers A to H IPRH). Lower priority interrupts are held pending. If two of these interrupts have th priority level or if multiple interrupts occur within a single module, the interrupt wi highest priority is selected, according to tables 6.4 and 6.5.
3. The priority level of the interrupt selected by the interrupt controller is compared wi interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the request pr is higher than the level in bits I3 to I0, the interrupt controller accepts the interrupt a an interrupt request signal to the CPU.
4. Detection timing: The INTC operates, and notifies the CPU of interrupt requests, in synchronization with the peripheral clock (Pϕ). The CPU receives an interrupt at a b instructions.

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

2. The interrupt source flag should be cleared in the interrupt handler. To ens
interrupt request that should have been cleared is not inadvertently accepte
the interrupt source flag after it has been cleared, and then execute an RTE

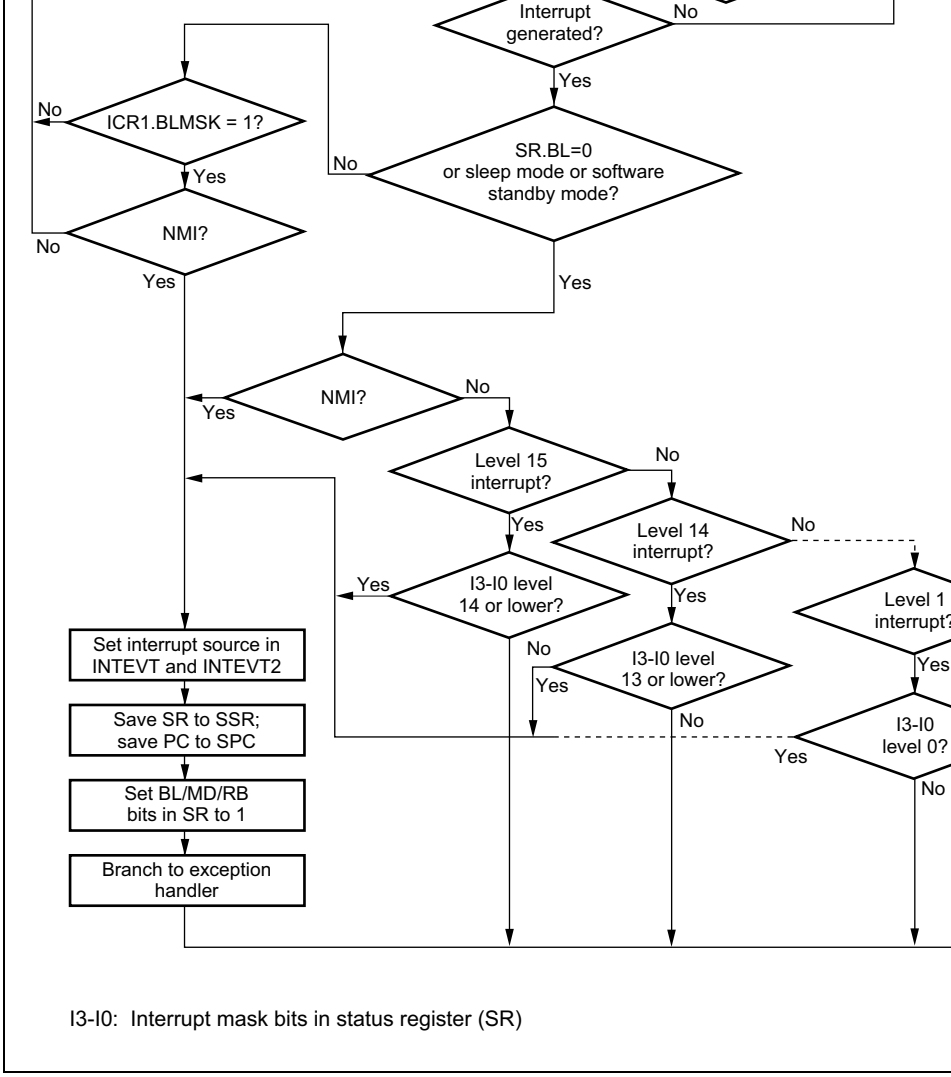


Figure 6.3 Interrupt Operation Flowchart

3. Save SSR and SPC to memory.
4. Clear the BL bit in SR, and set the accepted interrupt level in the interrupt mask bi
5. Handle the interrupt.
6. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the o handled can be accepted after clearing BL in step 4. Figure 6.3 shows a sample interru flowchart.

6.6 Usage Note

The interrupt accept timing in this LSI is not acknowledged externally. Thus, keep the note in mind when designing the system.

- Level interrupt

  The level interrupt request should be held until the CPU accepts it. The level interr needs to be cleared (released) within the specific interrupt handler. If the level inte is not held, the operation may branch to the interrupt handling routine when the va INTEVT/2 becomes H'000.

  When the standby state is cancelled, if the level interrupt request is not held, the op go back to the standby state again in the middle of WDT counting. When cancelin standby state again in such a condition by asserting the level interrupt request, the for the PLL or crystal oscillator is not secured enough and the operation may not r the standby state correctly.
- Interrupt flag update

  When an interrupt is acceptable and the generation of an interrupt request is enable or clearing the interrupt flag may branch the operation to the interrupt handling rou the value in INTEVT2 becomes H'000.

7.1.1 Features

The BSC has the following features:

- Physical address space is divided into eight areas
  - A maximum 32 or 64 Mbytes for each of the eight areas, CS0, CS2 to CS4, CS
    CS6A and CS6B, totally 384 Mbytes.
  - Can specify the normal space interface, byte-selection SRAM interface, burst I
    address/data multiplex I/O (MPX), or SDRAM for each address space.
  - Can select the data bus width (8, 16, or 32 bits) for each address space.
  - Controls the insertion of the wait state for each address space.
  - Controls the insertion of the wait state for each read access and write access.
  - Can set the independent idling cycle in the continuous access for five cases: re
    same space/different space), read-read (in same space/different space), the first
    write access.
- Normal space interface
  - Supports the interface that can directly connect to the SRAM.
- Burst ROM interface
  - High-speed access to the ROM, such as flash memory, that has the page mode
- Address/data multiplex I/O (MPX) interface
  - Can directly connect to a peripheral LSI that needs an address/data multiplexin
- SDRAM interface
  - Can set the SDRAM up to 2 areas.
  - Multiplex output for row address/column address.
  - Efficient access by single read/single write.
  - High-speed access by the bank-active mode.
  - Supports an auto-refresh and self-refresh.

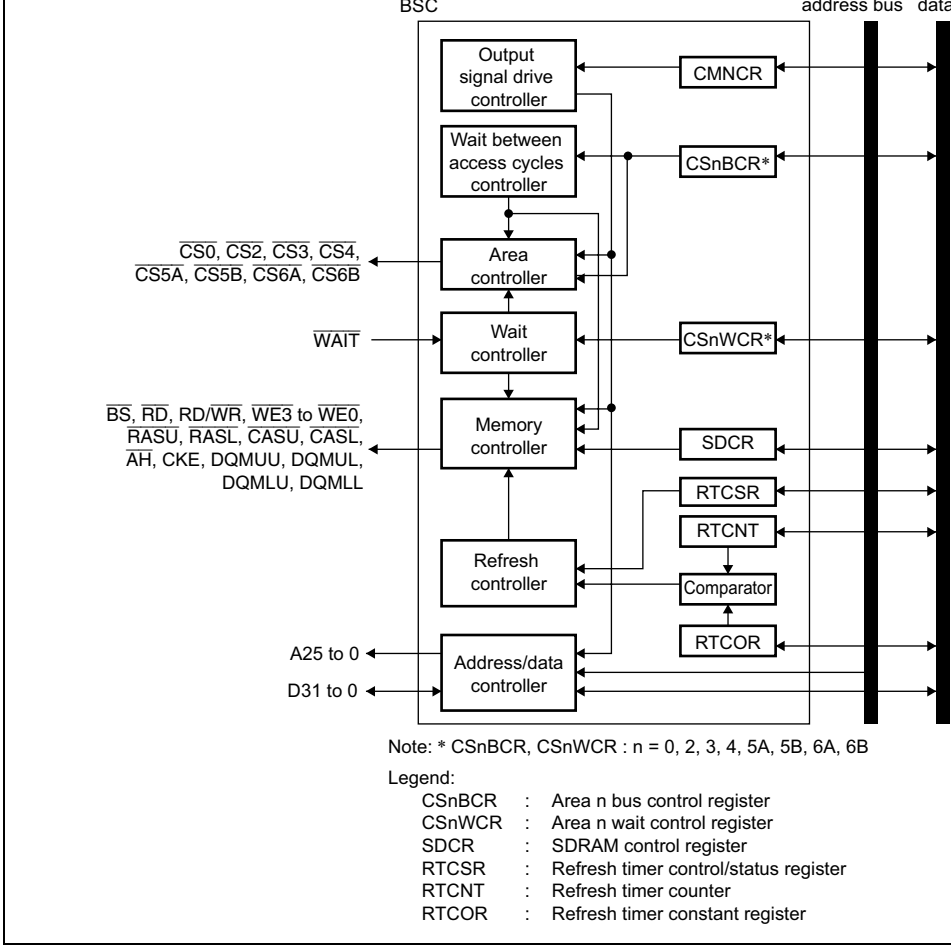


Figure 7.1 BSC Functional Block Diagram

		address/data multiplex I/O is accessed. Asserted by the as $\overline{CAS}$ in SDRAM access.
$\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A}$, $\overline{CS5B}$, $\overline{CS6A}$, $\overline{CS6B}$	O	Chip select
RD/$\overline{WR}$	O	Read/write Connects to $\overline{WE}$ pins when SDRAM or byte-selection SR connected.
$\overline{RD}$	O	Read
$\overline{WE3}$, DQMUU	O	Indicates that D31 to D24 are being written to when a no is set. Selects D31 to D24 when a byte-selection SRAM space Selects D31 to D24 when an SDRAM space is set.
$\overline{WE2}$, DQMUL	O	Indicates that D23 to D16 are being written to when a no is set. Selects D23 to D16 when a byte-selection SRAM space Selects D23 to D16 when an SDRAM space is set.
$\overline{WE1}$, DQMLU	O	Indicates that D15 to D8 are being written to when a nor and address/data multiplex I/O space are set. Selects D15 to D8 when a byte-selection SRAM space is Selects D15 to D8 when an SDRAM space is set.
$\overline{WE0}$, DQMLL	O	Indicates that D7 to D0 are being written to when a norm and address/data multiplex I/O space are set. Selects D7 to D0 when a byte-selection SRAM space is Selects D7 to D0 when an SDRAM space is set.
$\overline{RASU}$ $\overline{RASL}$	O	Connects to $\overline{RAS}$ pin when SDRAM is connected.
$\overline{CASU}$ $\overline{CASL}$	O	Connects to $\overline{CAS}$ pin when SDRAM is connected.

0: Big endian

1: Little endian

7.3 Area Overview

In the architecture of this LSI, both logical spaces and physical spaces have 32-bit addr
The cache access method is shown by the upper 3 bits. For details see section 4, Cache.
remaining 29 bits are used for division of the space into eight areas. The BSC performs
this 29-bit space.

As listed in table 7.2, this LSI can be connected directly to eight areas of memory, and
chip select signals ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A}$, $\overline{CS5B}$, $\overline{CS6A}$, and $\overline{CS6B}$) for each of ther
asserted during area 0 access; $\overline{CS5B}$ is asserted during area 5B access. When an SDRA
connected to area 2 or area 3, $\overline{RASU}$, $\overline{RASL}$, $\overline{CASU}$, $\overline{CASL}$, DQMUU, DQMUL, DQM
DQMLL are asserted.

7.3.1 Address Map

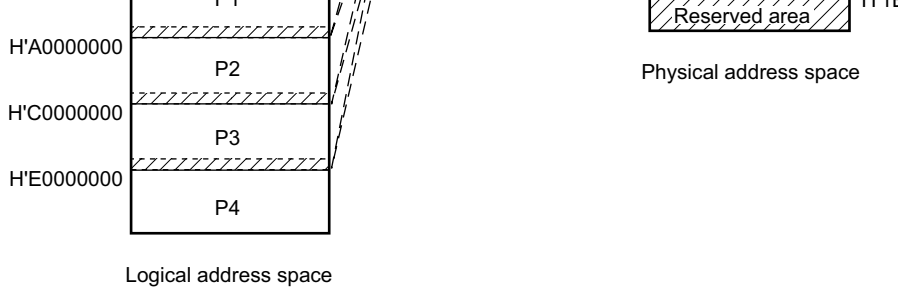
The external address space has a capacity of 384 Mbytes and is used by dividing 8 parti
The kind of memory to be connected and the data bus width are specified in each partia
The address map for the external address space is listed below.

Table 7.2 Physical Address Space Map

Area	Memory to be Connected	Physical Address		Capacity	Ac Si
Area 0	Normal memory*1, Burst ROM	H'00000000	to H'03FFFFFF	64 Mbytes	8,
		H'00000000 +H'20000000×n	to H'03FFFFFF +H'20000000×n	Shadow	(n

	Synchronous DRAM	H'0C000000 +H'20000000×n	to H'0FFFFFFF +H'20000000×n	Shadow	(
Area 4	Normal memory*1, Burst ROM, Byte-selection SRAM	H'10000000	to H'13FFFFFF	64 Mbytes	8
		H'10000000 +H'20000000×n	to H'13FFFFFF +H'20000000×n	Shadow	(
Area 5A	Normal memory*1	H'14000000	to H'15FFFFFF	32 Mbytes	8
		H'14000000 +H'20000000×n	to H'15FFFFFF +H'20000000×n	Shadow	(
Area 5B	Normal memory*1, Address/data multiplex I/O (MPX), Byte-selection SRAM	H'16000000	to H'17FFFFFF	32 Mbytes	8
		H'16000000 +H'20000000×n	to H'17FFFFFF +H'20000000×n	Shadow	(
Area 6A	Normal memory*1	H'18000000	to H'19FFFFFF	32 Mbytes	8
		H'18000000 +H'20000000×n	to H'19FFFFFF +H'20000000×n	Shadow	(
Area 6B	Normal memory*1	H'1A000000	to H'1BFFFFFF	32 Mbytes	8
		H'1A000000 +H'20000000×n	to H'1BFFFFFF +H'20000000×n	Shadow	(
Area 7*6	Reserved area	H'1C000000 +H'20000000×n	to H'1FFFFFFF +H'20000000×n		(

Notes: 1. Memory that has an interface such as SRAM or ROM.
2. Memory bus width is specified by an external pin.
3. Memory bus width is specified by a register.
4. With the address/data multiplex I/O (MPX) interface, the bus width must be
5. With the SDRAM, the bus width must be 16 bits or 32 bits.
6. Do not access the reserved area. If the reserved area is accessed, the ope be guaranteed.
7. When the addresses of the on-chip module control registers (internal I/O re area 1 are not translated by the MMU, set the top three bits of the logical a 101 to allocate in the P2 space.



Note: For logical address spaces P0 and P3, when the memory management unit (MMU) is on, optionally generate a physical address for the logical address. This figure can be applied v MMU is off and when the MMU is on and each physical address for the logical address is except for higher three bits. When translating a logical address to a physical address, refe table 7.2.

Figure 7.2 Address Space

7.3.2 Memory Bus Width

The memory bus width in this LSI can be set for each area. In area 0, external pins can select byte (8 bits), word (16 bits), or longword (32 bits) on power-on reset. The corres between the external pins (MD3, MD4) and memory size is listed in the table below.

Table 7.3 Correspondence between External Pins (MD3 and MD4) and Memor

MD4	MD3	Memory Size
0	0	Setting prohibited
	1	8 bits
1	0	16 bits
	1	32 bits

For areas other than area 0, byte, word, and longword may be chosen for the bus width CSnBCR that can be set in each area. The bus width that can be set differs according to connected interface. For more details, see the CSn Bus Control Register.

is the address space obtained by adding to it H'20000000 × n (n = 1 to 6) in areas P1 to

The address range for area 7 is H'1C000000 to H'1FFFFFFF. The address space H'1C
H'20000000 × n to H'1FFFFFFF + H'20000000 × n (n = 0 to 7) corresponding to the a
shadow space is reserved, so do not use it.

Area P4 (H'E0000000 to H'EFFFFFFF) is the I/O area where on-chip registers are allo
area has no shadow space.

7.4 Register Descriptions

The BSC has the following registers. Refer to section 24, List of Registers for the deta
addresses of these registers and the state of registers in each operating mode.

Do not access spaces other than CS0 until the termination of the setting the memory in

- Common control register (CMNCR)
- Bus control register for CS0 space (CS0BCR)
- Bus control register for CS2 space (CS2BCR)
- Bus control register for CS3 space (CS3BCR)
- Bus control register for CS4 space (CS4BCR)
- Bus control register for CS5A space (CS5ABCR)
- Bus control register for CS5B space (CS5BBCR)
- Bus control register for CS6A space (CS6ABCR)
- Bus control register for CS6B space (CS6BBCR)
- Wait control register for CS0 space (CS0WCR)
- Wait control register for CS2 space (CS2WCR)
- Wait control register for CS3 space (CS3WCR)
- Wait control register for CS4 space (CS4WCR)
- Wait control register for CS5A space (CS5AWCR)
- Wait control register for CS5B space (CS5BWCR)
- Wait control register for CS6A space (CS6AWCR)

Notes: 1. This register only accepts 32-bit writing to prevent incorrect writing. In this
upper 16 bits of the data must be H'A55A, otherwise writing cannot be perfo
When reading, the upper 16 bits are read as H'0000.

2. The contents of this register are stored in SDRAM. When this register space
accessed, the corresponding register in SDRAM is written to. For details, re
section 7.8.10, Power-On Sequence.

7.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each area. Do not acces
memory other than area 0 until the CMNCR register initialization is complete.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.
7	DMAIW1	0	R/W	Wait states between access cycles when DMA sin transfer is performed. Specify the number of idle cycles to be inserted af access to an external device with DACK when DM address transfer is performed. The method of inse cycles depends on the contents of DMAIWA. 00: No idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 4 idle cycled inserted
6	DMAIW0	0	R/W	

				device with DACK are performed.
4	—	1	R	Reserved This bit is always read as 1. The write value shou 1.
3	ENDIAN	0/1*	R	Endian Flag Samples the external pin for specifying endian or reset (MD5). All address spaces are defined by th a read-only bit. 0: The external pin for specifying endian (MD5) w on power-on reset. This LSI is being operated endian. 1: The external pin for specifying endian (MD5) w on power-on reset. This LSI is being operated endian.
2	—	0	R	Reserved This bit is always read as 0. The write value shou 0.
1	HIZMEM	0	R/W	High-Z Memory Control Specifies the pin state in software standby mode A0, $\overline{BS}$, $\overline{CS}$, RD/$\overline{WR}$, $\overline{WE}$, and $\overline{RD}$. 0: High impedance in software standby mode. 1: Driven in software standby mode
0	HIZCNT	0	R/W	High-Z Control Specifies the state in software standby mode and released for $\overline{RASU}$, $\overline{RASL}$, $\overline{CASU}$, and $\overline{CASL}$. 0: High impedance in software standby mode and released for $\overline{RASU}$, $\overline{RASL}$, $\overline{CASU}$, and $\overline{CASL}$. 1: Driven in standby mode and bus released for $\overline{R}$ $\overline{RASL}$, $\overline{CASU}$, and $\overline{CASL}$.

Note: * The external pin for specifying endian (MD5) is sampled on power-on reset When big endian is specified, this bit is read as 0 and when little endian is s this bit is read as 1.

Bit	Name	Value	R/W	Description
31, 30	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.
29	IWW1	1	R/W	Idle Cycles between Write-read Cycles and Write-
28	IWW0	1	R/W	These bits specify the number of idle cycles to be after the access to a memory that is connected to The target access cycles are the write-read cycle a write cycle. 00: No idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 4 idle cycles inserted
27	—	0	R	Reserved This bit is always read as 0. The write value shoul 0.
26	IWRWD1	1	R/W	Idle Cycles for Another Space Read-write
25	IWRWD2	1	R/W	Specify the number of idle cycles to be inserted af access to a memory that is connected to the space target access cycle is a read-write one in which co accesses switch between different spaces. 00: Setting prohibited 01: 2 idle cycles inserted 10: 3 idle cycles inserted 11: 5 idle cycles inserted

				accesses are for the same space. 00: Setting prohibited. 01: 2 idle cycles inserted 10: 3 idle cycles inserted 11: 5 idle cycles inserted
21	—	0	R	Reserved This bit is always read as 0. The write value shou 0.
20	IWRRD1	1	R/W	Idle Cycles for Read-read in Another Space
19	IWRRD0	1	R/W	Specify the number of idle cycles to be inserted a access to a memory that is connected to the spac target cycle is a read-read cycle of which continu accesses switch between different space. 00: 1 idle cycle inserted 01: 2 idle cycles inserted 10: 3 idle cycles inserted 11: 5 idle cycles inserted
18	—	0	R	Reserved This bit is always read as 0. The write value shou 0.
17	IWRRS1	1	R/W	Idle Cycles for Read-read in the Same Space
16	IWRRS0	1	R/W	Specify the number of idle cycles to be inserted a access to a memory that is connected to the spac target cycle is a read-read cycle of which continu accesses are for the same space. 00: No idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 4 idle cycles inserted

				001: Burst ROM 010: Address/data multiplex I/O (MPX) 011: Byte-selection SRAM 100: SDRAM 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited Note: SDRAM can be specified only in area 2 an SDRAM is connected to only one area, SD should be specified for area 3. In this case should be specified as normal space. Burs be specified only in area 0 and area 4. Add multiplex I/O (MPX) can be specified only i Byte-selection SRAM can be specified only and area 5B.
11	—	0	R	Reserved This bit is always read as 0. The write value shoul 0.
10	BSZ1	1	R/W	Data Bus Size
9	BSZ0	1	R/W	Specify the data bus sizes of spaces. The data bus sizes of areas 2, 3, 4 and 5A are sho 00: Setting prohibited. 01: 8-bit size 10: 16-bit size 11: 32-bit size The data bus sizes of areas 5B, 6A, and 6B are sh 00: Setting prohibited. 01: 8-bit size 10: 16-bit size 11: Setting Prohibited

3. When both the CS2 and CS3 spaces are specified as the SDRAM space, s same bus size for the CS2 and CS3 spaces.
4. When the CS2 or CS3 space is specified as the SDRAM space, specify the 16 bits or 32 bits.
5. The SDRAM bank active mode can only be used for the CS3 space. (Refer explanation of the BACTV bit in the SDRAM control register.)
6. The initial values of the bus size assignment for areas 5B, 6A, and 6B after reset is specified to prohibited setting. Therefore, specify the 8- or 16-bit siz accessing these areas.
7. When port A or B is used, specify the bus size of all areas to 8 bits or 16 bi

When the memory type is specified to an area other than the areas that can be specifie operation of this LSI is not guaranteed.

7.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0, 2, 3, 4, 5A, 5B, 6A

CSnWCR is a 32-bit readable/writable register that specifies various wait cycles for m accesses. The bit configuration of this register varies as shown below according to the type (TYPE 2, TYPE 1, or TYPE 0) specified by the CSn space bus control register (C Specify the CSnWCR register before accessing the target area. Specify CSnBCR regis then specify the CSnWCR register.

12	SW1	0	R/W	Number of Delay Cycles from Address, CSn Asser
11	SW0	0	R/W	WEn Assertion Specify the number of delay cycles from address a assertion to RD and WEn assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of cycles that are necessary fo access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited

				These bits are always read as 0. The write value always be 0.
1	HW1	0	R/W	Delay Cycles from RD, $\overline{WEn}$ negation to Address negation
0	HW0	0	R/W	Specify the number of delay cycles from RD and negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

8	WR1	1	R/W	access.
7	WR0	0	R/W	0000: 0 cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is va specification by this bit is valid even when the num access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 0	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.

16	WW0	0	R/W	access. 000: The same cycles as WR3 to WR0 setting (re wait) 001: 0 cycle 010: 1 cycles 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	0	R	Reserved These bits are always read as 0. The write value always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Asse
11	SW0	0	R/W	$\overline{WEn}$ Assertion Specify the number of delay cycles from address assertion to $\overline{RD}$ and $\overline{WEn}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is v specification by this bit is valid even when the num access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.
1	HW1	0	R/W	Delay Cycles from RD, $\overline{WEn}$ negation to Address,
0	HW0	0	R/W	negation Specify the number of delay cycles from RD and $\overline{W}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				address/data multiplex I/O. This specification is v when area 5B is specified to address/data multipl 0: No wait 1: 1 cycle wait inserted
19	—	0	R	Reserved This bit is always read as 0. The write value shou 0.
18	WW2	0	R/W	Number of Write Access Wait Cycles
17	WW1	0	R/W	Specify the number of cycles that are necessary
16	WW0	0	R/W	access. 000: The same cycles as WR3 to WR0 setting (re wait) 001: 0 cycle 010: 1 cycles 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	0	R	Reserved These bits are always read as 0. The write value always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Asse
11	SW0	0	R/W	$\overline{WEn}$ Assertion Specify the number of delay cycles from address assertion to $\overline{RD}$ and $\overline{WEn}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is va specification by this bit is valid even when the num access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.
1	HW1	0	R/W	Delay Cycles from RD, $\overline{WEn}$ negation to Address, negation
0	HW0	0	R/W	Specify the number of delay cycles from RD and $\overline{W}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted f second or later access cycles in burst access. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	0	R	Reserved These bits are always read as 0. The write value always be 0.
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted i read/write access cycle. 0000: 0 cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
8	W1	1	R/W	
7	W0	0	R/W	

				These bits are always read as 0. The write value s always be 0.

CS4WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted b second or later access cycles in burst access. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 13	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Asser
11	SW0	0	R/W	$\overline{WEn}$ Assertion Specify the number of delay cycles from address a assertion to $\overline{RD}$ and $\overline{WEn}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is specification by this bit is valid even when the nu access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	0	R	Reserved These bits are always read as 0. The write value always be 0.
1 0	HW1 HW0	0 0	R/W R/W	Delay Cycles from $\overline{RD}$, $\overline{WEn}$ negation to Address negation Specify the number of delay cycles from $\overline{RD}$ and negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

10	—	1	R	Reserved This bit is always read as 1. The write value should 1.
9	—	0	R	Reserved This bit is always read as 0. The write value should 0.
8	A2CL1	1	R/W	CAS Latency for Area 2
7	A2CL0	0	R/W	Specify the CAS latency for area 2. 00: Setting prohibited. 01: 2 cycles 10: 3 cycles 11: Setting prohibited
6 to 0	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.

				Specify the number of minimum cycles from the s precharge or issuing of PRE command to the issu command for the same bank. The setting for area common. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
12	—	0	R	Reserved This bit is always read as 0. The write value shou 0.
11	TRCD1	0	R/W	Number of Cycles from ACTV Command to READ(A)/WRIT(A) Command Specify the number of minimum cycles from issui command to issuing READ(A)/WRIT(A) comman setting for areas 2 and 3 is common. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
10	TRCD0	1	R/W	
9	—	0	R	Reserved This bit is always read as 0. The write value shou 0.
8	A3CL1	1	R/W	CAS Latency for Area 3 Specify the CAS latency for area 3. 00: Setting prohibited. 01: 2 cycles 10: 3 cycles 11: Setting prohibited
7	A3CL0	0	R/W	

				command. The setting for areas 2 and 3 is commo 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: Setting prohibited
2	—	0	R	Reserved This bit is always read as 0. The write value shoul 0.
1	TRC1	0	R/W	Number of Cycles from REF Command/Self-refres to ACTV Command Specify the number of cycles from issuing the REF or releasing self-refresh to issuing the ACTV comm setting for areas 2 and 3 is common. 00: 3 cycles 01: 4 cycles 10: 6 cycles 11: 9 cycles
0	TRC0	0	R/W	

Note: * Specify area 3 as SDRAM when only one area is connected with SDRAM. In specify area 2 as normal space.

7.4.4 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs t connected.

The bits other than RFSH and RMODE should be written in the initialization after a po reset and should not be modified after the initialization. When modifying these bits RF RMODE, do not change the values of other bits and write the previous values. Do not a 2 or 3 until the SDCR register setting is complete when using synchronous DRAM.

				01: 12 bits 10: 13 bits 11: Setting prohibited
18	—	0	R	Reserved This bit is always read as 0. The write value shou 0.
17	A2COL1	0	R/W	Number of Bits of Column Address for Area 2
16	A2COL0	0	R/W	Specifies the number of bits of column address fo 00: 8 bits 01: 9 bits 10: 10 bits 11: Setting prohibited
15 to 13	—	0	R	Reserved These bits are always read as 0. The write value always be 0.
12	SLOW	0	R/W	Low-Frequency Mode Specifies the output timing of command, address data for SDRAM and the latch timing of read data SDRAM. Setting this bit makes the hold time for c address, write and read data extended. This mod for SDRAM with low-frequency clock. 0: Command, address, and write data for SDRAM the rising edge of CKIO. Read data from SDR latched at the rising edge of CKIO. 1: Command, address, and write data for SDRAM the falling edge of CKIO. Read data from SDR latched at the falling edge of CKIO.

				Specifies whether to perform auto-refresh or self-r when the RFSH bit is 1. When the RFSH bit is 1 a 1, self-refresh starts immediately. When the RFSH this bit is 0, auto-refresh starts according to the co are set in registers RTCSR, RTCNT, and RTCOR. 0: Auto-refresh is performed 1: Self-refresh is performed
9	—	0	R	Reserved This bit is always read as 0. The write value shoul 0.
8	BACTV	0	R/W	Bank Active Mode Specifies to access whether in auto-precharge mo READA and WRITA commands) or in bank active (using READ and WRIT commands). 0: Auto-precharge mode (using READA and WRIT commands) 1: Bank active mode (using READ and WRIT com Note: Bank active mode can be used only when upper or lower bits of the CS3 space are u both the CS2 and CS3 spaces are set to S specify the auto-precharge mode.
7 to 5	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.
4	A3ROW1	0	R/W	Number of Bits of Row Address for Area 3
3	A3ROW0	0	R/W	Specifies the number of bits of the row address for 00: 11 bits 01: 12 bits 10: 13 bits 11: Setting prohibited

				01: 9 bits 10: 10 bits 11: Setting prohibited

7.4.5 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM.

This register only accepts 32-bit writing to prevent incorrect writing. In this case, the u
of the data must be H'A55A, otherwise writing cannot be performed. When reading, th
bits are read as H'0000.

RTCSR

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	Reserved
7	CMF	0	R/W	Compare Match Flag 0: Clearing condition When 0 is written in CMF a out RTCSR during CMF = 1. 1: Setting condition When the condition RTCNT satisfied.
6	CMIE	0	R/W	CMF Interrupt Enable 0: CMF interrupt request is disabled. 1: CMF interrupt request is enabled.

				011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/4096
2	RRC2	0	R/W	Refresh Count
1	RRC1	0	R/W	Specify the number of continuous refresh cycles, w
0	RRC0	0	R/W	refresh request occurs after the coincidence of the the refresh timer counter (RTCNT) and the refresh constant register (RTCOR). These bits can make t of occurrence of refresh long. 000: Once 001: Twice 010: 4 times 011: 6 times 100: 8 times 101: Setting prohibited. 110: Setting prohibited. 111: Setting prohibited.

When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT return
counting up to 255.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	Reserved
7 to 0	—	0	R/W	8-Bit Counter

7.4.7 Refresh Time Constant Register (RTCOR)

RTCOR is an 8-bit register. When RTCOR matches RTCNT, the CMF bit in RTCSR
and RTCNT is cleared to 0.

This register only accepts 32-bit writing to prevent incorrect writing. In this case, the u
of the data must be H'A55A, otherwise writing cannot be performed. When reading, th
bits are read as H'0000.

When the RFSH bit in SDCR is 1, a memory refresh request is issued by this matching
This request is maintained until the refresh operation is performed. If the request is no
when the next matching occurs, the previous request is ignored.

When the CMIE bit in RTCSR is 1, an interrupt request is issued by this matching sig
request signal is output until the CMF bit in RTCSR is cleared. Clearing the CMF bit
the interrupt and does not affect the refresh request. Accordingly, the refresh requests
timer interrupts can be used together. For example, the number of refresh requests can
by using interrupts while the refresh is performed.

RWTCNT is a 16-bit register. The lower seven bits of this register (bits 6 to 0) are valid
counter and the upper nine bits (bits 15 to 7) are reserved. This counter starts to count-u
synchronizing the CKIO after a power-on reset is released. This counter stops when the
reaches to H'007F. The access to an external bus has to wait when the counter is operat
counter is provided to minimize the time from releasing a reset for flash memory to the
access. This counter cannot be read or written into.

7.5 Endian/Access Size and Data Alignment

This LSI supports big endian, in which the 0 address is the most significant byte (MSB)
byte data and little endian, in which the 0 address is the least significant byte (LSByte)
data. Endian is specified on power-on reset by the external pin (MD5). When MD5 pin
level on power-on reset, the endian will become big endian and when MD5 pin is high
power-on reset, the endian will become little endian. Three data bus widths are availabl
normal memory (byte, word, and longword). Word and longword are available for SDR
bus width for address/data multiplex I/O (MPX) should be 16 bits. Data alignment is pe
accordance with the data bus width of the device and endian. This also means that when
data is read from a byte-width device, the read operation must be done four times. In th
alignment and conversion of data length is performed automatically between the respec
interfaces.

Tables 7.4 to 7.9 show the relationship between endian, device data width, and access u

Byte access at 2	—	—	Data 7 to Data 0	—	—	—	Assert
Byte access at 3	—	—	—	Data 7 to Data 0	—	—	—
Word access at 0	Data 15 to Data 8	Data 7 to Data 0	—	—	Assert	Assert	—
Word access at 2	—	—	Data 15 to Data 8	Data 7 to Data 0	—	—	Assert
Longword access at 0	Data 31 to Data 24	Data 23 to Data 16	Data 15 to Data 8	Data 7 to Data 0	Assert	Assert	Assert

Byte access at 2		—	—	Data 7 to Data 0	—	—	—	Assert
Byte access at 3		—	—	—	Data 7 to Data 0	—	—	—
Word access at 0		—	—	Data 15 to Data 8	Data 7 to Data 0	—	—	Assert
Word access at 2		—	—	Data 15 to Data 8	Data 7 to Data 0	—	—	Assert
Longword access at 0	1st time at 0	—	—	Data 31 to Data 24	Data 23 to Data 16	—	—	Assert
	2nd time at 2	—	—	Data 15 to Data 8	Data 7 to Data 0	—	—	Assert

Byte access at 2		—	—	—	Data 7 to Data 0	—	—	—
Byte access at 3		—	—	—	Data 7 to Data 0	—	—	—
Word access at 0	1st time at 0	—	—	—	Data 15 to Data 8	—	—	—
	2nd time at 1	—	—	—	Data 7 to Data 0	—	—	—
Word access at 2	1st time at 2	—	—	—	Data 15 to Data 8	—	—	—
	2nd time at 3	—	—	—	Data 7 to Data 0	—	—	—
Longword access at 0	1st time at 0	—	—	—	Data 31 to Data 24	—	—	—
	2nd time at 1	—	—	—	Data 23 to Data 16	—	—	—
	3rd time at 2	—	—	—	Data 15 to Data 8	—	—	—
	4th time at 3	—	—	—	Data 7 to Data 0	—	—	—

Byte access at 2	—	Data 7 to Data 0	—	—	—	Assert	—
Byte access at 3	Data 7 to Data 0	—	—	—	Assert	—	—
Word access at 0	—	—	Data 15 to Data 8	Data 7 to Data 0	—	—	Assert
Word access at 2	Data 15 to Data 8	Data 7 to Data 0	—	—	Assert	Assert	—
Longword access at 0	Data 31 to Data 24	Data 23 to Data 16	Data 15 to Data 8	Data 7 to Data 0	Assert	Assert	Assert

Byte access at 2		—	—	—	Data 7 to Data 0	—	—	—
Byte access at 3		—	—	Data7 to Data 0	—	—	—	Assert
Word access at 0		—	—	Data 15 to Data 8	Data 7 to Data 0	—	—	Assert
Word access at 2		—	—	Data 15 to Data 8	Data 7 to Data 0	—	—	Assert
Longword access at 0	1st time at 0	—	—	Data 15 to Data 8	Data 7 to Data 0	—	—	Assert
	2nd time at 1	—	—	Data 31 to Data 24	Data 23 to Data 16	—	—	Assert

Byte access at 2		—	—	—	Data 7 to Data 0	—	—	—
Byte access at 3		—	—	—	Data 7 to Data 0	—	—	—
Word access at 0	1st time at 0	—	—	—	Data 7 to Data 0	—	—	—
	2nd time at 1	—	—	—	Data 15 to Data 8	—	—	—
Word access at 2	1st time at 2	—	—	—	Data 7 to Data 0	—	—	—
	2nd time at 3	—	—	—	Data 15 to Data 8	—	—	—
Longword access at 0	1st time at 0	—	—	—	Data 7 to Data 0	—	—	—
	2nd time at 1	—	—	—	Data 15 to Data 8	—	—	—
	3rd time at 2	—	—	—	Data 23 to Data 16	—	—	—
	4th time at 3	—	—	—	Data 31 to Data 24	—	—	—

asserted for one cycle to indicate the start of a bus cycle.

There is no access size specification when reading. The correct access start address is least significant bit of the address, but since there is no access size specification, 32 bi read in case of a 32-bit device, and 16 bits in case of a 16-bit device. When writing, o signal for the byte to be written is asserted.

Read/write for cache fill or writeback follows the selected bus width and transfers a to bytes continuously. The bus is not released during this transfer. For cache misses that byte or word operand accesses or branching to odd word boundaries, the fill is always by longword accesses on the chip-external interface. Write-through-area write access cacheable read/write access are based on the actual address size.

It is necessary to output the read out data by using $\overline{RD}$ when a buffer is established in The RD/$\overline{WR}$ signal is in a read state (high output) when an access is not performed. T care must be taken about the collision of output in controlling the external data buffer.

When the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted to evaluate an wait. When the WM bit in CSnWCR is set to 1, an external wait is ignored and no Tn inserted.

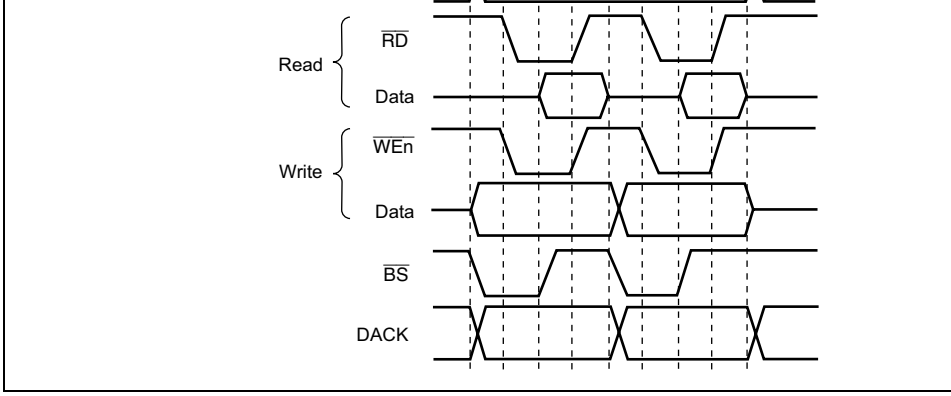


Figure 7.3 Continuous Access for Normal Space (No Wait, WM Bit in CSnW
16-Bit Bus Width, Longword Access, No Wait State between Cycles)

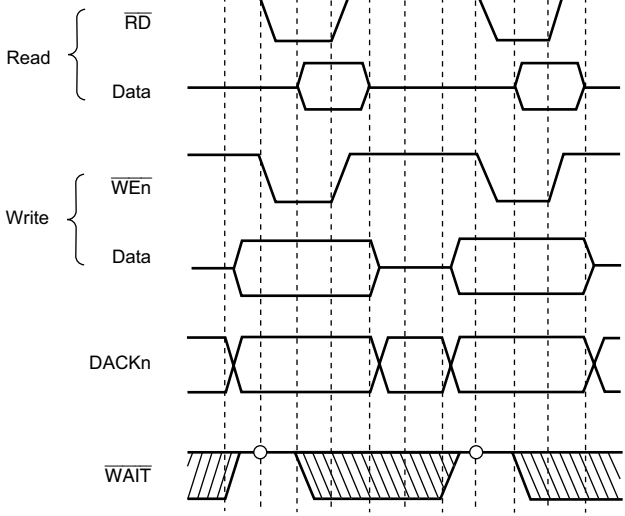


Figure 7.4 Continuous Access for Normal Space (No Wait, One Wait State between Cycles)

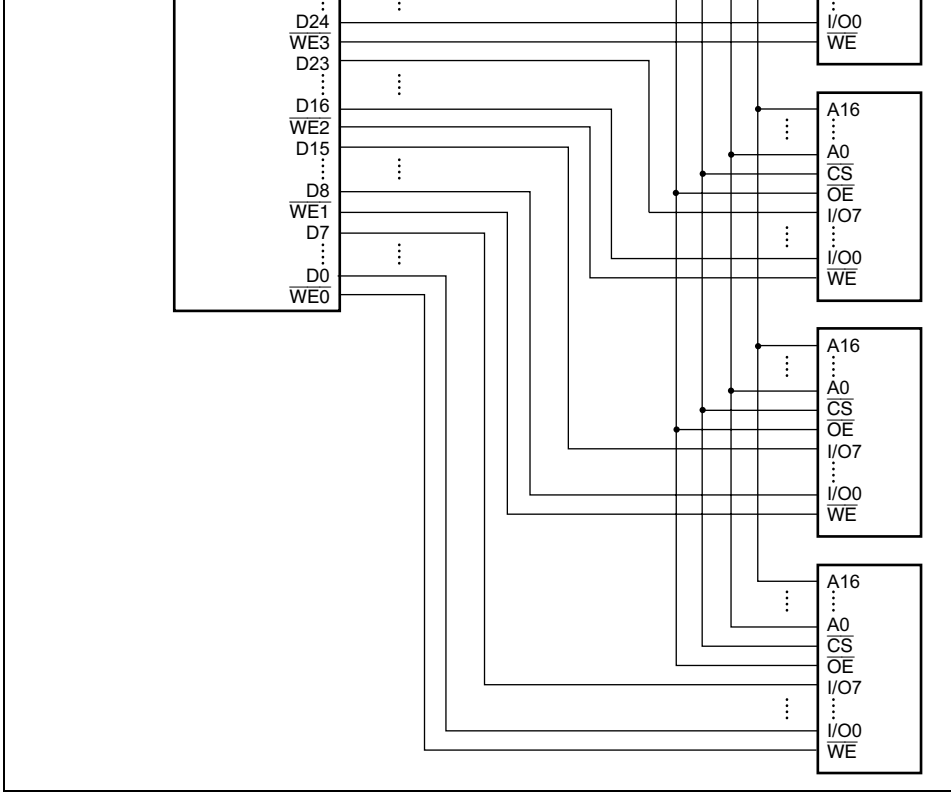


Figure 7.5 Example of 32-Bit Data-Width SRAM Connection

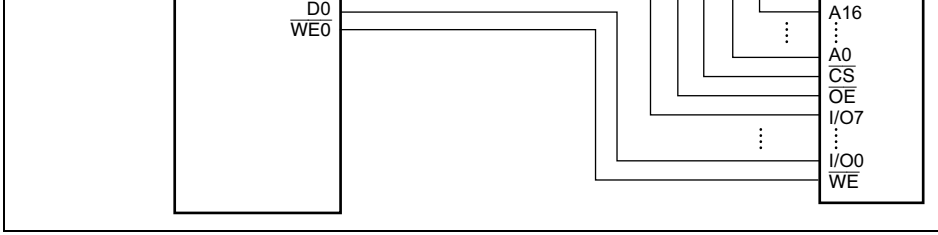


Figure 7.6 Example of 16-Bit Data-Width SRAM Connection

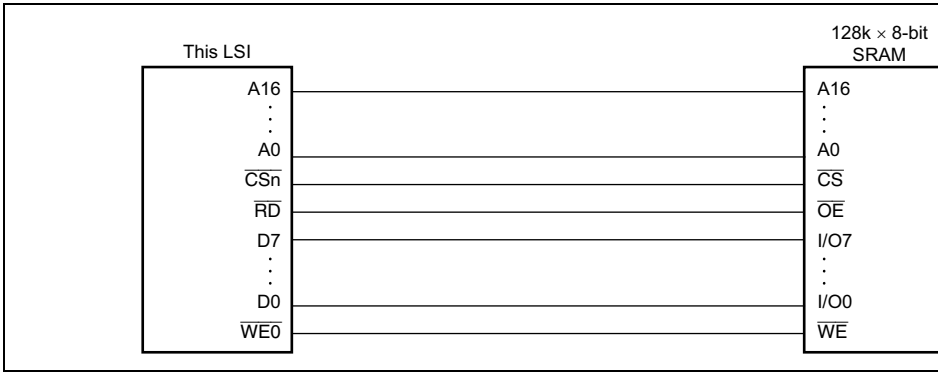


Figure 7.7 Example of 8-Bit Data-Width SRAM Connection

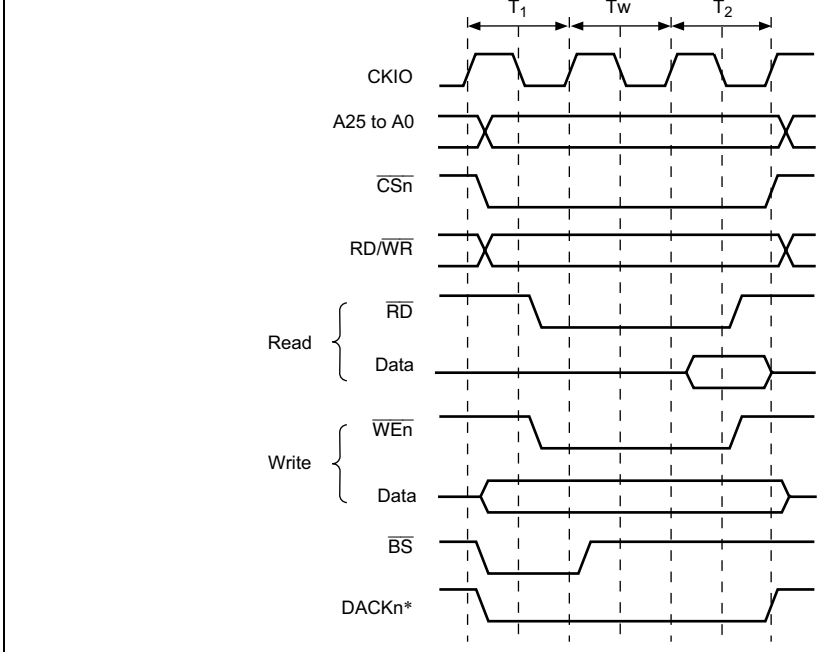
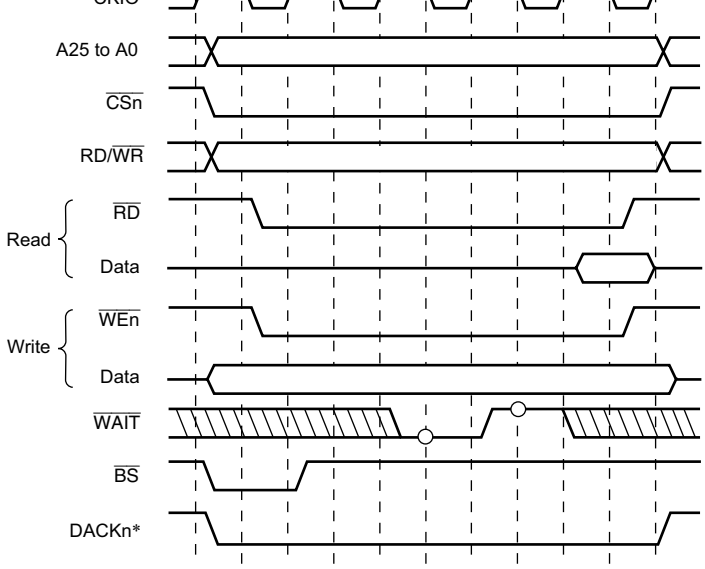


Figure 7.8 Wait Timing for Normal Space Access (Software Wait Only



Note: * The waveform for DACKn is when active low is specified.

Figure 7.9 Wait State Timing for Normal Space Access (Wait State Insertion by WAIT Signal)

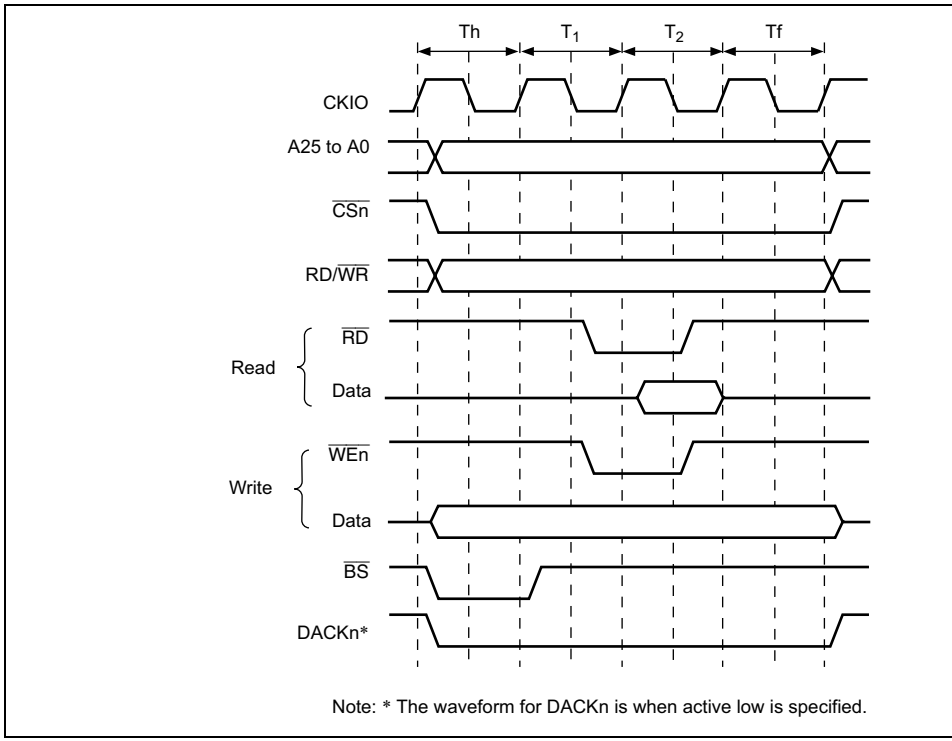


Figure 7.10 $\overline{\text{CSn}}$ Assert Period Expansion

The address output is performed from cycle Ta2 to cycle Ta3. Because cycle Ta1 has
impedance state, collisions of addresses and data can be avoided without inserting idl
even in continuous accesses. Address output is increased to 3 cycles by setting the MP
in CS5BWCR. The RD/$\overline{WR}$ signal is output at the same time as the $\overline{CSn}$ signal; it is h
read cycle and low in the write cycle.

The data cycle is the same as that in a normal space access.

Timing charts are shown in figures 7.11, 7.12, and 7.13.

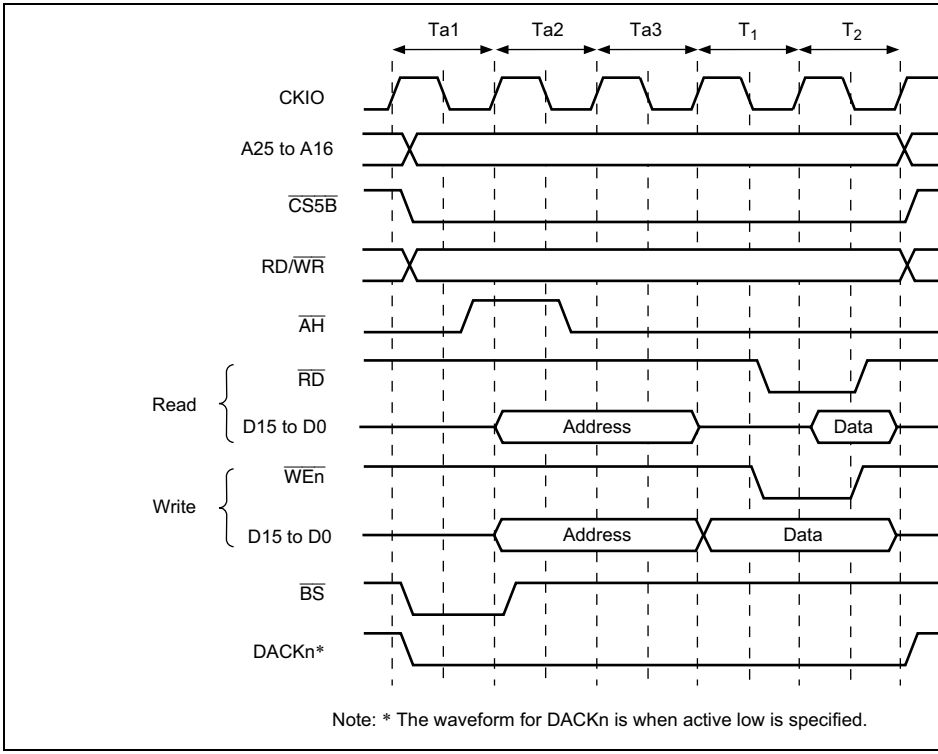
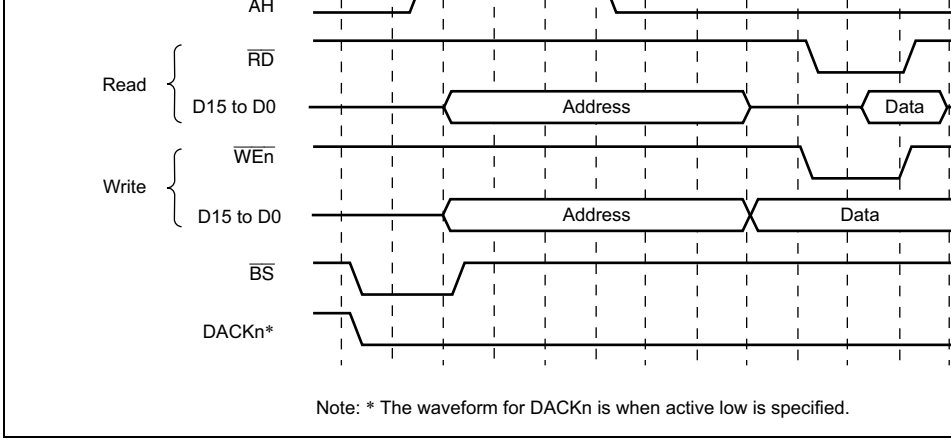
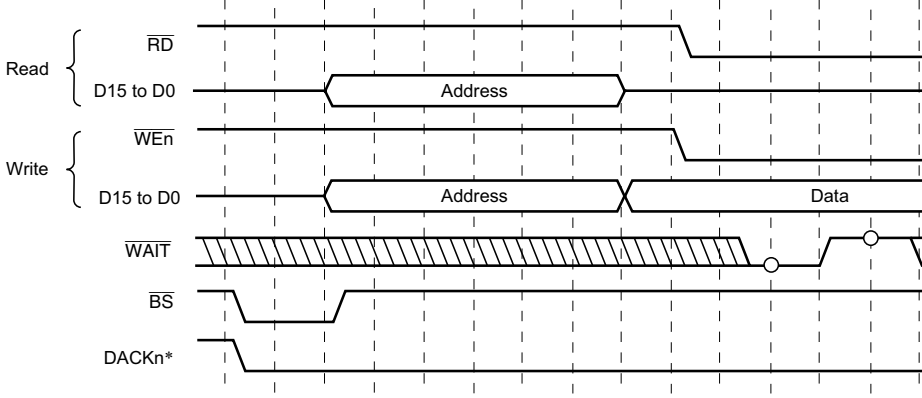


Figure 7.11 Access Timing for MPX Space (Address Cycle No Wait, Data Cycle No Wait)



Note: * The waveform for DACKn is when active low is specified.

Figure 7.12 Access Timing for MPX Space (Address Cycle Wait 1, Data Cycle No Wait)



Note: * The waveform for DACKn is when active low is specified.

Figure 7.13 Access Timing for MPX Space
(Address Cycle Access Wait 1, Data Cycle Wait 1, External Wait 1)

The SDRAM that can be connected to this LSI is a product that has 11/12/13 bits of rov
8/9/10 bits of column address, 4 or less banks, and uses the A10 pin for setting precharg
read and write command cycles. The control signals for direct connection of SDRAM a
$\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, RD/$\overline{\text{WR}}$, DQMUU, DQMUL, DQMLU, DQMLL, CKE, $\overline{\text{CS2}}$, a
All the signals other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than
valid when $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$ is asserted. SDRAM can be connected to up to 2 spaces. The da
width of the area that is connected to SDRAM can be set to 32 or 16 bits.

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are su
the SDRAM operating mode.

Commands for SDRAM can be specified by $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, RD/WR, ar
address signals. These commands are shown below.

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks pre-charge (PALL)
- Specified bank pre-charge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with pre-charge (READA)
- Write (WRIT)
- Write with pre-charge (WRITA)
- Write mode register (MRS)

The byte to be accessed is specified by DQMUU, DQMUL, DQMLU, and DQMLL. Fo
relationship between DQMxx and the byte to be accessed, refer to section 7.5, Endian/A
and Data Alignment.

Figures 7.14 and 7.15 show examples of the connection of SDRAM with the LSI.

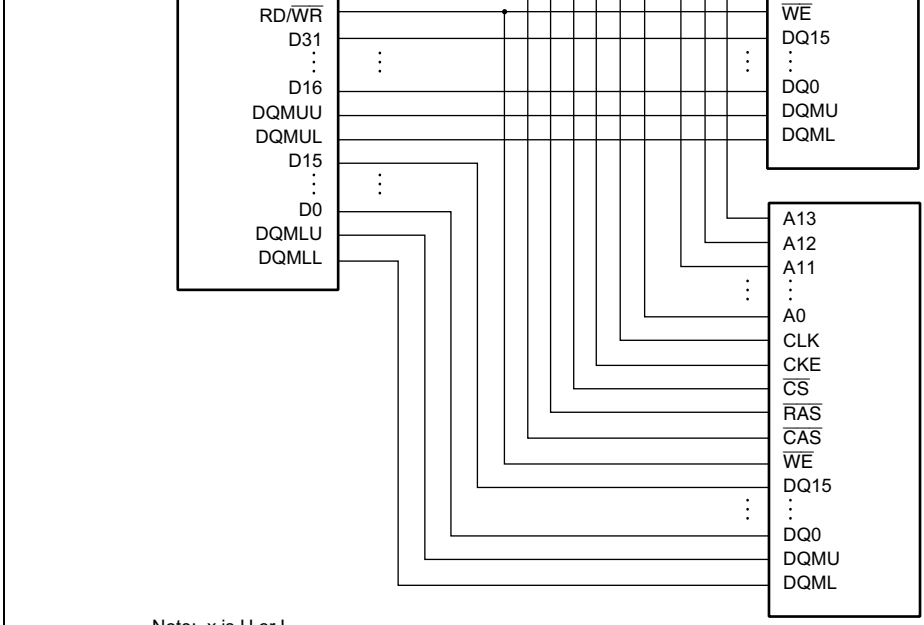


Figure 7.14 Example of 64-MBit Synchronous DRAM Connection (32-Bit D

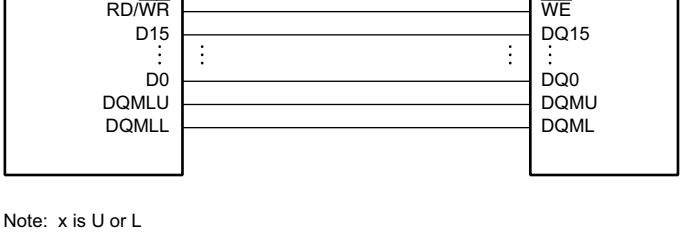


Figure 7.15 Example of 64-MBit Synchronous DRAM (16-Bit Data Bus

7.8.2 Address Multiplexing

An address multiplexing is specified so that SDRAM can be connected without externa multiplexing circuitry according to the setting of bits BSZ[1:0] in CSnBCR, AxROW[1 AxCOL[1:0] in SDCR. Tables 7.10 to 7.15 show the relationship between the settings o BSZ[1:0], AxROW[1:0], and AxCOL[1:0] and the bits output at the address pins. Do n those bits in the manner other than this table, otherwise the operation of this LSI is not A25 to A18 are not multiplexed and the original values of address are always output at

When the data bus width is 16 bits (BSZ[1:0] = 10), A0 of SDRAM specifies a word ac Therefore, connect this A0 pin of SDRAM to the A1 pin of the LSI; the A1 pin of SDR A2 pin of the LSI, and so on. When the data bus width is 32 bits (BSZ[1:0] = 11), the A SDRAM specifies a longword address. Therefore, connect this A0 pin of SDRAM to th the LSI; the A1 pin of SDRAM to the A3 pin of the LSI, and so on.

A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A12 (BA1)	Spec
A13	A21*2	A21*2	A11 (BA0)	
A12	A20	L/H*1	A10/AP	Spec addr
A11	A19	A11	A9	Addr
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unus
A0	A8	A0		
Example of connected memory				
64-Mbit product (512 kwords x 32 bits x 4 banks, column 8 bits product): 1 device				
16-Mbit product (512 kwords x 16 bits x 2 banks, column 8 bits product): 2 devices				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H accordin access mode.
2. Bank address specification

A16	A24	A16		
A15	A23*2	A23*2	A13 (BA1)	Specif
A14	A22*2	A22*2	A12 (BA0)	
A13	A21	A13	A11	Addre
A12	A20	L/H*1	A10/AP	Specif addres
A11	A19	A11	A9	Addre
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unuse
A0	A8	A0		

Example of connected memory

128-Mbit product (1 Mword x 32 bits x 4 banks, column 8 bits product): 1 device

64-Mbit product (1 Mword x 16 bits x 4 banks, column 8 bits product): 2 devices

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according access mode.

2. Bank address specification

A16	A25	A16		
A15	A24[*2]	A24[*2]	A13 (BA1)	Spec
A14	A23[*2]	A23[*2]	A12 (BA0)	
A13	A22	A13	A11	Addr
A12	A21	L/H[*1]	A10/AP	Spec addr
A11	A20	A11	A9	Addr
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unus
A0	A9	A0		
Example of connected memory				
256-Mbit product (2 Mwords x 32 bits x 4 banks, column 9 bits product): 1 device				
128-Mbit product (2 Mwords x 16 bits x 4 banks, column 9 bits product): 2 devices				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H accordin access mode.
2. Bank address specification

A16	A26	A16		
A15	A25[*2]	A25[*2*3]	A13 (BA1)	Specif
A14	A24[*2]	A24[*2]	A12 (BA0)	
A13	A23	A13	A11	Addre
A12	A22	L/H[*1]	A10/AP	Specif addres
A11	A21	A11	A9	Addre
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	
A2	A12	A2	A0	
A1	A11	A1		Unuse
A0	A10	A0		

Example of connected memory

512-Mbit product (4 Mwords x 32 bits x 4 banks, column 10 bits product): 1 device

256-Mbit product (4 Mwords x 16 bits x 4 banks, column 10 bits product): 2 devices

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank addre is not asserted.

A16	A25[*2*3]	A25[*2*3]	A14 (BA1)	Spec
A15	A24[*2]	A24[*2]	A13 (BA0)	
A14	A23	A14	A12	Addr
A13	A22	A13	A11	
A12	A21	L/H[*1]	A10/AP	Spec addr
A11	A20	A11	A9	Addr
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unus
A0	A9	A0		

Example of connected memory

512-Mbit product (4 Mwords x 32 bits x 4 banks, column 9 bits product): 1 device

256-Mbit product (4 Mwords x 16 bits x 4 banks, column 9 bits product): 2 devices

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H accordin access mode.

2. Bank address specification

3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank add $\overline{\text{RASU}}$ is not asserted.

A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21*2	A21*2	A12 (BA1)	Specif
A12	A20*2	A20*2	A11 (BA0)	
A11	A19	L/H*1	A10/AP	Specif addres
A10	A18	A10	A9	Addre
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unuse
Example of connected memory				
16-Mbit product (512 kwords x 16 bits x 2 banks, column 8 bits product): 1 device				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according access mode.
2. Bank address specification

A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A13 (BA1)	Spec
A13	A21*2	A21*2	A12 (BA0)	Addr
A12	A20	A12	A11	
A11	A19	L/H*1	A10/AP	Spec addr
A10	A18	A10	A9	Addr
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unus
Example of connected memory				
64-Mbit product (1 Mword x 16 bits x 4 banks, column 8 bits product): 1 device				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H accordin access mode.

2. Bank address specification

A16	A25	A16		
A15	A24	A15		
A14	A23*2	A23*2	A13 (BA1)	Specif
A13	A22*2	A22*2	A12 (BA0)	
A12	A21	A12	A11	Addre
A11	A20	L/H*1	A10/AP	Specif addres
A10	A19	A10	A9	Addre
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unuse
Example of connected memory				
128-Mbit product (2 Mwords x 16 bits x 4 banks, column 9 bits product): 1 device				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according access mode.
2. Bank address specification

A16	A26	A16		
A15	A25	A15		
A14	A24*2	A24*2	A13 (BA1)	Spec
A13	A23*2	A23*2	A12 (BA0)	
A12	A22	A12	A11	Addr
A11	A21	L/H*1	A10/AP	Spec addr
A10	A20	A10	A9	Addr
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unus
Example of connected memory				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 10 bits product): 1 device				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H accordin access mode.

2. Bank address specification

A16	A25	A16		
A15	A24*2	A24*2	A14 (BA1)	Specif
A14	A23*2	A23*2	A13 (BA0)	
A13	A22	A13	A12	Addre
A12	A21	A12	A11	
A11	A20	L/H*1	A10/AP	Specif addres
A10	A19	A10	A9	Addre
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unuse
Example of connected memory				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 9 bits product): 1 device				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according access mode.
2. Bank address specification

A16	A26	A16		
A15	A25*2*3	A25*2*3	A14 (BA1)	Spec
A14	A24*2	A24*2	A13 (BA0)	
A13	A23	A13	A12	Addr
A12	A22	A12	A11	
A11	A21	L/H*1	A10/AP	Spec addr
A10	A20	A10	A9	Addr
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unus
Example of connected memory				
512-Mbit product (8 Mwords x 16 bits x 4 banks, column 10 bits product): 1 device				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H accordin access mode.

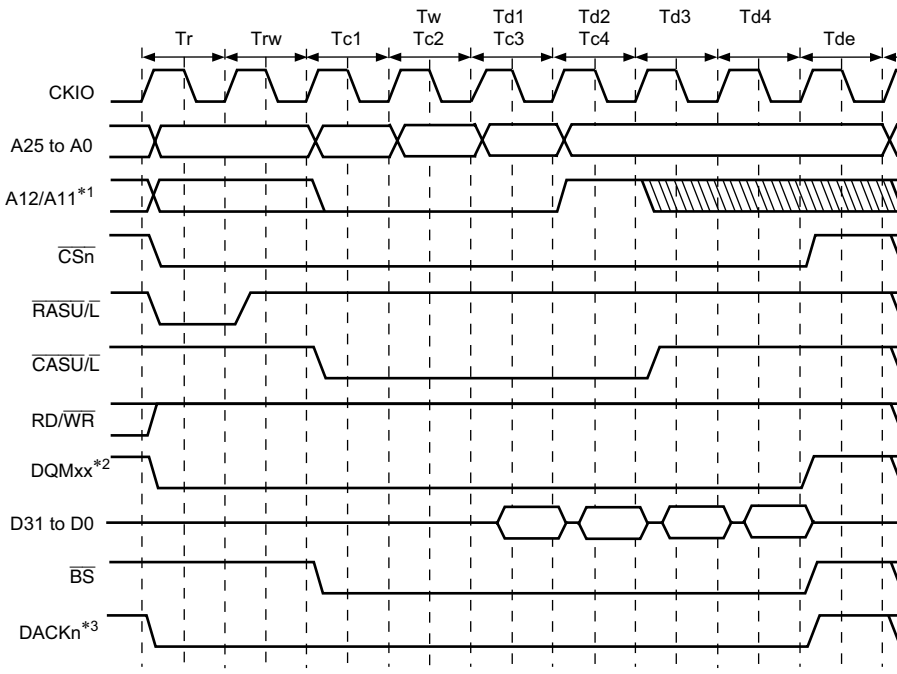
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank add is not asserted.

length 1 is performed consecutively 4 times to read 16-byte continuous data from the S is connected to a 32-bit data bus.

Table 7.16 shows the relationship between the access size and the number of bursts.

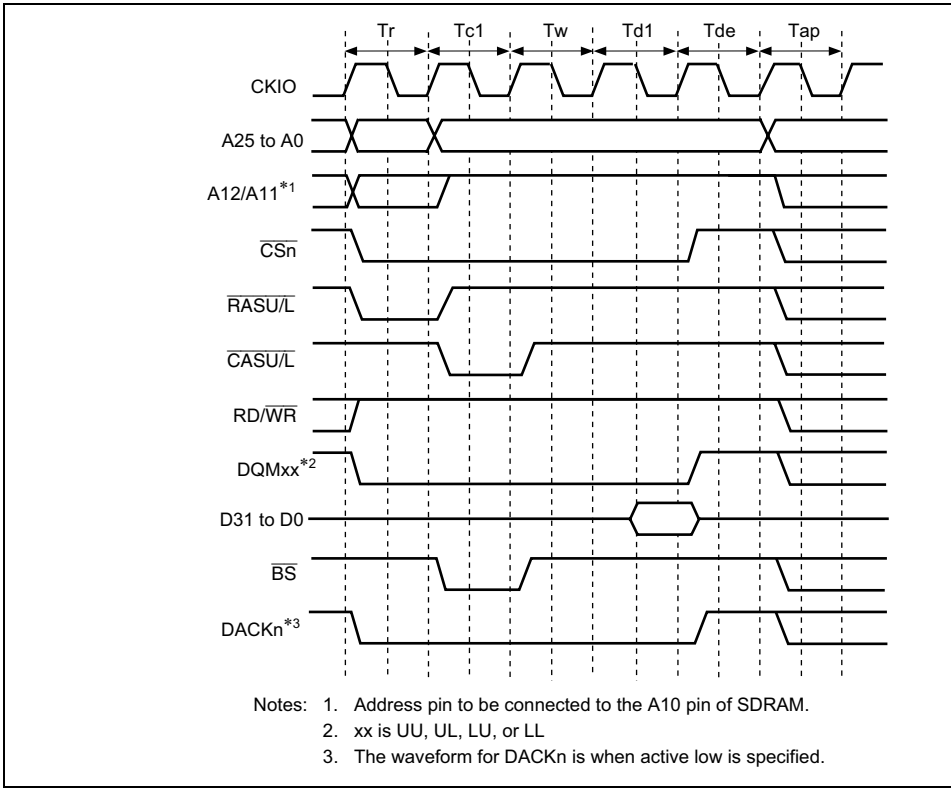
Table 7.16 Relationship between Access Size and Number of Bursts

Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bits	8
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	16 bits	4



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.16 Synchronous DRAM Burst Read Wait Specification Timi (Auto Precharge)



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL
3. The waveform for DACKn is when active low is specified.

Figure 7.17 Basic Timing for Single Read (Auto Precharge)

is performed continuously 4 times to write 16-byte continuous data to the SDRAM tha
connected to a 32-bit data bus.

The relationship between the access size and the number of bursts is shown in table 7.

specified by the TRP[1:0] bits of the CS3WCR register.

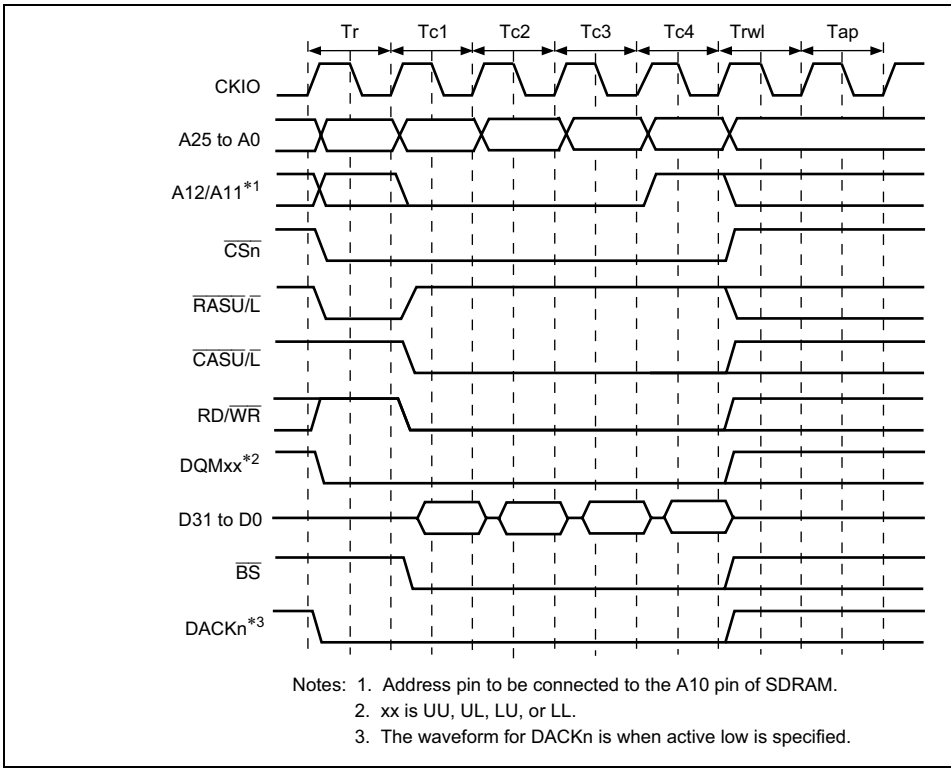
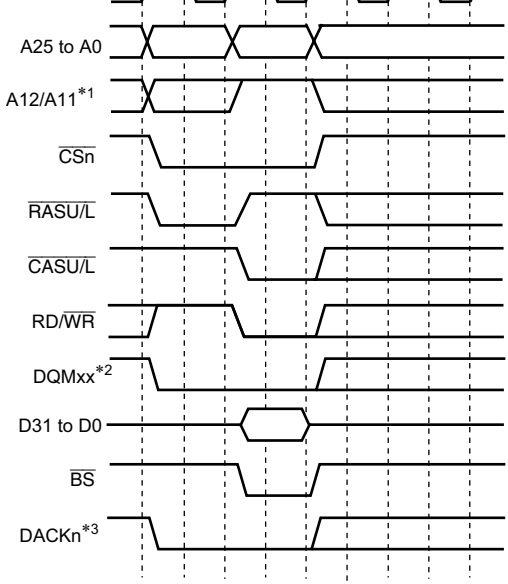


Figure 7.18 Basic Timing for Synchronous DRAM Burst Write (Auto Prech



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.19 Basic Timing for Single Write (Auto Precharge)

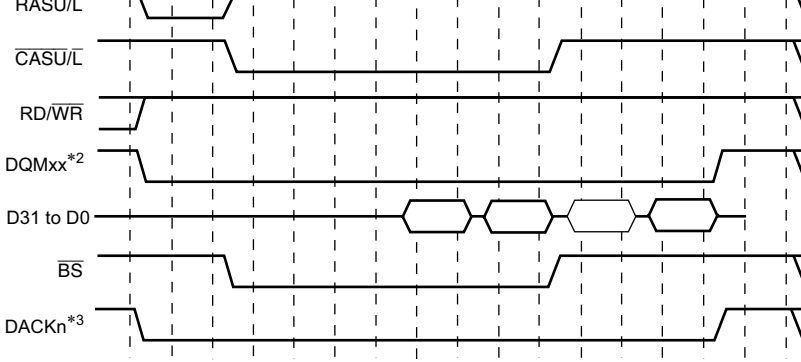
mode must be set. In this case, precharging is not performed when the access ends. Wh
accessing the same row address in the same bank, it is possible to issue the READ or W
command immediately, without issuing an ACTV command. As synchronous DRAM i
divided into several banks, it is possible to activate one row address in each bank. If the
access is to a different row address, a PRE command is first issued to precharge the rele
then when precharging is completed, the access is performed by issuing an ACTV com
followed by a READ or WRIT command. If this is followed by an access to a different
address, the access time will be longer because of the precharging performed after the a
request is issued.

In a write, when auto-precharge is performed, a command cannot be issued for a period
Tpc cycles after issuance of the WRITA command. When bank active mode is used, RE
WRIT commands can be issued successively if the row address is the same. The numbe
can thus be reduced by Trwl + Tpc cycles for each write.

There is a limit on t_{RAS}, the time for placing each bank in the active state. If there is no
that there will not be a cache hit and another row address will be accessed within the pe
which this value is maintained by program execution, it is necessary to set auto-refresh
refresh cycle to no more than the maximum value of t_{RAS}.

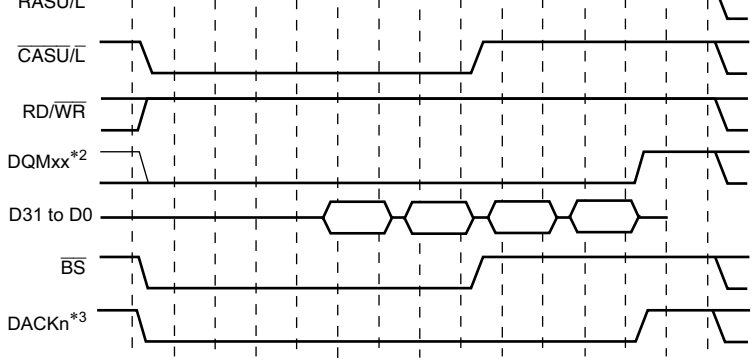
A burst read cycle without auto-precharge is shown in figure 7.20, a burst read cycle fo
row address in figure 7.21, and a burst read cycle for different row addresses in figure 7
Similarly, a burst write cycle without auto-precharge is shown in figure 7.23, a single w
for the same row address in figure 7.24, and a single write cycle for different row addre
figure 7.25.

When bank active mode is set, if only accesses to the respective banks in the area 3 spa
considered, as long as accesses to the same row address continue, the operation starts w
cycle in figure 7.20 or 7.23, followed by repetition of the cycle in figure 7.21 or 7.24. A
a different area during this time has no effect. If there is an access to a different row ad
bank active state, after this is detected the bus cycle in figure 7.22 or 7.25 is executed in
that in figure 7.21 or 7.24. In bank active mode, too, all banks become inactive after a r
cycle or after the bus is released as the result of bus arbitration.



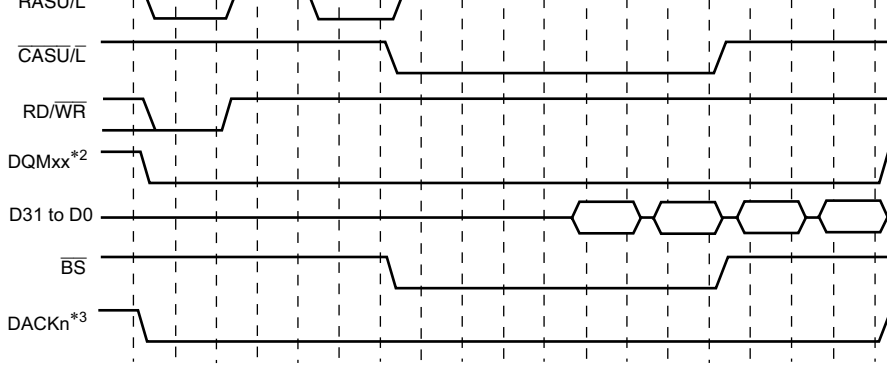
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.20 Burst Read Timing (No Auto Precharge)



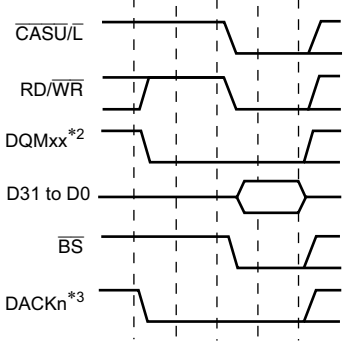
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.21 Burst Read Timing (Bank Active, Same Row Address)



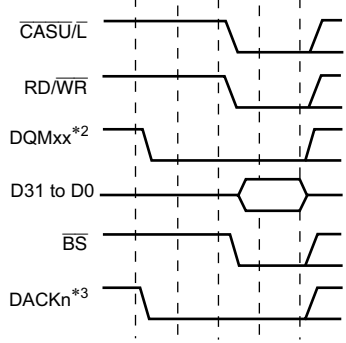
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.22 Burst Read Timing (Bank Active, Different Row Address



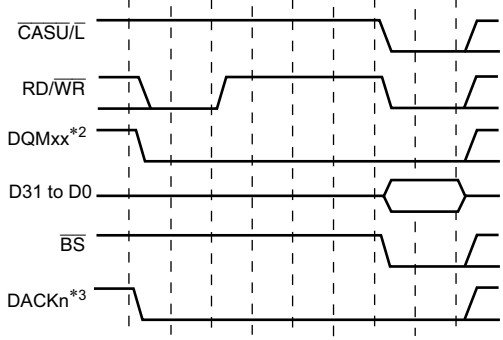
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.23 Single Write Timing (No Auto Precharge)



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.24 Single Write Timing (Bank Active, Same Row Address)

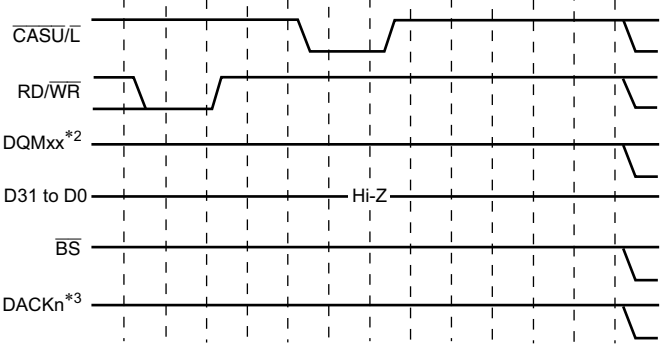


Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.25 Single Write Timing (Bank Active, Different Row Addresse

Refreshing is performed for the number of times specified by bits RRC[2:0] in RT
intervals determined by the input clock selected by bits CKS[2:0] in RTCSR, and
in RTCOR. The value of these bits should be set so as to satisfy the refresh interva
for the synchronous DRAM used. First make the settings for RTCOR, RTCNT, an
RMODE and RFSH bits in SDCR, then make bits CKS[2:0] and RRC[2:0] setting
When the clock is selected by bits CKS[2:0], RTCNT starts counting up from the
time. The RTCNT value is constantly compared with the RTCOR value, and if the
are the same, a refresh request is generated and auto-refresh is performed for the n
times specified by bits RRC[2:0]. At the same time, RTCNT is cleared to zero and
up is restarted. Figure 7.26 shows the auto-refresh cycle timing.

After starting, the auto refreshing, PALL command is issued in the Tp cycle to ma
banks to pre-charged state from active state when some bank is being pre-charged.
command is issued in the Trr cycle after inserting idle cycles of which number is s
the TRP[1:0] bits in CSnWCR. A new command is not issued for the duration of t
of cycles specified by the TRC[1:0] bits in CSnWCR after the Trr cycle. The TRC
must be set so as to satisfy the SDRAM refreshing cycle time stipulation (t_{RC}). A N
inserted between the Tp cycle and Trr cycle when the setting value of the TRP[1:0
CSnWCR is longer than or equal to 2 cycles.

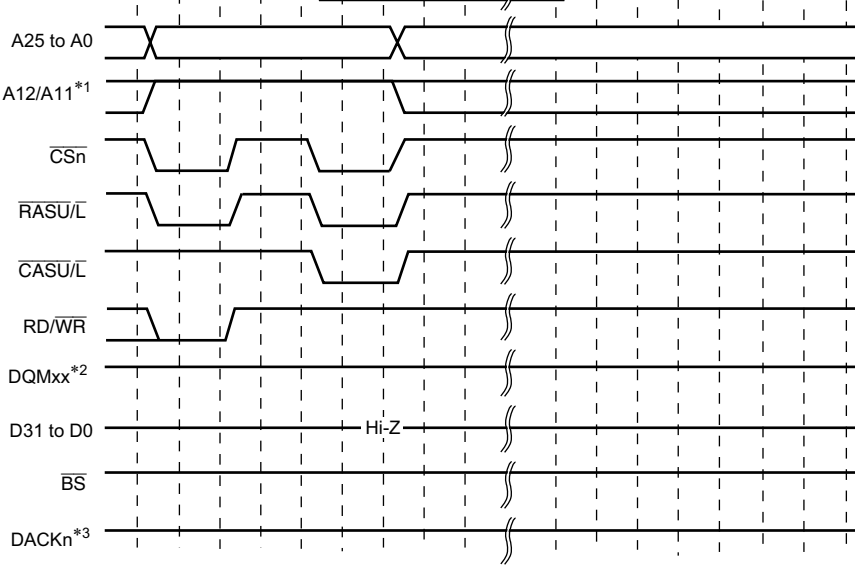


Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.26 Auto-Refresh Timing

2. Self-refreshing

Self-refresh mode in which the refresh timing and refresh addresses are generated w
synchronous DRAM. Self-refreshing is activated by setting both the RMODE bit ar
RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issue
cycle after the completion of the pre-charging bank. A SELF command is then issue
inserting idle cycles of which number is specified by the TRP[1:0] bits in CSnWSR
Synchronous DRAM cannot be accessed while in the self-refresh state. Self-refresh
cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared,
issuance is disabled for the number of cycles specified by the TRC[1:0] bits in CSn
refresh timing is shown in figure 7.27. Settings must be made so that self-refresh cl
data retention are performed correctly, and auto-refreshing is performed at the corre
intervals. When self-refreshing is activated from the state in which auto-refreshing i
refreshing is restarted if the RFSH bit is set to 1 and the RMODE bit is cleared to 0
refresh mode is cleared. If the transition from clearing of self-refresh mode to the st
refreshing takes time, this time should be taken into consideration when setting the
of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable ref
be started immediately.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.27 Self-Refresh Timing

3. Relationship between refresh requests and bus cycle

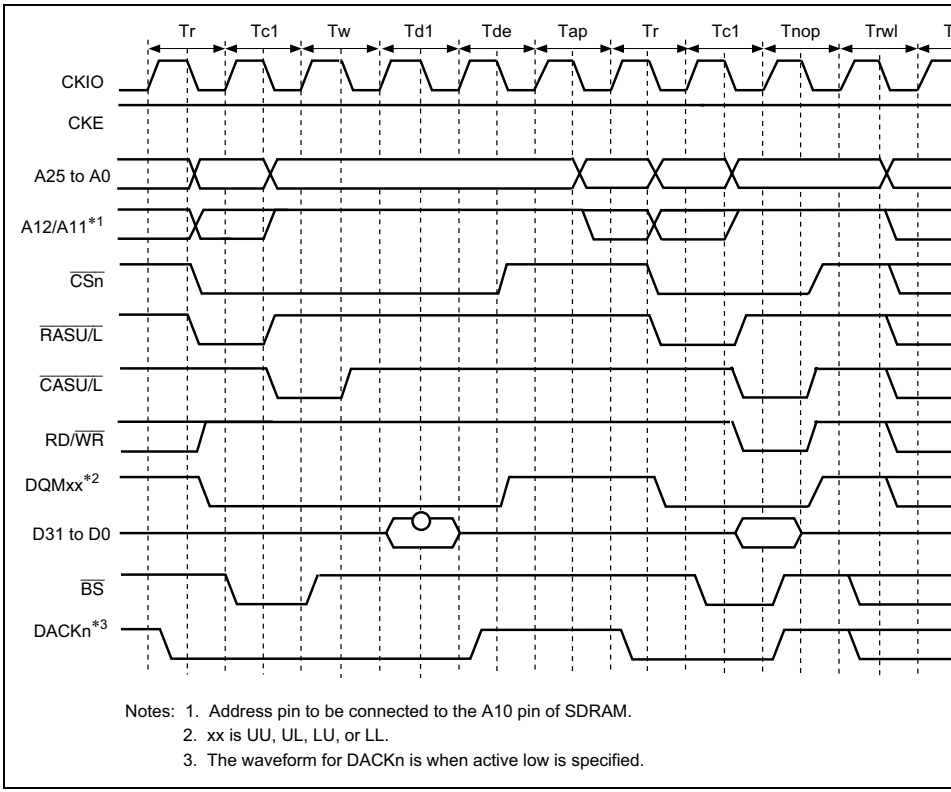
   If a refresh request is generated during a bus cycle, refresh waits for the bus cycle completed. If a refresh request is generated while the bus is released by the bus arb function, refresh waits for the bus mastership to be obtained.

   If a new refresh request is generated while refresh is waiting, the first refresh requ canceled. To perform refresh correctly, the bus cycle and the bus-owned period mu than the refresh interval.

   If a bus request is issued during self-refreshing, the bus is not released until the ref completed.

write data, and read data to be extended.

If SDRAM is operated at a high frequency with the SLOW bit set to 1, the setup time o commands, addresses, write data, and read data are not guaranteed. Take the operating and timing design into consideration when making the SLOW bit setting.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.28 Low-Frequency Mode Access Timing

synchronous DRAM. In this operation the data is ignored, but the mode write is perfo
word-size access. To set burst read/single write, CAS latency 2 to 3, wrap type = sequ
burst length 1 supported by the LSI, arbitrary data is written in a word-size access to t
shown in table 7.17. In this time 0 is output at the external address pins of A12 or late

Table 7.17 Access Address in SDRAM Mode Register Write

(1) Setting for Area 2 (SDMR2)

Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External A
16 bits	2	H'A4FD4440	H'0000440
	3	H'A4FD4460	H'0000460
32 bits	2	H'A4FD4880	H'0000880
	3	H'A4FD48C0	H'00008C0

Burst read/burst write (burst length 1):

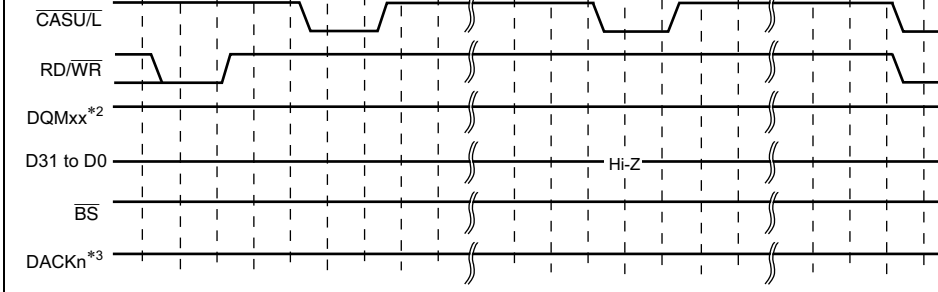
Data Bus Width	CAS Latency	Access Address	External A
16 bits	2	H'A4FD4040	H'0000040
	3	H'A4FD4060	H'0000060
32 bits	2	H'A4FD4080	H'0000080
	3	H'A4FD40C0	H'00000C0

Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Ad
16 bits	2	H'A4FD5040	H'0000040
	3	H'A4FD5060	H'0000060
32 bits	2	H'A4FD5080	H'0000080
	3	H'A4FD50C0	H'00000C0

Mode register setting timing is shown in figure 7.29. A PALL command (all bank pre-c
command) is firstly issued. A REF command (auto refresh command) is then issued 8 t
MRS command (mode register write command) is finally issued. Idle cycles, of which
specified by the TRP[1:0] bits in CSnWCR, are inserted between the PALL and the firs
cycles, of which number is specified by the TRC[1:0] bits in CSnWCR, are inserted be
and REF, and between the 8th REF and MRS. Idle cycles, of which number is one or m
inserted between the MRS and a command to be issued next.

It is necessary to keep idle time of certain cycles for SDRAM before issuing PALL com
power-on. Refer the manual of the SDRAM for the idle time to be needed. When the pu
of the reset signal is longer then the idle time, mode register setting can be started imm
after the reset, but care should be taken when the pulse width of the reset signal is short
idle time.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. xx is UU, UL, LU, or LL.
3. The waveform for DACKn is when active low is specified.

Figure 7.29 Synchronous DRAM Mode Write Timing (Based on JEDE

7.9 Burst ROM Interface

The burst ROM interface is provided to access ROM that has the page mode function, flash memory, in high speed. Basically the access to the ROM is performed in the sam normal space. When the first cycle is terminated, however, negation of the $\overline{RD}$ signal executed. The accesses after the 2nd access are performed by exchanging only the add accesses after the 2nd access, the address is changed at the falling edge of CKIO.

The number of wait cycles specified by the W[3:0] bits in CSnWCR are inserted for t access cycle. The number of wait cycles specified by the BW[1:0] bits in CSnWCR ar for the second and subsequent access cycles.

In the access to the burst ROM, the $\overline{BS}$ signal is asserted only to the first access cycle. wait input is valid only to the first access cycle. In the single access or write access tha perform the burst operation in the burst ROM interface, access timing is same as a nor Table 7.18 lists a relationship between bus width, access size, and the number of burst 7.30 shows a timing chart.

16 bits	1
32 bits	2
16 bytes	8

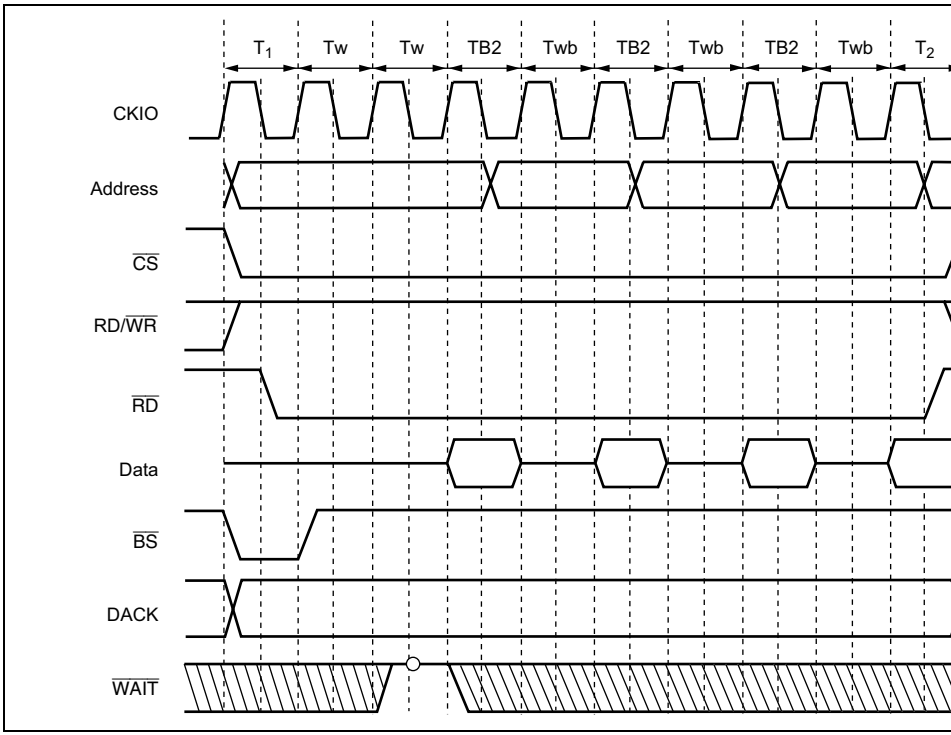


Figure 7.30 Burst ROM Access (Bus Width 8 Bits, Access Size 32 Bits (Number o Access Wait for the 1st Time 2, Access Wait for 2nd Time and after 1)

Note that in a write cycle, data is written in accordance with the byte-selection pin ($\overline{W}$
Check the data sheet of the memory to be used for the actual timing.

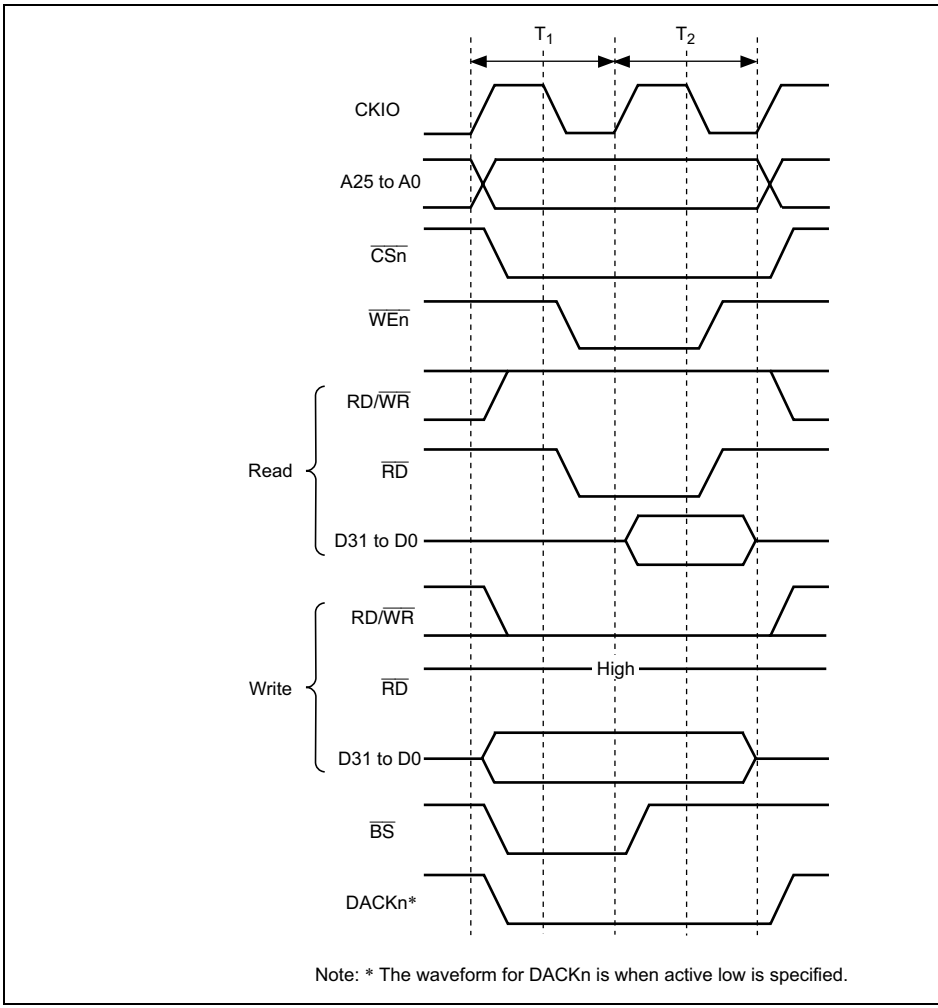


Figure 7.31 Byte-Selection SRAM Basic Access Timing

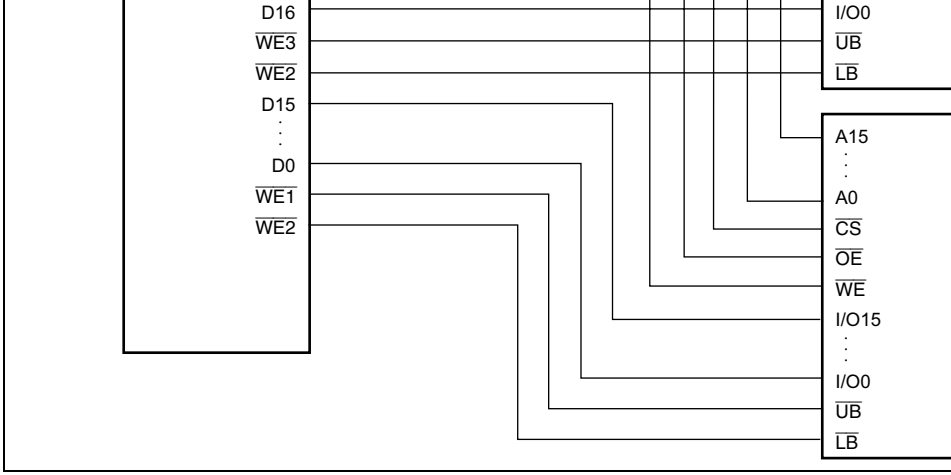


Figure 7.32 Example of Connection with 32-Bit Data-Width Byte-Selection S

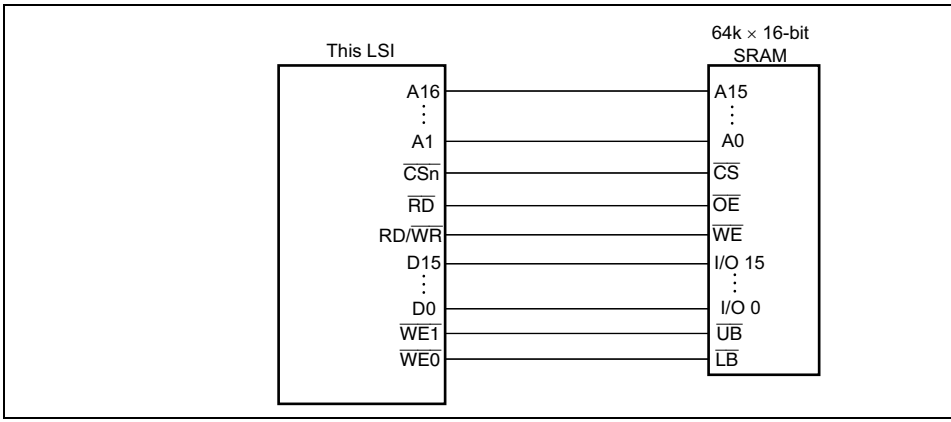


Figure 7.33 Example of Connection with 16-Bit Data-Width Byte-Selection S

IWRWS[1:0], IWRRD[1:0], and IWRRS[1:0] in CSnBCR, and bits DMAIW[1:0] and
in CMNCR. The conditions for setting the wait cycles between access cycles (idle cyc
shown below.

1. Continuous accesses are write-read or write-write
2. Continuous accesses are read-write for different spaces
3. Continuous accesses are read-write for the same space
4. Continuous accesses are read-read for different spaces
5. Continuous accesses are read-read for the same space
6. Data output from an external device caused by DMA single transfer is followed by
   from another device that includes this LSI (DMAIWA = 0)
7. Data output from an external device caused by DMA single transfer is followed by
   access (DMAIWA = 1)

7.12 Bus Arbitration

This LSI supports bus arbitration. This LSI has bus mastership in the normal state and
mastership after receiving a bus request from another device.

To prevent device malfunction while the bus mastership is transferred between master
the LSI negates all of the bus control signals before bus release. When the bus masters
received, all of the bus control signals are first negated and then driven appropriately.
output buffer conflicts can be prevented because the master and slave drive the same s
the same values. In addition, to prevent noise while the bus control signal is in the hig
state, pull-up resistors must be connected to these control signals.

Bus mastership is transferred at the boundary of bus cycles. Namely, bus mastership is
immediately after receiving a bus request when a bus cycle is not being performed. Th
bus mastership is delayed until the bus cycle is complete when a bus cycle is in progre
when from outside the LSI it looks like a bus cycle is not being performed, a bus cycle
performing internally, started by inserting wait cycles between access cycles. Therefor
be immediately determined whether or not bus mastership has been released by lookin
signal or other bus control signals. The states that do not allow bus mastership release
below.

master clock stops because a transition is underway to standby mode or the frequency i
or if this LSI is being reset, the master device cannot release the bus. To prevent the sla
issuing bus requests in such a case, the slave must be put into the sleep state so that no
access cycles are generated.

The refresh request and bus request are accepted during the DMA burst transfer.

Bus mastership is maintained until a new bus request is received. Bus mastership is rele
immediately after the completion of the bus cycle in progress when an external bus req
($\overline{\text{BREQ}}$) is asserted (low level) and a bus acknowledge signal ($\overline{\text{BACK}}$) is asserted (low
use is resumed when a negation (high level) of $\overline{\text{BREQ}}$, which shows that the slave has r
bus, has been received.

SDRAM issues all bank pre-charge commands (PALLs) when active banks exist and re
bus after completion of a PALL command.

The bus release sequence is as follows. The address bus and data bus are placed in a hig
impedance state synchronized with the rising edge of CKIO. The bus mastership enable
asserted 0.5 cycles after the above timing, synchronized with the falling edge of CKIO.
control signals ($\overline{\text{BS}}$, $\overline{\text{CS}}$n, $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, DQMxx, $\overline{\text{WE}}$n, $\overline{\text{RD}}$, and RD/$\overline{\text{W}}$
made to the high-impedance states at the subsequent rising edge of CKIO. Bus request
sampled at the falling edge of CKIO. The sequence for re-claiming bus mastership from
described below. After detecting the negation of $\overline{\text{BREQ}}$ at the falling edge of CKIO, the
enable signal is negated at the subsequent falling edge of the clock. The address and da
are driven at the subsequent rising edge of CKIO. Figure 7.34 shows the bus arbitration

Figure 7.34 Bus Arbitration

In an original slave device designed by the user, multiple bus accesses are generated c
to reduce the overhead caused by bus arbitration. In this case, to execute SDRAM refr
correctly, the slave device must be designed to release the bus mastership within the r
interval time.

The bus release by the $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ signal handshaking requires some overhead
has many tasks, multiple bus cycles should be executed in a bus mastership acquisitio
the cycles required for master to slave bus mastership transitions streamlines the syste

7.13 Others

Reset: The bus state controller (BSC) can be initialized completely only at power-on r
power-on reset, all signals are negated and output buffers are turned off regardless of t
state. All control registers are initialized.

In standby, sleep, and manual reset, control registers of the bus state controller are not
At manual reset, the current bus cycle being executed is completed and then the acces
is entered. Since the RTCNT continues counting up during manual reset signal asserti
request occurs to initiate the refresh cycle.

Note that arbitration requests using $\overline{\text{BREQ}}$ are not accepted during manual reset signa

On-Chip Peripheral Module Access: To access an on-chip module register, two or r
peripheral module clock (Pϕ) cycles are required. Care must be taken in system desig

- Four channels (two channels can receive an external request)
- 4-Gbyte physical address space
- Data transfer unit: Byte, word (2 bytes), longword (4 bytes), and 16 bytes (longwo
- Maximum transfer count: 16777216 transfers
- Address mode: Dual address mode or single address mode can be selected.
- Transfer requests: External request, on-chip peripheral module request, or auto req selected.

  The following modules can issue an on-chip peripheral module request.

  SCIF0, SCIF2, CMT, USB, and A/D converter
- Bus modes: Cycle steal mode (normal mode and intermittent mode 16/64) or burst be selected.
- Selectable channel priority levels:

  The channel priority levels are selectable between fixed mode and round-robin mo
- Interrupt request: An interrupt request can be generated to the CPU after transfers
- External request detection: Low-/high-level or rising/falling edge detection of DRI be selected.
- Transfer request acknowledge signal: Active levels for DACK can be set independ
- Transfer end signal: Active level for TEND can be set. TEND is output at the sam DACK in the last DMA transfer. (Only channel 0)

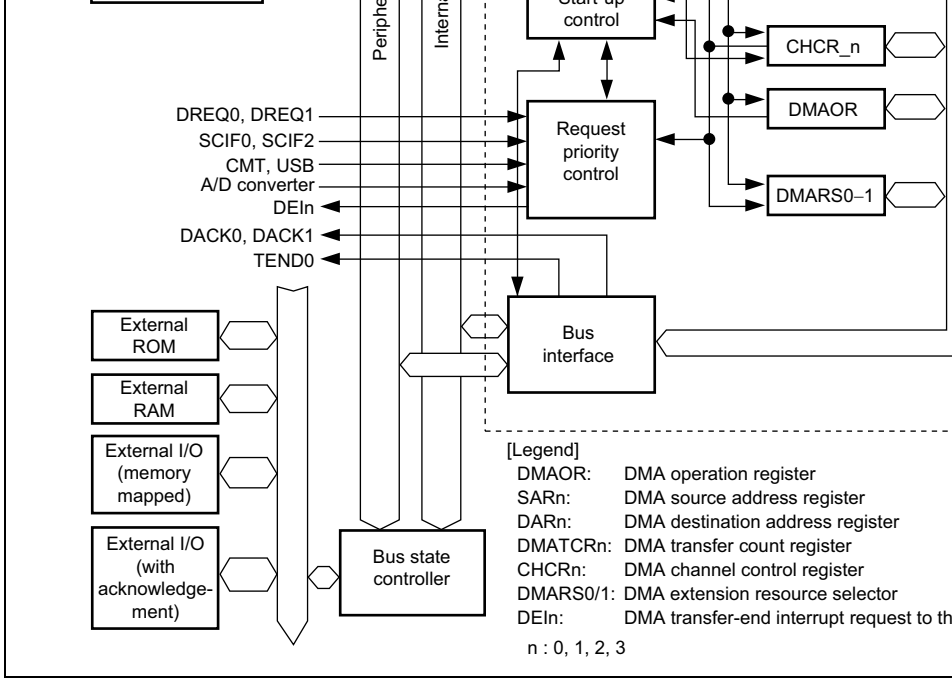


Figure 8.1 Block Diagram of DMAC

Channel	Name	Symbol	I/O	Function
0	DMA transfer request	DREQ0	I	DMA transfer request input external device to channel 0
	DMA transfer request acknowledge	DACK0	O	DMA transfer request ackno output from channel 0 to ex
	DMA transfer end	TEND0	O	Transfer end output in chan
1	DMA transfer request	DREQ1	I	DMA transfer request input external device to channel 1
	DMA transfer request acknowledge	DACK1	O	DMA transfer request ackno output from channel 1 to ex

8.3 Register Descriptions

The DMAC has the following registers. See section 24, List of Registers, for the addre these registers and the states of them in each processing state. The SAR for channel 0 such as SAR_0.

1. Channel 0

- DMA source address register_0 (SAR_0)
- DMA destination address register_0 (DAR_0)
- DMA transfer count register_0 (DMATCR_0)
- DMA channel control register_0 (CHCR_0)

2. Channel 1

- DMA source address register_1 (SAR_1)
- DMA destination address register_1 (DAR_1)
- DMA transfer count register_1 (DMATCR_1)
- DMA channel control register _1 (CHCR_1)

- DMA destination address register_3 (DAR_3)
- DMA transfer count register_3 (DMATCR_3)
- DMA channel control register_3 (CHCR_3)

5. Common
- DMA operation register (DMAOR)
- DMA extended resource selector 0 (DMARS0)
- DMA extended resource selector 1 (DMARS1)

8.3.1 DMA Source Address Registers (SAR)

SAR are 32-bit readable/writable registers that specify the source address of a DMA tra
During a DMA transfer, these registers indicate the next source address. When the data
external device with DACK is transferred in single address mode, SAR is ignored.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit addres
When transferring data in 16-byte units, a 16-byte boundary must be set for the source
value. The initial value is undefined. The SAR retains the current value in software sta
module standby mode.

8.3.2 DMA Destination Address Registers (DAR)

DAR are 32-bit readable/writable registers that specify the destination address of a DM
During a DMA transfer, these registers indicate the next destination address. When the
external device with DACK is transferred in single address mode, DAR is ignored.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit addres
When transferring data in 16-byte units, a 16-byte boundary must be set for the source
value. The initial value is undefined. The DAR retains the current value in software sta
module standby mode.

The DMATCR retains the current value in software standby or module standby mode.

8.3.4 DMA Channel Control Registers (CHCR)

CHCR are 32-bit readable/writable registers that control the DMA transfer mode.

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 24	—	0	R	Reserved These bits are always read as 0. The write value s be 0.
23	DO	0	R/W	DMA Overrun Selects whether DREQ is detected by overrun 0 o 1. This bit is valid only in CHCR_0 and CHCR_1. always read as 0 in CHCR_2 and CHCR_3. The w should always be 0. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1
22	TL	0	R/W	Transfer End Level Selects whether the TEND signal output is high ac active. This bit is valid only in CHCR_0. There are pins in CHCR_1 to CHCR_3. Therefore this settin This bit is always read as 0. The write value shoul 0. 0: Low-active output of TEND 1: High-active output of TEND
21 to 18	—	0	R	Reserved These bits are always read as 0. The write value s be 0.

				should always be 0. 0: DACK output in read cycle (Dual address mode) 1: DACK output in write cycle (Dual address mode)
16	AL	0	R/W	Acknowledge Level Specifies the DACK signal output is high active or l This bit is valid only in CHCR_0 and CHCR_1. This always read as 0 in CHCR_2 and CHCR_3. The wr should always be 0. 0: Low-active output of DACK 1: High-active output of DACK
15	DM1	0	R/W	Destination Address Mode
14	DM0	0	R/W	Specify whether the DMA destination address is inc decremented, or left fixed. (In single address mode and DM0 bits are ignored when data is transferred external device with DACK.) 00: Fixed destination address (setting prohibited in 16-byte transfer) 01: Destination address is incremented (+1 in byte- transfer, +2 in word-size transfer, +4 in longwor transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in byte transfer, –2 in word-size transfer, –4 in longwor transfer, setting prohibited in 16-byte transfer) 11: Setting prohibited

				01: Source address is incremented (+1 in byte-siz in word-size transfer, +4 in longword-size trans 16-byte transfer) 10: Source address is decremented (–1 in byte-siz in word-size transfer, –4 in longword-size trans prohibited in 16-byte transfer) 11: Setting prohibited
11	RS3	0	R/W	Resource Select
10	RS2	0	R/W	Specifies which transfer requests will be sent to th changing of transfer request source should be don that the DMA enable bit (DE) is set to 0. 0000: External request, dual address mode 0001: Setting prohibited 0010: External request/single address mode External address space → external device w 0011: External request/single address mode External device with DACK → external addr 0100: Auto request 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: DMA extended resource selector specificati 1001: Setting prohibited 1010: Setting prohibited 1011: Setting prohibited 1100: Setting prohibited 1101: Setting prohibited 1110: Peripheral module request, A/D converter 1111: Peripheral module request, CMT Note: External request specification is valid only in and CHCR_1. None of the external request can be set in channels CHCR_2 and CHCR
9	RS1	0	R/W	
8	RS0	0	R/W	

				is specified, these bits are invalid. 00: DREQ detected in low level 01: DREQ detected at falling edge 10: DREQ detected in high level 11: DREQ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode Specifies the bus mode when DMA transfers data. 0: Cycle steal mode 1: Burst mode
4	TS1	0	R/W	Transfer Size
3	TS0	0	R/W	Specify the size of data to be transferred. Select the size of data to be transferred when the s destination is an on-chip peripheral module register transfer size is specified. 00: Byte size 01: Word size (two bytes) 10: Longword size (four bytes) 11: 16-byte unit (four longword transfers)
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is gen CPU at the end of the DMA transfer. Setting this bit generates an interrupt request (DEI) to the CPU wh bit is set to 1. 0: Interrupt request disabled 1: Interrupt request enabled

				• DMA transfer is ended by clearing the DE bit a bit in the DMA operation register (DMAOR). This bit can only be cleared by writing 0 after read the DE bit is set to 1 while this bit is set to 1, trans enabled. 0: During the DMA transfer or DMA transfer has b [Clearing conditions] • Writing 0 after reading TE = 1 • Power-on reset • Manual reset 1: Data transfer ends by the specified count (DMA
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In auto req DMA transfer starts by setting the DE bit and DME DMAOR to 1. In this time, all of the bits TE, NMIF and AE must be 0. In an external request or peripl request, DMA transfer starts if DMA transfer reque generated by the devices or peripheral modules a the bits DE and DME to 1. In this case, however, a TE, NMIF, and AE must be 0 an in the case of aut mode. Clearing the DE bit to 0 can terminate the D 0: DMA transfer disabled 1: DMA transfer enabled

Note: * Only 0 can be written for clearing the flags.

				always be 0.
13	CMS1	0	R/W	Cycle Steal Mode Select
12	CMS0	0	R/W	Select either normal mode or intermittent mode in mode. It is necessary that all channels' bus modes are se steal mode to make valid intermittent mode. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Executes one DMA transfer in each of 16 clock external bus clock. 11: Intermittent mode 64 Executes one DMA transfer in each of 64 clock external bus clock.
11, 10	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.
9	PR1	0	R/W	Priority Mode
8	PR0	0	R/W	Select the priority level between channels when th transfer requests for multiple channels simultaneo 00: CH0 > CH1 > CH2 > CH3 01: CH0 > CH2 > CH3 > CH1 10: Setting prohibited 11: Round-robin mode
7 to 3	—	0	R	Reserved These bits are always read as 0. The write value s always be 0.

				• Writing 0 after reading AE = 1 • Power-on reset • Manual reset 1: DMAC address error. DMA transfer disabled. [Setting condition] DMAC address error occurrence
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. If this bi transfer is not enabled even if the DE bit in CHCR DME bit in DMAOR are set to 1. This bit can only by writing 0 after reading 1. When the NMI is input, the DMA transfer in progr done in one transfer unit. When the DMAC is not the NMIF bit is set to 1 even if the NMI interrupt w 0: No NMI interrupt [Clearing conditions] • Writing 0 after reading NMIF = 1 • Power-on reset • Manual reset 1: NMI input. DMA transfer disabled. [Setting condition] NMI interrupt occurrence
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfers on all channe DME bit and the DE bit in CHCR are set to 1, DM are enabled. In this time, all of the bits TE in CHC DMAOR, and AE must be 0. If this bit is cleared c transfer, transfers in all the channels can be term 0: Disable DMA transfers on all channels 1: Enable DMA transfers on all channels

Note: * Only 0 can be written for clearing the flags.

transfer request source is not accepted.

- DMARS0

Bit	Bit Name	Initial Value	R/W	Description
15	C1MID5	0	R/W	Transfer request source module ID5 to ID0 for D
14	C1MID4	0	R/W	1 (MID)
13	C1MID3	0	R/W	See table 8.2.
12	C1MID2	0	R/W	
11	C1MID1	0	R/W	
10	C1MID0	0	R/W	
9	C1RID1	0	R/W	Transfer request resource register ID1 to ID0 for
8	C1RID0	0	R/W	channel 1 (RID)
				See table 8.2.
7	C0MID5	0	R/W	Transfer request source module ID5 to ID0 for D
6	C0MID4	0	R/W	0 (MID)
5	C0MID3	0	R/W	See table 8.2
4	C0MID2	0	R/W	
3	C0MID1	0	R/W	
2	C0MID0	0	R/W	
1	C0RID1	0	R/W	Transfer request resource register ID1 to ID0 for
0	C0RID0	0	R/W	channel 0 (RID)
				See table 8.2.

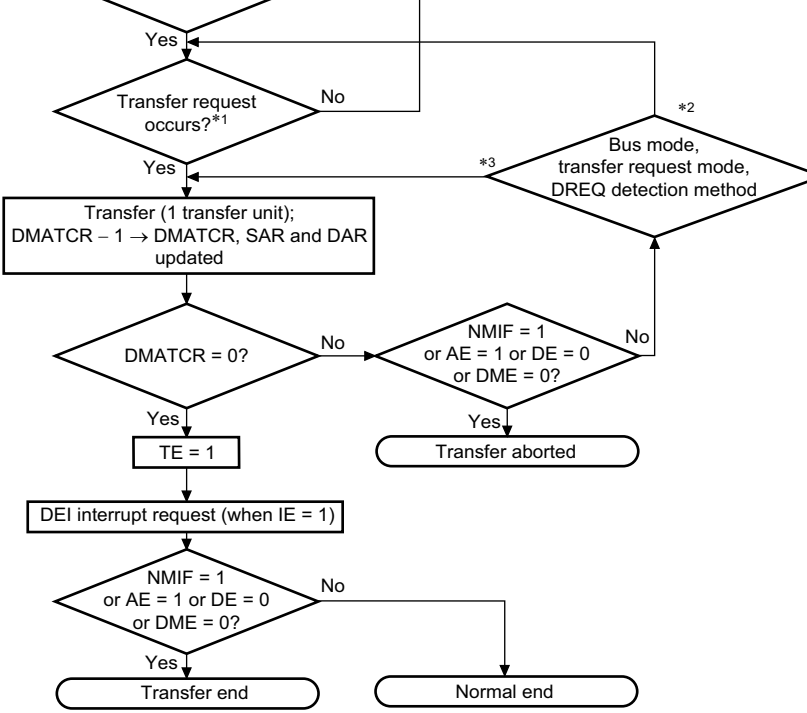
7 to 2	C2MID5 to C2MID0	0	R/W	Transfer request resource module ID5 to ID0 for channel 2 (MID) See table 8.2.
1	C2RID1	0	R/W	Transfer request resource register ID1 and ID0 f channel 2 (RID) See table 8.2.
0	C2RID0	0	R/W	

Table 8.2 Transfer Request Sources

Peripheral Module	Setting Value for One Channel (MID + RID)	MID	RID	Function
SCIF0	H'21	B'001000	B'01	Transmit
	H'22		B'10	Receive
SCIF2	H'29	B'001010	B'01	Transmit
	H'2A		B'10	Receive
USB	H'73	B'011100	B'11	Transmit
	H'70		B'00	Receive

After the DMA source address registers (SAR), DMA destination address registers (DA
transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA op
register (DMAOR), and DMA extended resource selector (DMARS) are set, the DMAC
data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request is generated and transfer is enabled, the DMAC transfers 1
unit of data (depending on the TS0 and TS1 settings). For an auto request, the transf
automatically when the DE bit and DME bit are set to 1. The DMATCR value will
decremented for each transfer. The actual transfer flows vary by address mode and
3. When the specified number of transfer have been completed (when DMATCR reach
transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt
the CPU.
4. When an address error or an NMI interrupt is generated, the transfer is aborted. Tra
also aborted when the DE bit in CHCR or the DME bit in DMAOR are cleared to 0



Notes: 1. In auto-request mode, transfer begins when NMIF, AE, and TE bits are 0 and the DE and
bits are set to 1.
2. DREQ = level detection in burst mode (external request), or cycle-steal mode.
3. DREQ = edge detection in burst mode (external request), or auto-request mode in burst

Figure 8.2 DMAC Transfer Flowchart

memory-to-memory transfer or a transfer between memory and an on-chip peripheral m
unable to request a transfer, auto-request mode allows the DMAC to automatically gen
transfer request signal internally. When the DE bit in CHCR and the DME bit in DMA
to 1, the transfer begins so long as the AE and NMIF bits in DMAOR are 0.

External Request Mode: In this mode a transfer is performed at the request signals (D
DREQ1) of an external device. This mode is valid only in channels 0 and 1. Choose on
modes shown in table 8.3 according to the application system. When this mode is selec
DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer is
upon a request at the DREQ input.

Table 8.3 Selecting External Request Modes with RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destinatio
0	0	0	0	Dual address mode	Any	Any
		1	0	Single address mode	External memory, memory-mapped external device	External de DACK
			1		External device with DACK	External m memory-m external de

Choose to detect DREQ by either the edge or level of the signal input with the DL bit a
of CHCR_0 and CHCR_1 as shown in table 8.4. The source of the transfer request doe
to be the data transfer source or destination.

When DREQ is accepted, the DREQ pin becomes request accept disabled state. After
acknowledge signal DACK for the accepted DREQ, the DREQ pin again becomes req
enabled state.

When DREQ is used for level detection, there are the following two cases depending
to detect the next DREQ after outputting DACK. A case wherein transfer is aborted af
number of transfers has been performed as requests (overrun 0) and wherein another t
aborted after transfers have been performed for (the number of requests plus 1) times
The DO bit in CHCR selects overrun 0 or overrun 1.

Table 8.5 Selecting External Request Detection with DO Bit

CHCR_0 or CHCR_1

DO	External Request
0	Overrun 0
1	Overrun 1

When a transmit data empty transfer request of the SCIF is set as the transfer request, t
destination must be the SCIF's transmit data register. Likewise, when receive data full
request of the SCIF is set as the transfer request, the transfer source must be the SCIF's
data register. These conditions also apply to the USB. Any address can be specified for
and destination, when transfer request is generated by the CMT.

Table 8.6 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS

RS3	RS2	RS1	RS0	DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination
1	1	1	0	ADC	AD-conversion end request	ADDR	Any
1	1	1	1	CMT	Compare-match transfer request	Any	Any

Table 8.7 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destinat
RS[3:0]	MID	RID				
1000	001000	01	SCIF0 transmitter	TXI0 (transmit FIFO data empty)	Any	SCFTDR
		10	SCIF0 receiver	RXI0 (receive FIFO data full)	SCFRDR_0	Any
	001010	01	SCIF2 transmitter	TXI2 (transmit FIFO data empty)	Any	SCFTDR
		10	SCIF2 receiver	RXI2 (receive FIFO data full)	SCFRDR_2	Any
	011100	11	USB transmitter	EP2 FIFO empty transfer request	Any	EPDR2
		00	USB receiver	EP1 FIFO full transfer request	EPDR1	Any

CH0 > CH2 > CH3 > CH1

These are selected by the PR1 and the PR0 bits in the DMA operation register (DMAC

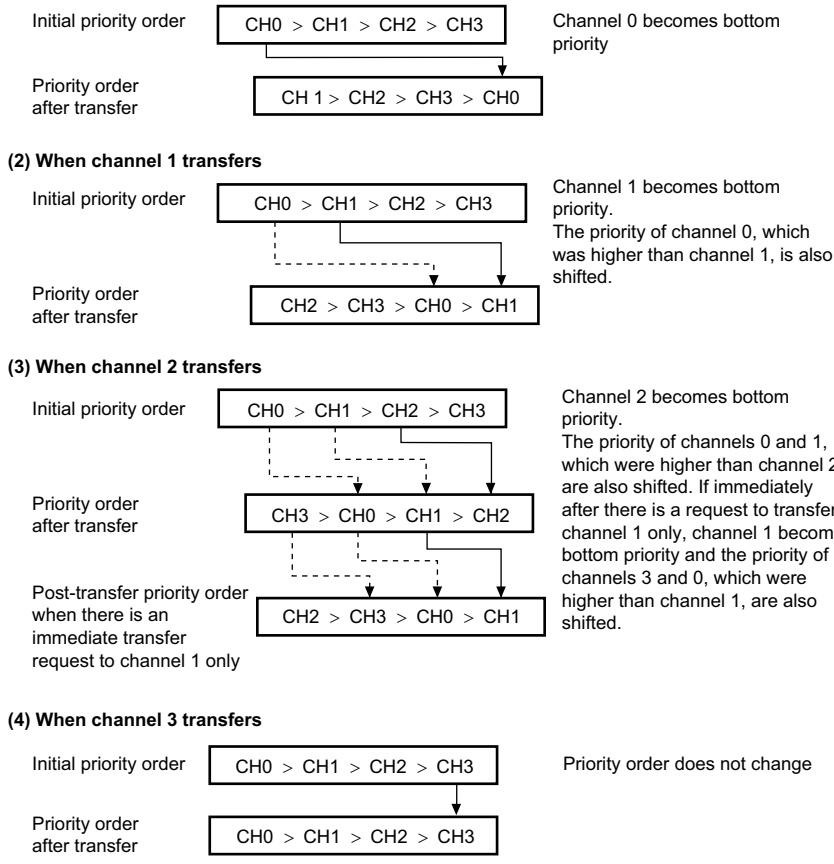


Figure 8.3 Round-Robin Mode

4. When the channel 0 transfer ends, channel 0 becomes lowest priority.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 trans (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 becomes lowest priority.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 shift downward in priority so t 3 becomes the lowest priority.

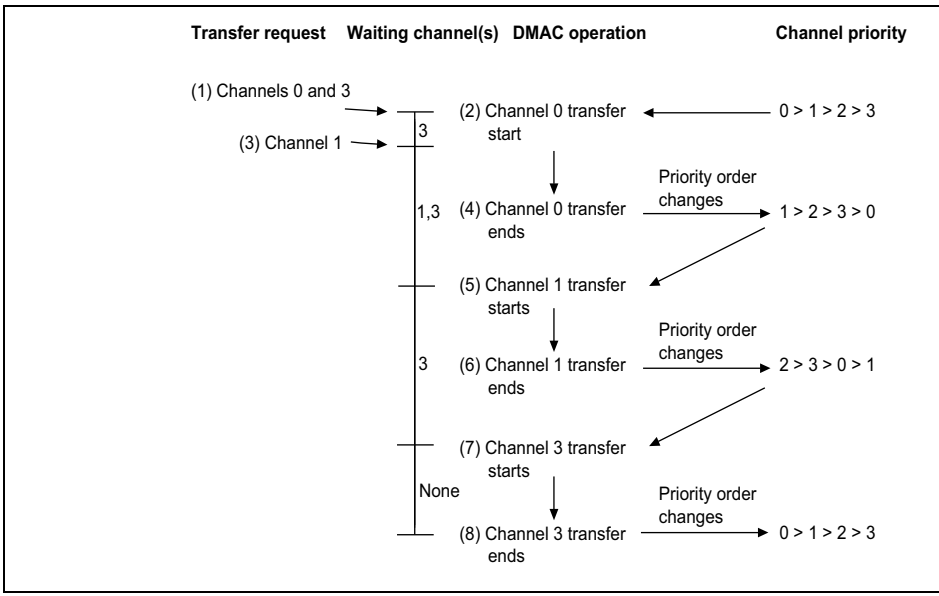
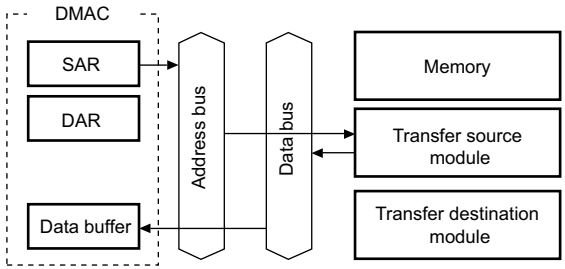


Figure 8.4 Channel Priority in Round-Robin Mode

	Destination			
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip P Module**
External device with DACK	Not available	Dual, single	Dual, single	Not availab
External memory	Dual, single	Dual	Dual	Dual
Memory-mapped external device	Dual, single	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual

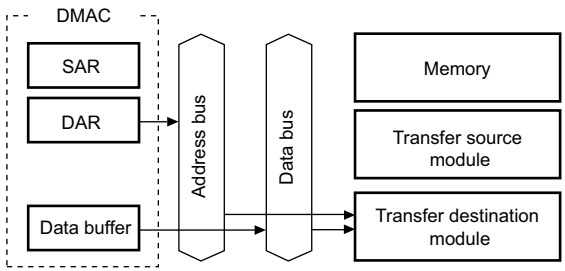
Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. A 16-byte transfer is available only for the registers to which longword-size a enabled in the on-chip peripheral modules.

then that data is written to the other external memory in a write cycle.



The SAR value is an address, data is read from the transfer source module, and the data is temporarily stored in the DMAC.

First bus cycle



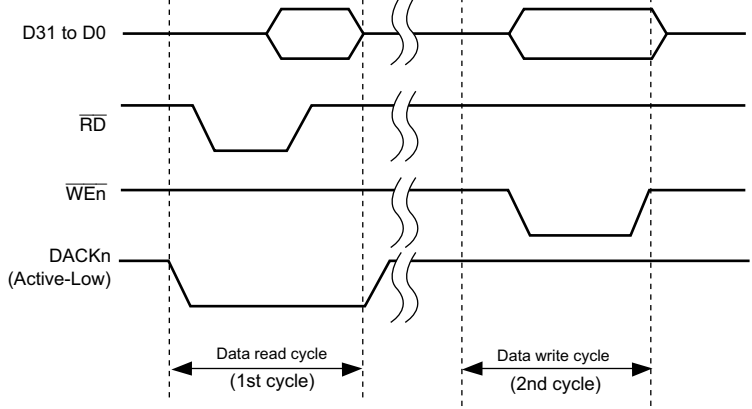
The DAR value is an address and the value stored in the data buffer in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 8.5 Data Flow of Dual Address Mode

Auto request, external request, and on-chip peripheral module request are available transfer request. DACK can be output in read cycle or write cycle in dual address m Channel control register (CHCR) can specify whether the DACK is output in read write cycle.

Figure 8.6 shows an example of DMA transfer timing in dual address mode.



Note: In transfer between external memories, with DACK output in the read cycle, DACK output timing is the same as that of CSn.

Figure 8.6 Example of DMA Transfer Timing in Dual Mode (Source: Ordinary Memory, Destination: Ordinary Memory)

- Single Address Mode

  In single address mode, either the transfer source or transfer destination peripheral c accessed (selected) by means of the DACK signal, and the other device is accessed In this mode, the DMAC performs one DMA transfer in one bus cycle, accessing or external devices by outputting the DACK transfer request acknowledge signal to it, same time outputting an address to the other device involved in the transfer. For exa the case of transfer between external memory and an external device with DACK sh figure 8.7, when the external device outputs data to the data bus, that data is written external memory in the same bus cycle.

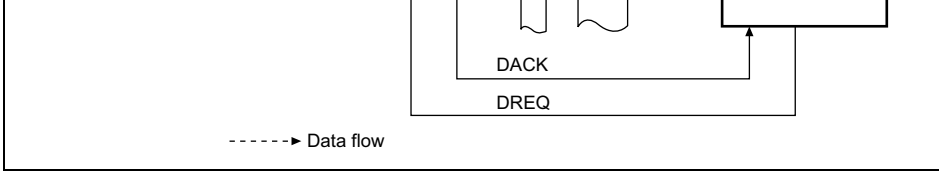


Figure 8.7 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an device with DACK and a memory-mapped external device, and (2) transfer betwe external device with DACK and external memory. In both cases, only the external signal (DREQ) is used for transfer requests.

Figures 8.8 shows example of DMA transfer timing in single address mode.

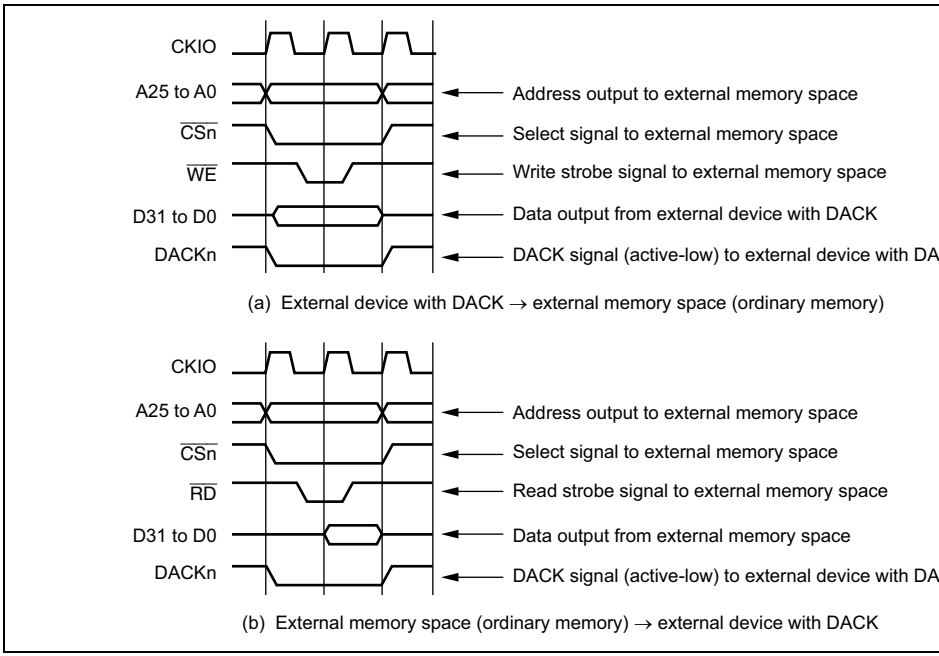


Figure 8.8 Example of DMA Transfer Timing in Single Address Mod

bus master. This is repeated until the transfer end conditions are satisfied.

In cycle-steal mode, transfer areas are not affected regardless of settings of the trans source, transfer source, and transfer destination. Figure 8.9 shows an example of DM timing in cycle steal mode. Transfer conditions shown in the figure are:

— Dual address mode

— DREQ low level detection

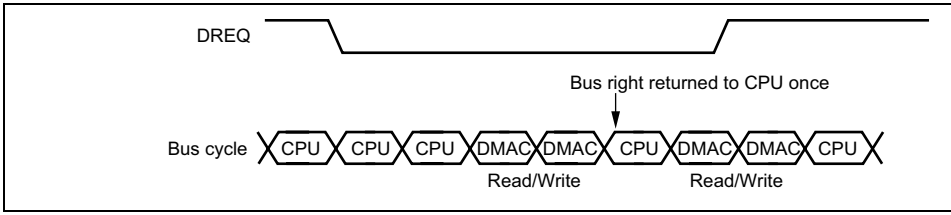


Figure 8.9 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection)

- Intermittent Mode 16 and Intermittent Mode 64

  In intermittent mode of cycle steal, DMAC returns the bus right to other bus master a unit of transfer (byte, word, longword, or 16 bytes) is complete. If the next transfe occurs after that, DMAC gets the bus right from other bus master after waiting for 1 clocks in Bϕ count. DMAC then transfers data of one unit and returns the bus right master. These operations are repeated until the transfer end condition is satisfied. It possible to make lower the ratio of bus occupation by DMA transfer than the norma cycle steal.

  When DMAC gets again the bus right, DMA transfer can be postponed in case of er updating due to cache miss.

  This intermittent mode can be used for all transfer section; transfer requester, source destination. The bus modes, however, must be cycle steal mode in all channels.

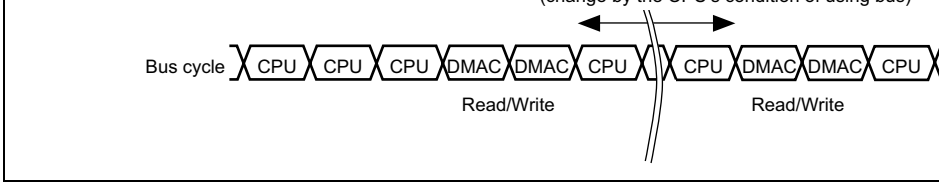


Figure 8.10 Example of DMA Transfer in Cycle Steal Intermittent Mo (Dual Address, DREQ Low Level Detection)

b. Burst Mode

Once the bus right is obtained, the transfer is performed continuously until the transfe condition is satisfied. In external request mode with low level detection of the DREQ however, when the DREQ pin is driven high, the bus passes to the other bus master af DMAC transfer request that has already been accepted ends, even if the transfer end c have not been satisfied.

Burst mode cannot be used for other than the CMT when the on-chip peripheral modu transfer request source. Figure 8.11 shows DMA transfer timing in burst mode.

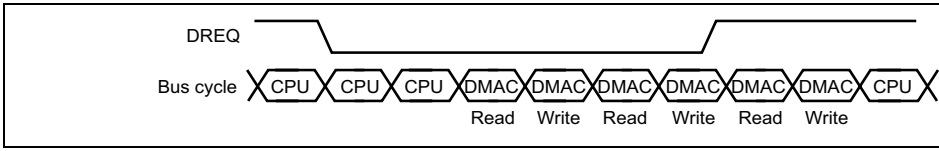


Figure 8.11 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

	mapped external device			
	External memory and external memory	All*1	B/C	8/16/32/128
	External memory and memory-mapped external device	All*1	B/C	8/16/32/128
	Memory-mapped external device and memory-mapped external device	All*1	B/C	8/16/32/128
	External memory and on-chip peripheral module	All*2	B/C*3	8/16/32/128*
	Memory-mapped external device and on-chip peripheral module	All*2	B/C*3	8/16/32/128*
	On-chip peripheral module and on-chip peripheral module	All*2	B/C*3	8/16/32/128*
Single	External device with DACK and external memory	External	B/C	8/16/32
	External device with DACK and memory-mapped external device	External	B/C	8/16/32

B: Burst, C: Cycle steal

Notes: 1. External requests, auto requests, and on-chip peripheral module requests ar available. In the case of on-chip peripheral module requests, however, the C available.

2. External requests, auto requests, and on-chip peripheral module requests ar available. However, with the exception of the CMT, the module must be desi the transfer request source or the transfer destination.
3. Only cycle steal except for the CMT as the transfer source.
4. Access size permitted for the on-chip peripheral module register functioning transfer source or transfer destination.
5. If the transfer request is an external request, channels 0 and 1 are only avail

Bus Mode and Channel Priority Order: If, when the priority is set to fixed mode (CH and channel 1 is transferring in burst mode, a transfer request to channel 0, which has h priority, is generated, the transfer on channel 0 will begin immediately.

At this time, if channel 0 is set to burst mode, the transfer on channel 1 will resume wh channel 0 transfer has completely finished.

master until all of the competing burst mode transfers have completed.

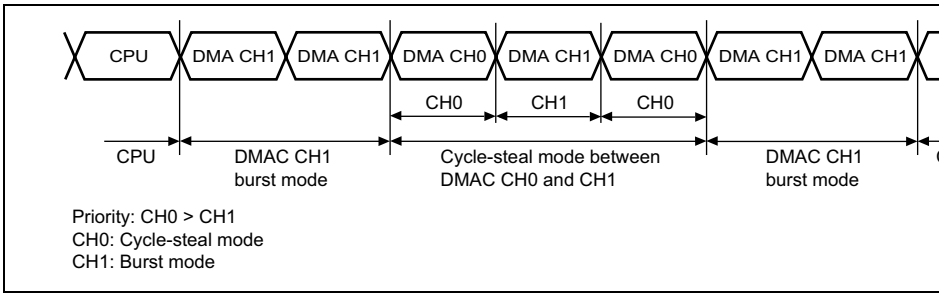


Figure 8.12 Bus State when Multiple Channels are Operating

Cycle-steal mode channels and burst mode channels should not be mixed in round-rob
Doing so runs the risk that priority changes may not be made properly, although the in
channel transfer operations will be performed correctly.

8.4.5 Number of Bus Cycle States and DREQ Pin Sampling Timing

Number of Bus Cycle States: When the DMAC is the bus master, the number of bus
is controlled by the bus state controller (BSC) in the same way as when the CPU is the
For details, see section 7, Bus State Controller (BSC).

DREQ Pin Sampling Timing:

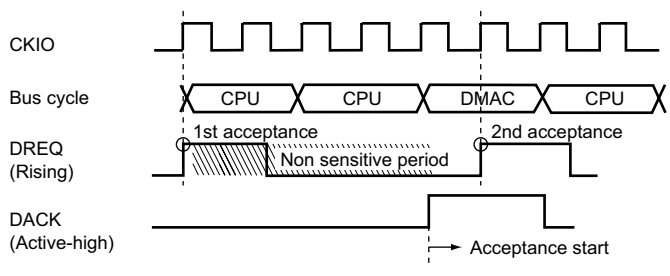


Figure 8.13 Example of DREQ Input Detection in Cycle Steal Mode Edge D

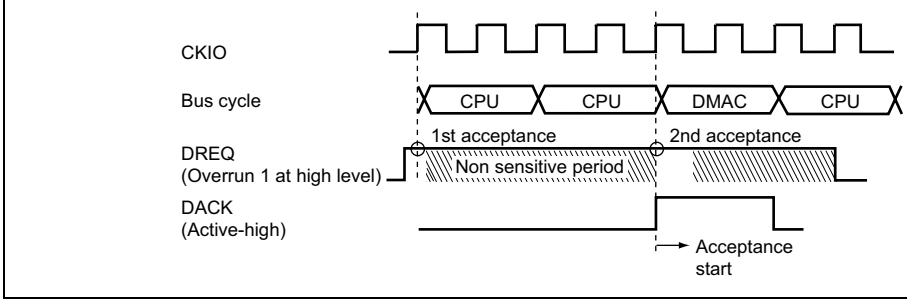


Figure 8.14 Example of DREQ Input Detection in Cycle Steal Mode Level De

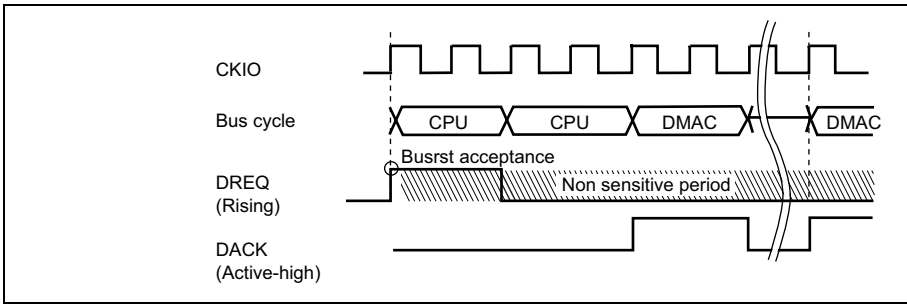


Figure 8.15 Example of DREQ Input Detection in Burst Mode Edge Detec

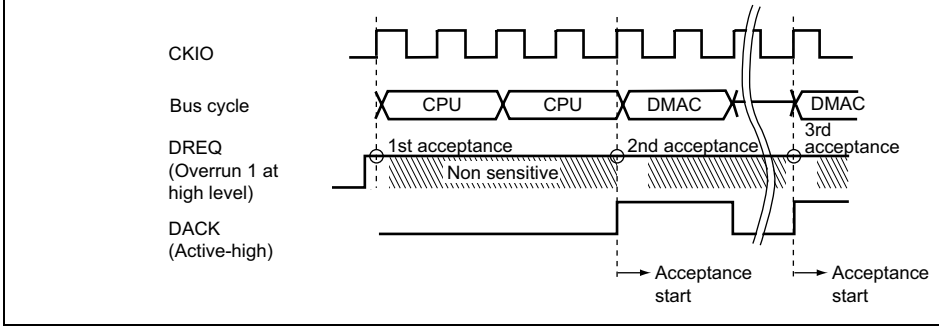


Figure 8.16 Example of DREQ Input Detection in Burst Mode Level Dete

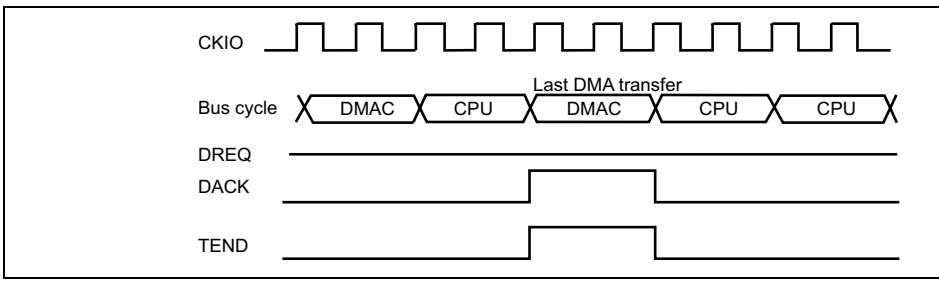
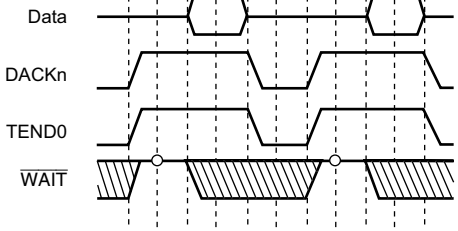


Figure 8.17 Example of DMA Transfer End Signal (in Cycle Steal Level De



Note: TEND0 is asserted during the last transfer unit of DMA transfer. When the transfer unit is divided into several bus cycles and $\overline{CS}$ is negated between bus cycles, TEND0 is also divided.

Figure 8.18 BSC Ordinary Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

8.5 Precautions

8.5.1 Precautions when Mixing Cycle-Steal Mode Channels and Burst Mode C

Transfer mode settings should not fulfill conditions (1) and (2) below at the same time.

(1) DMA transfer takes place using multiple channels, some of which operate in the bu and some in the cycle-steal mode.

(2) A channel that uses the burst mode is set to dual address mode and DACK is output write cycle.

Selection of seven clock modes depending on the frequency ranges and crystal osc
external clock input.

- Three clocks generated independently

  An internal clock for the CPU and cache (Iϕ); a peripheral clock (Pϕ) for the perip
  modules; a bus clock (Bϕ = CKIO) for the external bus interface.

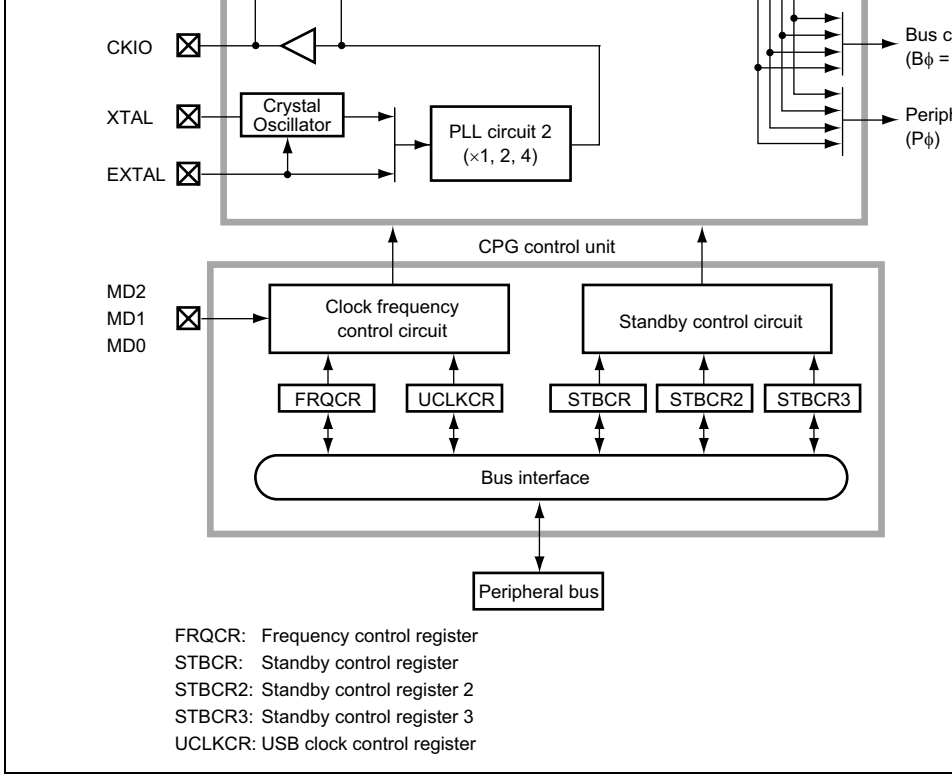
- Frequency change function

  Internal and peripheral clock frequencies can be changed independently using the
  loop (PLL) circuit and divider circuit within the CPG. Frequencies are changed by
  using the frequency control register (FRQCR) settings.

- Power-down mode control

  The clock can be stopped for sleep mode and software standby mode and specific
  be stopped using the module standby function.

A block diagram of the CPG is shown in figure 9.1.



FRQCR: Frequency control register
STBCR: Standby control register
STBCR2: Standby control register 2
STBCR3: Standby control register 3
UCLKCR: USB clock control register

Figure 9.1 Block Diagram of Clock Pulse Generator

the crystal oscillator or EXTAL pin. The multiplication ratio is fixed by the clock-
mode. The clock-operating mode is set by pins MD0, MD1, and MD2. For more d
clock operating modes, refer to table 9.2.

3. Crystal Oscillator

   This oscillator circuit is used when a crystal resonator is connected to the XTAL a
   pins. This crystal oscillator operates according to the clock operating mode setting

4. Divider 1

   Divider 1 generates a clock at the operating frequency used by the internal or perip
   The operating frequency can be 1, 1/2, 1/3, or 1/4 times the output frequency of PI
   as long as it stays at or above the clock frequency of the CKIO pin. The division ra
   the frequency control register.

5. Clock Frequency Control Circuit

   The clock frequency control circuit controls the clock frequency using the MD0, M
   MD2 pins and the frequency control register.

6. Standby Control Circuit

   The standby control circuit controls the state of the on-chip oscillator and other mo
   clock switching and software/standby modes.

7. Frequency Control Register

   The frequency control register has control bits assigned for the following functions
   output/non-output from the CKIO pin, the frequency multiplication ratio of PLL ci
   the frequency division ratio of the internal clock and the peripheral clock.

8. Standby Control Register

   The standby control register has bits for controlling the power-down modes. See se
   Power-Down Modes, for more information.

9. USB Clock Control Register

   The source clock generating the USB clock is set in the USB clock control register

	MD2	I	Set the clock-operating mode.
Crystal oscillator pins for system clock (clock input pins)	XTAL	O	Connects a crystal resonator.
	EXTAL	I	Connects a crystal resonator. Also used to in external clock.
Clock I/O pin	CKIO	I/O	Inputs or outputs an external clock.
Crystal oscillator pins for USB (clock input pins)	XTAL_USB	O	Connects a crystal resonator for the USB.
	EXTAL_USB	I	Connects a crystal resonator for the USB. Al input an external clock.

Note: The values of the mode control pins are sampled only in a power-on reset. This the erroneous operation of the LSI.

Mode	MD2	MD1	MD0	Source	Output	PLL2 On/Off	PLL1 On/Off
0	0	0	0	EXTAL	CKIO	ON (× 1)	ON (× 1, 2, 3, 4)
1	0	0	1	EXTAL	CKIO	ON (× 4)	ON (× 1, 2, 3, 4)
2	0	1	0	Crystal resonator	CKIO	ON (× 4)	ON (× 1, 2, 3, 4)
4	1	0	0	Crystal resonator	CKIO	ON (× 1)	ON (× 1, 2, 3, 4)
5	1	0	1	EXTAL	CKIO	ON (× 2)	ON (× 1, 2, 3, 4)
6	1	1	0	Crystal resonator	CKIO	ON (× 2)	ON (× 1, 2, 3, 4)
7	1	1	1	CKIO	—	OFF	ON (× 1, 2, 3, 4)

Mode 0: An external clock is input from the EXTAL pin and executes waveform shap
circuit 2 before being supplied inside this LSI.

Mode 1: An external clock is input from the EXTAL pin and its frequency is multipli
PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency externa
used.

Mode 2: The on-chip crystal oscillator operates, with the oscillation frequency being
4 by PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency ex
to be used.

Mode 4: The on-chip crystal oscillator operates and executes waveform shaping by P
before being supplied inside this LSI.

Mode 5: An external clock is input from the EXTAL pin and its frequency is multipli
PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency externa
used.

Mode 6: The on-chip crystal oscillator operates, with the oscillation frequency being
2 by PLL circuit 2 before being supplied inside this LSI, allowing a low crystal freque
used.

0	H'1000	ON (× 1)	ON (× 1)	1:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1001	ON (× 1)	ON (× 1)	1:1:1/2	20.00 MHz to 66.67 MHz	20.00 MHz
	H'1003	ON (× 1)	ON (× 1)	1:1:1/4	20.00 MHz to 66.67 MHz	20.00 MHz
	H'1101	ON (× 2)	ON (× 1)	2:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1103	ON (× 2)	ON (× 1)	2:1:1/2	20.00 MHz to 66.67 MHz	20.00 MHz
	H'1111	ON (× 2)	ON (× 1)	1:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1113	ON (× 2)	ON (× 1)	1:1:1/2	20.00 MHz to 66.67 MHz	20.00 MHz
	H'1202	ON (× 3)	ON (× 1)	3:1:1	26.70 MHz to 33.34 MHz	26.70 MHz
	H'1222	ON (× 3)	ON (× 1)	1:1:1	26.70 MHz to 33.34 MHz	26.70 MHz
	H'1303	ON (× 4)	ON (× 1)	4:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1313	ON (× 4)	ON (× 1)	2:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1333	ON (× 4)	ON (× 1)	1:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
1, 2	H'1001	ON (× 1)	ON (× 4)	4:4:2	10.00 MHz to 16.67 MHz	40.00 MHz
	H'1003	ON (× 1)	ON (× 4)	4:4:1	10.00 MHz to 16.67 MHz	40.00 MHz
	H'1103	ON (× 2)	ON (× 4)	8:4:2	10.00 MHz to 16.67 MHz	40.00 MHz
	H'1113	ON (× 2)	ON (× 4)	4:4:2	10.00 MHz to 16.67 MHz	40.00 MHz
4	H'1000	ON (× 1)	ON (× 1)	1:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1001	ON (× 1)	ON (× 1)	1:1:1/2	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1003	ON (× 1)	ON (× 1)	1:1:1/4	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1101	ON (× 2)	ON (× 1)	2:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1103	ON (× 2)	ON (× 1)	2:1:1/2	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1111	ON (× 2)	ON (× 1)	1:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1113	ON (× 2)	ON (× 1)	1:1:1/2	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1202	ON (× 3)	ON (× 1)	3:1:1	26.70 MHz to 33.34 MHz	26.70 MHz
	H'1222	ON (× 3)	ON (× 1)	1:1:1	26.70 MHz to 33.34 MHz	26.70 MHz
	H'1303	ON (× 4)	ON (× 1)	4:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1313	ON (× 4)	ON (× 1)	2:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
	H'1333	ON (× 4)	ON (× 1)	1:1:1	20.00 MHz to 33.34 MHz	20.00 MHz

	H'1113	ON (× 2)	ON (× 2)	2:2:1	10.00 MHz to 33.34 MHz	20.00 MHz
	H'1202	ON (× 3)	ON (× 2)	6:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1222	ON (× 3)	ON (× 2)	2:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1303	ON (× 4)	ON (× 2)	8:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1313	ON (× 4)	ON (× 2)	4:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1333	ON (× 4)	ON (× 2)	2:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
6	H'1000	ON (× 1)	ON (× 2)	2:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1001	ON (× 1)	ON (× 2)	2:2:1	10.00 MHz to 33.34 MHz	20.00 MHz
	H'1003	ON (× 1)	ON (× 2)	2:2:1/2	10.00 MHz to 33.34 MHz	20.00 MHz
	H'1101	ON (× 2)	ON (× 2)	4:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1103	ON (× 2)	ON (× 2)	4:2:1	10.00 MHz to 33.34 MHz	20.00 MHz
	H'1111	ON (× 2)	ON (× 2)	2:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1113	ON (× 2)	ON (× 2)	2:2:1	10.00 MHz to 33.34 MHz	20.00 MHz
	H'1202	ON (× 3)	ON (× 2)	6:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1222	ON (× 3)	ON (× 2)	2:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1303	ON (× 4)	ON (× 2)	8:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1313	ON (× 4)	ON (× 2)	4:2:2	10.00 MHz to 16.67 MHz	20.00 MHz
	H'1333	ON (× 4)	ON (× 2)	2:2:2	10.00 MHz to 16.67 MHz	20.00 MHz

H'1113	ON (× 2)	OFF	1:1:1/2	20.00 MHz to 66.67 MHz	20.00 MHz
H'1202	ON (× 3)	OFF	3:1:1	26.70 MHz to 33.34 MHz	26.70 MHz
H'1222	ON (× 3)	OFF	1:1:1	26.70 MHz to 33.34 MHz	26.70 MHz
H'1303	ON (× 4)	OFF	4:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
H'1313	ON (× 4)	OFF	2:1:1	20.00 MHz to 33.34 MHz	20.00 MHz
H'1333	ON (× 4)	OFF	1:1:1	20.00 MHz to 33.34 MHz	20.00 MHz

Notes: 1. This LSI cannot operate in an FRQCR value other than that listed in table 9.
2. Taking input clock frequency ratio as 1.

Cautions:

1. The input to divider 1 is the output of the PLL circuit 1.
2. The frequency of the internal clock (Iϕ) is:
   - The product of the frequency of the CKIO pin, the frequency multiplication ratio
     circuit 1, and the division ratio of divider 1.
   - Do not set the internal clock frequency lower than the CKIO pin frequency.
3. The frequency of the peripheral clock (Pϕ) is:
   - The product of the frequency of the CKIO pin, the frequency multiplication ratio
     circuit 1, and the division ratio of divider 1.
   - The peripheral clock frequency should not be set higher than the frequency of th
     pin, higher than 33.34 MHz, or lower than 13 MHz when the USB is used.
4. ×1, ×2, ×3, or ×4 can be used as the multiplication ratio of PLL circuit 1. ×1, ×1/2, ×
   ×1/4 can be selected as the division ratios of divider 1. Set the rate in the frequency
   register.
5. The output frequency of PLL circuit 1 is the product of the CKIO frequency and the
   multiplication ratio of PLL circuit 1. Use the output frequency under 133.34 MHz.

The frequency control register (FRQCR) is a 16-bit readable/writable register used to
whether a clock is output from the CKIO pin, the on/off state of PLL circuit 1, PLL st
frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the
clock and the peripheral clock. Only word access can be used on FRQCR. As for the c
of the clock rate, refer to table 9.3. The combinations listed in table 9.3 should only be
FRQCR.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	0	R	Reserved These bits are always read as 0. The write should always be 0.
12	CKOEN	1	R/W	Clock Output Enable Specifies to output a clock from the CKIO the CKIO pin low when software standby i (after an interrupt before STATUS1 becom STATUS0 becomes low). The CKIO pin is during STATUS1 = low and STATUS0 = h the CKOEN bit is cleared to 0. Therefore, malfunction of an external circuit because unstable CKIO clock in releasing software mode can be prevented. In clock operating mode 7, the CKIO pin is state regardless of this bit. 0: Fixes the CKIO pin low in software stan 1: Outputs a clock from the CKIO pin.
11, 10	—	0	R	Reserved These bits are always read as 0. The write should always be 0.

				These bits are always read as 0. The write should always be 0.
5	IFC1	0	R/W	Internal Clock Frequency Division Ratio
4	IFC0	0	R/W	Specify the frequency division ratio of the in clock with respect to the output frequency o circuit 1. 00: × 1 time 01: × 1/2 time 10: × 1/3 time 11: × 1/4 time
3, 2	—	0	R	Reserved These bits are always read as 0. The write should always be 0.
1	PFC1	1	R/W	Peripheral Clock Frequency Division Ratio
0	PFC0	1	R/W	Specify the frequency division ratio of the p clock with respect to the output frequency o circuit 1. 00: × 1 time 01: × 1/2 time 10: × 1/3 time 11: × 1/4 time

7	USSCS1	1	R/W	Source Clock Selection Bit
6	USSCS0	1	R/W	These bits select the source clock. 00: Clock stopped 01: Setting prohibited 10: Setting prohibited 11: External input clock
5	USBEN	1	R/W	USB On-chip Oscillator Enable This bit controls the operation of the USB oscillator. 0: USB on-chip oscillator stopped 1: USB on-chip oscillator operates
4 to 0	—	0	R	Reserved These bits are always read as 0. The write should always be 0.

9.4.3 Usage Notes

Note the following when using the USB. If these are used incorrectly, the correct clock be generated, causing faulty operation of the USB.

1. UCLKCR is used only for generation of the USB clocks. When the USB is not used recommended that UCLKCR be cleared to H'00 to halt the clock.
2. Halt the USB before changing the values of UCLKCR. Halt the USB by stopping the supply using the USB module stop bits in STBCR3.
3. UCLKCR is initialized only by a power-on reset. In a manual reset, they retain the values.
4. Use the USB module with Pϕ > 13 MHz. Otherwise, the operation of this LSI is not guaranteed.

A PLL settling time is required when the multiplication rate of PLL circuit 1 is changed chip WDT counts the settling time.

1. In the initial state, the multiplication rate of PLL circuit 1 is 1.
2. Set a value that will become the specified oscillation settling time in the WDT and s WDT. The following must be set:
   WTCSR.TME = 0: WDT stops
   WTCSR.CKS[2:0]: Division ratio of WDT count clock
   WTCNT: Initial counter value
3. Set the desired value in the STC[1:0] bits. The division ratio can also be set in the I and PFC[1:0] bits.
4. The processor pauses internally and the WDT starts incrementing. The internal and clocks both stop and the WDT is supplied with the clock. The clock will continue to at the CKIO pin.
5. Supply of the clock that has been set begins at WDT count overflow, and the proces operating again. The WDT stops after it overflows.

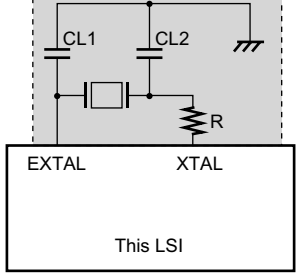
9.5.2 Changing Division Ratio

The WDT will not count unless the multiplication rate is changed simultaneously.

1. In the initial state, IFC[1:0] = 00 and PFC[1:0] = 11.
2. Set the IFC[1:0], PFC[1:0] bits to the new division ratio. The values that can be set by the clock mode and the multiplication rate of PLL circuit 1. Note that if the wron set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.

9.5.3 Modification of Clock Operating Mode

The values of the mode control pins (MD2 to MD0) that define a clock-operating mode reflected at power-on reset.



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

Figure 9.2 Points for Attention when Using Crystal Resonator

Decoupling Capacitors: As far as possible, insert a laminated ceramic capacitor of 0.
a passive capacitor for each $V_{SS}/V_{SS}Q$ and $V_{CC}/V_{CC}Q$ pair. Mount the passive capacito
as possible to the chip's power supply pins, and use components with a frequency cha
suitable for the chip's operating frequency, as well as a suitable capacitance value.

Digital system $V_{SS}/V_{SS}Q$ and $V_{CC}/V_{CC}Q$ pairs: 2-5, 17-19, 26-28, 32-34, 44-46, 57-59,
80, 87-89, 111-113, 130-132, 133-138, 159-161, 178-180, 182-184, 199-204

On-chip oscillator $V_{SS}/V_{SS}Q$ and $V_{CC}/V_{CC}Q$ pairs: 6-9, 149-150, 151-152, 205-208

When Using a PLL Oscillator Circuit: Keep the wiring from the PLL V_{CC} and PLL
connection pattern to the power supply pins short, and make the pattern width wide, to
the inductance value.

In clock mode 7, connect the EXTAL pin to $V_{CC}Q$ (3.3-V power) with pull-up resistor
the XTAL pin open.

Figure 9.3 Points for Attention when Using PLL Oscillator Circuit

Notes on Wiring Power Supply Pins: To avoid crossing signal lines, wire V_{CC}–PLL1, V_{CC}–PLL2, V_{SS}–PLL1, and V_{SS}–PLL2 as three patterns from the power supply source on the board so that they are independent of digital V_{CC} and V_{SS}.

10.1 Features

- Can be used to ensure the clock settling time

  Use the WDT to clear software standby mode and the temporary standbys which o
  the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode

  Internal resets occur after counter overflow.

  Power-on reset and manual reset are available.
- Interrupt generation in interval timer mode

  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks

  Eight clocks (×1 to ×1/4096) that are obtained by dividing the peripheral clock car

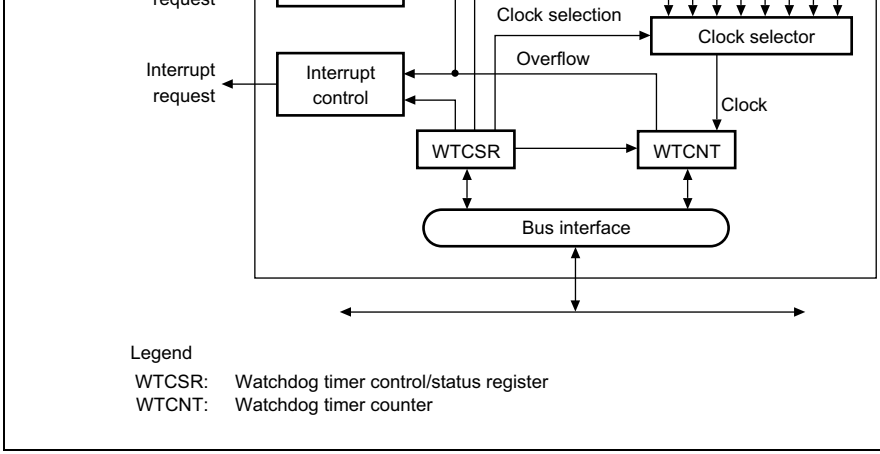


Figure 10.1 Block Diagram of WDT

10.2 Register Descriptions

The WDT has the following two registers. Refer to section 24, List of Registers for the the addresses of these registers and the state of registers in each operating mode.

- Watchdog timer counter (WTCNT)
- Watchdog timer control/status register (WTCSR)

10.2.1 Watchdog Timer Counter (WTCNT)

The watchdog timer counter (WTCNT) is an 8-bit readable/writable register that increm selected clock. When an overflow occurs, it generates a reset in watchdog timer mode a interrupt in interval timer mode. The WTCNT counter is not initialized by an internal r the WDT overflow. The WTCNT counter is initialized to H'00 only by a power-on rese $\overline{\text{RESETP}}$ pin.

Use a word access to write to the WTCNT counter, with H'5A in the upper byte. Use a to read WTCNT.

H'00 only by a power-on reset using the RESETP pin.

When used to count the clock settling time for canceling a software standby, it retains after counter overflow. Use a word access to write to the WTCSR counter, with H'A5 byte. Use a byte access to read WTCSR.

Note: WTCSR differs from other registers in that it is more difficult to write to. See 10.2.3, Notes on Register Access, for details.

6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether to use the WDT as a wat or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: If WT/$\overline{IT}$ is modified when the WDT the up-count may not be performe
5	RSTS	0	R/W	Reset Select Selects the type of reset when WTCNT ov watchdog timer mode. In interval timer mo setting is ignored. 0: Power-on reset 1: Manual reset
4	WOVF	0	R/W	Watchdog Timer Overflow Indicates that WTCNT has overflowed in w timer mode. This bit is not set in interval ti 0: No overflow 1: WTCNT has overflowed in watchdog tim
3	IOVF	0	R/W	Interval Timer Overflow Indicates that WTCNT has overflowed in i mode. This bit is not set in watchdog timer 0: No overflow 1: WTCNT has overflowed in interval time

001	Pϕ/4	(68 μs)
010	Pϕ/16	(273 μs)
011	Pϕ/32	(546 μs)
100	Pϕ/64	(1.09 ms)
101	Pϕ/256	(4.36 ms)
110	Pϕ/1024	(17.48 ms)
111	Pϕ/4096	(69.91 ms)

Note: If bits CKS2 to CKS0 are modifie
WDT is running, the up-count ma
performed correctly. Ensure that t
modified only when the WDT is n

Note: If manual reset is selected using the RSTS bit, a frequency division ratio of 1/16
1/256, 1/1,024, or 1/4,096 is selected using bits CKS2 to CKS0, and a watchdo
counter overflow occurs, resulting in a manual reset, the LSI will generate two r
resets in succession. This will not affect its operation but will cause change in th
the STATUS pin.

10.2.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (W
more difficult to write to than other registers. The procedure for writing to these regist
below.

- These registers must be written by a word transfer instruction. They cannot be writ
byte or longword transfer instruction.

When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as th
as shown in figure 10.2. When writing to WTCSR, set the upper byte to H'A5 and tran
lower byte as the write data. This transfer procedure writes the lower byte data to WT
WTCSR.

10.3 Operation

10.3.1 Canceling Software Standbys

The WDT is used to cancel software standby mode with an interrupt such as an NMI. T
procedure when using an NMI interrupt is described below. (The WDT does not run wh
are used for canceling, so keep the $\overline{\text{RESETP}}$ or $\overline{\text{RESETM}}$ pin low until the clock stabili

1. Before transitioning to software standby mode, always clear the TME bit in WTCSR
   When the TME bit is 1, an erroneous reset or interval timer interrupt may be genera
   the count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the initia
   the counter in the WTCNT counter. These values should ensure that the time till co
   overflow is longer than the clock oscillation settling time.
3. Move to software standby mode by executing a SLEEP instruction, after that clock
4. The WDT starts counting by detecting the edge change of the NMI signal.
5. When the WDT count overflows, the CPG starts supplying the clock and the proces
   resumes operation. The WOVF flag in WTCSR is not set.
6. Since the WDT continues counting from H'00, clear the STBY bit in the STBCR re
   in the interrupt processing program and this will stop the WDT. When the STBY bi
   the LSI again enters the software standby mode when the WDT has counted up to H
   software standby mode can be canceled by power-on resets.

overflow is longer than the clock oscillation settling time. The divided clock set by CKS0 bits in WTCSR will be used for the base clock of Pϕ after the frequency is c

3. When the frequency control register (FRQCR) is written, the processor stops temp WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the pr resumes operation. The WOVF flag in WTCSR is not set.
5. The counter stops at the values H'00.
6. Before changing WTCNT after the execution of the frequency change instruction, confirm that the value of WTCNT is H'00 by reading WTCNT.

10.3.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{IT}$ bit in WTCSR to 1, set the reset type in the RSTS bit, set the type o clock in the CKS2 to CKS0 bits, and set the initial value of the counter in the WTC
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and ge reset signal specified by the RSTS bit. The counter then resumes counting.

10.3.4 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{IT}$ bit in WTCSR to 0, set the type of count clock in the CKS2 to CK set the initial value of the counter in the WTCNT counter.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1 and an i timer interrupt request is sent to the INTC. The counter then resumes counting.

1. Sleep mode
2. Software standby mode
3. Module standby function (Cache, TLB, UBC, DMAC, UDI, and on-chip peripher
4. Hardware standby mode

Table 11.1 shows the transition conditions for entering the modes from the program e state, as well as the CPU and peripheral module states in each mode and the procedur canceling each mode.

Software standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halt	Halt	Held	Halt[*1]	[*3]	Self-refresh	1. I 2. I
Module standby function	Set MSTP bit of STBCR, STBCR2, and STBCR3 to 1	Run	Run	Held	Specified module halts	[*2]	Refresh	1. C I 2. I r
Hardware standby mode	Drive CA pin low	Halt	Halt	Held	Halt[*1]	[*4]	—	Pow

Notes: 1. The RTC still runs if the START bit in RCR2 is set to 1 (see section 15, Real (RTC)).
2. Depends on the on-chip peripheral module.
3. Refer to table A.1, in Appendix.
4. Hi-Z except EXTAL, XTAL, EXTAL2, XTAL2, EXTAL_USB, XTAL_USB, STA STATUS0.

			HL: Sleep mode LH: Standby mode LL: Normal operation Note: H means high level, and L means lo
Power-on reset	$\overline{\text{RESETP}}$	I	Reset input signal. Power-on reset occurs
Manual reset	$\overline{\text{RESETM}}$	I	Reset input signal. Manual reset occurs a
Hardware standby	CA	I	Normal operation at high-level and hardw mode is entered at low-level.

11.3 Register Descriptions

There are following five registers used for the power-down modes. Refer to section 24 Registers, for the details of the addresses of these registers and the state of registers in operating mode.

- Standby control register (STBCR)
- Standby control register 2 (STBCR2)
- Standby control register 3 (STBCR3)

				1: Executing SLEEP instruction puts chip software standby mode
6, 5	—	0	R	Reserved These bits are always read as 0. The writ should always be 0.
4	STBXTL	0	R/W	Standby Crystal Specifies stop/start of the crystal oscillato standby mode. 0: Crystal oscillator stops in standby mod 1: Crystal oscillator continues operation i mode.
3	—	0	R	Reserved This bit is always read as 0. The write val always be 0.
2	MSTP2	0	R/W	Module Stop 2 Specifies halting the clock supply to the T the MSTP2 bit has been set to 1. 0: TMU runs 1: Clock supply to TMU halted
1	MSTP1	0	R/W	Module Stop 1 Specifies halting the clock supply to the R the MSTP1 bit has been set to 1. 0: RTC runs 1: Clock supply to RTC halted
0	—	0	R	Reserved This bit is always read as 0. The write val always be 0.

				0: UDI runs 1: Clock supply to UDI is halted
6	MSTP9	0	R/W	Module Stop Bit 9 When the MSTP9 bit is set to 1, the cloc the UBC is halted. 0: UBC runs 1: Clock supply to UBC is halted
5	MSTP8	0	R/W	Module Stop Bit 8 When the MSTP8 bit is set to 1, the cloc the DMAC is halted. 0: DMAC runs 1: Clock supply to DMAC is halted
4	—	0	R	Reserved This bit is always read as 0. The write va always be 0.
3	MSTP6	0	R/W	Module Stop Bit 6 When the MSTP6 bit is set to 1, the cloc the TLB is halted. 0: TLB runs 1: Clock supply to TLB is halted
2	MSTP5	0	R/W	Module Stop Bit 5 When the MSTP5 bit is set to 1, the cloc cache memory is halted. 0: Cache memory runs 1: Clock supply to cache memory is halt

down mode.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP37	0	R/W	Module Stop Bit 37 When the MSTP37 bit is set to 1, the cloc the USB is halted. 0: USB runs 1: Clock supply to USB is halted
6	—	0	R	Reserved This bit is always read as 0. The write val always be 0.
5	MSTP35	0	R/W	Module Stop Bit 35 When the MSTP35 bit is set to 1, the cloc the CMT is halted. 0: CMT runs 1: Clock supply to CMT is halted
4	MSTP34	0	R/W	Module Stop Bit 34 When the MSTP34 bit is set to 1, the cloc the TPU is halted. 0: TPU runs 1: Clock supply to TPU is halted
3	MSTP33	0	R/W	Module Stop Bit 33 When the MSTP33 bit is set to 1, the cloc the ADC is halted. 0: ADC runs 1: Clock supply to ADC is halted

				the SCIF2 is halted. 0: SCIF2 runs 1: Clock supply to SCIF2 is halted
0	MSTP30	0	R/W	Module Stop Bit 30 When the MSTP30 bit is set to 1, the clo the SCIF0 is halted. 0: SCIF0 runs 1: Clock supply to SCIF0 is halted

11.4 Sleep Mode

11.4.1 Transition to Sleep Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transitio program execution state to sleep mode. Although the CPU halts immediately after exe SLEEP instruction, the contents of its internal registers remain unchanged. The on-chi modules continue to run in sleep mode and the clock continues to be output to the CK

In sleep mode, the STATUS1 pin is set high and the STATUS0 pin low.

11.4.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, IRL, PINT, and on-chip periphera reset. Interrupts are accepted in sleep mode even when the BL bit in SR is 1. If necess SPC and SSR to the stack before executing the SLEEP instruction.

Canceling with an Interrupt: When an NMI, IRQ, IRL, PINT, or on-chip peripheral interrupt occurs, sleep mode is canceled and interrupt exception processing is executed indicating the interrupt source is set in INTEVT and INTEVT2.

Canceling with a Reset: Sleep mode is canceled by a power-on reset or a manual res

The contents of the CPU and cache registers remain unchanged. Some registers of on-c peripheral modules are, however, initialized. For more details on the states of on-chip p modules registers in software standby mode, refer to section 24.3, Register States in Ea Operating Mode.

The procedure for moving to software standby mode is as follows:

1. Clear the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the W
2. Clear the WDT's timer counter (WTCNT) to 0 and set the CKS2 to CKS0 bits in W appropriate values to secure the specified oscillation settling time.
3. After the STBY bit in STBCR is set to 1, a SLEEP instruction is executed.
4. Software standby mode is entered and the clocks within the LSI are halted. The ST output goes low and the STATUS0 pin output goes high.

11.5.2 Canceling Software Standby Mode

Software standby mode is canceled by an interrupt (NMI, IRQ, IRL, PINT, or RTC) or

Canceling with an Interrupt: The on-chip WDT can be used for hot starts. When the an NMI, IRQ*[1], IRL*[1], PINT*[1], or RTC*[1] interrupt, the clock will be supplied to the en and software standby mode canceled after the time set in the WDT's timer control/statu has elapsed. The STATUS1 and STATUS0 pins both go low. Interrupt exception handl begins and a code indicating the interrupt source is set in INTEVT and INTEVT2. Afte to the interrupt handling routine occurs, clear the STBY bit in STBCR. WTCNT stops automatically. If the STBY bit is not cleared, WTCNT continues operation and transits standby mode*[2] when it reaches H'80. This function prevents data from being broken in voltage rise when the power supply is unstable. At this time, a manual reset is not accep the STBY bit is cleared to 0.

Interrupts are accepted in software standby mode even when the BL bit in SR is 1. If ne save SPC and SSR to the stack before executing the SLEEP instruction.

Immediately after an interrupt is detected, the phase of the clock output of the CKIO pi unstable, until software standby mode is cancelled.

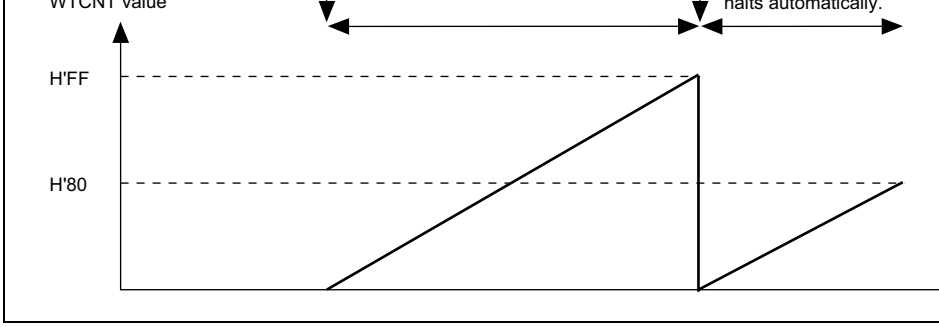


Figure 11.1 Canceling Standby Mode with STBY Bit in STBCR

Canceling with a Reset: Software standby mode is canceled by a reset (power-on or m
Keep the $\overline{\text{RESETP}}$ or $\overline{\text{RESETM}}$ pins low until the clock oscillation settles. The interna
continue to be output to the CKIO pin.

11.6 Module Standby Function

11.6.1 Transition to Module Standby Function

Setting the standby control register MSTP bits to 1 halts the supply of clocks to the co
on-chip peripheral modules. This function can be used to reduce the power consumpti
normal mode and sleep mode. Before making a transition to module standby state, be
disable the relevant module.

In module standby state, the functions of the external pins of the on-chip peripheral m
change depending on the on-chip peripheral module and port settings. With a few exc
registers hold their values prior to halt.

11.7 Hardware Standby Mode

11.7.1 Transition to Hardware Standby Mode

The LSI enters hardware standby mode by driving the CA pin low. In hardware standby
the same as software standby mode entered by executing the SLEEP instruction, all mo
except ones operated by the RTC clock. Even in hardware standby mode, supply power
power supply pins including the RTC power supply pins.

As differing from software standby mode, an interrupt or manual reset cannot be accept
hardware standby mode.

When the CA pin is driven low, the LSI enters hardware standby mode in the following
depending on the state of CPG.

During Software Standby Mode: The LSI enters the hardware standby state with the
halted. An interrupt or manual reset cannot be accepted.

During WDT Operation for Canceling Software Standby Mode by an Interrupt: T
restarts the operation after software standby mode is canceled. Then, the LSI enters har
standby mode.

During Sleep Mode: The CPU restarts the operation after sleep mode is canceled. The
enters hardware standby mode.

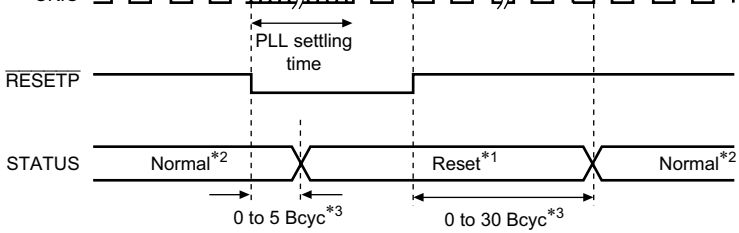
In hardware standby mode, the CA pin must be held low.

11.7.2 Canceling Hardware Standby Mode

The hardware standby function can be canceled only by the power-on reset.

When the CA pin is driven high while the $\overline{\text{RESETP}}$ pin is low, the clock starts oscillati
sure to hold the $\overline{\text{RESETP}}$ pin low until the oscillation stabilizes. Then, drive the $\overline{\text{RESET}}$
to start the power-on resetting by the CPU.

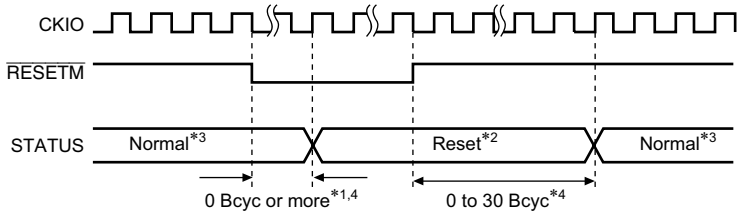
The operation is not guaranteed when an interrupt or manual reset is input.



Notes: 1. Reset: HH (STATUS1 high, STATUS0 high)
2. Normal: LL (STATUS1 low, STATUS0 low)
3. Bcyc: Bus clock cycle

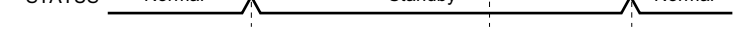
Figure 11.2 Power-On Reset STATUS Output

b. Manual reset



Notes: 1. During manual reset, STATUS becomes HH (reset) and the internal reset begins after waiting for the executing bus cycle to end.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Normal: LL (STATUS1 low, STATUS0 low)
4. Bcyc: Bus clock cycle

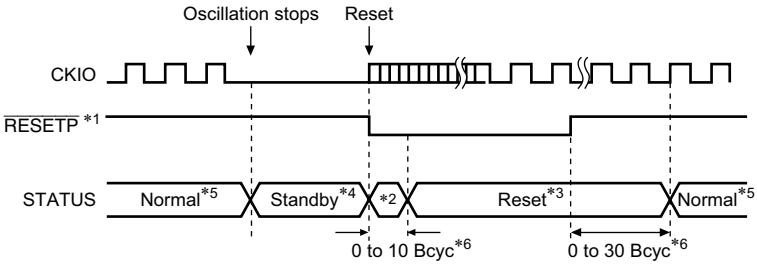
Figure 11.3 Manual Reset STATUS Output



Notes: 1. Standby: LH (STATUS1 low, STATUS0 high)
2. Normal: LL (STATUS1 low, STATUS0 low)

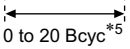
Figure 11.4 Canceling Software Standby by Interrupt STATUS Outpu

b. Canceling software standby by power-on reset



Notes: 1. When software standby mode is cleared with a power-on reset, the WDT does not count. Keep RESETP low during the PLL's oscillation settling time.
2. Undefined
3. Reset: HH (STATUS1 high, STATUS0 high)
4. Standby: LH (STATUS1 low, STATUS0 high)
5. Normal: LL (STATUS1 low, STATUS0 low)
6. Bcyc: Bus clock cycle

Figure 11.5 Canceling Software Standby by Power-On Reset STATUS Ou

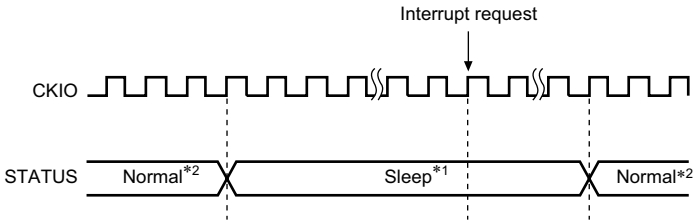


Notes: 1. When software standby mode is cleared with a manual reset, the WDT does not coun
Keep RESETM low during the PLL's oscillation settling time.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Standby: LH (STATUS1 low, STATUS0 high)
4. Normal: LL (STATUS1 low, STATUS0 low)
5. Bcyc: Bus clock cycle

Figure 11.6 Canceling Software Standby by Manual Reset STATUS Ou

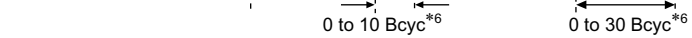
In Case of Canceling Sleep:

a. Canceling sleep to interrupt



Notes: 1. Sleep: HL (STATUS1 high, STATUS0 low)
2. Normal: LL (STATUS1 low, STATUS0 low)

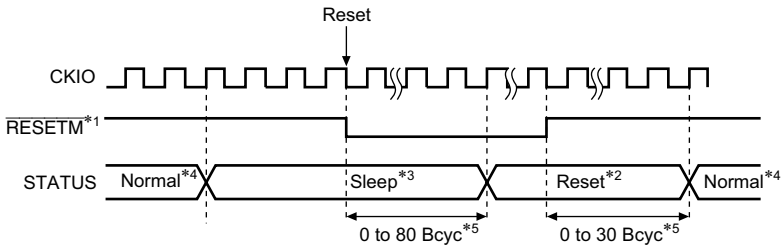
Figure 11.7 Canceling Sleep by Interrupt STATUS Output



Notes: 1. When the PLL1's multiplication ratio is changed by a power-on reset, keep $\overline{\text{RESETP}}$ low during the PLL's oscillation settling time.
2. Undefined
3. Reset: HH (STATUS1 high, STATUS0 high)
4. Sleep: HL (STATUS1 high, STATUS0 low)
5. Normal: LL (STATUS1 low, STATUS0 low)
6. Bcyc: Bus clock cycle

Figure 11.8 Canceling Sleep by Power-On Reset STATUS Output

c. Canceling sleep by manual reset



Notes: 1. Keep $\overline{\text{RESETM}}$ low until STATUS becomes reset.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Sleep: HL (STATUS1 high, STATUS0 low)
4. Normal: LL (STATUS1 low, STATUS0 low)
5. Bcyc: Bus clock cycle

Figure 11.9 Canceling Sleep by Manual Reset STATUS Output

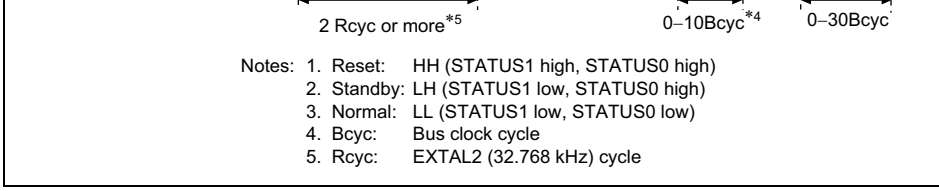
In Case of Hardware Standby:

Figures 11.10 and 11.11 show examples of pin timing in hardware standby mode.

The CA pin is sampled using EXTAL2 (32.768 kHz), and a hardware standby request i detected when the pin is low for two consecutive clock cycles.

The CA pin must be held low while the chip is in hardware standby mode.

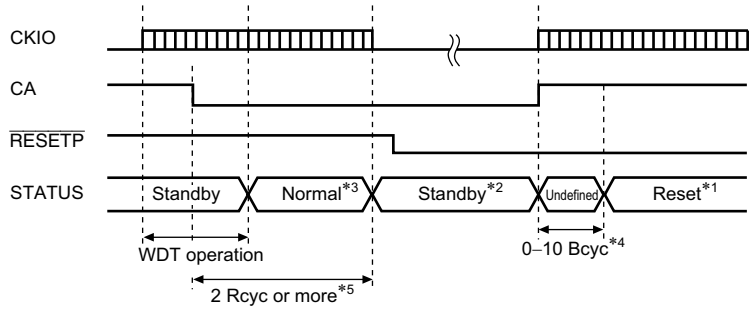
Clock oscillation starts when the CA pin is driven high after the $\overline{\text{RESETP}}$ pin is driven



Notes: 1. Reset: HH (STATUS1 high, STATUS0 high)
2. Standby: LH (STATUS1 low, STATUS0 high)
3. Normal: LL (STATUS1 low, STATUS0 low)
4. Bcyc: Bus clock cycle
5. Rcyc: EXTAL2 (32.768 kHz) cycle

Figure 11.10 Hardware Standby Mode (When CA Goes Low in Normal Operation)

b. Canceling software standby (during WDT operation) to hardware standby



Notes: 1. Reset: HH (STATUS1 high, STATUS0 high)
2. Standby: LH (STATUS1 low, STATUS0 high)
3. Normal: LL (STATUS1 low, STATUS0 low)
4. Bcyc: Bus clock cycle
5. Rcyc: EXTAL2 (32.768 kHz) cycle

Figure 11.11 Hardware Standby Mode Timing (When CA Goes Low during WDT Operation while Standby Mode Is Can

reload function that can be read or written to at any time

- All channels generate interrupt requests when the 32-bit down counter underflows (H'00000000 → H'FFFFFFFF)
- Only channel 2 is provided with an input capture function
- Allows selection among five counter input clocks: External clock (TCLK), Pϕ/4, P and Pϕ/256

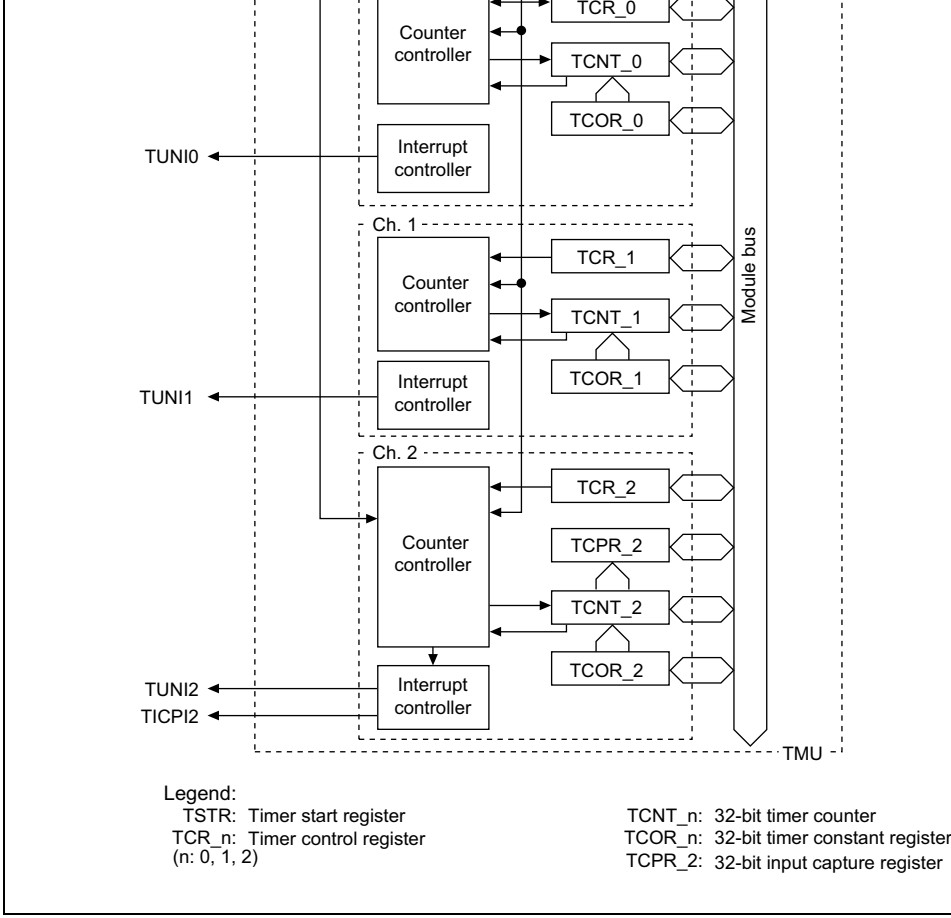


Figure 12.1 TMU Block Diagram

12.3 Register Descriptions

The TMU has the following registers. Refer to section 24, List of Registers, for more d
addresses of these registers and state of these registers in each processing state. For th
name for each channel, TCOR for channel 0 is noted as TCOR_0.

1. Common
- Timer start register (TSTR)

2. Channel 0
- Timer constant register_0 (TCOR_0)
- Timer counter_0 (TCNT_0)
- Timer control register_0 (TCR_0)

3. Channel 1
- Timer constant register_1 (TCOR_1)
- Timer counter_1 (TCNT_1)
- Timer control register_1 (TCR_1)

4. Channel 2
- Timer constant register_2 (TCOR_2)
- Timer counter_2 (TCNT_2)
- Timer control register_2 (TCR_2)
- Input capture register_2 (TCPR_2)

7 to 3	—	0	R	Reserved These bits are always read as 0. The write v always be 0.
2	STR2	0	R/W	Counter Start 2 Selects whether to run or halt timer counter 2 (TCNT_2). 0: TCNT_2 count halted 1: TCNT_2 counts
1	STR1	0	R/W	Counter Start 1 Selects whether to run or halt timer counter 1 (TCNT_1). 0: TCNT_1 count halted 1: TCNT_1 counts
0	STR0	0	R/W	Counter Start 0 Selects whether to run or halt timer counter 0 (TCNT_0). 0: TCNT_0 count halted 1: TCNT_0 counts

capture.

TCR_0 and TCR_1:

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	0	R	Reserved These bits are always read as 0. The write always be 0.
8	UNF	0	R/(W)*	Underflow Flag Status flag that indicates occurrence of a T underflow. 0: TCNT has not underflowed [Clearing condition] 0 is written to UNF 1: TCNT has underflowed [Setting condition] TCNT underflows
7, 6	—	0	R	Reserved These bits are always read as 0. The write always be 0.
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling of interrupt generation wh status flag (UNF) indicating TCNT underflow set to 1. 0: Interrupt due to UNF (TUNI) is disabled 1: Interrupt due to UNF (TUNI) is enabled

2	TPSC2	0	R/W	Timer Prescaler
1	TPSC1	0	R/W	Select the TCNT count clock.
0	TPSC0	0	R/W	000: Count on Pϕ/4
				001: Count on Pϕ/16
				010: Count on Pϕ/64
				011: Count on Pϕ/256
				100: Setting prohibited
				101: Count on TCLK pin input
				110: Setting prohibited
				111: Setting prohibited

Note: * Only 0 can be written for clearing the flags. If 1 is written to this bit, the prior valu
retained.

				0: No input capture request has been
				[Clearing condition]
				0 is written to ICPF
				1: Input capture has been requested pin.
				[Setting condition]
				When an input capture is requested v pin
8	UNF	0	R/(W)*	Underflow Flag
				Status flag that indicates occurrence underflow.
				0: TCNT_2 has not underflowed
				[Clearing condition]
				0 is written to UNF
				1: TCNT_2 has underflowed
				[Setting condition]
				TCNT_2 underflows

				01: Setting prohibited 10: Input capture function is used. Inte ICPF (TICPI2) are not enabled. 11: Input capture function is used. Inte ICPF (TICPI2) are enabled.
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling of interrupt generati the status flag (UNF) indicating TCNT underflow has been set to 1. 0: Interrupt due to UNF (TUNI2) is not 1: Interrupt due to UNF (TUNI2) is ena
4	CKEG1	0	R/W	Clock Edge
3	CKEG0	0	R/W	Select an input edge of the external cl the external clock is selected, or when capture function is used. 00: Count/capture register set on risin 01: Count/capture register set on fallin 1X: Count/capture register set on both falling edge Note: X: Don't care.

101: Count on TCLK pin input
110: Setting prohibited
111: Setting prohibited

Note: * Only 0 can be written for clearing the flags. If 1 is written to this bit, the prior val retained.

12.3.3 Timer Constant Registers (TCOR)

TCOR set the value to be set in TCNT when TCNT underflows.

TCOR are 32-bit readable/writable registers. Their initial value is H'FFFFFFFF.

12.3.4 Timer Counters (TCNT)

TCNT counts down upon input of a clock. The clock input is selected using the TPSC bits in the timer control register (TCR).

When a TCNT countdown results in an underflow (H'00000000 → H'FFFFFFFF), the flag (UNF) in the timer control register (TCR) of the relevant channel is set. The TCO simultaneously set in TCNT itself and the countdown continues from that value.

Initial value of TCNT is H'FFFFFFFF.

12.3.5 Input Capture Register_2 (TCPR_2)

TCPR_2 is a read-only 32-bit register used for the input capture function built only int The TCPR_2 setting conditions due to the TCLK pin are controlled by the input captu bits (ICPE1/ICPE0 and CKEG1/CKEG0) in TCR_2. When a TCPR_2 setting indicati TCLK pin occurs, the value of TCNT_2 is copied into TCPR_2.

Initial value of TCPR_2 is undefined.

When the STR0 to STR2 bits in the timer start register (TSTR) are set to 1, the correspo
timer counter (TCNT) starts counting. When a TCNT underflows, the UNF flag of the
corresponding timer control register (TCR) is set. At this time, if the UNIE bit in TCR
interrupt request is sent to the CPU. Also at this time, the value is copied from TCOR t
and the down-count operation is continued.

Count Operation Setting Procedure: An example of the procedure for setting the cou
operation is shown in figure 12.2.

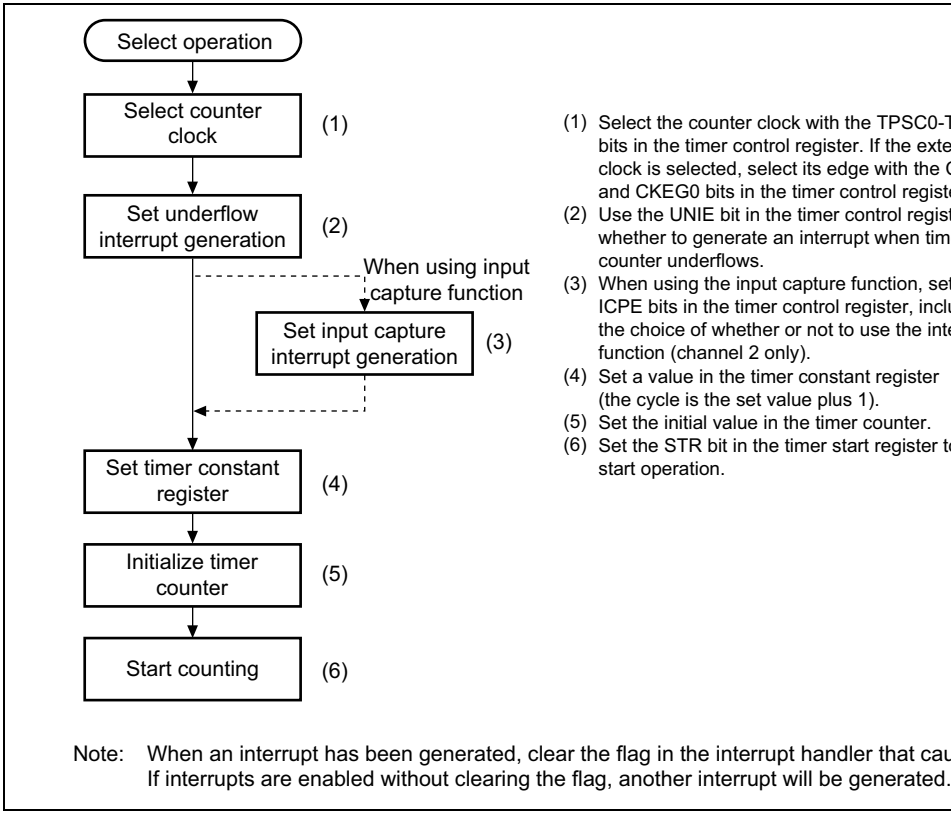


Figure 12.2 Setting Count Operation

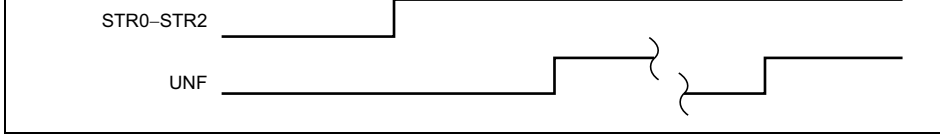


Figure 12.3 Auto-Reload Count Operation

TCNT Count Timing:

1. Internal Clock Operation: Set the TPSC2 to TPSC0 bits in TCR to select whether four internal clocks created by dividing the peripheral module clock is used (Pϕ/4, Pϕ/64, Pϕ/256). Figure 12.4 shows the timing.

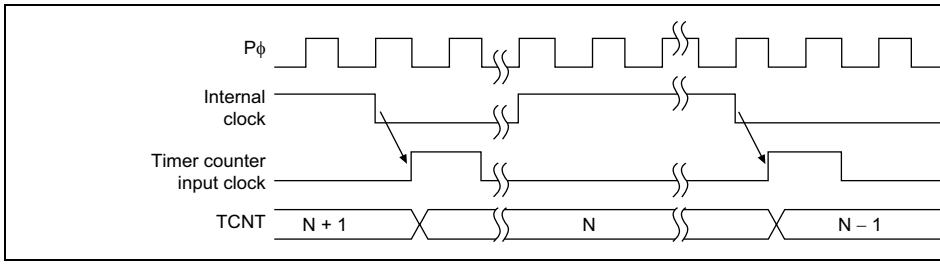


Figure 12.4 Count Timing when Internal Clock Is Operating

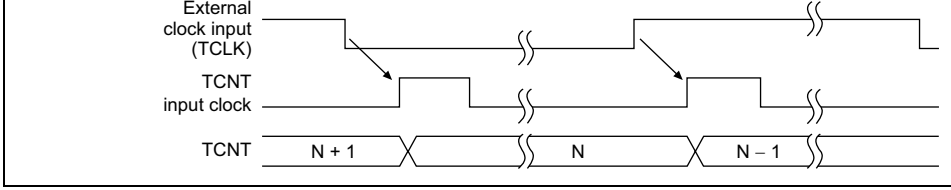


Figure 12.5 Count Timing when External Clock Is Operating (Both Edges De

12.4.2 Input Capture Function

Channel 2 has an input capture function. When using the input capture function, set the operation clock to internal clock with the TPSC2 to TPSC0 bits in TCR_2. Also, specif the input capture function and whether to generate interrupts on using it with the ICPE1 bits in TCR_2, and specifies the use of either the rising or falling edge of the TCLK pin TCNT_2 value into TCPR_2 with the CKEG1 to CKEG0 bits in TCR_2. The input cap function cannot be used in standby mode.

Figure 12.6 shows the timing at the rising edge of the TCLK pin input.

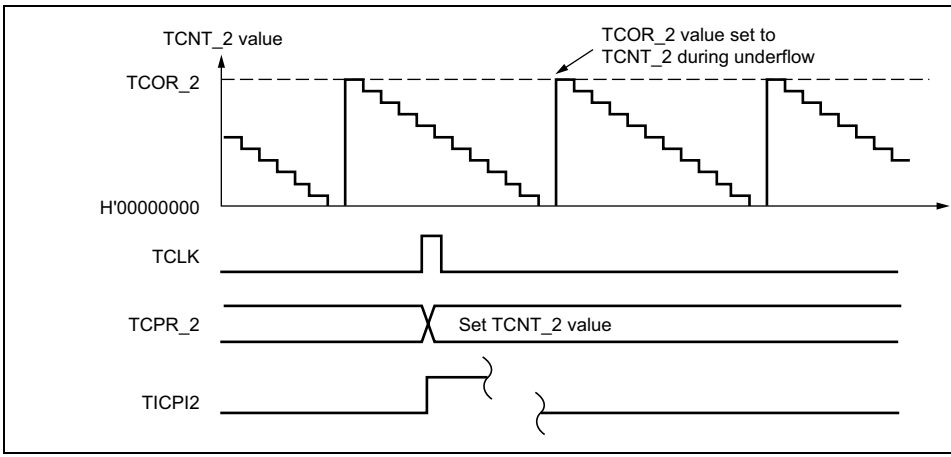


Figure 12.6 Operation Timing when Using Input Capture Function (Using TCLK Rising Edge)

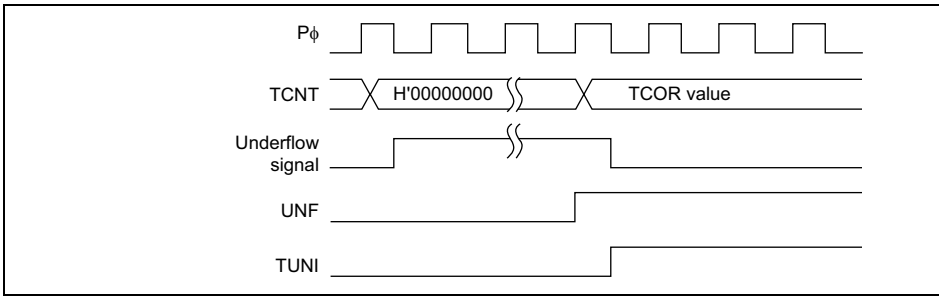


Figure 12.7 UNF Set Timing

12.5.2 Status Flag Clear Timing

The status flag can be cleared by writing 0 from the CPU. Figure 12.8 shows the timin

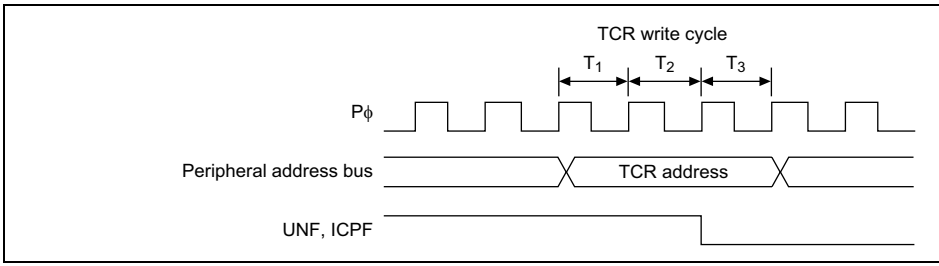


Figure 12.8 Status Flag Clear Timing

Table 12.2 lists TMU interrupt sources.

Table 12.2 TMU Interrupt Sources

Channel	Interrupt Source	Description	Priority
0	TUNI0	Underflow interrupt 0	High
1	TUNI1	Underflow interrupt 1	
2	TUNI2	Underflow interrupt 2	
	TICPI2	Input capture interrupt 2	Low

12.6 Usage Notes

12.6.1 Writing to Registers

Synchronization processing is not performed for timer counting during register writes. W
writing to registers, always clear the appropriate start bits for the channel (STR2 to STR
timer start register (TSTR) to halt timer counting.

12.6.2 Reading Registers

Synchronization processing is performed for timer counting during register reads. When
counting and register read processing are performed simultaneously, the register value b
TCNT counting down is read.

- Four types of counter input clock can be selected.
  One of four internal clocks (Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/64) can be selected.
- Generates a DMA transfer request when compare match occurs. (The CPU interru
  supported.)
- When the CMT is not used, the operation can be halted by stopping the clock supp
  CMT so that the power consumption can be reduced.

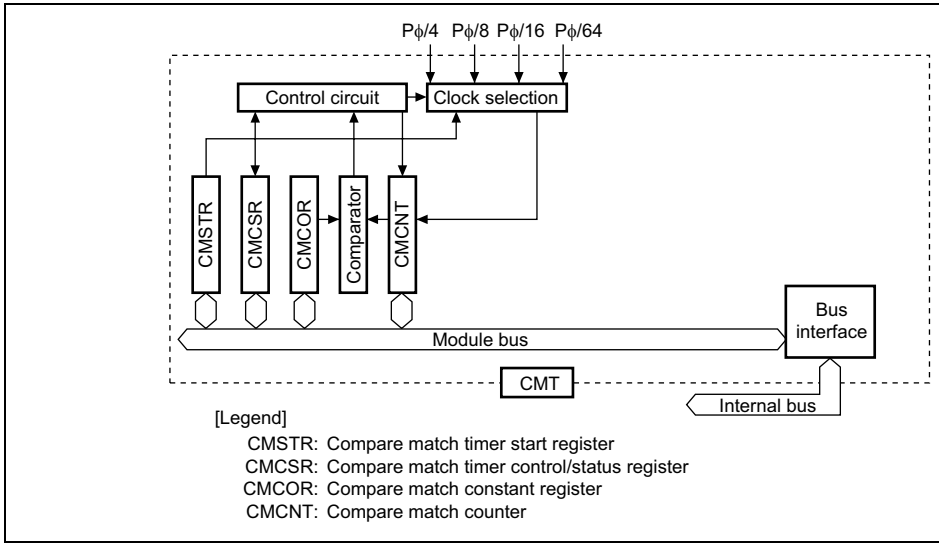


Figure 13.1 CMT Block Diagram

- Compare match constant register (CMCOR)

13.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether to operate or halt the counter (CMCNT

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	0	R	Reserved These bits are always read as 0. The v should always be 0.
0	STR	0	R/W	Count Start Selects whether to operate or halt the match counter. 0: CMCNT count operation halted 1: CMCNT count operation

7	CMF	0	R/(W)*	Compare Match Flag Indicates whether CMCNT and CMC have matched or not. 0: CMCNT and CMCOR values have [Clearing condition] Write 0 to CMF after reading CMF = 1: CMCNT and CMCOR values have
6, 5	—	0	R	Reserved These bits are always read as 0. The should always be 0.
4	CMR	0	R/W	Compare Match Request 0: Disables a DMA transfer request 1: Enables a DMA transfer request
3, 2	—	0	R	Reserved These bits are always read as 0. The should always be 0.
1	CKS1	0	R/W	Clock Select
0	CKS0	0	R/W	Select the clock input to CMCNT fron four internal clocks obtained by dividi peripheral clock (Pϕ). When the STR CMSTR is set to 1, CMCNT begins in with the clock selected by the CKS1 bits. 00: P ϕ/4 01: P ϕ/8 10: P ϕ/16 11: P ϕ/64

Note: *Only 0 can be written for clearing the flags.

The initial value of CMCNT is H'0000.

13.2.4 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the compare match period with CMCNT.

The initial value of CMCOR is H'FFFF.

13.3 Operation

13.3.1 Period Count Operation

When an internal clock is selected with the CKS1 and CKS0 bits in CMCSR and the ST
CMSTR is set to 1, CMCNT begins incrementing with the selected clock. When the CM
value matches that of CMCOR, CMCNT is cleared to H'0000 and the CMF flag in CM
to 1. CMCNT begins counting up again from H'0000.

Figure 13.2 shows the compare match counter operation.

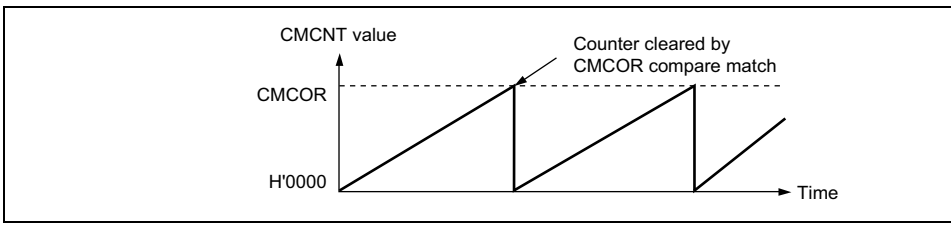


Figure 13.2 Counter Operation

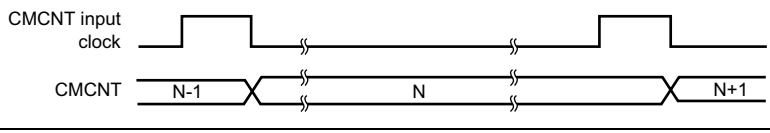


Figure 13.3 Count Timing

13.3.3 Compare Match Flag Set Timing

The CMF bit in CMCSR is set to 1 by the compare match signal generated when CMC
CMCNT match. The compare match signal is generated upon the final state of the mat
at which the CMCNT matching count value is updated to H'0000). Consequently, afte
and CMCNT match, a compare match signal will not be generated until a CMCNT clo
Figure 13.4 shows the CMF bit set timing.

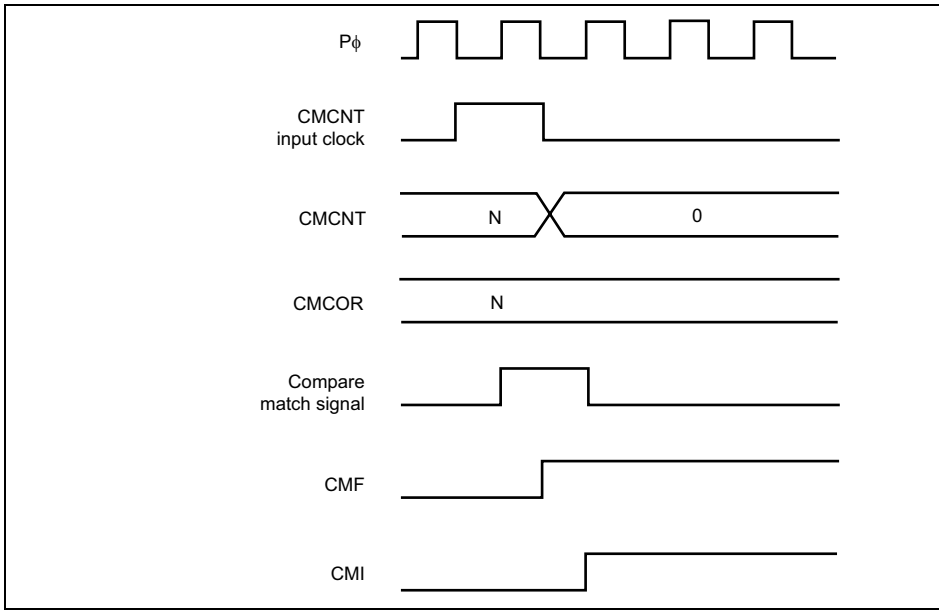


Figure 13.4 CMF Set Timing

channels). TGRA can be set as an output compare register.

TGRB, TGRC, and TGRD for each channel can also be used as timer counter clea registers. TGRC and TGRD can also be used as buffer registers.

- Selection of four counter input clocks for channels 0 to 3
- The following operations can be set for each channel:

  Waveform output at compare match: Selection of 0, 1, or toggle output

  Counter clear operation: Counter clearing possible by compare match

  PWM mode: Any PWM output duty cycle can be set

  Maximum of 4-phase PWM output possible
- Buffer operation settable for each channel

  Automatic rewriting of output compare register possible
- An interrupt request for each channel

  Compare match and overflow interrupt requests can be enabled or disabled for eac independently

		TGR0B	TGR1B	TGR2B	TGR3B
General registers/ buffer registers		TGR0C TGR0D	TGR1C TGR1D	TGR2C TGR2D	TGR3C TGR3D
Output pins		TO0	TO1	TO2	TO3
Counter clear function		TGR compare match	TGR compare match	TGR compare match	TGR co match
Compare match output	0 output	◯	◯	◯	◯
	1 output	◯	◯	◯	◯
	Toggle output	◯	◯	◯	◯
PWM mode		◯	◯	◯	◯
Buffer operation		◯	◯	◯	◯
Interrupt sources		5 sources • Compare match • Overflow	5 sources • Compare match • Overflow	5 sources • Compare match • Overflow	5 sourc • Co ma • Ov

Legend

◯: Possible

—: Not possible

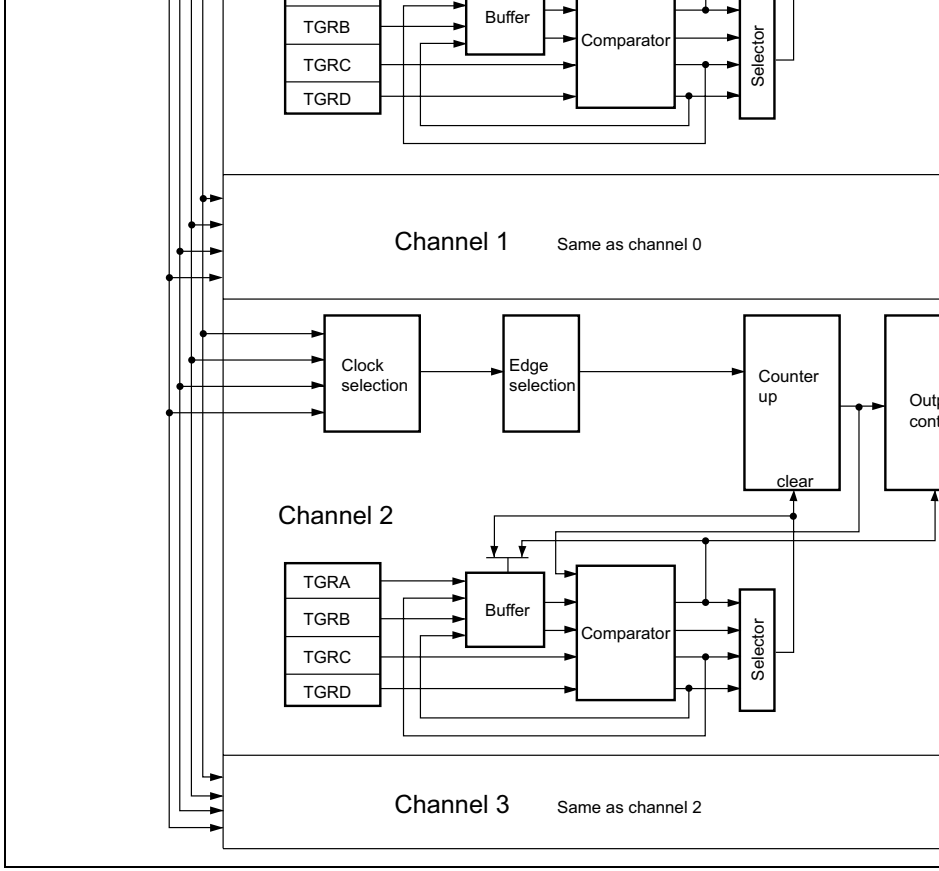


Figure 14.1 Block Diagram of TPU

1	Output compare match 1	TO1	O	TGR1A output compare output/PW pin
2	Output compare match 2	TO2	O	TGR2A output compare output/PW pin
3	Output compare match 3	TO3	O	TGR3A output compare output/PW pin

14.3 Register Descriptions

The TPU has the following registers. Refer to section 24, List of Registers, for more de addresses of these registers and state of these registers in each processing state. For the name for each channel, TCR for channel 0 is noted as TCR_0.

1. Channel 0

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register_0 (TIOR_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)

2. Channel 1

- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register_1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)

- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register 2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)
- Timer general register C_2 (TGRC_2)
- Timer general register D_2 (TGRD_2)

4. Channel 3

- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)
- Timer I/O control register_3 (TIOR_3)
- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)

5. Common

- Timer start register (TSTR)

				These bits are always read as 0 and cannot be m
7	CCLR2	0	R/W	Counter Clear
6	CCLR1	0	R/W	Select the TCNT clearing source.
5	CCLR0	0	R/W	000: TCNT clearing disabled 001: TCNT cleared by TGRA compare match 010: TCNT cleared by TGRB compare match 011: Setting prohibited 100: TCNT clearing disabled 101: TCNT cleared by TGRC compare match 110: TCNT cleared by TGRD compare match 111: Setting prohibited
4	CKEG1	0	R/W	Clock Edge
3	CKEG0	0	R/W	Select the input clock edge. When the internal clo counted using both edges, the input clock period i (e.g. Pϕ/4 both edges = Pϕ/2 rising edge). 00: Count at rising edge 01: Count at falling edge 1X: Count at both edges* [Legend] X: Don't care Note: * Internal-clock edge selection is valid when clock is Pϕ/4 or slower. If the input clock is operation is not performed.
2	TPSC2	0	R/W	Timer Prescaler
1	TPSC1	0	R/W	Select the TCNT count clock. The clock source ca
0	TPSC0	0	R/W	selected independently for each channel. Table 1 the clock sources that can be set for each channe information on count clock selection, see table 14

[Legend]

○ : Setting

Blank: No setting

Table 14.4 TPSC2 to TPSC0 (1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	X	X	Setting prohibited

Note: X: Don't care

Table 14.4 TPSC2 to TPSC0 (2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	X	X	Setting prohibited

Note: X: Don't care

	1	X	X	Setting prohibited

Note: X: Don't care

Table 14.4 TPSC2 to TPSC0 (4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	X	X	Setting prohibited

Note: X: Don't care

15 to 7	—	0	R	Reserved These bits are always read as 0 and cannot be r
6	BFWT	0	R/W	Buffer Write Timing Specifies TGRA and TGRB update timing when TGRD are used as a compare match buffer. Wh and TGRD are not used as a compare match bu this bit is ignored. 0: TGRA and TGRB are rewritten at compare ma register. 1: TGRA and TGRB are rewritten in counter clea
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the nor TGRB and TGRD are to be used together for bu operation. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer ope
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the nor TGRA and TGRC are to be used together for bu operation. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer ope
3	—	0	R	Reserved This bit is always read as 0 and cannot be modif
2	MD2	0	R/W	Timer Operating Mode Set the timer-operating mode. 000: Normal operation 001: Setting prohibited 010: PWM mode 011: Setting prohibited 1XX: Setting prohibited Note: X: Don't care
1	MD1	0	R/W	
0	MD0	0	R/W	

Bit	Bit Name	Value	R/W	Description
15 to 3	—	0	R	Reserved These bits are always read as 0 and cannot be m
2	IOA2	0	R/W	I/O Control
1	IOA1	0	R/W	Bits IOA2 to IOA0 specify the functions of TGRA a
0	IOA0	0	R/W	pin. For details, refer to table 14.5.

Table 14.5 IOA2 to IOA0

Channel	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description	
0 to 3	0	0	0	Always 0 output	
			1	Initial output is 0 output for TO pin	0 output at TGRA compare ma
		1	0		1 output at TGRA compare ma
			1		Toggle output at TGRA compa
	1	0	0	Always 1 output	
			1	Initial output is 1 output for TO pin	0 output at TGRA compare ma
		1	0		1 output at TGRA compare ma
			1		Toggle output at TGRA compa

Note: * This setting is invalid in PWM mode.

4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests by the TC the TCFV bit in TSR is set to 1 (TCNT overflow). 0: Interrupt requests by TCFV disabled 1: Interrupt requests by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests by the TG the TGFD bit in TSR is set to 1 (TCNT and TGR match). 0: Interrupt requests by TGFD disabled 1: Interrupt requests by TGFD enabled
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests by the TG the TGFC bit in TSR is set to 1 (TCNT and TGR match). 0: Interrupt requests by TGFC disabled 1: Interrupt requests by TGFC enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests by the TG the TGFB bit in TSR is set to 1 (TCNT and TGR match). 0: Interrupt requests by TGFB disabled 1: Interrupt requests by TGFB enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests by the TG the TGFA bit in TSR is set to 1 (TCNT and TGR match). 0: Interrupt requests by TGFA disabled 1: Interrupt requests by TGFA enabled

4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has [Clearing condition] When 0 is written to TCFV after reading TCFV = 1 [Setting condition] When the TCNT value overflows (changes from H H'0000)
3	TGFD	0	R/(W)*	Output Compare Flag D Status flag that indicates the occurrence of TGRD match. [Clearing condition] When 0 is written to TGFD after reading TGFD = [Setting condition] When TCNT = TGRD
2	TGFC	0	R/(W)*	Output Compare Flag C Status flag that indicates the occurrence of TGRC match. [Clearing condition] When 0 is written to TGFC after reading TGFC = [Setting condition] When TCNT = TGRC
1	TGFB	0	R/(W)*	Output Compare Flag B Status flag that indicates the occurrence of TGRB match. [Clearing condition] When 0 is written to TGFB after reading TGFB = 1 [Setting condition] When TCNT = TGRB

				When TCNT = TGRA

Note: * Only 0 can be written for clearing the flags.

14.3.6 Timer Counters (TCNT)

TCNT are 16-bit counters.

The initial value of TCNT is H'0000.

14.3.7 Timer General Registers (TGR)

TGR are 16-bit registers.

TGRC and TGRD can also be designated for operation as buffer registers*. The initial
TGR is H'FFFF.

Note: *TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

14.3.8 Timer Start Register (TSTR)

TSTR is a 16-bit readable/writable register that selects TCNT operation/stoppage for c
3.

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	0	R	Reserved
				These bits are always read as 0 and cannot be r
3	CST3	0	R/W	Counter Start
2	CST2	0	R/W	Select operation or stoppage for TCNT.
1	CST1	0	R/W	0: TCNTn count operation is stopped
0	CST0	0	R/W	1: TCNTn performs count operation
				[Legend] n = 3 to 0

Buffer Operation: When a compare match occurs, the value in the buffer register for t
channel is transferred to TGR. For update timing from a buffer register, rewriting on co
match occurrence or on counter clearing can be selected.

PWM Mode: In this mode, a PWM waveform is output. The output level can be set by
TIOR. A PWM waveform with a duty cycle of between 0% and 100% can be output, ac
the setting of each TGR register.

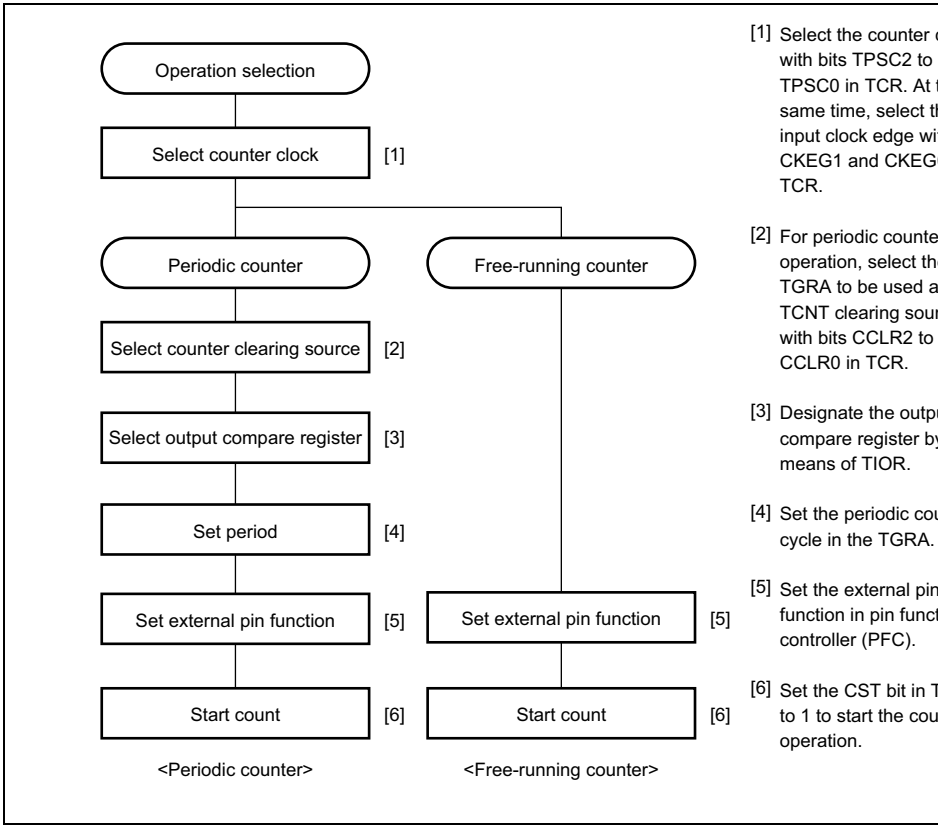


Figure 14.2 Example of Counter Operation Setting Procedure

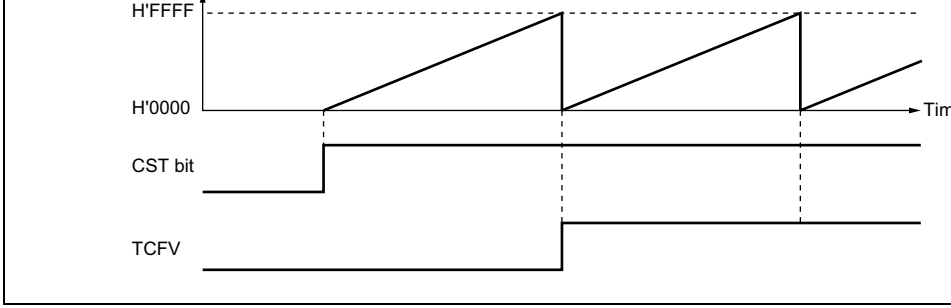


Figure 14.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for th
channel performs periodic count operation. The TGR register for setting the period is d
as an output compare register, and counter clearing by compare match is selected by m
CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count o
a periodic counter when the corresponding bit in TSTR is set to 1. When the count valu
the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

After a compare match, TCNT starts counting up again from H'0000.

Figure 14.4 illustrates periodic counter operation.

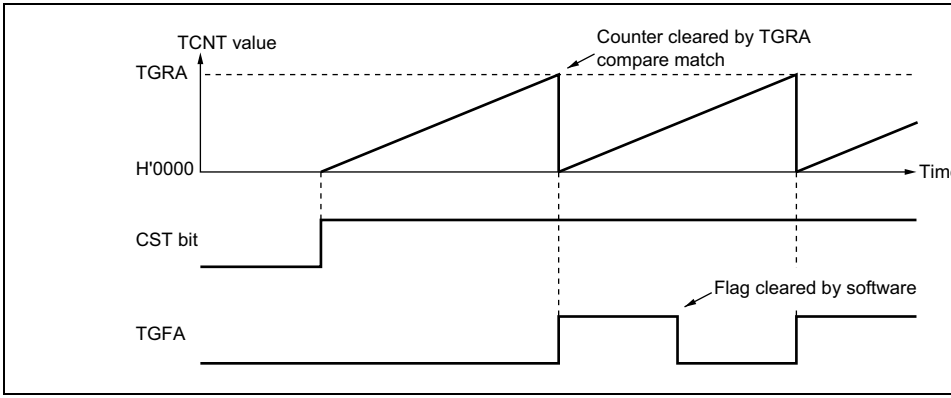
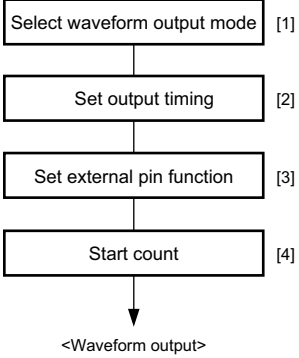


Figure 14.4 Periodic Counter Operation



initial value is output at the TO pin until the first compare match occurs.

[2] Set the timing for compare match generation in TGRA.

[3] Set the external pin function in pin function controller (PFC).

[4] Set the CST bit in TSTR to 1 to start the count operation.

Figure 14.5 Example of Setting Procedure for Waveform Output by Compar

- Examples of waveform output operation

  Figure 14.6 shows an example of 0 output/1 output.

  In this example TCNT has been designated as a free-running counter, and settings made so that 1 is output by compare match A, and 0 is output by compare match B set level and the pin level coincide, the pin level does not change.

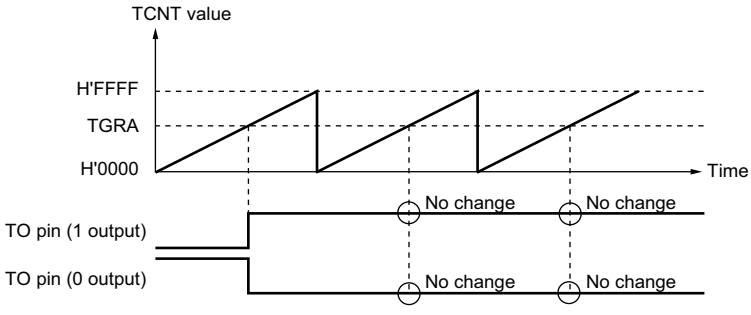


Figure 14.6 Example of 0 Output/1 Output Operation

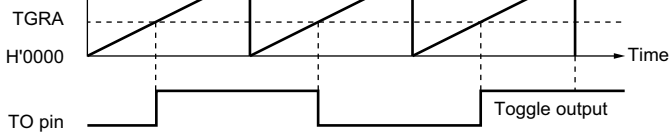


Figure 14.7 Example of Toggle Output Operation

14.4.3 Buffer Operation

Buffer operation, enables TGRC and TGRD to be used as buffer registers.

Table 14.6 shows the register combinations used in buffer operation.

Table 14.6 Register Combinations in Buffer Operation

Timer General Register	Buffer Register
TGRA	TGRC
TGRB	TGRD

When a compare match occurs, the value in the buffer register for the corresponding ch
transferred to the timer general register. For update timing from a buffer register, rewrit
compare match occurrence or on counter cleaning can be selected.

This operation is illustrated in figure 14.8.

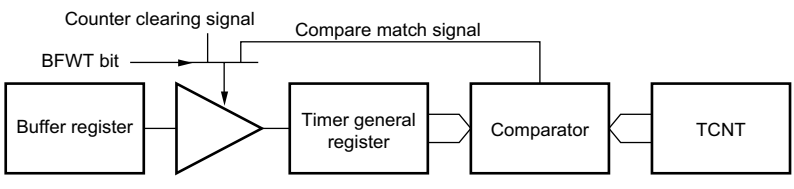


Figure 14.8 Compare Match Buffer Operation

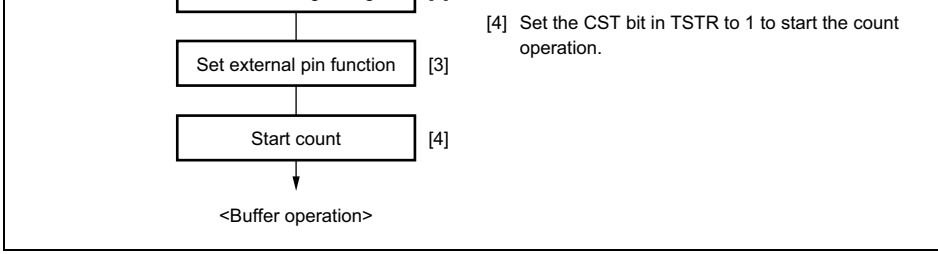


Figure 14.9 Example of Buffer Operation Setting Procedure

Example of Buffer Operation

Figure 14.10 shows an operation example in which PWM mode has been designated f and buffer operation has been designated for TGRA and TGRC. The settings used in t are TCNT clearing by compare match B, 1 output at compare match A, and 0 output a clearing. Rewriting timing from the buffer register is set at counter clearing.

As buffer operation has been set, when compare match A occurs the output changes. W counter clearing occurs by TGRB, the output changes and the value in buffer register T simultaneously transferred to timer general register TGRA. This operation is repeated compare match A occurs.

For details of PWM modes, see section 14.4.4, PWM Modes.

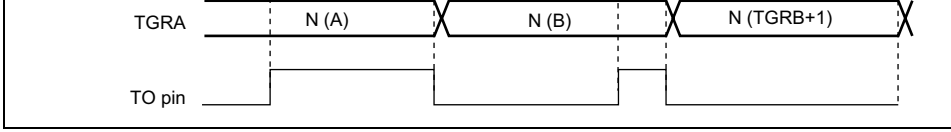


Figure 14.10 Example of Buffer Operation

14.4.4 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0 or 1 output can be
the output level in response to compare match of each TGRA.

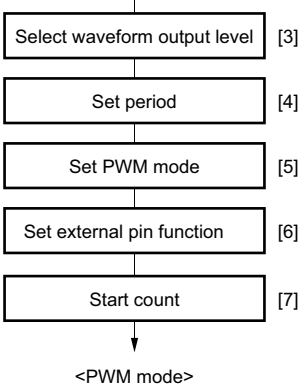
Designating TGRB compare match as the counter-clearing source enables the period to
that register. All channels can be designated for PWM mode independently.

PWM output is generated from the TO pin using TGRB as the period register and TGR
cycle registers. The output specified in TIOR is performed by means of compare match
counter clearing by a period register compare match, the output value of each pin is the
value set in TIOR. Set TIOR so that the initial output and an output value by compare r
different. If the same levels or toggle outputs are selected, operation is disabled.

Conditions of duty cycle 0% and 100% are shown below.

- Duty cycle 0%: The set value of the period register (TGRB) is TGRA + 1 for the
  register (TGRA).
- Duty cycle 100%: The set value of the duty register (TGRA) is 0.

In PWM mode, a maximum 4-phase PWM output is possible.



[3] Use TIOR to select the initial value and output value.

[4] Set the period in TGRB, and set the duty in TGRA.

[5] Select the PWM mode with bits MD2 to MD0 in TMDR.

[6] Set the external pin function in pin function controller (PFC).

[7] Set the CST bit in TSTR to 1 to start the count operation.

Figure 14.11 Example of PWM Mode Setting Procedure

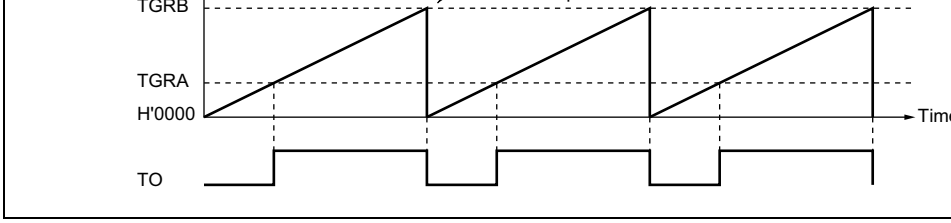


Figure 14.12 Example of PWM Mode Operation (1)

Figure 14.13 shows examples of PWM waveform output with 0% duty and 100% duty mode.

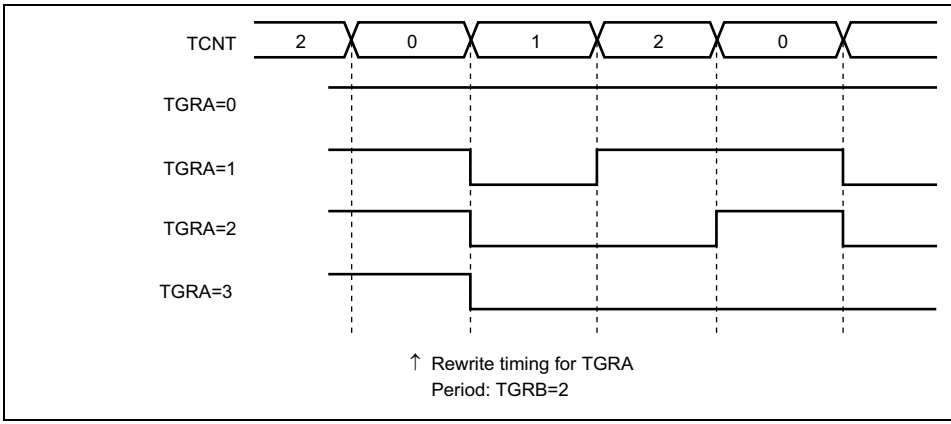


Figure 14.13 Examples of PWM Mode Operation (2)

- Clock and calendar functions (BCD format): seconds, minutes, hours, date, day of month, and year
- 1-Hz to 64-Hz timer (binary format)
- Start/stop function
- 30-second adjust function
- Alarm interrupt: frame comparison of seconds, minutes, hours, date, day of the we and year can be used as conditions for the alarm interrupt
- Periodic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 sec second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter re
- Automatic leap year adjustment

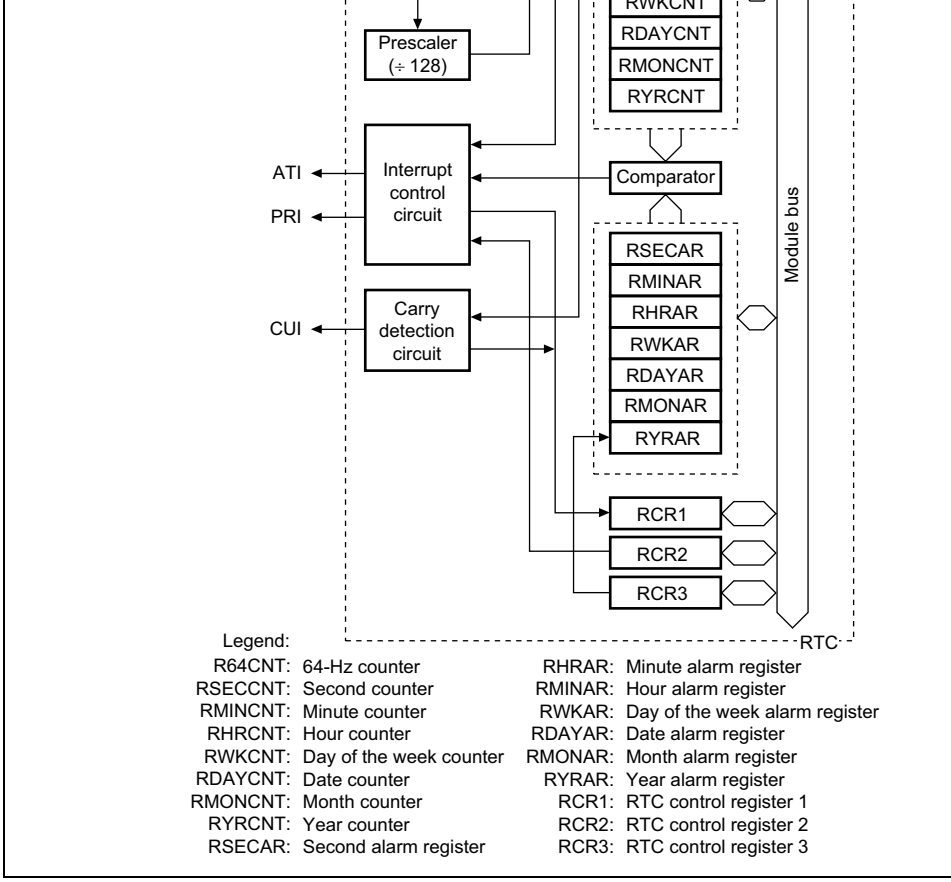


Figure 15.1 RTC Block Diagram

Power-supply for RTC	V_{CC}-RTC	—	Power-supply pin for RTC
GND for RTC	V_{SS}-RTC	—	GND pin for RTC

Note: * Pull up (VccQ (3.3 V power)) EXTAL2 and leave XTAL2 open, when the RTC is

15.3 Register Descriptions

The RTC has the following registers. Refer to section 24, List of Registers, for more d address and access size.

- 64-Hz counter (R64CNT)
- Second counter (RSECCNT)
- Minute counter (RMINCNT)
- Hour counter (RHRCNT)
- Day of week counter (RWKCNT)
- Date counter (RDAYCNT)
- Month counter (RMONCNT)
- Year counter (RYRCNT)
- Second alarm register (RSECAR)
- Minute alarm register (RMINAR)
- Hour alarm register (RHRAR)
- Day of week alarm register (RWKAR)
- Date alarm register (RDAYAR)
- Month alarm register (RMONAR)
- Year alarm register (RYRAR)
- RTC control register 1 (RCR1)
- RTC control register 2 (RCR2)
- RTC control register 3 (RCR3)

7	—	0	R	Reserved This bit is always read as 0.
6 to 0	—	—	R	64-Hz Counter Each bit (bits 6 to 0) indicates the stat RTC divider circuit between 64 Hz an Bit Frequency 6: 1 Hz 5: 2 Hz 4: 4 Hz 3: 8 Hz 2: 16 Hz 1: 32 Hz 0: 64 Hz

15.3.2 Second Counter (RSECCNT)

The second counter (RSECCNT) is an 8-bit readable/writable register used for setting/c the BCD-coded second section. The count operation is performed by a carry for each se 64-Hz counter.

The range of second that can be set is 00 to 59 (decimal). Errant operation will result if value is set. Carry out write processing after stopping the count operation with the STA RCR2.

RSECCNT is not initialized by a power-on reset or manual reset, or in standby mode.

15.3.3 Minute Counter (RMINCNT)

The minute counter (RMINCNT) is an 8-bit readable/writable register used for setting
the BCD-coded minute section. The count operation is performed by a carry for each
second counter.

The range of minute that can be set is 00 to 59 (decimal). Errant operation will result i
value is set. Carry out write processing after stopping the count operation with the ST
RCR2.

RMINCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The writ should always be 0.
6 to 4	—	—	R/W	10-unit of the minute counter in the B The range that can be set is 0 to 5 (d
3 to 0	—	—	R/W	1-unit of the minute counter in the BC The range that can be set is 0 to 9 (d

15.3.4 Hour Counter (RHRCNT)

The hour counter (RHRCNT) is an 8-bit readable/writable register used for setting/cou
BCD-coded hour section. The count operation is performed by a carry for each 1 hour
minute counter.

The range of hour that can be set is 00 to 23 (decimal). Errant operation will result if a
value is set. Carry out write processing after stopping the count operation with the ST
RCR2.

RHRCNT is not initialized by a power-on reset or manual reset, or in standby mode.

15.3.5 Day of Week Counter (RWKCNT)

The day of week counter (RWKCNT) is an 8-bit readable/writable register used for setting/counting in the day of week section. The count operation is performed by a carr day of the date counter.

The range for day of the week that can be set is 0 to 6 (decimal). Errant operation will r other value is set. Carry out write processing after stopping the count operation with the bit in RCR2.

RWKCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	0	R	Reserved These bits are always read as 0. The should always be 0.
2 to 0	—	—	R/W	Counter for the day of week in the BC The range that can be set is 0 to 6 (de Code Day of Week 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday

RDAYCNT is not initialized by a power-on reset or manual reset, or in standby mode

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	0	R	Reserved These bits are always read as 0. The should always be 0.
5, 4	—	—	R/W	10-unit of the date counter in the BCI The range that can be set is 0 to 3 (d
3 to 0	—	—	R/W	1-unit of the date counter in the BCD The range that can be set is 0 to 9 (d

15.3.7 Month Counter (RMONCNT)

The month counter (RMONCNT) is an 8-bit readable/writable register used for setting the BCD-coded month section. The count operation is performed by a carry for each m date counter.

The range of month that can be set is 01 to 12 (decimal). Errant operation will result if value is set. Carry out write processing after stopping the count operation with the STA RCR2.

RMONCNT is not initialized by a power-on reset or manual reset, or in standby mode

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	0	R	Reserved These bits are always read as 0. The should always be 0.
4	—	—	R/W	10-unit of the month counter in the B The range that can be set is 0 to 1 (d
3 to 0	—	—	R/W	1-unit of the month counter in the BC The range that can be set is 0 to 9 (d

RYRCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Leap years are recognized by dividing the year counter value by 4 and obtaining a fract
of 0. The year counter value 0000 is recognized as a leap year.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	—	R/W	1000-unit of the year counter in the BC The range that can be set is 0 to 9 (de
11 to 8	—	—	R/W	100-unit of the year counter in the BCI The range that can be set is 0 to 9 (de
7 to 4	—	—	R/W	10-unit of the year counter in the BCD The range that can be set is 0 to 9 (de
3 to 0	—	—	R/W	1-unit of the year counter in the BCD-c The range that can be set is 0 to 9 (de

15.3.9 Second Alarm Register (RSECAR)

The second alarm register (RSECAR) is an 8-bit readable/writable register, and an aları
corresponding to the second counter RSECCNT. When the ENB bit is set to 1, a compa
the RSECCNT value is performed. For alarm registers RSECAR, RMINAR, RHRAR,
RDAYAR, and RMONAR, a comparison with the corresponding counter value is perfo
those whose ENB bit is set to 1, and for RCR3, a comparison is performed when the YA
set to 1. If all of those match, an RTC alarm interrupt is generated.

The range of second alarm that can be set is 00 to 59 (decimal). Errant operation will re
other value is set.

The ENB bit in RSECAR is initialized to 0 by a power-on reset, and it is not initialized
reset and standby mode. The remaining RSECAR fields are not initialized by a power-c
manual reset, or in standby mode.

3 to 0	—	—	R/W	1-unit of second alarm setting in the The range that can be set is 0 to 9 (d

15.3.10 Minute Alarm Register (RMINAR)

The minute alarm register (RMINAR) is an 8-bit readable/writable register, and an ala corresponding to the minute counter RMINCNT. When the ENB bit is set to 1, a comp the RMINCNT value is performed. For alarm registers RSECAR, RMINAR, RHRAR RDAYAR, and RMONAR, a comparison with the corresponding counter value is per those whose ENB bit is set to 1, and for RCR3, a comparison is performed when the Y set to 1. If all of those match, an RTC alarm interrupt is generated.

The range of minute alarm that can be set is 00 to 59 (decimal). Errant operation will r other value is set.

The ENB bit in RMINAR is initialized by a power-on reset, and it is not initialized by reset and standby mode. The remaining RMINAR fields are not initialized by a power manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Minute Alarm Enable Specifies whether to compare RMINC RMINAR to generate a second alarm 0: Not compared 1: Compared
6 to 4	—	—	R/W	10-unit of minute alarm setting in the The range that can be set is 0 to 5 (d
3 to 0	—	—	R/W	1-unit of minute alarm setting in the B The range that can be set is 0 to 9 (d

The range of hour alarm that can be set is 00 to 23 (decimal). Errant operation will resu
other value is set.

The ENB bit in RHRAR is initialized by a power-on reset, and it is not initialized by m
and standby mode. The remaining RHRAR fields are not initialized by a power-on rese
reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Hour Alarm Enable Specifies whether to compare RHRCN RHRAR to generate a second alarm. 0: Not compared 1: Compared
6	—	0	R	Reserved This bit is always read as 0. The write should always be 0.
5, 4	—	—	R/W	10-unit of hour alarm setting in the BC The range that can be set is 0 to 2 (de
3 to 0	—	—	R/W	1-unit of hour alarm setting in the BCD The range that can be set is 0 to 9 (de

The range of day of the week alarm that can be set is 0 to 6 (decimal). Errant operation if any other value is set.

The ENB bit in RWKAR is initialized by a power-on reset, and it is not initialized by and standby mode. The remaining RWKAR fields are not initialized by a power-on re manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Day of Week Alarm Enable Specifies whether to compare RWKC RWKAR to generate a second alarm 0: Not compared 1: Compared
6 to 3	—	0	R	Reserved These bits are always read as 0. The should always be 0.
2 to 0	—	—	R/W	Day of Week Alarm Code The range that can be set is 0 to 6 (d Code Day of the Week 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday

The range of date alarm that can be set is 01 to 31 (decimal). Errant operation will resul other value is set. The RDAYCNT range that can be set changes with some months and years. Please confirm the correct setting.

The ENB bit in RDAYAR is initialized by a power-on reset, and it is not initialized by reset and standby mode. The remaining RDAYAR fields are not initialized by a power- manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Date Alarm Enable Specifies whether to compare RDAYC RDAYAR to generate a second alarm. 0: Not compared 1: Compared
6	—	0	R	Reserved This bit is always read as 0. The write should always be 0.
5, 4	—	—	R/W	10-unit of date alarm setting in the BC The range that can be set is 0 to 3 (de
3 to 0	—	—	R/W	1-unit of date alarm setting in the BCD The range that can be set is 0 to 9 (de

The range of month alarm that can be set is 01 to 12 (decimal). Errant operation will r
other value is set.

The ENB bit in RMONAR is initialized by a power-on reset, and it is not initialized b
reset and standby mode. The remaining RMONAR fields are not initialized by a powe
manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Month Alarm Enable Specifies whether to compare RMON RMONAR to generate a second alar 0: Not compared 1: Compared
6, 5	—	0	R	Reserved These bits are always read as 0. The should always be 0.
4	—	—	R/W	10-unit of month alarm setting in the The range that can be set is 0 to 1 (d
3 to 0	—	—	R/W	1-unit of month alarm setting in the B The range that can be set is 0 to 9 (d

The range of year alarm that can be set is 0000 to 9999 (decimal). Errant operation will
any other value is set.

The RYRAR contents are not initialized by a power-on reset or manual reset, or in stan

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	—	R/W	1000-unit of year alarm setting in the The range that can be set is 0 to 9 (de
11 to 8	—	—	R/W	100-unit of year alarm setting in the B The range that can be set is 0 to 9 (de
7 to 4	—	—	R/W	10-unit of year alarm setting in the BC The range that can be set is 0 to 9 (de
3 to 0	—	—	R/W	1-unit of year alarm setting in the BCD The range that can be set is 0 to 9 (de

Bit	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag Status flag that indicates that a carry occurred. CF is set to 1 when R64CN RSECCNT is read during a carry occ R64CNT or RSECCNT. A count regis read at this time cannot be guarantee read is required. 0: No carry by R64CNT or RSECCNT [Clearing condition] When 0 is writ 1: [Setting condition] When R64CNT RSECCNT is read during a carry o R64CNT or RSECCNT, or 1 is writ
6, 5	—	0	R	Reserved These bits are always read as 0. The should always be 0.
4	CIE	0	R/W	Carry Interrupt Enable Flag When the carry flag (CF) is set to 1, t enables interrupts. 0: A carry interrupt is not generated w flag is set to 1 1: A carry interrupt is generated when is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag When the alarm flag (AF) is set to 1, enables interrupts. 0: An alarm interrupt is not generated AF flag is set to 1 1: An alarm interrupt is generated wh flag is set to 1

				time. This flag is cleared to 0 when 0 but holds the previous value when 1 is 0: Clock/calendar and alarm register h matched. [Clearing condition] When 0 is writte 1: [Setting condition] Clock/calendar a register have matched (only registe bit and YAEN bit is 1)

15.3.17 RTC Control Register 2 (RCR2)

The RTC control register 2 (RCR2) is an 8-bit readable/writable register for periodic in
control, 30-second adjustment ADJ, divider circuit RESET, and RTC count start/stop c
initialized to H'09 by a power-on reset. It is initialized except for RTCEN and START
manual reset. It is not initialized in standby mode, and retains its contents.

Bit	Bit Name	Initial Value	R/W	Description
7	PEF	0	R/W	Periodic Interrupt Flag Indicates interrupt generation with the designated by the PES2 to PES0 bits. to 1, PEF generates periodic interrupts 0: Interrupts not generated with the pe designated by the PES bits. [Clearing condition] When 0 is writte 1: [Setting condition] When interrupts generated with the period designate PES bits or 1 is written to PEF

				011: Periodic interrupt generated eve second 100: Periodic interrupt generated eve second 101: Periodic interrupt generated eve second 110: Periodic interrupt generated eve 111: Periodic interrupt generated eve
3	RTCEN	1	R/W	Controls the operation of the crystal the RTC. 0: Halts the crystal oscillator for the R 1: Runs the crystal oscillator for the R
2	ADJ	0	R/W	30 Second Adjustment When 1 is written to the ADJ bit, time seconds or less will be rounded to 00 and 30 seconds or more to 1 minute. circuit will be simultaneously reset. T always read as 0. 0: Runs normally. 1: 30-second adjustment.
1	RESET	0	R/W	Reset When 1 is written, initializes the divid (RTC prescaler and R64CNT). This b read as 0. 0: Runs normally. 1: Divider circuit is reset.

15.3.18 RTC Control Register 3 (RCR3)

The RTC control register 3 (RCR3) is an 8-bit readable/writable register that controls th comparison between the BCD-coded year section counter RYRCNT of the RTC and th alarm register RYRAR.

Bit	Bit Name	Initial Value	R/W	Description
7	YAEN	0	R/W	Year Alarm Enable When this bit is set to 1, the year alarm (RYRAR) is compared with the year co (RYRCNT). For alarm registers RSEC RMINAR, RHRAR, RWKAR, RDAYAR RMONAR, a comparison with the corr counter value is performed for those w bit is set to 1, and for RCR3, a compar performed when this bit is set to 1. If a match, an RTC alarm interrupt is gene
6 to 0	—	0	R	Reserved These bits are always read as 0. The v should always be 0.

Figure 15.2 shows how to set the time when the clock is stopped.

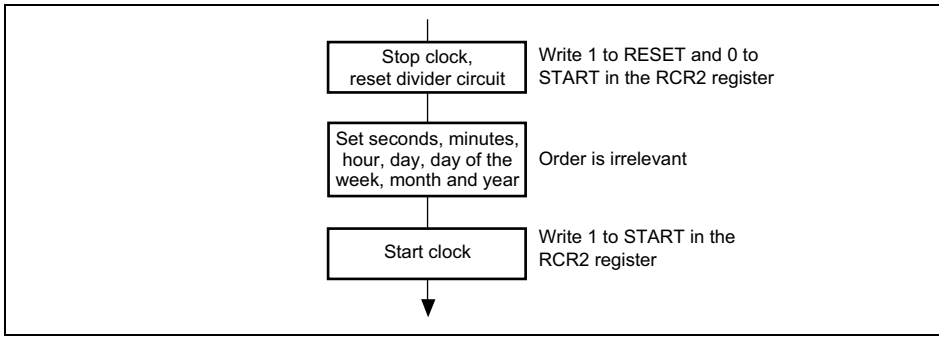


Figure 15.2 Setting Time

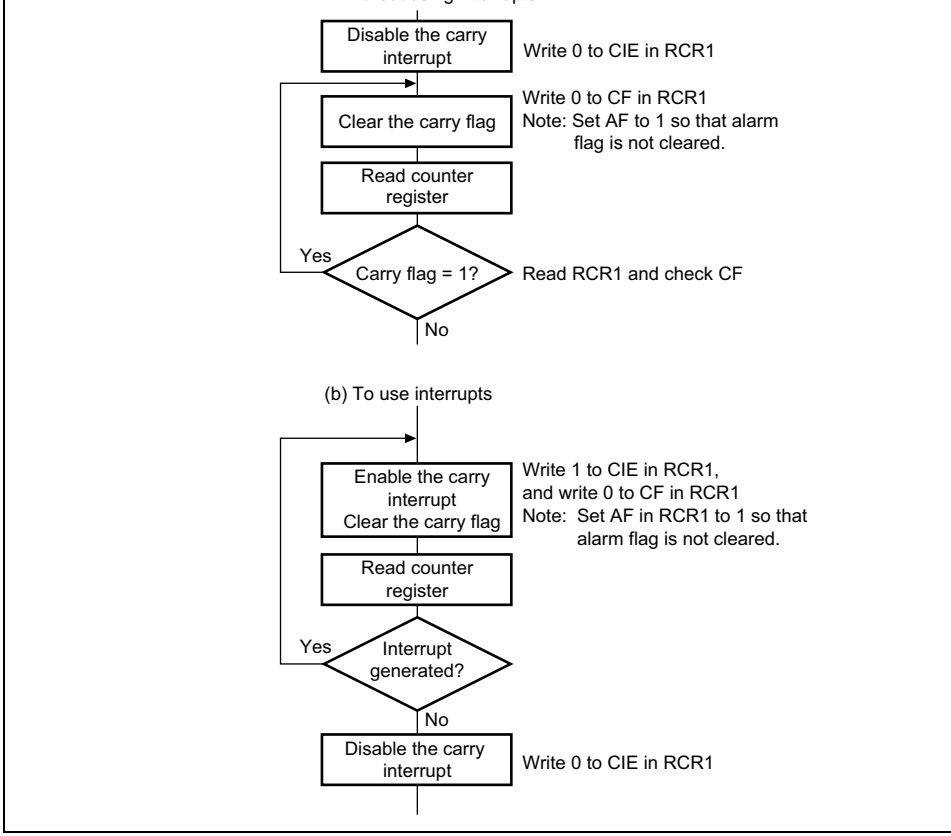


Figure 15.3 Reading the Time

When the clock and alarm times match, 1 is set in the AF bit in RCR1. Alarm detectio
checked by reading this bit, but normally it is done by interrupt. If 1 is placed in the A
RCR1, an interrupt is generated when an alarm occurs.

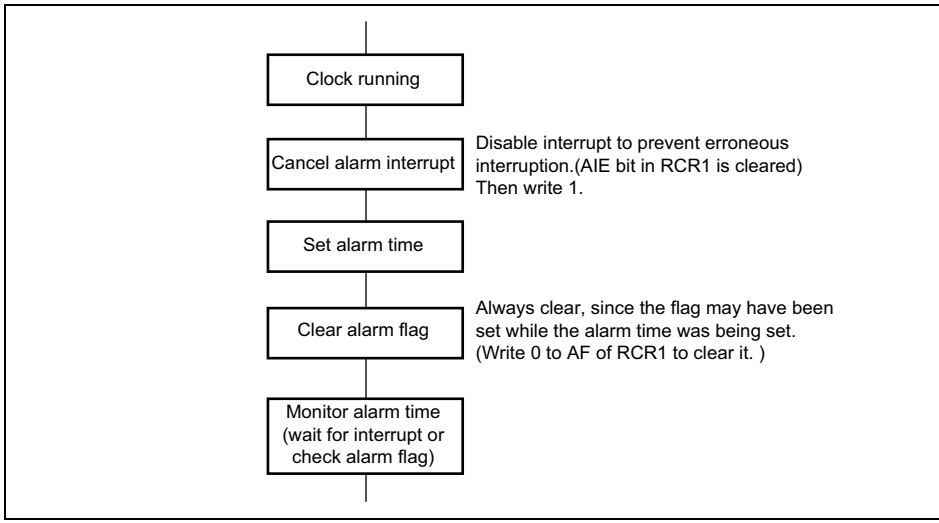
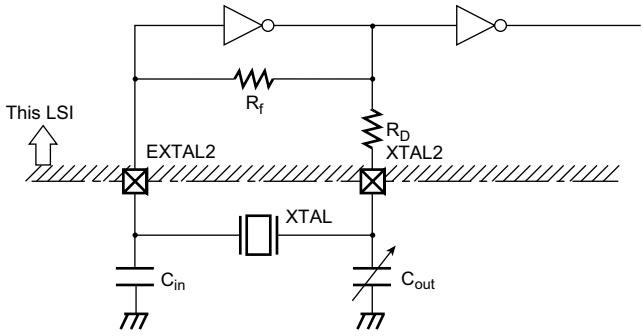


Figure 15.4 Using the Alarm Function



Notes:
1. Select either the C_{in} or C_{out} side for frequency adjustment variable capacitor according to requirements such as frequency range, stability, etc.
2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 kΩ
3. C_{in} and C_{out} values include stray capacitance due to the wiring. Take care when using a ground plane.
4. The crystal oscillation settling time depends on the mounted circuit constants, stray capacitance, etc., and should be decided after consultation with the crystal resonator manufacturer.
5. Place the crystal resonator and load capacitors C_{in} and C_{out} as close as possible to the chip.
   (Correct oscillation may not be possible if there is externally induced noise in the EXTAL2 and XTAL2 pins.)
6. Ensure that the crystal resonator connection pin (EXTAL2, XTAL2) wiring is routed as far away as possible from other power lines (except GND) and signal lines.

Figure 15.5 Example of Crystal Oscillator Circuit Connection

The RTC count must be halted before writing to any of the above registers.

15.5.2 Use of Realtime Clock (RTC) Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 15.6.

A periodic interrupt can be generated periodically at the interval set by the periodic in interval bits (PES0 to PES2) in RCR2. When the time set by the PES0 to PES2 bits ha the PEF bit is set to 1.

The PEF is cleared to 0 upon periodic interrupt generation when the periodic interrupt (PES0 to PES2) is set. Periodic interrupt generation can be confirmed by reading this normally the interrupt function is used.

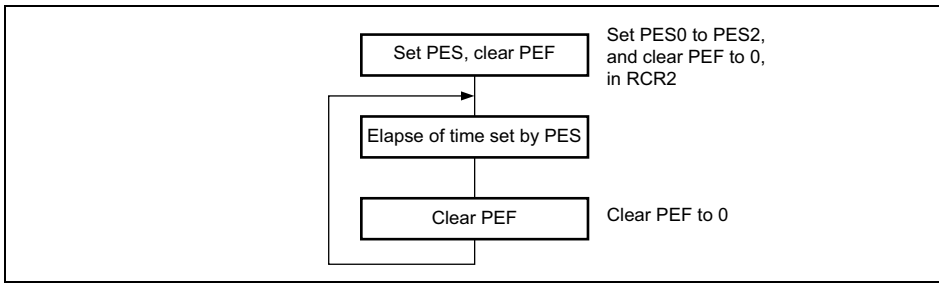


Figure 15.6 Using Periodic Interrupt Function

15.5.3 Standby Mode after Register Setting

If the standby mode is entered after the RTC registers are set, the time cannot be coun correctly. After setting the registers, wait for 2 RTC clock cycles or longer before the mode is entered.

and continuous communication.

16.1 Features

- Asynchronous mode

  Serial data communication is executed using an asynchronous system in which synchronization is achieved character by character. Serial data communication can out with standard asynchronous communication chips such as a Universal Asynchr Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapte

  There is a choice of eight serial data communication formats.

  — Data length: 7 or 8 bits
  — Stop bit length: 1 or 2 bits
  — Parity: Even/odd/none
  — LSB-first transfer
  — Receive error detection: Parity, framing, and overrun errors
  — Break detection: If a framing error is followed by at least one frame at the spac level, a break is detected.
- Clock synchronous mode

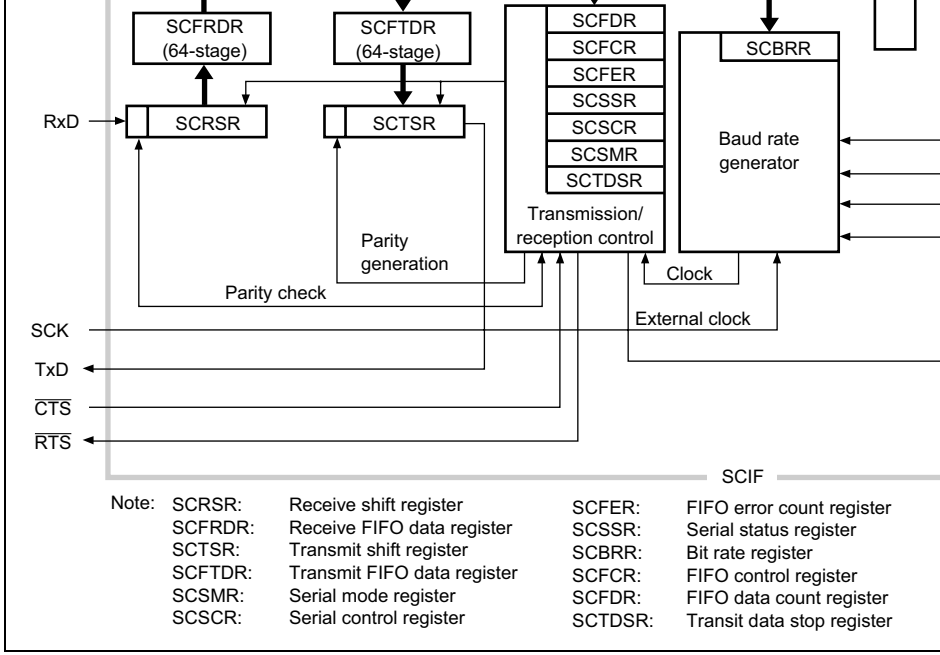
  Serial data communication is synchronized with a clock. Serial data communicatio carried out with other chips that have a synchronous communication function.

  — Data length: 8 bits
  — LSB-first transfer
- Full-duplex communication capability

  The transmitter and receiver are independent units, enabling transmission and rece performed simultaneously.

  The transmitter and receiver both have a 64-stage FIFO buffer structure, enabling continuous serial data transmission and reception.
- On-chip baud rate generator allows any bit rate to be selected.
- Choice of serial clock source: internal clock from baud rate generator or external c SCK pin.
- Six interrupt sources in asynchronous mode

- On-chip modem control functions (CTS and RTS)
- On-chip transmit-data-stop functions (only in asynchronous mode)
- When not in use, the SCIF can be stopped by halting its clock supply to reduce pow consumption.
- The amount of data in the transmit/receive FIFO registers and the number of receiv the receive data in the receive FIFO register can be ascertained.



Note: SCRSR: Receive shift register
SCFRDR: Receive FIFO data register
SCTSR: Transmit shift register
SCFTDR: Transmit FIFO data register
SCSMR: Serial mode register
SCSCR: Serial control register
SCFER: FIFO error count register
SCSSR: Serial status register
SCBRR: Bit rate register
SCFCR: FIFO control register
SCFDR: FIFO data count register
SCTDSR: Transit data stop register

Figure 16.1 Block Diagram of SCIF

	Transmit data	TxD0	TxD*2	Output	Transmit data
	Modem control	$\overline{\text{CTS0}}$	$\overline{\text{CTS}}$	Input	Transmission
	Modem control	$\overline{\text{RTS0}}$	$\overline{\text{RTS}}$	Output	Transmit requ
2	Serial clock	SCK2	SCK	Input/output	Clock input/ou
	Receive data	RxD2	RxD*2	Input	Receive data
	Transmit data	TxD2	TxD*2	Output	Transmit data
	Modem control	$\overline{\text{CTS2}}$	$\overline{\text{CTS}}$	Input	Transmission
	Modem control	$\overline{\text{RTS2}}$	$\overline{\text{RTS}}$	Output	Transmit requ

Notes: 1. The pins are collectively called SCK, RxD, TxD, $\overline{\text{CTS}}$, and $\overline{\text{RTS}}$ without chan in the following descriptions.

2. These pins are made to function as serial pins by setting SCIF operation wit and RE bits in SCSCR.

- Serial control register 0 (SCSCR_0)
- Transmit data stop register 0 (SCTDSR_0)
- FIFO error count register 0 (SCFER_0)
- Serial status register 0 (SCSSR_0)
- FIFO control register 0 (SCFCR_0)
- FIFO data count register 0 (SCFDR_0)
- Transmit FIFO data register 0 (SCFTDR_0)
- Receive FIFO data register 0 (SCFRDR_0)

2. Channel 2

- Serial mode register 2 (SCSMR_2)
- Bit rate register 2 (SCBRR_2)
- Serial control register 2 (SCSCR_2)
- Transmit data stop register 2 (SCTDSR_2)
- FIFO error count register 2 (SCFER_2)
- Serial status register 2 (SCSSR_2)
- FIFO control register 2 (SCFCR_2)
- FIFO data count register 2 (SCFDR_2)
- Transmit FIFO data register 2 (SCFTDR_2)
- Receive FIFO data register 2 (SCFRDR_2)

16.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a 64-stage 8-bit FIFO register that stores 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 enable
reception, and consecutive receive operations can be performed until the receive FIFO d
register is full (64 data bytes).

SCFRDR is a read-only register, and cannot be written to by the CPU.

If a read is performed when there is no receive data in the receive FIFO data register, an
value will be returned. When the receive FIFO data register is full of receive data, subs
serial data is lost.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCFRD7 to SCFRD0	Undefined	R	Serial Receive Data FIFO

16.3.3 Transmit Shift Register (SCTSR)

SCTSR is the register used to transmit serial data.

To perform serial data transmission, the SCIF first transfers transmit data from SCFTD
SCTSR, then sends the data sequentially to the TxD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from
to SCTSR, and transmission is started automatically.

SCTSR cannot be directly read or written to by the CPU.

The next data cannot be written when SCFTDR is filled with 64 bytes of transmit data written in this case is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCFTD7 to SCFTD0	Undefined	W	Serial Transmit Data FIFO

16.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit readable/writable register used to set the SCIF's serial transfer form select the baud rate generator clock source and the sampling rate.

				001: Sampling rate 1/5 010: Sampling rate 1/11 011: Sampling rate 1/13 100: Sampling rate 1/29 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7	C/A	0	R/W	Communication Mode Selects whether the SCI operates in the asyn or clock synchronous mode. 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects seven or eight bits as the data length This setting is only valid in asynchronous mod clock synchronous mode, the data length is a eight bits, regardless of the CHR setting. 0: 8-bit data 1: 7-bit data*
				Note: * When the 7-bit data is selected, the M 7) in the transmit FIFO data register (S is not transmitted.

				1: Parity bit addition and checking enabled*
				Note: * When the PE bit is set to 1, the parity odd) specified by the O/E bit is adde transmit data before transmission. In the parity bit is checked for the parity odd) specified by the O/E bit.
4	O/E	0	R/W	Parity Mode
				Selects either even or odd parity for use in p addition and checking. The O/E bit setting is when the PE bit is set to 1, enabling parity b and checking. The O/E bit setting is invalid w addition and checking is disabled in asynchr clock synchronous mode.
				0: Even parity*1
				1: Odd parity*2
				Notes: 1. When even parity is set, parity bi performed in transmission so tha number of 1-bits in the transmit d plus the parity bit is even. In rece check is performed to see if the t number of 1-bits in the receive cl plus the parity bit is even.
				2. When odd parity is set, parity bit performed in transmission so tha number of 1-bits in the transmit d plus the parity bit is odd. In recep check is performed to see if the t number of 1-bits in the receive cl plus the parity bit is odd.

				clock synchronous mode, this setting is invalid stop bits are not added. 0: One stop bit*1 1: Two stop bits*2 Notes: 1. In transmission, a single 1-bit (sto added to the end of a transmit cha before it is sent. 2. In transmission, two 1-bits (stop b added to the end of a transmit cha before it is sent.
2	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
1 0	CKS1 CKS0	0 0	R/W R/W	Clock Select Select the clock source for the on-chip baud r generator. 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64

Note: When the clock synchronous mode is selected (C/A bit = 1), the bits other than C
CKS0 bits are all fixed to 0.

				always be 0.
11	TSIE	0	R/W	Transmit Data Stop Interrupt Enable Enables or disables generation of a transmit interrupt when the TSE bit in SCFCR is enal TSF flag in SCSSR is set to 1. 0: Transmit-data-stop interrupt disabled* 1: Transmit-data-stop interrupt enabled Note: * The interrupt request is cleared by cl TSF flag to 0 after reading 1 from it c the TSIE bit to 0.
10	ERIE	0	R/W	Receive Error Interrupt Enable Enables or disables generation of a receive- (framing or parity error) interrupt when the E SCSSR is set to 1. 0: Receive-error interrupt disabled* 1: Receive-error interrupt enabled Note: * The interrupt request is cleared by cl ER flag to 0 after reading 1 from it or ERIE bit to 0.

				the BRIE bit to 0.
8	DRIE	0	R/W	Receive Data Ready Interrupt Enable Enables or disables generation of a receive-d interrupt when the DR flag in SCSSR is set to 0: Receive-data-ready interrupt disabled* 1: Receive-data-ready interrupt enabled Note: * The interrupt request is cleared by cle DR flag to 0 after reading 1 from it or DRIE bit to 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables generation of a transmit- empty interrupt request when the TDFE flag i set to 1. 0: Transmit-FIFO-data-empty interrupt reque disabled* 1: Transmit-FIFO-data-empty interrupt reque Note: * The interrupt request is cleared by wri transmit data exceeding the transmit t number to SCFTDR, reading 1 from th flag, then clearing it to 0, or clearing th 0.
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables generation of a receive-F full interrupt request when the RDF flag in SC to 1. 0: Receive-FIFO-data-full interrupt request di 1: Receive-FIFO-data-full interrupt request en Note: * The interrupt requests is cleared by re from the RDF flag, then clearing the fl clearing the RIE bit to 0.

				the transmit format decided, and the FIFO reset, before the TE bit is set t
4	RE	0	R/W	Receive Enable Enables or disables the start of serial recept SCIF. 0: Reception disabled*1 1: Reception enabled*2 Notes: 1. Clearing the RE bit to 0 does not DR, ER, BRK, RDF, FER, PER, flags, which retain their state. 2. The serial mode register (SCSM control register (SCFCR) setting made, the receive format decide receive FIFO reset, before the R 1.
3, 2	—	0	R	Reserved These bits are always read as 0. The write v always be 0.

output*1

10: External clock/SCK pin functions as clock

11: External clock/SCK pin functions as clock

When data is sampled by the on-chip baud ra
generator, set bits CKE1 and CKE0 to B'00 (i
clock/SCK pin functions as input pin (input sig
ignored)).

When using the SCK pin as a port, set bits CK
CKE0 to B’00.

Notes: 1. In synchronous mode, a clock witl
frequency equal to the bit rate is o

2. In asynchronous mode, a clock wi
sampling rate should be input. For
when the sampling rate is 1/16, a
frequency of 8 times the bit rate sl
input. When an external clock is n
bits CKE1 and CKE0 to B'00 or B'

				always be 0.
13 to 8	PER5 to PER0	0	R	Parity Error Count Indicates the number of data, in which parity generated, in receive data stored in the rece data register (SCFRDR) in asynchronous m After setting the ER bit in SCSSR, the value to 8 indicates the number of parity error gen When all 64 bytes of receive data in SCFRD parity errors, the PER5 to PER0 bits indicate
7, 6	—	0	R	Reserved These bits are always read as 0. The write v always be 0.
5 to 0	FER5 to FER0	0	R	Framing Error Count Indicates the number of data, in which framin are generated, in receive data stored in the FIFO data register (SCFRDR) in asynchron After setting the ER bit in SCSSR, the value 0 indicates the number of framing error gene When all 64 bytes of receive data in SCFRD framing errors, the FER5 to FER0 bits indica

Bit	Name	Value	R/W	Description
15 to 10	—	0	R	Reserved These bits are always read as 0. The write va should always be 0.
9	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error occurred durin reception. This bit is only valid in asynchronous mode. 0: Reception in progress, or reception has en successfully*1 [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ORER after reading 1 1: An overrun error occurred during reception [Setting condition] When serial reception is completed while rece FIFO is full Notes: 1. The ORER flag is not affected and its previous state when the RE bit SCSCR is cleared to 0. 2. The receive data prior to the overr is retained in SCFRDR, and the d received subsequently is lost. Ser reception cannot be continued wh ORER flag is set to 1.

- When 0 is written to TSF after reading TSF = 1

1: Number of transmit data matches the value of S

7	ER	0	R/(W)*	Receive Error Indicates that a framing error or parity error occurr reception in asynchronous mode.*1 0: No framing error or parity error occurred during [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ER after reading ER = 1 1: A framing error or parity error occurred during r [Setting conditions] • When the SCIF checks whether the stop bit at of the receive data is 1 when reception ends, a stop bit is 0*2 • When, in reception, the number of 1-bits in the data plus the parity bit does not match the pari (even or odd) specified by the O/E bit in SCSM Notes: 1. The ER flag is not affected and retains previous state when the RE bit in SCS cleared to 0. When a receive error occ receive data is still transferred to SCFI reception continues. The FER and PER bits in SCSSR can to determine whether there is a receiv the data read from SCFRDR. 2. When the stop length is two bits, only stop bit is checked for a value of 1; the stop bit is not checked.

				1: Transmission has been ended [Setting condition] When there is no transmit data in SCFTDR on trans of a 1-byte serial transmit character
5	TDFE	1	R/(W)*	Transmit FIFO Data Empty Indicates that data has been transferred from SCFT SCTSR, the number of data bytes in SCFTDR has or below the transmit trigger data number set by bit TTRG1 and TTRG0 in the FIFO control register (SC and new transmit data can be written to SCFTDR. 0: A number of transmit data bytes exceeding the tr trigger set number have been written to SCFTDR [Clearing condition] When transmit data exceeding the transmit trigger s number is written to SCFTDR, and 0 is written to TI after reading TDFE = 1 1: The number of transmit data bytes in SCFTDR d exceed the transmit trigger set number [Setting conditions] • Power-on reset or manual reset • When the number of SCFTDR transmit data byt to or below the transmit trigger set number as th of a transmit operation*1 Note: 1. As SCFTDR is a 64-byte FIFO register, th maximum number of bytes that can be wri when TDFE = 1 is 64 – (transmit trigger se number). Data written in excess of this wil ignored. The number of data bytes in SCF indicated by SCFDR.

				1: A break signal has been received[*1] [Setting condition] When data with a framing error is received, follow space 0 level (low level) for at least one frame len Note: 1. When a break is detected, the receive (H'00) following detection is not transfe SCFRDR. When the break ends and t signal returns to mark 1, receive data t resumed.
3	FER	0	R	Framing Error Indicates a framing error in the data read from SC asynchronous mode. 0: There is no framing error in the receive data rea SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no framing error in SCFRDR rea 1: There is a framing error in the receive data read SCFRDR [Setting condition] When there is a framing error in SCFRDR read da

				• When there is no parity error in SCFRDR read d 1: There is a parity error in the receive data read fro SCFRDR [Setting condition] When there is a parity error in SCFRDR read data
1	RDF	0	R/(W)*	Receive FIFO Data Full Indicates that the received data has been transferre SCRSR to SCFRDR, and the number of receive da in SCFRDR is equal to or greater than the receive t number set by bits RTRG1 and RTRG0 in the FIFC register (SCFCR). 0: The number of receive data bytes in SCFRDR is than the receive trigger set number [Clearing conditions] • Power-on reset or manual reset • When SCFRDR is read until the number of rece bytes in SCFRDR falls below the receive trigger number, and 0 is written to RDF after reading R 1: The number of receive data bytes in SCFRDR is or greater than the receive trigger set number [Setting condition] When SCFRDR contains at least the receive trigge number of receive data bytes*1 Note: 1. SCFRDR is a 64-byte FIFO register. Wher 1, at least the receive trigger set number o bytes can be read. If data is read when SC is empty, an undefined value will be return number of receive data bytes in SCFRDR indicated by the lower bits of SCFDR.

- Power-on reset or manual reset
- When all the receive data in SCFRDR ha
  read, and 0 is written to DR after reading

1: No further receive data has arrived

[Setting condition]

When SCFRDR contains fewer than the rec
set number of receive data bytes and no furt
will arrive.* 1

Note: 1. The DR bit is set 15 etu after the la
received at a sampling rate of 1/16
of the setting of the sampling contr
SCSMR.
etu: Elementary time unit (time for
one bit)

Note: * Only 0 can be written for clearing the flags.

16.3.9 Bit Rate Register (SCBRR)

SCBRR is an 8-bit readable/writable register that sets the serial transfer bit rate in acc
the baud rate generator operating clock selected by bits CKS1 and CKS0 in SCSMR.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCBR7 to SCBR0	H'FF	R/W	Bit Rate Setting

The SCBRR setting is found from the following equation.

3. When sampling rate is 1/11

$$N = \frac{P\phi}{22 \times 2^{2n-1} \times B} \times 10^6 - 1$$

4. When sampling rate is 1/13

$$N = \frac{P\phi}{26 \times 2^{2n-1} \times B} \times 10^6 - 1$$

5. When sampling rate is 1/29

$$N = \frac{P\phi}{58 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clock Synchronous Mode:

$$N = \frac{P\phi}{4 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator
Asynchronous mode: $0 \leq N \leq 255$
Clock synchronous mode: $1 \leq N \leq 255$
Pϕ: Peripheral module operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

		SCSMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

$$\left((1+N) \times B \times 10 \times 2 \right)$$

3. When sampling rate is 1/11

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 22 \times 2^{2n-1}} - 1 \right) \times 100$$

4. When sampling rate is 1/13

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 26 \times 2^{2n-1}} - 1 \right) \times 100$$

5. When sampling rate is 1/27

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 58 \times 2^{2n-1}} - 1 \right) \times 100$$

				This function is enabled only in asynchronous Since this function is not supported in clock s mode, clear this bit to 0 in clock synchronous 0: Transmit data stop function disabled 1: Transmit data stop function enabled
14	TCRST	0	R/W	Transmit Count Reset Clears the transmit count to 0. This bit is valid the transmit data stop function is used. 0: Transmit count reset disabled* 1: Transmit count reset enabled (clearing to 0 Note:* The transmit count is reset (clearing to performed in power-on reset or manua
13 to 11	—	0	R	Reserved These bits are always read as 0. The write va always be 0.
10 9 8	RSTRG2 RSTRG1 RSTRG0	0 0 0	R/W R/W R/W	$\overline{\text{RTS}}$ Output Active Trigger The $\overline{\text{RTS}}$ signal goes high when the number o data bytes in SCFRDR is equal to or greater t trigger set number shown in below. 000: 63 001: 1 010: 8 011: 16 100: 32 101: 48 110: 54 111: 60

				01: 16 10: 32 11: 48
5 4	TTRG1 TTRG0	0 0	R/W R/W	Transmit FIFO Data Number Trigger Set the number of remaining transmit data bytes th the transmit FIFO data register empty (TDFE) flag serial status register (SCSSR). The TDFE flag is set when, as the result of a trans operation, the number of transmit data bytes in the FIFO data register (SCFTDR) falls to or below the set number shown in below. 00: 32 (32) 01: 16 (48) 10: 2 (62) 11: 0 (64) Note: The values in parentheses are the number bytes in SCFTDR when the flag is set.
3	MCE	0	R/W	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. This setting is only valid in asynchronous mode. 0: Modem signal disabled* 1: Modem signal enabled Note: * $\overline{\text{CTS}}$ is fixed at active 0 regardless of the in value, and $\overline{\text{RTS}}$ is also fixed at 0.

1	RFRST	0	R/W	Receive FIFO Data Register Reset Invalidates the receive data in the receive FIFO dat register and resets it to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note:* A reset operation is performed in the event power-on reset or manual reset.
0	LOOP	0	R/W	Loopback Test Internally connects the transmit output pin (TxD) an receive input pin (RxD), and $\overline{\text{RTS}}$ pin and $\overline{\text{CTS}}$ pin, loopback testing. 0: Loopback test disabled 1: Loopback test enabled

Bit	Name	Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
14 to 8	T6 to T0	0	R	These bits show the number of untransmitte bytes in SCFTDR. A value of H'00 means that there is no trans and a value of H'40 means that SCFTDR is transmit data.
7	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
6 to 0	R6 to R0	0	R	These bits show the number of receive data SCFRDR. A value of H'00 means that there is no recei and a value of H'40 means that SCFRDR is receive data.

16.3.12 Transmit Data Stop Register (SCTDSR)

SCTDSR is an 8-bit readable/writable register that sets the number of transmit data by SCTDSR is valid only when the TSE bit in the FIFO control register (SCFCR) is enab Transmit operation is stopped when the number of data bytes set in SCTDSR is transm setting value should be H'00 (one byte) to H'FF (256 bytes). This function is only enab asynchronous mode.

SCTDSR is initialized to H'FF.

64-stage FIFO buffers are provided for both transmission and reception, reducing the C
overhead and enabling fast, continuous communication to be performed.

16.4.2 Asynchronous Mode

The transfer format is selected using the serial mode register (SCSMR), as shown in tab
The SCIF clock source is determined by the CKE1 and CKE0 bits in the serial control r
(SCSCR).

- Data length: Choice of seven or eight bits
- Choice of parity addition and addition of one or two stop bits (the combination of th
parameters determines the transfer format and character length)
- Detection of framing errors, parity errors, overrun errors, receive-FIFO-data-full sta
data-ready state, and breaks, during reception
- Indication of the number of data bytes stored in the transmit and receive FIFO regis
- Choice of internal or external clock as the SCIF clock source
  - When internal clock is selected: the SCIF operates on the baud rate generator cl
  - When external clock is selected: A clock must be input according to the samplin
example, when the sampling rate is 1/16, a clock with a frequency of 8 times the
must be input (the on-chip baud rate generator is not used.)

		1			
1	0	0		7-bit data	No
		1			
	1	0			Yes
		1			

SCSMR Settings			Serial Transfer Format and Frame Length									
CHR	**PE**	**STOP**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	0	0	S	8-bit data							STOP	
		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

S: Start bit
STOP: Stop bit
P: Parity bit

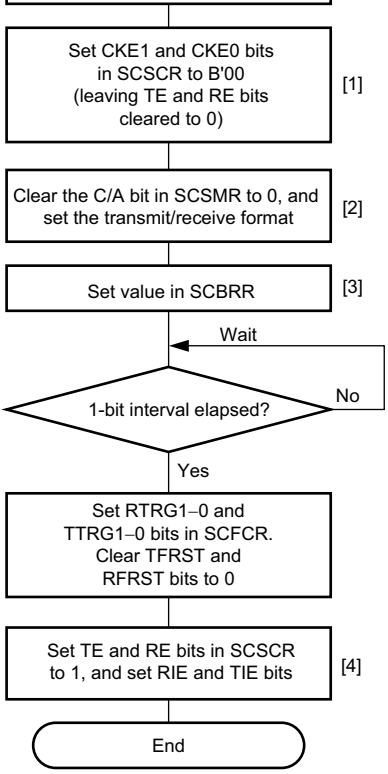
3. Data Transfer Operations

a. SCIF Initialization

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in S then initialize the SCIF as described below.

When the transfer format, etc., is changed, the TE and RE bits must be cleared to 0 be the change using the following procedure. When the TE bit is cleared to 0, the transmi register (SCTSR) is initialized. Note that clearing the TE and RE bits to 0 does not cha contents of SCSSR, SCFTDR, or SCFRDR. The TE bit should be cleared to 0 after all data has been sent and the TEND bit in SCSSR has been set to 1. Clearing to 0 can als performed during transmission, but the data being transmitted will go to the high-impe after the clearance. Before setting TE to 1 again to start transmission, the TFRST bit in should first be set to 1 to reset SCFTDR.

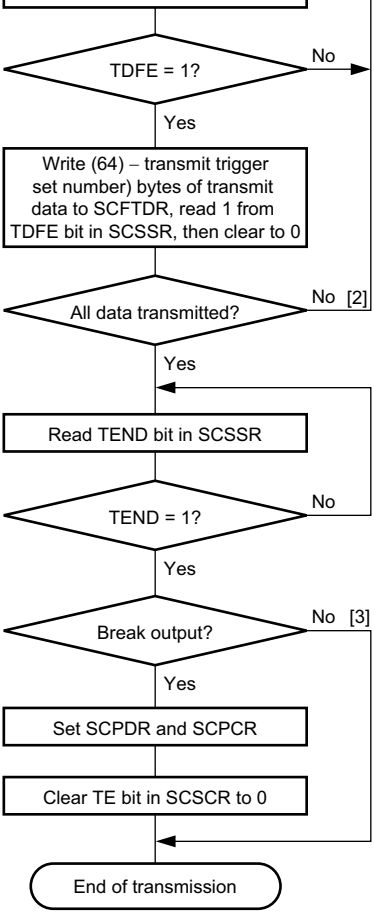
When an external clock is used, the clock should not be stopped during operation, incl initialization, since operation will be unreliable in this case.



[3] Write a value corresponding to the bit rate into SCBRR. (Not necessary if an external clock is used.)

[4] Wait at least one bit interval, then set the TE bi and RE bits in SCSCR to 1. Also set the RIE and TIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used. When transmitting, the TxD pin will go to the mark state; when receiving, the RxD pin will go to the idle state.

Figure 16.2 Sample SCIF Initialization Flowchart



Read the serial status register (SCSSR) and check that the TDFE flag is set to 1, then writ transmit data to SCFTDR, read 1 from the TD flag, then clear the flag to 0.
The number of data bytes that can be written 64 – (transmit trigger set number).

[2] Serial transmission continuation procedure:
To continue serial transmission, read 1 from TDFE flag to confirm that writing is possible, write data to SCFTDR, and then clear the TD bit to 0.

[3] Break output at the end of serial transmissio
To output a break in serial transmission, set t port SC data register (SCPDR) and port SC control register (SCPCR), then clear the TE b SCSCR to 0.
In steps 1 and 2, it is possible to ascertain th number of data bytes that can be written from number of transmit data bytes in SCFTDR indicated by the upper 8 bits of SCFDR.

Figure 16.3 Sample Serial Transmission Flowchart

number of transmit data bytes in SCFTDR falls to or below the transmit trigger num
the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in SCSCR is
this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

When the transmit data stop function is used and the number of data bytes set in the
data stop register (SCTDSR) is matched, transmit operation is stopped, and the TSF
serial status register (SCSSR) is set. If the TSIE bit in the serial control register (SC
to 1, a transmit-data-stop-interrupt (TDI) request is generated. The vectors of transm
data-empty and transmit-data-stop interrupts are the same.

The serial transmit data is sent from the TxD pin in the following order.

a. Start bit: One 0-bit is output.
b. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
c. Parity bit: One parity bit (even or odd parity) is output.
d. A format in which a parity bit is not output can also be selected.
e. Stop bit(s): One or two 1-bits (stop bits) are output.
f. Mark state: 1 is output continuously until the start bit that starts the next transmi
sent.

3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. I
present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and the
transmission of the next frame is started.

   If there is no transmit data, the TEND flag in SCSSR is set to 1, the stop bit is sent,
the line goes to the mark state in which 1 is output.

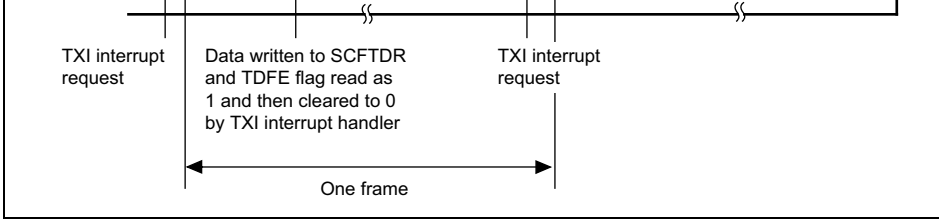


Figure 16.4 Example of Transmit Operation (Example with 8-Bit Data, Parity, One Stop Bit)

- Transmit Data Stop Function

When a value in the SCTDSR register is matched with the number of transmit data by function stops the transmit operation. Interrupts can be generated and the DMAC can by setting the TSIE bit (interrupt enable bit).

Figure 16.5 shows an example of operation for the transmit data stop function.

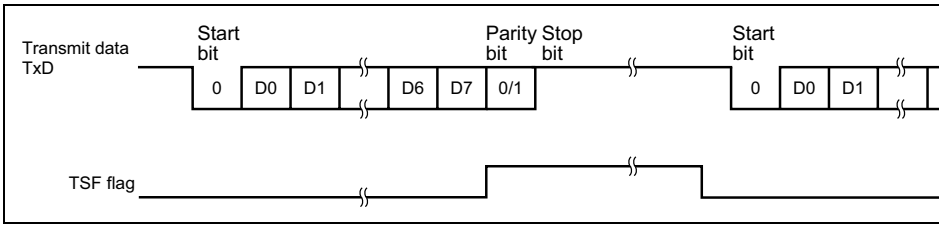
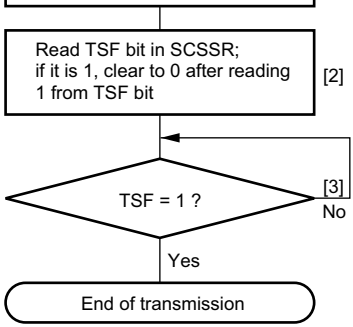


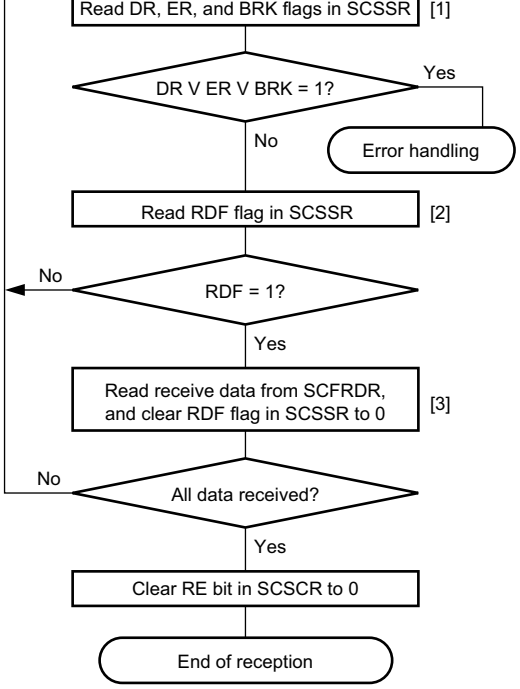
Figure 16.5 Example of Transmit Data Stop Function



is stopped. If the TSIE bit is set to 1, an interrupt is generated.

Serial transmission continuation procedure:
Set the TCRST bit in SCFCR to 1, clear transmit count, clear the TCRST bit to 0. Then follow steps 1, 2, and 3.

Figure 16.6 Transmit Data Stop Function Flowchart

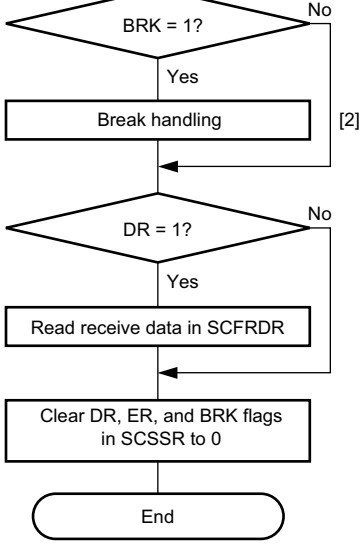


In the case of a framing error, a break c
detected by reading the value of the RxI

[2] SCIF status check and receive data rea
Read SCSSR and check that RDF = 1,
receive data in SCFRDR, read 1 from th
and then clear the RDF flag to 0. The tra
RDF flag from 0 to 1 can also be identifi
interrupt.

[3] Serial reception continuation procedure:
To continue serial reception, read at lea
trigger set number of data bytes from SC
1 from the RDF flag, and then clear the
0. The number of receive data bytes in S
be ascertained by reading the lower bits

Figure 16.7 Sample Serial Reception Flowchart (1)



[2] When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00 and the break data in which a framing error occurred is stored.

Figure 16.8 Sample Serial Reception Flowchart (2)

a. Stop bit check: the SCIF checks whether the stop bit is 1. If there are two stop
first is checked.

b. The SCIF checks whether receive data can be transferred from the receive shif
(SCRSR) to SCFRDR.

c. Break check: the SCIF checks that the BRK flag is 0, indicating that the break
set.

If all the above checks are passed, the receive data is stored in SCFRDR.

Note: Reception continues when a receive error (a framing error or parity error) occu

4. If the RIE bit in SCSCR is set to 1 when the RDF flag changes to 1, a receive-FIF
interrupt (RXI) request is generated.
If the ERIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-erro
(ERI) request is generated.
If the BRIE bit in SCSCR is set to 1 when the BRK flag changes to 1, a break rece
interrupt (BRI) request is generated.
If the DRIE bit in SCSCR is set to 1 when the DR flag changes to 1, a receive-data
interrupt (DRI) request is generated.
The vectors of receive-FIFO-data-full and receive-data-ready interrupts are the sa
vectors of receive-error and break reception interrupts are the same.

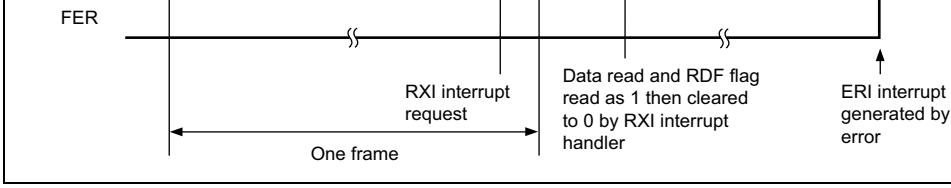


Figure 16.9 Example of SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

- Modem Function

When using a modem function, transmission can be stopped and started again accordin $\overline{\text{CTS}}$ input value. When the $\overline{\text{CTS}}$ is set to 1 during transmission, the data enters a mark s transmitting one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is output starting w bit.

Figure 16.10 shows an example of operation for the $\overline{\text{CTS}}$ control.

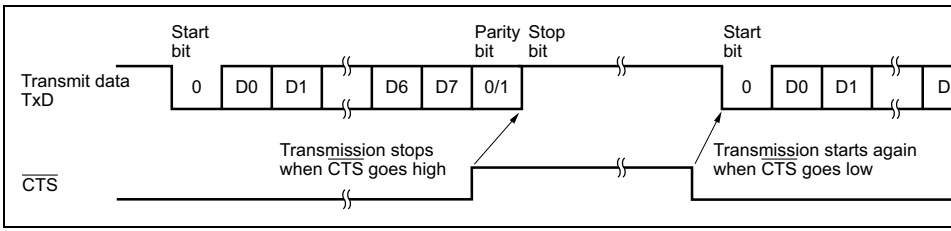


Figure 16.10 $\overline{\text{CTS}}$ Control Operation

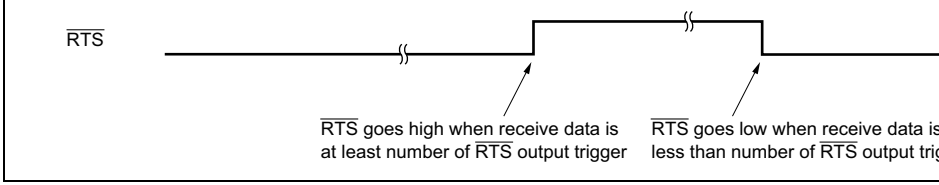


Figure 16.11 RTS Control Operation

16.4.4 Clock Synchronous Mode

64-stage FIFO buffers are provided for both transmission and reception, reducing the overhead and enabling fast, continuous communication to be performed.

The operating clock source is selected using the serial mode register (SCSMR). The S source is determined by the CKE1 and CKE0 bits in the serial control register (SCSCR

- Transmit/receive format: Fixed 8-bit data
- Indication of the number of data bytes stored in the transmit and receive FIFO regi
- Internal clock or external clock used as the SCIF clock source

  When the internal clock is selected:
  The SCIF operates on the baud rate generator clock and outputs a serial clock from

  When the external clock is selected:
  The SCIF operates on the external clock input through the SCK pin.

Figure 16.12 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, data on the communication line is output f
falling edge of the serial clock to the next falling edge. Data is guaranteed valid at the r
of the serial clock.

In serial communication, each character is output starting with the LSB and ending witl
After the MSB is output, the communication line remains in the state of the MSB.

In clock synchronous mode, the SCIF receives data in synchronization with the rising e
serial clock.

1. Data Transfer Format

A fixed 8-bit data format is used. No parity or multiprocessor bits are added.

2. Clock

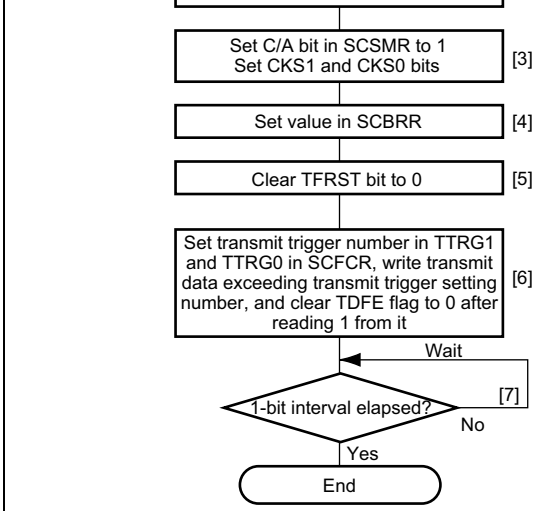
An internal clock generated by the on-chip baud rate generator or an external clock inp
the SCK pin can be selected as the serial clock for the SCIF, according to the setting of
and CKE0 bits in SCSCR.

Eight serial clock pulses are output in the transfer of one character, and when no
transmission/reception is performed, the clock is fixed high. However, when the operat
reception only, the synchronous clock output continues while the RE bit is set to 1. To
clock high every time one character is transferred, write to the transmit FIFO data regis
(SCFTDR) the same number of dummy data bytes as the data bytes to be received and
and RE bits to 1 at the same time to transmit the dummy data. When the specified numb
bytes are transmitted, the clock is fixed high.

3. Data Transfer Operations

a. SCIF Initialization:

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SC
then initialize the SCIF as described below.



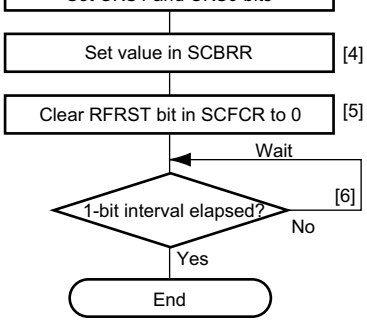
[4] Write a value corresponding to the bit rate into SCBRR.

[5] Clear the TFRST bit in SCFCR to 0.

[6] Set the transmit trigger number, write transmit data exceeding the transmit trigger setting number, and clear the TDFE flag to 0 after reading it.

[7] Wait one bit interval.

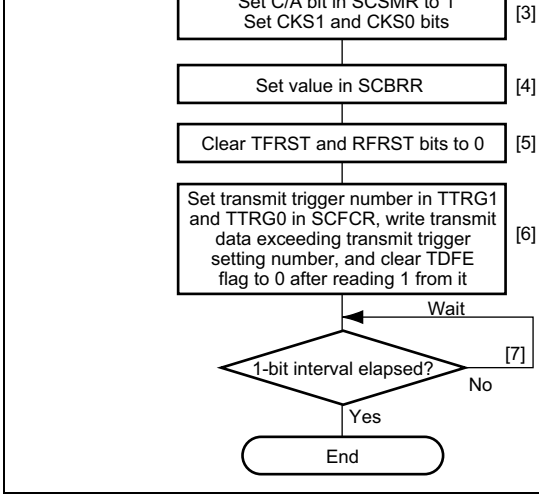
Figure 16.13 Sample SCIF Initialization Flowchart (1) (Transmission)



[5] Clear the RFRST bit in SCFCR to 0.

[6] Wait one bit interval.

Figure 16.13 Sample SCIF Initialization Flowchart (2) (Reception)



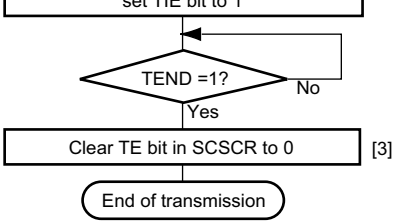
[4] Write a value corresponding to the bit rate into SCBRR.

[5] Clear the TFRST and RFRST bits in SCFCR to 0.

[6] Set the transmit trigger number, write transmit data exceeding the transmit trigger setting number, and clear the TDFE flag to 0 after reading it.

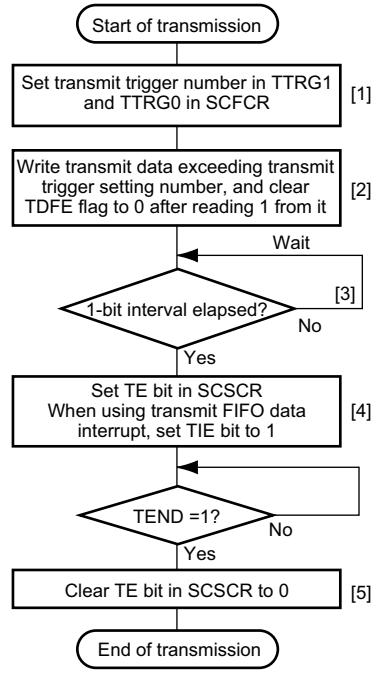
[7] Wait one bit interval.

Figure 16.13 Sample SCIF Initialization Flowchart (3) (Simultaneous Transmission and Reception)



[5] After the end of transmission, clear the TE bit to 0.

Figure 16.14 Sample Serial Transmission Flowchart (1)
(First Transmission after Initialization)



[1] Set the transmit trigger number in SCFCR.

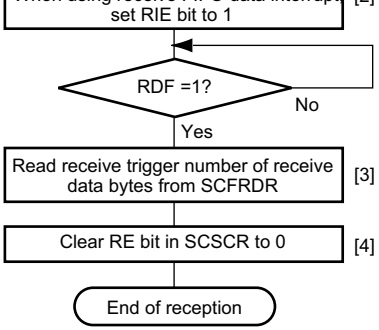
[2] Write transmit data to SCFTDR, and clear the TDFE flag to 0 after reading 1 from it.

[3] Wait for one bit interval.

[4] Transmission is started when the TE bit in SCSCR is set to 1.

[5] After the end of transmission, clear the TE bit to 0.

Figure 16.14 Sample Serial Transmission Flowchart (2)
(Second and Subsequent Transmission)

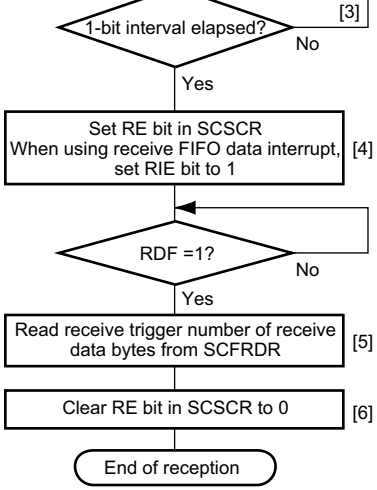


RE bit in SCSCR is set to 1.

[3] Read receive data while the RDF bit is 1.

[4] After the end of reception, clear the RE bit to 0.

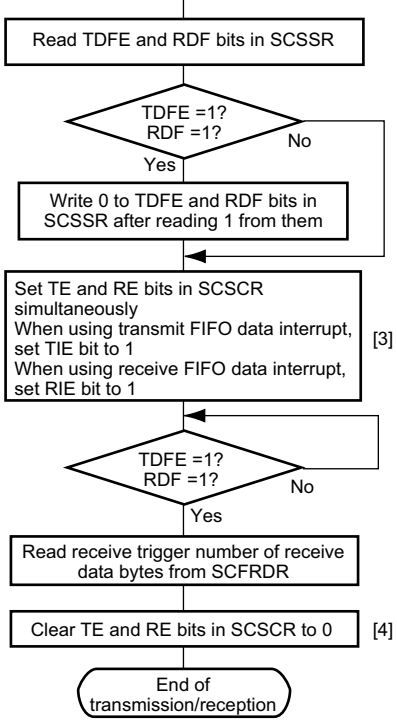
Figure 16.15 Sample Serial Reception Flowchart (1)
(First Reception after Initialization)



[5] Read receive data while the RDF bit is 1.

[6] After the end of reception, clear the RE bit to 0.

Figure 16.15 Sample Serial Reception Flowchart (2)
(Second and Subsequent Reception)

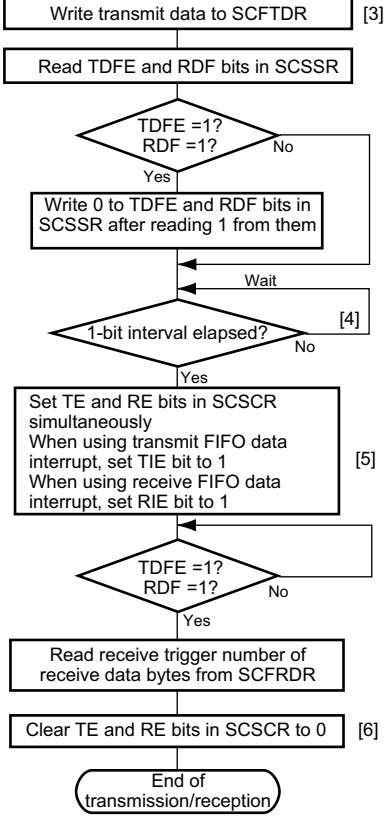


to SCFTDR, and if there is receive data in the FIFO, read receive data until there is less than the receive trigger setting number, read the TDFE and RDF bits in SCSSR, and if 1, clear to 0.

[3] Transmission/reception is started when the TE and RE bits in SCSCR are set to 1. The TE and RE bits must be set simultaneously.

[4] After the end of transmission/reception, clear the TE and RE bits to 0.

Figure 16.16 Sample Simultaneous Serial Transmission and Reception Flowch (First Transfer after Initialization)



read the TDFE and RDF bits in SCSSR and if 1, clear to 0.

[4] Wait for one bit interval.

[5] Transmission/reception is started when the TE and RE bits in SCSCR are set to 1. The TE and RE bits must be set simultaneously.

[6] After the end of transmission/reception, clear the TE and RE bits to 0.

Figure 16.16 Sample Simultaneous Serial Transmission and Reception Flow (Second and Subsequent Transfer)

and receive-FIFO-data-full (RXI).

Table 16.4 shows the interrupt sources. The interrupt sources can be enabled or disable
of the TIE, RIE, ERIE, BRIE, DRIE, and TSIE bits in SCSCR.

When the TDFE flag in SCSSR is set to 1, a TXI interrupt request is generated. When t
flag in SCSSR is set to 1, a TDI interrupt request is generated. The DMAC can be activ
data transfer performed on generation of TXI and TDI interrupt requests. The DMAC r
TXI and TDI are assigned to the same vector.

When the RDF flag in SCSSR is set to 1, an RXI interrupt request is generated. The DM
be activated and data transfer performed on generation of an RXI interrupt request.

When using the DMAC for transmission/reception, set and enable the DMAC before m
settings. See section 8, Direct Memory Access Controller (DMAC), for details of the D
setting procedure.

When the ER flag in SCSSR is set to 1, an ERI interrupt request is generated. When the
in SCSSR is set to 1, a BRI interrupt request is generated. When the DR flag in SCSSR
a DRI interrupt request is generated. When the TSF flag in SCSSR is set to 1, a TDI int
request is generated.

The vectors of TXI and TDI, ERI and BRI, and RXI and DRI are the same.

The DMAC activation and interrupts cannot be generated simultaneously by the same s
following procedure should be used for the DMAC activation.

1. Set the interrupt enable bits (TIE and RIE) corresponding to the generated source t
2. Mask the corresponding interrupt requests by using the interrupt mask register of th
   controller.

Notes: 1. The DMAC can be activated only by a receive-FIFO-data-full interrupt requ
2. The DMAC can be activated by a transmit-FIFO-data-empty (TDFE) or tran
stop (TSF) interrupt request. When the DMAC is activated by the TSF inter
cleared by either of two cases listed below.
(1) The TSF flag is read by the CPU.
(2) The transmit FIFO is full.

See section 5, Exception Handling, for priorities and the relationship with non-SCIF i

continuous transmission.

However, if the number of data bytes written in SCFTDR is equal to or less than the tra
trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to
clearing should therefore be carried out when SCFTDR contains more than the transmit
number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from the 14 to 8 bits of the
count register (SCFDR).

b. SCFRDR Reading and the RDF Flag:

The RDF flag in the serial status register (SCSSR) is set when the number of receive da
the receive FIFO data register (SCFRDR) has become equal to or greater than the recei
number set by bits RTRG1 and RTRG0 in the FIFO control register (SCFCR). After RI
receive data equivalent to the trigger number can be read from SCFRDR, allowing effic
continuous reception.

However, if the number of data bytes in SCFRDR is still equal to or greater than the tri
number after a read, the RDF flag will be set to 1 again if it is cleared to 0. RDF should
be cleared to 0 after being read as 1 after all receive data has been read.

The number of receive data bytes in SCFRDR can be found from the 6 to 0 bits of the R
count register (SCFDR).

c. Break Detection and Processing:

Break signals can be detected by reading the RxD pin directly when a framing error (FE
detected. In the break state the input from the RxD pin consists of all 0s, so the FER fla
the parity error flag (PER) may also be set.

Although the SCIF stops transferring receive data to SCFRDR after receiving a break, t
operation continues.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{1 + F}{N} (1 + F) \right| \times 100\% \quad \text{..... (1)}$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by eq

When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\% = 46.875\% \quad \text{..... (2)}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 3

17.1 Features

- Conforms to the IrDA 1.0 system
- Asynchronous serial communication
  - Data length: 8 bits
  - Stop bit length: 1 bit
  - Parity bit: None
- On-chip 64-stage FIFO buffers for both transmit and receive operations
- On-chip baud rate generator with selectable bit rates
- Guard functions to protect the receiver during transmission
- Clock supply halted to reduce power consumption when not using the IrDA interfa

Figure 17.1 shows a block diagram of the IrDA.

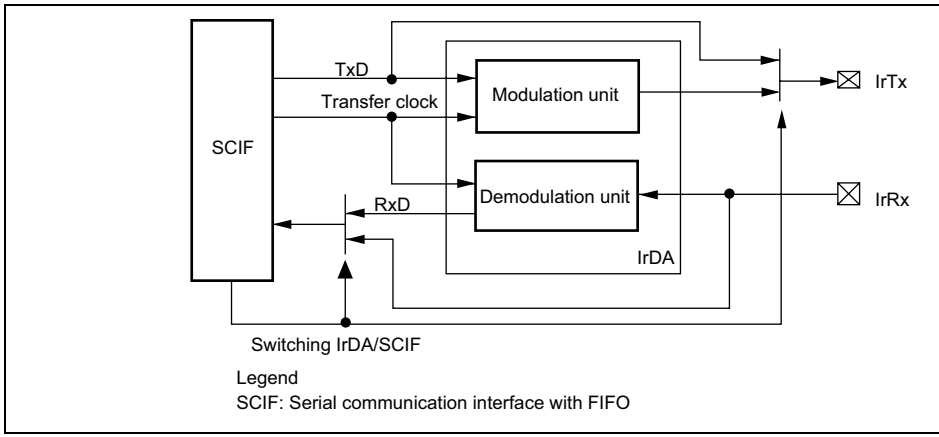


Figure 17.1 Block Diagram of IrDA

Note: Clock input from the serial clock pin cannot be set in IrDA mode.

17.3 Register Description

The IrDA has the following internal registers. For details on register addresses and regi
in each processing state, refer to section 24, List of Registers.

- IrDA mode register (SCSMR_Ir)

17.3.1 IrDA Mode Register (SCSMR_Ir)

SCSMR_Ir is a 16-bit register that selects IrDA or SCIF mode and selects the IrDA out
width.

This module operates as IrDA when the IRMOD bit is set to 1. When the IRMOD bit is
0, this module can also operate as an SCIF.

				1: Operates as an IrDA
6 to 3	ICK3 to ICK0	0	R/W	Output Pulse Division Ratio Specifies the ratio for dividing the clock (Pϕ) to generate the IRCLK to be used for IrDA. IRCLK is obta follows: IRCLK = 1/(2N + 2) × Pϕ N = Value set by ICK3 to ICK0
2	PSEL	0	R/W	Output Pulse Width Select PSEL selects an IrDA output puls is 3/16 of the bit length for 115 kb the bit length for the selected bau 0: Pulse width is 3/16 of the bit le 1: Pulse width is 3/16 of 115 kbps the baud rate selected by ICK3
1, 0	—	0	R	Reserved These bits are always read as 0. value should always be 0.

Example:

Pϕ clock: 14.7456 MHz

IRCLK: 921.6 kHz (fixed)

N: Setting of ICK3 to ICK0 ($0 \leq N \leq 15$)

$$N \geq \frac{P\phi}{2 \times IRCLK} - 1 \geq 7$$

Accordingly, N is 7.

The IrDA module modifies IrTx/IrRx transmit/receive data waveforms to satisfy the IrI specification for infrared communication.

In the IrDA 1.0 specification, communication is first performed at a speed of 9600 bps, communication speed is changed. However, the communication rate cannot be automat changed in this module, so the communication speed should be confirmed, and the appr speed set for this module by software.

17.4.2 Transmitting

The waveforms of a serial output signal (UART frame) from the SCIF are modified and is converted into the IR frame serial output signal by the IrDA module, as shown in figu

When serial data is 0, a pulse of 3/16 the IR frame bit width is generated and output. W data is 1, no pulse is output.

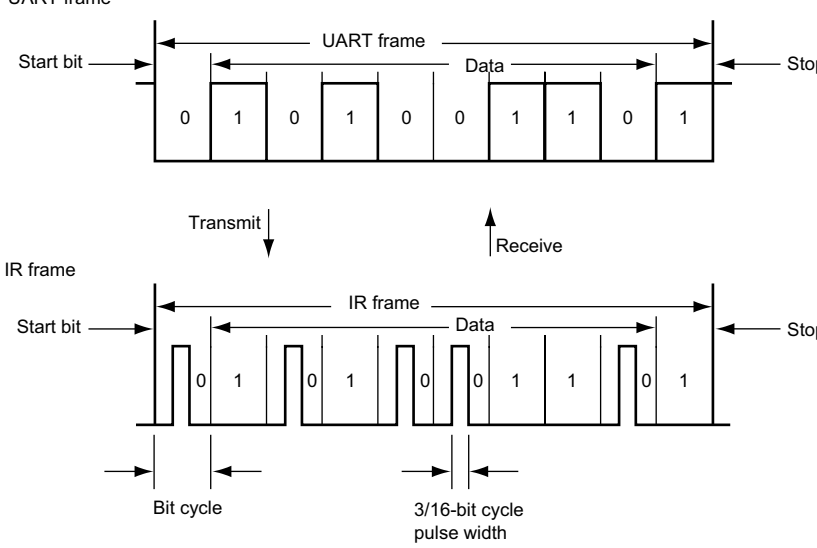


Figure 17.2 Transmit/Receive Operation

17.4.4 Data Format Specification

The data format of UART frames used for IrDA communication must be specified by registers. The UART frame has eight data bits, no parity bit, and one stop bit.

IrDA communication is performed in asynchronous mode, and this mode must also be by the SCIF0 registers. The sampling rate must be set to 1/16.

The internal clock must be selected for the SCIF0 operation clock and the SCK0 pin m specified for the synchronizing clock output pin.

The IrDA communication rate is the same as the SCIF0 bit rate, which is specified by registers.

For details on SCIF0 registers, refer to section 16, Serial Communication Interface wi (SCIF).

Automatic processing of USB standard commands for endpoint 0 (some command
class/vendor commands require decoding and processing by firmware)

- Transfer speed: Full-speed
- Endpoint configuration:

Endpoint Name	Abbreviation	Transfer Type	Maximum Packet Size	FIFO Buffer Capacity (Byte)	DMA
Endpoint 0	EP0s	Setup	8	8	—
	EP0i	Control-in	8	8	—
	EP0o	Control-out	8	8	—
Endpoint 1	EP1	Bulk-out	64	128	Possib
Endpoint 2	EP2	Bulk-in	64	128	Possib
Endpoint 3	EP3	Interrupt	8	8	—

- Interrupt requests: Generates various interrupt signals necessary for USB transmission/reception
- Clock: External input (48 MHz) (Refer to section 9.4.1, Frequency Control Regist and section 9.4.2, USB Clock Frequency Control Register (UCLKCR))
- Power-down mode

  Power consumption can be reduced by stopping UDC internal clock when USB ca disconnected

  Automatic transition to/recovery from suspend state
- Can be connected to a Philips PDIUSBP11 Series transceiver or compatible produ transceiver bypass mode (the XVEROFF bit in XVERCR is set to 1)

  When using a compatible product, carry out evaluation and investigation with the manufacturer supplying the transceiver beforehand.
- Power mode: Self-powered

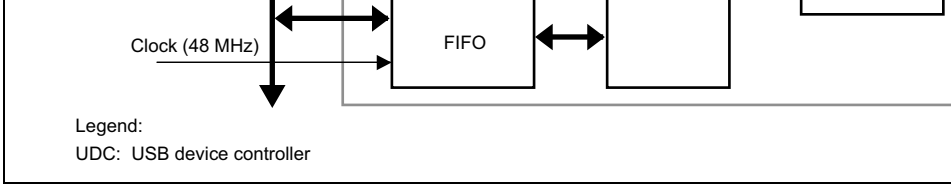


Figure 18.1 Block Diagram of USB

DPLS	Input	Input pin to driver for D+ signal from receiver	1
DMNS	Input	Input pin to driver for D– signal from receiver	1
TXDPLS	Output	D+ transmit output pin to driver	1
TXDMNS	Output	D- transmit output pin to driver	1
TXENL	Output	Driver output enable pin	1
VBUS	Input	USB cable connection monitor pin	1 or
SUSPND	Output	Transceiver suspend state output pin	1
EXTAL_USB	Input	USB clock input pin (external clock input/crystal resonator connect)	
XTAL_USB	Output	USB clock pin (crystal resonator connect)	
D+	I/O	USB internal transceiver D+	
D-	I/O	USB internal transceiver D–	
Vcc-USB	Input	Power supply for USB	
Vss-USB	Input	Ground for USB	

Note: The USB can be connected to a Philips PDIUSBP11 Series transceiver or com
product in internal transceiver bypass mode (the XVEROFF bit in XVERCR is s
When using a compatible product, carry out evaluation and investigation with th
manufacturer supplying the transceiver beforehand.

- Interrupt select register 1 (ISR1)
- Interrupt enable register 0 (IER0)
- Interrupt enable register 1 (IER1)
- EP0i data register (EPDR0i)
- EP0o data register (EPDR0o)
- EP0s data register (EPDR0s)
- EP1 data register (EPDR1)
- EP2 data register (EPDR2)
- EP3 data register (EPDR3)
- EP0o receive data size register (EPSZ0o)
- EP1 receive data size register (EPSZ1)
- Trigger register (TRG)
- Data status register (DASTS)
- FIFO clear register (FCLR)
- DMA transfer setting register (DMAR)
- Endpoint stall register (EPSTL)
- Transceiver control register (XVERCR)

Bit	Bit Name	Value	R/W	Description
7	BRST	0	R/W	Bus Reset This bit is set to 1 when a bus reset signal is the USB bus.
6	EP1FULL	0	R	EP1 FIFO Full This bit is set when endpoint 1 receives one data successfully from the host, and holds a long as there is valid data in the FIFO buffer. This is a status bit, and cannot be cleared.
5	EP2TR	0	R/W	EP2 Transfer Request This bit is set if there is no valid transmit data buffer when an IN token for endpoint 2 is rece the host. A NACK handshake is returned to t data is written to the FIFO buffer and packet is enabled.
4	EP2EMPTY	1	R	EP2 FIFO Empty This bit is set when at least one of the dual e transmit FIFO buffers is ready for transmit da written. This is a status bit, and cannot be cleared.
3	SETUPTS	0	R/W	Setup Command Receive Complete This bit is set to 1 when endpoint 0 receives a setup command requiring decoding on the side, and returns an ACK handshake to the h
2	EP0oTS	0	R/W	EP0o Receive Complete This bit is set to 1 when endpoint 0 receives host successfully, stores the data in the FIFO returns an ACK handshake to the host.

				This bit is set when data is transmitted to the h endpoint 0 and an ACK handshake is returned

18.3.2 Interrupt Flag Register 1 (IFR1)

IFR1, together with interrupt flag register 0 (IFR0), indicates interrupt status informatio by the application. When an interrupt source is generated, the corresponding bit is set to interrupt request is sent to the CPU according to the combination with interrupt enable (IER1). Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bi

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	0	R	Reserved These bits are always read as 0. The write val always be 0.
3	VBUSMN	0	R	This is a status bit which monitors the state of pin. This bit reflects the state of the VBUS pin.
2	EP3TR	0	R/W	EP3 Transfer Request This bit is set if there is no valid transmit data i buffer when an IN token for endpoint 3 is recei the host. A NACK handshake is returned to th data is written to the FIFO buffer and packet tr is enabled.
1	EP3TS	0	R/W	EP3 Transmit Complete This bit is set when data is transmitted to the h endpoint 3 and an ACK handshake is returned
0	VBUS	0	R/W	USB Disconnection Detection When the function is connected to the USB bu disconnected from it, this bit is set to 1. The VB this module is used for detecting connection o disconnection.

7	BRST	0	R/W	Bus Reset
6	EP1FULL	0	R/W	EP1 FIFO Full
5	EP2TR	0	R/W	EP2 Transfer Request
4	EP2EMPTY	0	R/W	EP2 FIFO Empty
3	SETUPTS	0	R/W	Setup Command Receive C
2	EP0oTS	0	R/W	EP0o Receive Complete
1	EP0iTR	0	R/W	EP0i Transfer Request
0	EP0iTS	0	R/W	EP0i Transmit Complete

18.3.4 Interrupt Select Register 1 (ISR1)

ISR1 selects the vector numbers of the interrupt requests indicated in interrupt flag reg (IFR1). If the USB issues an interrupt request to the INTC when a bit in ISR1 is cleare interrupt corresponding to the bit will be USI0 (USB interrupt 0). If the USB issues an request to the INTC when a bit in ISR1 is set to 1, the corresponding interrupt will be interrupt 1). If interrupts occur simultaneously, USI0 has priority by default.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	0	R	Reserved These bits are always read as 0. The writ should always be 0.
2	EP3TR	1	R/W	EP3 Transfer Request
1	EP3TS	1	R/W	EP3 Transmit Complete
0	VBUS	1	R/W	USB Bus Connect

6	EP1FULL	0	R/W	EP1 FIFO Full
5	EP2TR	0	R/W	EP2 Transfer Request
4	EP2EMPTY	1	R/W	EP2 FIFO Empty
3	SETUPTS	0	R/W	Setup Command Receive Complet
2	EP0oTS	0	R/W	EP0o Receive Complete
1	EP0iTR	0	R/W	EP0i Transfer Request
0	EP0iTS	0	R/W	EP0i Transmit Complete

18.3.6 Interrupt Enable Register 1 (IER1)

IER1 enables the interrupt requests of interrupt flag register 1 (IFR1). When an interrup to 1 while the corresponding bit of each interrupt is set to 1, an interrupt request is sent CPU. The interrupt vector number is determined by the contents of interrupt select regi (ISR1).

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	0	R	Reserved These bits are always read as 0. The write should always be 0.
2	EP3TR	0	R/W	EP3 Transfer Request
1	EP3TS	0	R/W	EP3 Transmit Complete
0	VBUS	0	R/W	USB Bus Connect

7 to 0	D7 to D0	Undefined	W	Data register for control-in transfer

18.3.8 EP0o Data Register (EPDR0o)

EPDR0o is an 8-byte receive FIFO buffer for endpoint 0. EPDR0o holds endpoint 0 re other than setup commands. When data is received successfully, EP0oTS in interrupt 0 is set, and the number of receive bytes is indicated in the EP0o receive data size regi the data has been read, setting EP0oRDFN in the trigger register enables the next pack received. This FIFO buffer can be initialized by means of BP0oCLR in the FCLR regi

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	R	Data register for control-out transfe

18.3.9 EP0s Data Register (EPDR0s)

EPDR0s is an 8-byte FIFO buffer specifically for receiving endpoint 0 setup comman setup command to be processed by the application is received. When command data is successfully, the SETUPTS bit in interrupt flag register 0 is set.

As a latest setup command must be received in high priority, if data is left in this buffe overwritten with new data. If reception of the next command is started while the curre is being read, command reception has priority, the read by the application is forcibly s the read data is invalid.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	R	Data register for storing the setup the control-out transfer

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	R	Data register for endpoint 1 transfer

18.3.11 EP2 Data Register (EPDR2)

EPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. EPDR2 has a dual-buffer co and has a capacity of twice the maximum packet size. When transmit data is written to buffer and EP2PKTE in the trigger register is set, one packet of transmit data is fixed, a dual-FIFO buffer is switched over. The transmit data for this FIFO buffer can be transfe DMA. This FIFO buffer can be initialized by means of EP2CLR in the FCLR register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 2 transfer

18.3.12 EP3 Data Register (EPDR3)

EPDR3 is an 8-byte transmit FIFO buffer for endpoint 3. EPDR3 holds one packet of tr for the interrupt transfer of endpoint 3. Transmit data is fixed by writing one packet of setting EP3PKTE in the trigger register. When an ACK handshake is returned from the one packet of data has been transmitted successfully, EP3TS in interrupt flag register 0 FIFO buffer can be initialized by means of EP3CLR in the FCLR register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	R	Data register for endpoint 3 transfer

EPSZ1 is a receive data size resister for endpoint 1. EPSZ1 indicates the number of by
from the host. The FIFO for endpoint 1 has a dual-buffer configuration. The size of th
data indicated by this register is the size of the currently selected side (can be read by

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	0	R	Number of received bytes for endp

				After one packet of data has been written endpoint 3 transmit FIFO buffer, the trans fixed by writing 1 to this bit.
5	EP1RDFN	Undefined	W	EP1 Read Complete Write 1 to this bit after one packet of data read from the endpoint 1 FIFO buffer. Th 1 receive FIFO buffer has a dual-buffer configuration. Writing 1 to this bit initializ that was read, enabling the next packet t received.
4	EP2PKTE	Undefined	W	EP2 Packet Enable After one packet of data has been written endpoint 2 transmit FIFO buffer, the trans fixed by writing 1 to this bit.
3	—	Undefined	—	Reserved The write value should always be 0.
2	EP0sRDFN	Undefined	W	EP0s Read Complete Write 1 to this bit after data for the EP0s FIFO has been read. Writing 1 to this bit transfer of data in the following data stag handshake is returned in response to tra requests from the host in the data stage written to this bit.
1	EP0oRDFN	Undefined	W	EP0o Read Complete Writing 1 to this bit after one packet of da been read from the endpoint 0 transmit F initializes the FIFO buffer, enabling the n to be received.
0	EP0iPKTE	Undefined	W	EP0i Packet Enable After one packet of data has been written endpoint 0 transmit FIFO buffer, the trans fixed by writing 1 to this bit.

5	EP3DE	0	R	EP3 Data Present This bit is set when the endpoint 3 FIFO contains valid data.
4	EP2DE	0	R	EP2 Data Present This bit is set when the endpoint 2 FIFO contains valid data.
3 to 1	—	0	R	Reserved This bit is always read as 0.
0	EP0iDE	0	R	EP0i Data Present This bit is set when the endpoint 0 FIFO contains valid data.

18.3.17 FIFO Clear Register (FCLR)

FCLR is a register to initialize the FIFO buffers for each endpoint. Writing 1 to a bit c data in the corresponding FIFO buffer. Note that the corresponding interrupt flag is no Do not clear a FIFO buffer during transfer.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved The write value should always be 0.
6	EP3CLR	Undefined	W	EP3 Clear Writing 1 to this bit initializes the endpoi FIFO buffer.
5	EP1CLR	Undefined	W	EP1 Clear Writing 1 to this bit initializes both sides endpoint 1 receive FIFO buffer.

				FIFO buffer.
0	EP0iCLR	Undefined	W	EP0i Clear Writing 1 to this bit initializes the endpoin FIFO buffer.

18.3.18 DMA Transfer Setting Register (DMAR)

DMA transfer can be carried out between the endpoint 1 and 2 data registers and memo
means of the on-chip direct memory access controller (DMA). Dual address transfer is
in bytes. To start DMA transfer, DMAC settings must be made in addition to the setting
register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	0	R	Reserved These bits are always read as 0. The write va always be 0.

DMAC again. However, if the size of the dat
be transmitted is less than 64 bytes, the EP2
enable bit is not set automatically, and so sh
by the CPU with a DMA transfer end interrup

As EP2-related interrupt requests to the CPU
automatically masked, interrupt requests sho
masked as necessary in the interrupt enable

- Operating procedure

1. Write of 1 to the EP2 DMAE bit in DMAR
2. Transfer count setting in the DMAC
3. DMAC activation
4. DMA transfer
5. DMA transfer end interrupt generated

Refer to section 18.7.3, DMA Transfer for En

automatically masked.

- Operating procedure:

1. Write of 1 to the EP1 DMAE bit in DMAR
2. Transfer count setting in the DMAC
3. DMAC activation
4. DMA transfer
5. DMA transfer end interrupt generated

Refer to section 18.7.2, DMA Transfer for End

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	0	R	Reserved These bits are always read as 0. The w should always be 0.
3	EP3STL	0	R/W	EP3 Stall When this bit is set to 1, endpoint 3 is p stall state.
2	EP2STL	0	R/W	EP2 Stall When this bit is set to 1, endpoint 2 is p stall state.
1	EP1STL	0	R/W	EP1 Stall When this bit is set to 1, endpoint 1 is p stall state.
0	EP0STL	0	R/W	EP0 Stall When this bit is set to 1, endpoint 0 is p stall state.

18.3.20 Transceiver Control Register (XVERCR)

The Transceiver Control Register sets either the internal transceiver or external transc

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	0	R	Reserved These bits are always read as 0. The w should always be 0.
0	XVEROFF	0	R/W	Transceiver Control 1: The internal transceiver function is st digital signal for the external transcei from the port. 0: The internal transceiver is operated.

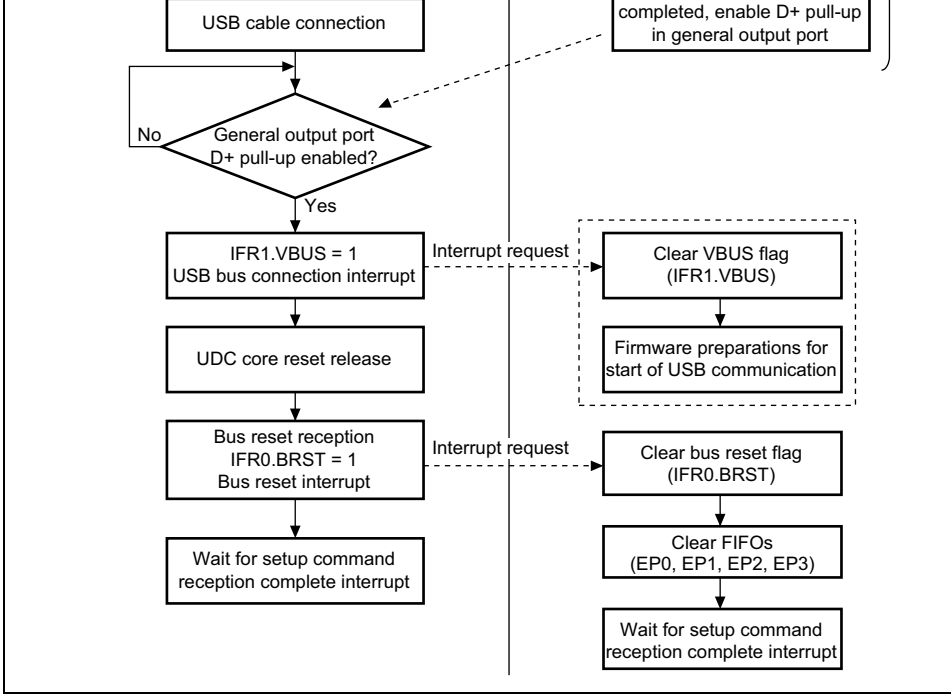


Figure 18.2 Cable Connection Operation

The above flowchart shows the operation in the case of in section 18.8, Example of US
Circuitry.

In applications that do not require USB cable connection to be detected, processing by t
bus connection interrupt is not necessary. Preparations should be made with the bus-res
interrupt.

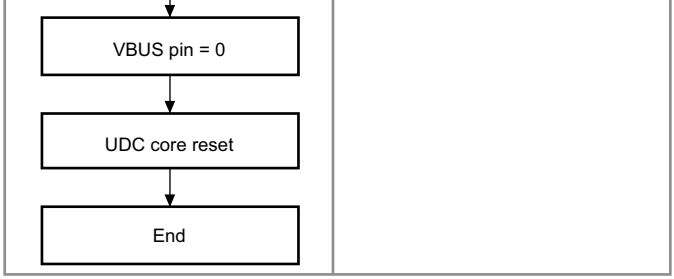


Figure 18.3 Cable Disconnection Operation

The above flowchart shows the operation in section 18.8, Example of USB External C

18.4.3 Control Transfer

Control transfer consists of three stages: setup, data (not always included), and status
The data stage comprises a number of bus transactions. Operation flowcharts for each
shown below.

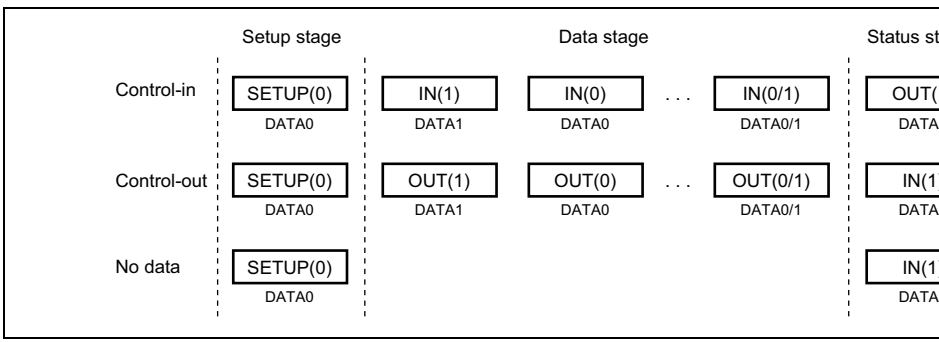
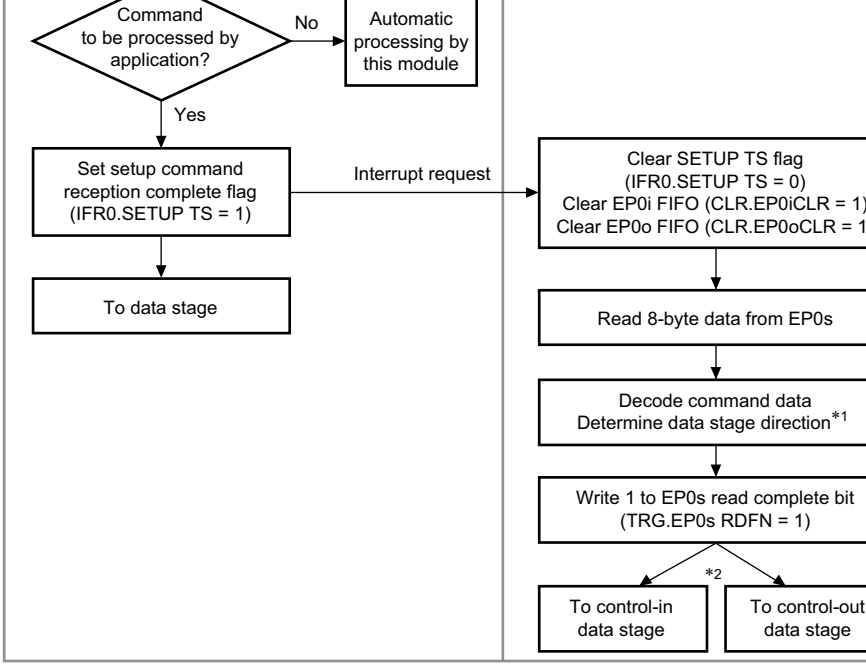


Figure 18.4 Transfer Stages in Control Transfer



Notes: 1. In the setup stage, the application analyzes command data from the host requiring processing the application, and determines the subsequent processing (for example, data stage direction
2. When the transfer direction is control-out, the EP0i transfer request interrupt required in the st stage should be enabled here. When the transfer direction is control-in, this interrupt is not re and should be disabled.

Figure 18.5 Setup Stage Operation

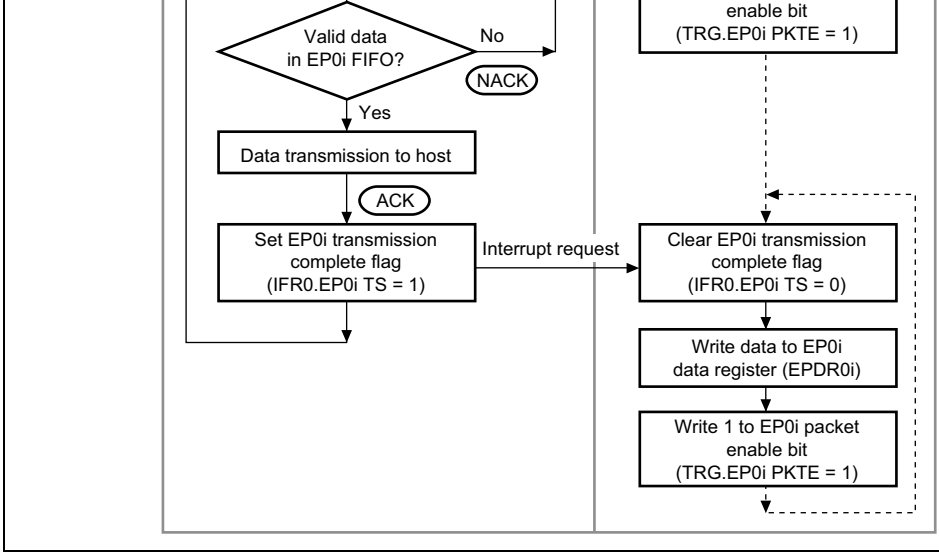


Figure 18.6 Data Stage (Control-In) Operation

The application first analyzes command data from the host in the setup stage, and dete
subsequent data stage direction. If the result of command data analysis is that the data
transfer, one packet of data to be sent to the host is written to the FIFO. If there is mor
sent, this data is written to the FIFO after the data written first has been sent to the hos
bit in IFR0 = 1).

The end of the data stage is identified when the host transmits an OUT token and the s
is entered.

Note: If the size of the data transmitted by the function is smaller than the data size r
the host, the function indicates the end of the data stage by returning to the hos
shorter than the maximum packet size. If the size of the data transmitted by th
an integral multiple of the maximum packet size, the function indicates the en
stage by transmitting a zero-length packet.

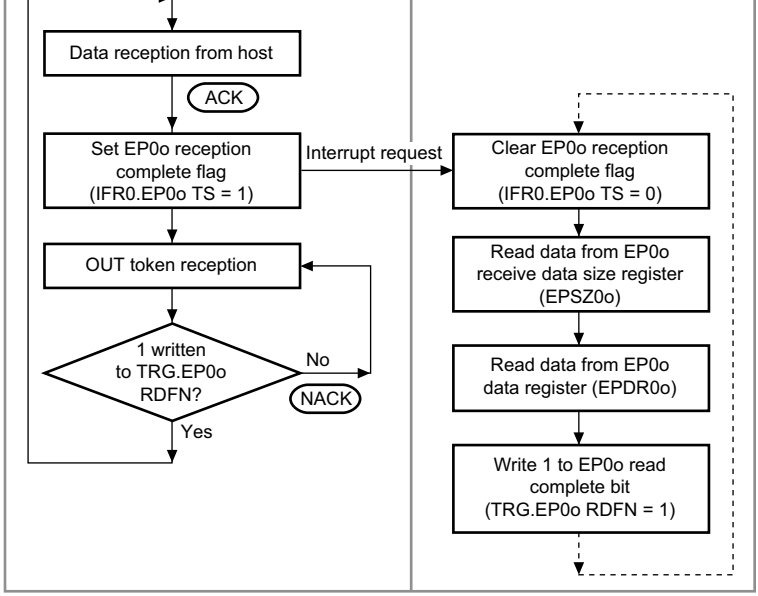


Figure 18.7 Data Stage (Control-Out) Operation

The application first analyzes command data from the host in the setup stage, and deter subsequent data stage direction. If the result of command data analysis is that the data s transfer, the application waits for data from the host, and after data is received (EP0oTS IFR0 = 1), reads data from the FIFO. Next, the application writes 1 to the EP0o read co empties the receive FIFO, and waits for reception of the next data.

The end of the data stage is identified when the host transmits an IN token and the statu entered.

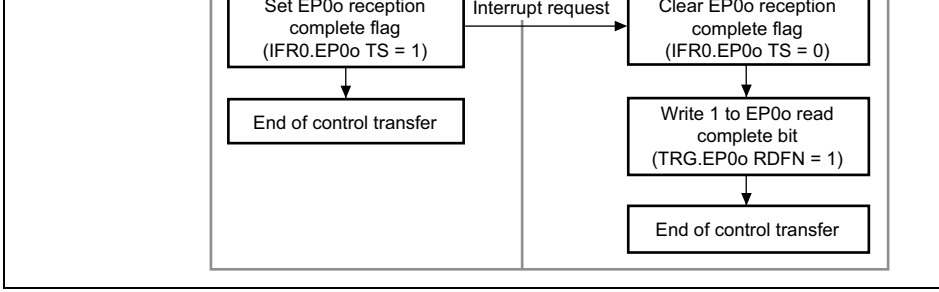


Figure 18.8 Status Stage (Control-In) Operation

The control-in status stage starts with an OUT token from the host. The application re
data from the host, and ends control transfer.

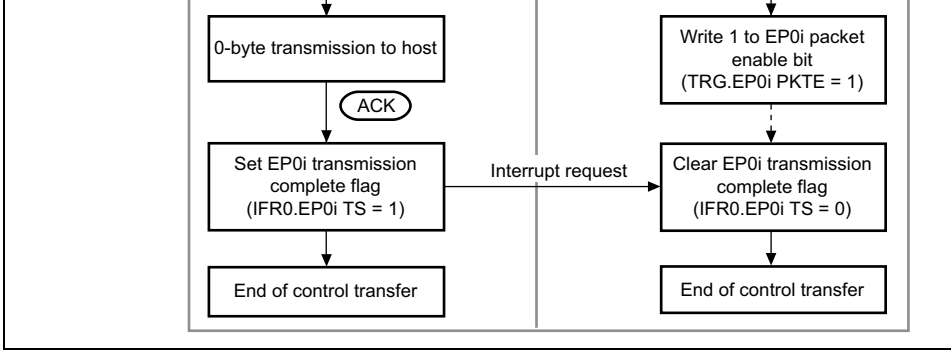


Figure 18.9 Status Stage (Control-Out) Operation

The control-out status stage starts with an IN token from the host. When an IN-token is
the start of the status stage, there is not yet any data in the EP0i FIFO, and so an EP0i t
request interrupt is generated. The application recognizes from this interrupt that the sta
has started. Next, in order to transmit 0-byte data to the host, 1 is written to the EP0i pa
bit but no data is written to the EP0i FIFO. As a result, the next IN token causes 0-byte
transmitted to the host, and control transfer ends.

After the application has finished all processing relating to the data stage, 1 should be v
the EP0i packet enable bit.

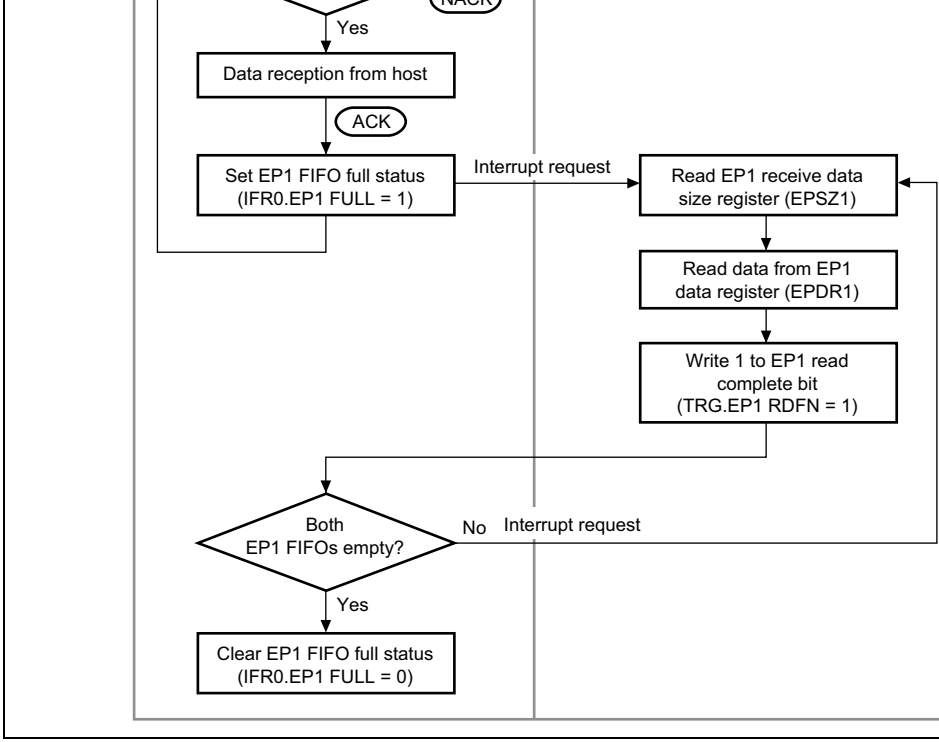


Figure 18.10 EP1 Bulk-Out Transfer Operation

EP1 has two 64-byte FIFOs, but the user can receive data and read receive data withou
aware of this dual-FIFO configuration.

When one FIFO is full after reception is completed, the EP1FULL bit in IFR0 is set. A
receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is
so the next packet can be received immediately. When both FIFOs are full, NACK is r
the host automatically. When reading of the receive data is completed following data r
is written to the EP1RDFN bit in TRG. This operation empties the FIFO that has just b
and makes it ready to receive the next packet.

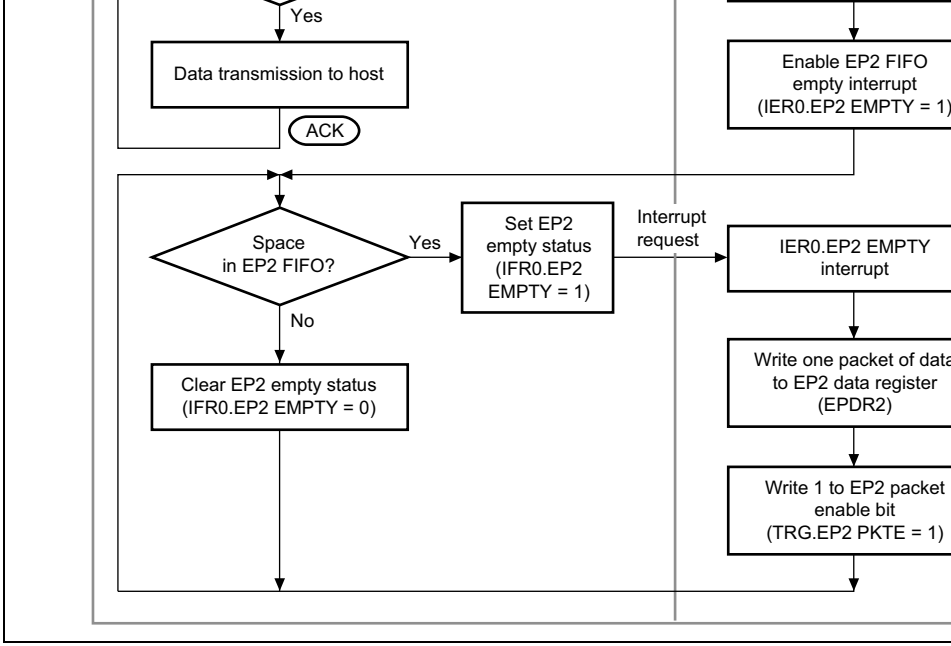
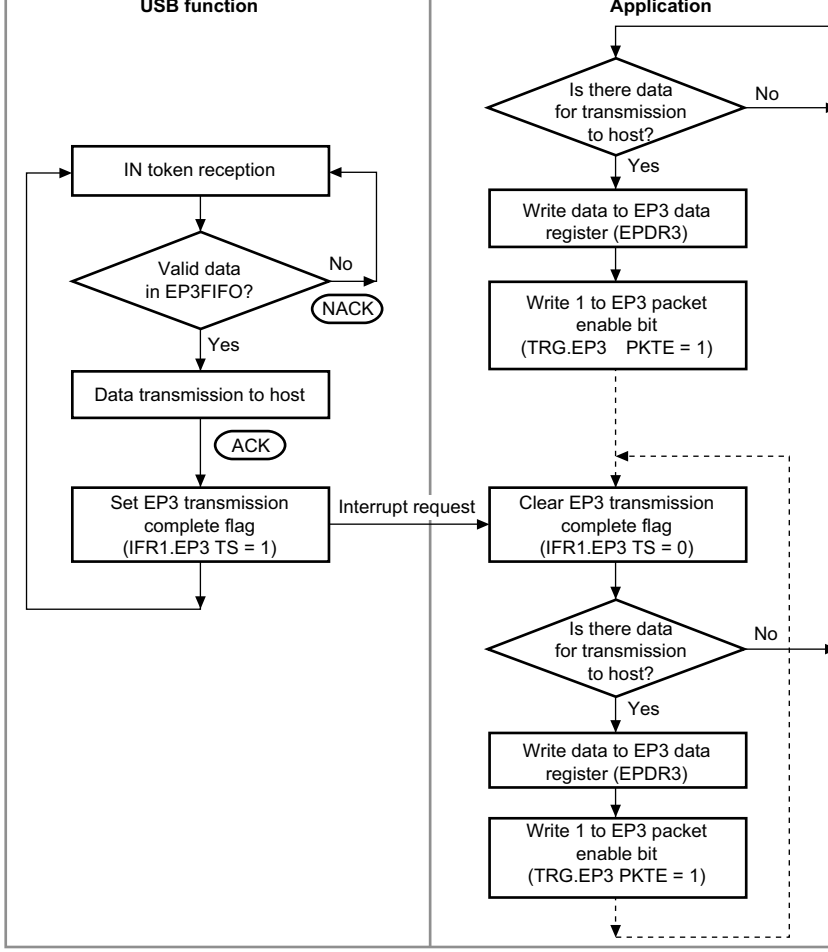


Figure 18.11 EP2 Bulk-In Transfer Operation

EP2 has two 64-byte FIFOs, but the user can transmit data and write transmit data with aware of this dual-FIFO configuration. However, one data write is performed for one F example, even if both FIFOs are empty, it is not possible to perform EP2PKTE at one t consecutively writing 128 bytes of data. EP2PKTE must be performed for each 64-byte

When performing bulk-in transfer, as there is no valid data in the FIFOs on reception o IN token, an EP2TR bit interrupt in IFR0 is requested. With this interrupt, 1 is written t EP2EMPTY bit in IER0, and the EP2 FIFO empty interrupt is enabled. At first, both E are empty, and so an EP2 FIFO empty interrupt is generated immediately.

The data to be transmitted is written to the data register using this interrupt. After the fi data write for one FIFO, the other FIFO is empty, and so the next transmit data can be w the other FIFO immediately. When both FIFOs are full, EP2 EMPTY is cleared to 0. If FIFO is empty, the EP2EMPTY bit in IFR0 is set to 1. When ACK is returned from the



Note: This flowchart shows just one example of interrupt transfer processing. Other possibilities includ
operation flow in which, if there is data to be transferred, the EP3 DE bit in the data status regist
referenced to confirm that the FIFO is empty, and then data is written to the FIFO.

Figure 18.12 Operation of EP3 Interrupt-In Transfer

Table 18.2 Command Decoding on Application Side

Decoding not Necessary on Application Side	Decoding Necessary on Applicati
Clear Feature	Get Descriptor
Get Configuration	Class/Vendor command
Get Interface	Set Descriptor
Get Status	Sync Frame
Set Address	
Set Configuration	
Set Feature	
Set Interface	

If decoding is not necessary on the application side, command decoding and data stage stage processing are performed automatically. No processing is necessary by the user. A is not generated in this case.

If decoding is necessary on the application side, this module stores the command in the FIFO. After reception is completed successfully, the IFR0/SETUP TS flag is set and an request is generated. In the interrupt routine, eight bytes of data must be read from the E register (EPDR0s) and decoded by firmware. The necessary data stage and status stage should then be carried out according to the result of the decoding operation.

- When a stall is performed automatically within the USB function module due to a specification violation

The USB function module has internal status bits that hold the status (stall or non-stall endpoint. When a transaction is sent from the host, the module references these interna and determines whether to return a stall to the host. These bits cannot be cleared by th application; they must be cleared with a Clear Feature command from the host.

18.6.2 Forcible Stall by Application

The application uses the EPSTL register to issue a stall request for the USB function m When the application wishes to stall a specific endpoint, it sets the corresponding bit i 1 in figure 18.13). The internal status bits are not changed at this time. When a transac from the host for the endpoint for which the EPSTL bit was set, the USB function mod references the internal status bit, and if this is not set, references the corresponding bit (1-2 in figure 18.13). If the corresponding bit in EPSTL is set, the USB function modu internal status bit and returns a stall handshake to the host (1-3 in figure 18.13). If the corresponding bit in EPSTL is not set, the internal status bit is not changed and the tra accepted.

Once an internal status bit is set, it remains set until cleared by a Clear Feature comma host, without regard to the EPSTL register. Even after a bit is cleared by the Clear Fea command (3-1 in figure 18.13), the USB function module continues to return a stall ha while the bit in EPSTL is set, since the internal status bit is set each time a transaction for the corresponding endpoint (1-2 in figure 18.13). To clear a stall, therefore, it is ne the corresponding bit in EPSTL to be cleared by the application, and also for the inter to be cleared with a Clear Feature command (2-1, 2-2, and 2-3 in figure 18.13).

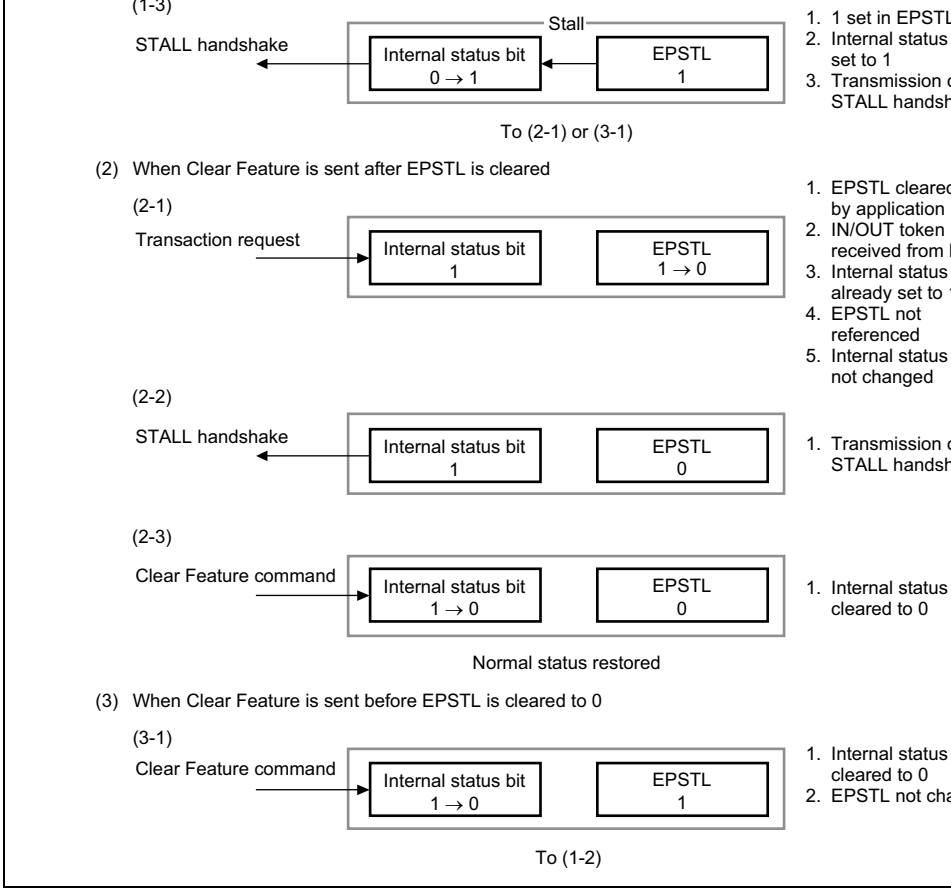


Figure 18.13 Forcible Stall by Application

handshake while the internal status bit is set, since the internal status bit is set even if a
is executed for the corresponding endpoint (2-1 and 2-2 in figure 18.14). To clear a sta
the internal status bit must be cleared with a Clear Feature command (3-1 in figure 18.
by the application, EPSTL should also be cleared (2-1 in figure 18.14).

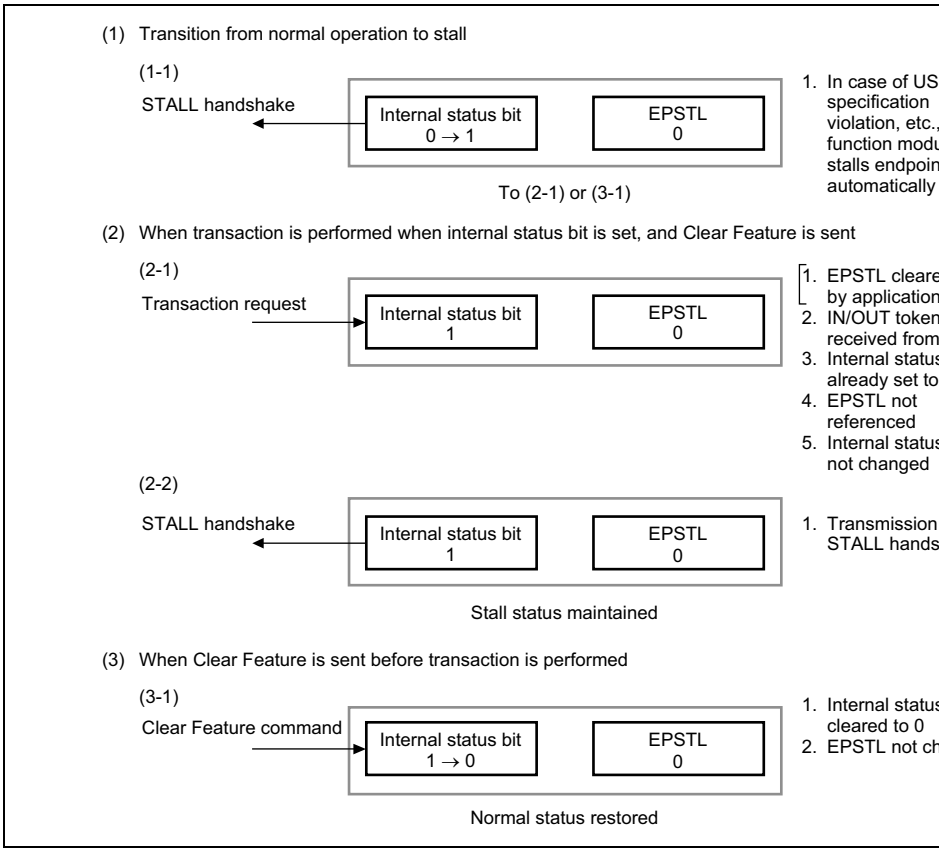


Figure 18.14 Automatic Stall by USB Function Module

If the DMA transfer is enabled by setting the EP1DMAE bit to 1 in the DMA transfer s register, zero-length data reception at endpoint 1 is ignored. When the DMA transfer is the RDFN bit for EP1 and PKTE bit for EP2 do not need to be set to 1 in TRG (note tha PKTE bit must be set to 1 when the transfer data is less than the maximum number of b When all the data received at EP1 is read, the FIFO automatically enters the EMPTY st the maximum number of bytes (64 bytes) are written to the EP2 FIFO, the FIFO autom enters the FULL state, and the data in the FIFO can be transmitted (see figures 18.15 an

18.7.2 DMA Transfer for Endpoint 1

When the data received at EP1 is transferred by the DMAC, the USB function module automatically performs the same processing as writing 1 to the RDFN bit in TRG if the selected FIFO becomes empty. Accordingly, in DMA transfer, do not write 1 to the RD TRG. If the user writes 1 to the RDFN bit in DMA transfer, correct operation cannot be guaranteed.

Figure 18.15 shows an example of receiving 150 bytes of data from the host. In this cas processing which is the same as writing 1 to the RDFN bit in TRG is automatically perf three times. This internal processing is performed when the currently selected data FIF empty. Accordingly, this processing is automatically performed both when 64-byte data and when data less than 64 bytes is sent.

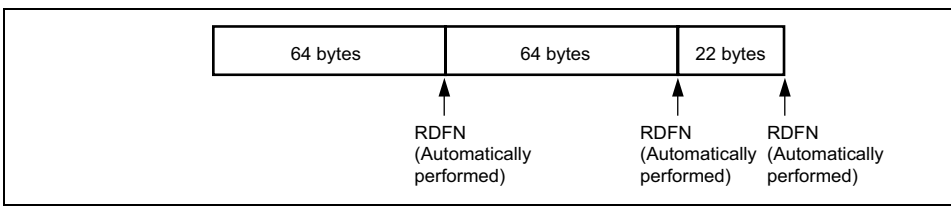


Figure 18.15 RDFN Bit Operation for EP1

Figure 18.16 shows an example for transmitting 150 bytes of data to the host. In this c
processing which is the same as writing 1 to the PKTE bit in TRG is automatically per
twice. This internal processing is performed when the currently selected data FIFO be
Accordingly, this processing is automatically performed only when 64-byte data is ser

When the last 22 bytes are sent, the internal processing for writing 1 to the PKTE bit i
performed, and the user must write 1 to the PKTE bit by software. In this case, the app
no more data to transfer but the USB function module continues to output DMA reque
as long as the FIFO has an empty space. When all data has been transferred, write 0 to
EP2DMAE bit in DMAR to cancel DMA requests for EP2.

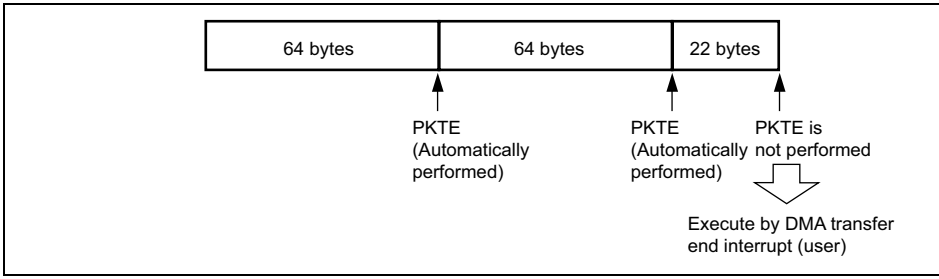
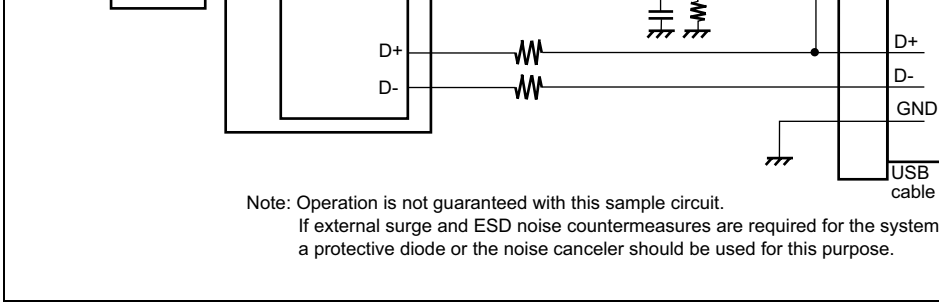


Figure 18.16 PKTE Bit Operation for EP2

(during high-priority processing or initialization processing, for example), D+ pull-
be controlled using a general output port. However, if a USB cable is already conne
host/hub and D+ pull-up is prohibited, D+ and D– will both go low (both of D+ and
down on the host/hub side) and the USB module will mistakenly identify this as rec
USB bus reset from the host. Therefore, the D+ pull-up control signal and VBUS pi
signal should be controlled using a general output port and the USB cable VBUS (A
circuit) as shown in figure 18.17. (The UDC core in this LSI maintains the powered
the VBUS pin is low, regardless of the D+/D– state.)

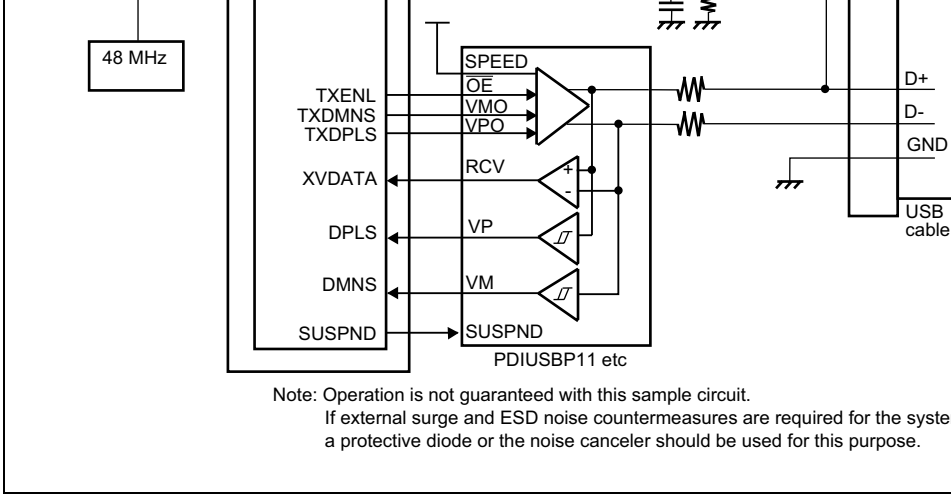
3. Detection of USB Cable Connection/Disconnection

   As USB states, etc., are managed by hardware in this module, a VBUS signal that r
   connection/disconnection is necessary. The power supply signal (VBUS) in the USI
   used for this purpose. However, if the cable is connected to the USB host/hub when
   function (system installing this LSI) power is off, a voltage (5 V) will be applied fr
   host/hub. Therefore, an IC (such as an HD74LV1G08A or 2G08A) that allows volta
   application when the system power is off should be connected externally.



Note: Operation is not guaranteed with this sample circuit.
If external surge and ESD noise countermeasures are required for the system
a protective diode or the noise canceler should be used for this purpose.

Figure 18.17 Example of USB Function Module External Circuitry (Internal Transceiver)



Note: Operation is not guaranteed with this sample circuit.
If external surge and ESD noise countermeasures are required for the syste
a protective diode or the noise canceler should be used for this purpose.

Figure 18.18 Example of USB Function Module External Circuitry (External Transceiver)

CPU is reading data after the data is received, the read from the CPU is forcibly te
Therefore, the data read after reception is started becomes invalid.

2. EPDR0s must always be read in 8-byte units. If the read is terminated at a midpoin
received at the next setup cannot be read correctly.

18.9.2 Clearing the FIFO

If a USB cable is disconnected during data transfer, the data being received or transmi
remain in the FIFO. When disconnecting a USB cable, clear the FIFO.

While a FIFO is transferring data, it must not be cleared.

18.9.3 Overreading and Overwriting the Data Registers

Note the following when reading or writing to a data register of this module.

(1) Receive data registers

The receive data registers must not be read exceeding the valid amount of receive data
number of bytes indicated by the receive data size register. Even for EPDR1 which ha
FIFO buffers, the maximum data to be read at one time is 64 bytes. After the data is re
current valid FIFO buffer, be sure to write 1 to EP1RDFN in TRG, which switches the
buffer, updates the receive data size to the new number of bytes, and enables the next
received.

(2) Transmit data registers

The transmit data registers must not be written to exceeding the maximum packet size
EPDR2 which has double FIFO buffers, write data within the maximum packet size at
After the data is written, write 1 to PKTE in TRG to switch the valid buffer and enable
data to be written. Data must not be continuously written to the two FIFO buffers.

18.9.6 Notes on TR Interrupt

Note the following when using the transfer request interrupt (TR interrupt) for IN trans
EP2, or EP3.

The TR interrupt flag is set if the FIFO for the target EP has no data when the IN token
from the USB host. However, at the timing shown in figure 18.19, multiple TR interrup
successively. Take appropriate measures against malfunction in such a case.

Note: This module determines whether to return NAK if the FIFO of the target EP ha
when receiving the IN token, but the TR interrupt flag is set only after a NAK h
is sent. If the next IN token is sent before PKTE of TRG is written to, the TR in
is set again.

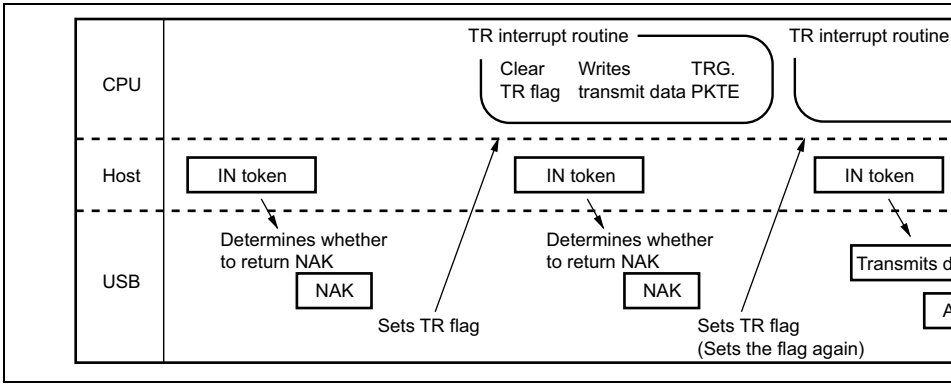


Figure 18.19 TR Interrupt Flag Set Timing

Table 19.1 Multiplex Pins

Port	Port Function (Related Module)	Other Functions (Related Modu
A	PTA7 input/output (port)/PINT7 input (INTC)	D23 input/output (BSC)
A	PTA6 input/output (port)/PINT6 input (INTC)	D22 input/output (BSC)
A	PTA5 input/output (port)/PINT5 input (INTC)	D21 input/output (BSC)
A	PTA4 input/output (port)/PINT4 input (INTC)	D20 input/output (BSC)
A	PTA3 input/output (port)/PINT3 input (INTC)	D19 input/output (BSC)
A	PTA2 input/output (port)/PINT2 input (INTC)	D18 input/output (BSC)
A	PTA1 input/output (port)/PINT1 input (INTC)	D17 input/output (BSC)
A	PTA0 input/output (port)/PINT0 input (INTC)	D16 input/output (BSC)
B	PTB7 input/output (port)/PINT15 input (INTC)	D31 input/output (BSC)
B	PTB6 input/output (port)/PINT14 input (INTC)	D30 input/output (BSC)
B	PTB5 input/output (port)/PINT13 input (INTC)	D29 input/output (BSC)
B	PTB4 input/output (port)/PINT12 input (INTC)	D28 input/output (BSC)
B	PTB3 input/output (port)/PINT11 input (INTC)	D27 input/output (BSC)
B	PTB2 input/output (port)/PINT10 input (INTC)	D26 input/output (BSC)
B	PTB1 input/output (port)/PINT9 input (INTC)	D25 input/output (BSC)
B	PTB0 input/output (port)/PINT8 input (INTC)	D24 input/output (BSC)
C	PTC7 input/output (port)	$\overline{\text{CS6A}}$ output (BSC)
C	PTC6 input/output (port)	$\overline{\text{CS5A}}$ output (BSC)
C	PTC5 input/output (port)	$\overline{\text{CS4}}$ output (BSC)
C	PTC4 input/output (port)	$\overline{\text{CS3}}$ output (BSC)
C	PTC3 input/output (port)	$\overline{\text{CS2}}$ output (BSC)
C	PTC2 input/output (port)	$\overline{\text{WE3}}$ output (BSC)/DQMUU outp $\overline{\text{AH}}$ output (BSC)
C	PTC1 input/output (port)	$\overline{\text{WE2}}$ output (BSC)/DQMUL outp
C	PTC0 input/output (port)	$\overline{\text{BS}}$ output (BSC)

D	PTD0 input/output (port)	$\overline{\text{RASL}}$ output (BSC)
E	PTE7 input/output (port)	—
E	PTE6 input/output (port)	TCLK input (TMU)
E	PTE5 input/output (port)	STATUS1 output (CPG)/$\overline{\text{CTS0}}$ input
E	PTE4 input/output (port)	STATUS0 output (CPG)/$\overline{\text{RTS0}}$ outpu
E	PTE3 input/output (port)	TEND0 output (DMAC)
E	PTE2 input/output (port)	IRQ5 input (INTC)
E	PTE1 input/output (port)	DACK1 output (DMAC)
E	PTE0 input/output (port)	DACK0 output (DMAC)
F	PTF7 input/output (port)	$\overline{\text{ASEMD0}}$ input
F	PTF6 input/output (port)	$\overline{\text{ASEBRKAK}}$ output
F	PTF5 input/output (port)	TDO output (UDI)
F	PTF4 input/output (port)	$\overline{\text{AUDSYNC}}$ output (AUD)
F	PTF3 input/output (port)	AUDATA3 output (AUD)/TO3 output
F	PTF2 input/output (port)	AUDATA2 output (AUD)/TO2 output
F	PTF1 input/output (port)	AUDATA1 output (AUD)/TO1 output
F	PTF0 input/output (port)	AUDATA0 output (AUD)/TO0 output
G	PTG7 input/output (port)	$\overline{\text{WAIT}}$ input (BSC)
G	PTG6 input/output (port)	$\overline{\text{BREQ}}$ input (BSC)
G	PTG5 input/output (port)	$\overline{\text{BACK}}$ output (BSC)
G	PTG4 input/output (port)	AUDCK output (AUD)
G	PTG3 input/output (port)	$\overline{\text{TRST}}$ input (UDI)
G	PTG2 input/output (port)	TMS input (UDI)
G	PTG1 input/output (port)	TCK input (UDI)
G	PTG0 input/output (port)	TDI input (UDI)

J	PTJ7 output (port)	NF[*1]
J	PTJ6 output (port)	NF[*1]
J	PTJ5 output (port)	NF[*1]
J	PTJ4 output (port)	NF[*1]
J	PTJ3 output (port)	NF[*1]
J	PTJ2 output (port)	NF[*1]
J	PTJ1 output (port)	NF[*1]
J	PTJ0 output (port)	NF[*1]
K	PTK7 input/output (port)	A25 output (BSC)
K	PTK6 input/output (port)	A24 output (BSC)
K	PTK5 input/output (port)	A23 output (BSC)
K	PTK4 input/output (port)	A22 output (BSC)
K	PTK3 input/output (port)	A21 output (BSC)
K	PTK2 input/output (port)	A20 output (BSC)
K	PTK1 input/output (port)	A19 output (BSC)
K	PTK0 input/output (port)	A0 output (BSC)
L	PTL3 input (port)	AN3 input (ADC)
L	PTL2 input (port)	AN2 input (ADC)
L	PTL1 input (port)	AN1 input (ADC)
L	PTL0 input (port)	AN0 input (ADC)
M	PTM6 input/output (port)	VBUS input (USB)
M	PTM4 input (port)	NF[*1]
M	PTM3 input/output (port)	—
M	PTM2 input/output (port)	—
M	PTM1 input/output (port)	—
M	PTM0 input/output (port)	—

N	PTN0 input/output (port)	SUSPND output (USB)
SCPT	SCPT5 input/output (port)	$\overline{CTS2}$ input (SCIF2)
SCPT	SCPT4 input/output (port)	$\overline{RTS2}$ output (SCIF2)
SCPT	SCPT3 input/output (port)	SCK2 input/output (SCIF2)
SCPT	SCPT2 input (port)*2	RXD2 input (SCIF2)
SCPT	SCPT2 output (port)*2	TXD2 output (SCIF2)
SCPT	SCPT1 input/output (port)	SCK0 input/output (SCIF0)
SCPT	SCPT0 input (port)*2	RXD0 input (SCIF0)/IrRX input (Ir
SCPT	SCPT0 output (port)*2	TXD0 output (SCIF0)/IrTX output

Notes: 1. The initial functions of NF (No Function) pins are not assigned after power-o
Specifies the functions with Pin Function Controller (PFC).
PTD5 and PTM4 must be pulled up.
PTJ[7:0] must be open except for the pins specified as port output pins.
The values of PTJ6, PTJ1, and PTJ0 differ during power-on reset and after t
on reset state is released. They conform to the port J data register value afte
switched to port status by the pin function controller (PFC).

		After Power-On Reset Releas	
	During Power-On Reset	**PTD5/NF = 1**	**PTD5/NF**
PTJ6/NF	1	0	1
PTJ1/NF	1	1	0
PTJ0/NF	1	0	1

2. SCPT0 and SCPT2 each have two separate pins for input and output, howe
common data register is accessed.

In table 19.1, pin functions in the shaded column can be used immediately after a powe

- Port D control register (PDCR)
- Port E control register (PECR)
- Port E control register 2 (PECR2)
- Port F control register (PFCR)
- Port F control register 2 (PFCR2)
- Port G control register (PGCR)
- Port H control register (PHCR)
- Port J control register (PJCR)
- Port K control register (PKCR)
- Port L control register (PLCR)
- Port M control register (PMCR)
- Port N control register (PNCR)
- Port N control register 2 (PNCR2)
- Port SC control register (SCPCR)

				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
13	PA6MD1	0	R/W	PTA6 Mode
12	PA6MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
11	PA5MD1	0	R/W	PTA5 Mode
10	PA5MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
9	PA4MD1	0	R/W	PTA4 Mode
8	PA4MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
7	PA3MD1	0	R/W	PTA3 Mode
6	PA3MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)

2	PA1MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
1	PA0MD1	0	R/W	PTA0 Mode
0	PA0MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

19.2.2 Port B Control Register (PBCR)

PBCR is a 16-bit readable/writable register that selects the pin function and input pull control.

Bit	Bit Name	Initial Value	R/W	Description
15	PB7MD1	0	R/W	PTB7 Mode
14	PB7MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
13	PB6MD1	0	R/W	PTB6 Mode
12	PB6MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

8	PB4MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
7	PB3MD1	0	R/W	PTB3 Mode
6	PB3MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
5	PB2MD1	0	R/W	PTB2 Mode
4	PB2MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
3	PB1MD1	0	R/W	PTB1 Mode
2	PB1MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
1	PB0MD1	0	R/W	PTB0 Mode
0	PB0MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
13	PC6MD1	1	R/W	PTC6 Mode
12	PC6MD0	1	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
11	PC5MD1	0	R/W	PTC5 Mode
10	PC5MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
9	PC4MD1	0	R/W	PTC4 Mode
8	PC4MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
7	PC3MD1	0	R/W	PTC3 Mode
6	PC3MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)

2	PC1MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
1	PC0MD1	0	R/W	PTC0 Mode
0	PC0MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
13	PD6MD1	1	R/W	PTD6 Mode
12	PD6MD0	1	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
11	PD5MD1	0	R/W	PTD5 Mode
10	PD5MD0	0	R/W	00: NF
				01: Setting prohibited
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
9	PD4MD1	0	R/W	PTD4 Mode
8	PD4MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
7	PD3MD1	1	R/W	PTD3 Mode
6	PD3MD0	1	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)

2	PD1MD0	1	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
1	PD0MD1	0	R/W	PTD0 Mode
0	PD0MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
13	PE6MD1	1	R/W	PTE6 Mode
12	PE6MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
11	PE5MD1	0	R/W	PTE5 Mode
10	PE5MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
9	PE4MD1	0	R/W	PTE4 Mode
8	PE4MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
7	PE3MD1	1	R/W	PTE3 Mode
6	PE3MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
5	PE2MD1	1	R/W	PTE2 Mode
4	PE2MD0	1	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)

0	PE0MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

19.2.6 Port E Control Register 2 (PECR2)

PECR2 is an 8-bit readable/writable register that selects the pin function.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	0	R	Reserved These bits are always read as 0. The write valu always be 0.
5	PE5MD2	0	R/W	PE5 Mode 2 This bit is valid when the PE5MD[1:0] bits in PE to B'00 (other functions). 0: STATUS1 (CPG) 1: $\overline{\text{CTS0}}$ (SCIF0)
4	PE4MD2	0	R/W	PE4 Mode 2 This bit is valid when the PE4MD[1:0] bits in PE to B'00 (other functions). 0: STATUS0 (CPG) 1: $\overline{\text{RTS0}}$ (SCIF0)
3 to 0	—	0	R	Reserved These bits are always read as 0. The write valu always be 0.

				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
13	PF6MD1	1[*1]	R/W	PTF6 Mode
12	PF6MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
11	PF5MD1	0	R/W	PTF5 Mode
10	PF5MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
9	PF4MD1	1[*1]	R/W	PTF4 Mode
8	PF4MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
7	PF3MD1	1[*1]	R/W	PTF3 Mode
6	PF3MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
5	PF2MD1	1[*1]	R/W	PTF2 Mode
4	PF2MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)

0	PF0MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

Notes: 1. Indicates the initial value when $\overline{\text{ASEMD0}}$ = 1. When $\overline{\text{ASEMD0}}$ = 0, the releva becomes 0, and other functions is selected.
2. Pull-up MOS on.

19.2.8 Port F Control Register 2 (PFCR2)

PFCR2 is an 8-bit readable/writable register that selects the pin function.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	0	R	Reserved These bits are always read as 0. The write valu always be 0.
3	PF3MD2	0	R/W	PTF3 Mode 2 This bit is valid when the PF3MD[1:0] bits in PF to B'00 (other functions). 0: AUDATA3 (AUD) 1: TO3 (TPU)
2	PF2MD2	0	R/W	PTF2 Mode 2 This bit is valid when the PF2MD[1:0] bits in PF to B'00 (other functions). 0: AUDATA2 (AUD) 1: TO2 (TPU)

				This bit is valid when the PF0MD[1:0] bits in PF B'00 (other functions). 0: AUDATA0 (AUD) 1: TO0 (TPU)

19.2.9 Port G Control Register (PGCR)

PGCR is a 16-bit readable/writable register that selects the pin function and input pull control.

Bit	Bit Name	Initial Value	R/W	Description
15	PG7MD1	0	R/W	PTG7 Mode
14	PG7MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
13	PG6MD1	0	R/W	PTG6 Mode
12	PG6MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
11	PG5MD1	0	R/W	PTG5 Mode
10	PG5MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

6	PG3MD0	0	R/W	00: Other functions*2 (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
5	PG2MD1	0	R/W	PTG2 Mode
4	PG2MD0	0	R/W	00: Other functions*2 (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
3	PG1MD1	0	R/W	PTG1 Mode
2	PG1MD0	0	R/W	00: Other functions*2 (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
1	PG0MD1	0	R/W	PTG0 Mode
0	PG0MD0	0	R/W	00: Other functions*2 (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

Notes: 1. Indicates the initial value when $\overline{\text{ASEMD0}}$ = 1. When $\overline{\text{ASEMD0}}$ = 0, the releva becomes 0, and other functions is selected.

2. Pull-up MOS on.

				always be 0.
13	PH6MD1	1	R/W	PTH6 Mode
12	PH6MD0	1	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
11	PH5MD1	1	R/W	PTH5 Mode
10	PH5MD0	1	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
9	PH4MD1	0	R/W	PTH4 Mode
8	PH4MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
7	PH3MD1	0	R/W	PTH3 Mode
6	PH3MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
5	PH2MD1	0	R/W	PTH2 Mode
4	PH2MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

0	PH0MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

19.2.11 Port J Control Register (PJCR)

PJCR is a 16-bit readable/writable register that selects the pin function.

Bit	Bit Name	Initial Value	R/W	Description
15	PJ7MD1	0	R/W	PTJ7 Mode
14	PJ7MD0	0	R/W	00: NF 01: Port output 10: Setting prohibited 11: Setting prohibited
13	PJ6MD1	0	R/W	PTJ6 Mode
12	PJ6MD0	0	R/W	00: NF 01: Port output 10: Setting prohibited 11: Setting prohibited
11	PJ5MD1	0	R/W	PTJ5 Mode
10	PJ5MD0	0	R/W	00: NF 01: Port output 10: Setting prohibited 11: Setting prohibited

6	PJ3MD0	0	R/W	00: NF 01: Port output 10: Setting prohibited 11: Setting prohibited
5	PJ2MD1	0	R/W	PTJ2 Mode
4	PJ2MD0	0	R/W	00: NF 01: Port output 10: Setting prohibited 11: Setting prohibited
3	PJ1MD1	0	R/W	PTJ1 Mode
2	PJ1MD0	0	R/W	00: NF 01: Port output 10: Setting prohibited 11: Setting prohibited
1	PJ0MD1	0	R/W	PTJ0 Mode
0	PJ0MD0	0	R/W	00: NF 01: Port output 10: Setting prohibited 11: Setting prohibited

				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
13	PK6MD1	0	R/W	PTK6 Mode
12	PK6MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
11	PK5MD1	0	R/W	PTK5 Mode
10	PK5MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
9	PK4MD1	0	R/W	PTK4 Mode
8	PK4MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
7	PK3MD1	0	R/W	PTK3 Mode
6	PK3MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
5	PK2MD1	0	R/W	PTK2 Mode
4	PK2MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)

0	PK0MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

7	PL3MD1	0	R/W	PTL3 Mode
6	PL3MD0	0	R/W	00: Other functions (see table 19.1)
				01: Setting prohibited
				10: Setting prohibited
				11: Port input (pull-up MOS: Off)
5	PL2MD1	0	R/W	PTL2 Mode
4	PL2MD0	0	R/W	00: Other functions (see table 19.1)
				01: Setting prohibited
				10: Setting prohibited
				11: Port input (pull-up MOS: Off)
3	PL1MD1	0	R/W	PTL1 Mode
2	PL1MD0	0	R/W	00: Other functions (see table 19.1)
				01: Setting prohibited
				10: Setting prohibited
				11: Port input (pull-up MOS: Off)
1	PL0MD1	0	R/W	PTL0 Mode
0	PL0MD0	0	R/W	00: Other functions (see table 19.1)
				01: Setting prohibited
				10: Setting prohibited
				11: Port input (pull-up MOS: Off)

				always be 0.
13	PM6MD1	1	R/W	PTM6 Mode
12	PM6MD0	0	R/W	00: Other functions (see table 19.1)
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
11, 10	—	0	R	Reserved
				These bits are always read as 0. The write v always be 0.
9	PM4MD1	0	R/W	PTM4 Mode
8	PM4MD0	0	R/W	00: NF
				01: Setting prohibited
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
7	PM3MD1	1	R/W	PTM3 Mode
6	PM3MD0	0	R/W	00: Setting prohibited
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)
5	PM2MD1	1	R/W	PTM2 Mode
4	PM2MD0	0	R/W	00: Setting prohibited
				01: Port output
				10: Port input (pull-up MOS: On)
				11: Port input (pull-up MOS: Off)

0	PM0MD0	0	R/W	00: Setting prohibited 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

19.2.15 Port N Control Register (PNCR)

PNCR is a 16-bit readable/writable register that selects the pin function and input pull-u control.

Bit	Bit Name	Initial Value	R/W	Description
15	PN7MD1	1	R/W	PTN7 Mode
14	PN7MD0	0	R/W	00: Setting prohibited 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
13	PN6MD1	1	R/W	PTN6 Mode
12	PN6MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
11	PN5MD1	1	R/W	PTN5 Mode
10	PN5MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

6	PN3MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
5	PN2MD1	1	R/W	PTN2 Mode
4	PN2MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
3	PN1MD1	1	R/W	PTN1 Mode
2	PN1MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
1	PN0MD1	1	R/W	PTN0 Mode
0	PN0MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

6	PN6MD2	0	R/W	PTN6 Mode 2 This bit is valid when the PN6MD[1:0] bits in PN set to B'00 (other functions). 0: Setting prohibited 1: DPLS (USB)
5	PN5MD2	0	R/W	PTN5 Mode 2 This bit is valid when the PN5MD[1:0] bits in PN set to B'00 (other functions). 0: Setting prohibited 1: DMNS (USB)
4	PN4MD2	0	R/W	PTN4 Mode 2 This bit is valid when the PN4MD[1:0] bits in PN set to B'00 (other functions). 0: Setting prohibited 1: TXDPLS (USB)
3	PN3MD2	0	R/W	PTN3 Mode 2 This bit is valid when the PN3MD[1:0] bits in PN set to B'00 (other functions). 0: Setting prohibited 1: TXDMNS (USB)
2	PN2MD2	0	R/W	PTN2 Mode 2 This bit is valid when the PN2MD[1:0] bits in PN set to B'00 (other functions). 0: Setting prohibited 1: XVDATA (USB)

				This bit is valid when the PN0MD[1:0] bits in P set to B'00 (other functions). 0: Setting prohibited 1: SUSPND (USB)

19.2.17 Port SC Control Register (SCPCR)

SCPCR is a 16-bit readable/writable register that selects the pin function and input pu control. The settings of SCPCR become valid only when transmission/reception opera disabled by the settings of SCSCR in the on-chip serial communication interface (SCI

When the TE bit in SCSCR_0 or SCSCR_2 of the SCIF is set to 1, the output status o functions: TxD0 or TxD2" has priority for the setting of SCPCR.

Similarly, when the RE bit in SCSCR_0 or SCSCR_2 is set to 1, the input status of "o functions: RxD0 or RxD2" has priority for the setting of SCPCR.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	0	R	Reserved These bits are always read as 0. The write always be 0.
11	SCP5MD1	0	R/W	SCPT5 Mode
10	SCP5MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)

6	SCP3MD0	0	R/W	00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off)
5	SCP2MD1	0	R/W	SCPT2 Mode
4	SCP2MD0	0	R/W	These bits select pin function and input pull-up control. When TE = 0 and RE = 0 in SCSCR_2, operat follows: 00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off) When TE = 1 in SCSCR_2, the SCPT2/TxD2 p as TxD2. When RE = 1 in SCSCR_2, the SCPT2/RxD2 functions as RxD2. Note: Since two pins (TxD2 and RxD2) are us one bit (SCPT2), there is no combinatio simultaneous input/output of SCPT2. When port input is set (when bit SCP2MD1 is TxD2 pin enters an output state when the TE b SCSCR_2 is set to 1, whereas it enters high-i state when the TE bit is cleared to 0.

				11: Port input (pull-up MOS: Off)
1	SCP0MD1	0	R/W	SCPT0 Mode
0	SCP0MD0	0	R/W	These bits select pin function and input pull-up control. When TE = 0 and RE = 0 in SCSCR_0, operat follows: 00: Other functions (see table 19.1) 01: Port output 10: Port input (pull-up MOS: On) 11: Port input (pull-up MOS: Off) When TE = 1 in SCSCR_0, the SCPT0/TxD0 p functions as TxD0. When RE = 1 in SCSCR_0, the SCPT0/RxD0 functions as RxD0. Note: Since two pins (TxD0 and RxD0) are us access one bit (SCPT0), there is no com simultaneous input/output of SCPT0. When port input is set (when bit SCP0MD1 is s the TxD0 pin enters an output state when the T SCSCR_0 is set to 1, whereas it enters high-im state when the TE bit is cleared to 0.

Port A is an 8-bit input/output port with the pin configuration shown in figure 20.1. Ea
an input pull-up MOS, which is controlled by the port A control register (PACR) in th

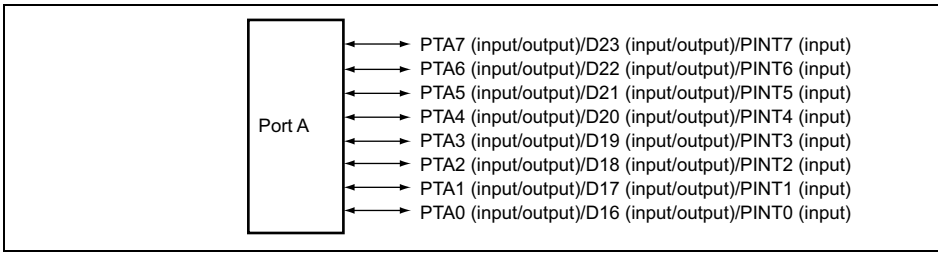


Figure 20.1 Port A

20.1.1 Register Description

Port A has the following register. For details on the register address and access size, s
24, List of Registers.

- Port A data register (PADR)

7 to 0	PA7DT to PA0DT	0	R/W	Table 20.1 shows the function of PADR.

Table 20.1 Port A Data Register (PADR) Read/Write Operations

PACR State				
PAnMD1	**PAnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PADR value	Data can be written to PADR but pin state.
	1	Output	PADR value	Written data is output from the pin
1	0	Input (Pull-up MOS on)	Pin state	Data can be written to PADR but pin state.
	1	Input (Pull-up MOS off)	Pin state	Data can be written to PADR but pin state.

Note: n = 0 to 7

20.2 Port B

Port B is an 8-bit input/output port with the pin configuration shown in figure 20.2. Eac an input pull-up MOS, which is controlled by the port B control register (PBCR) in the

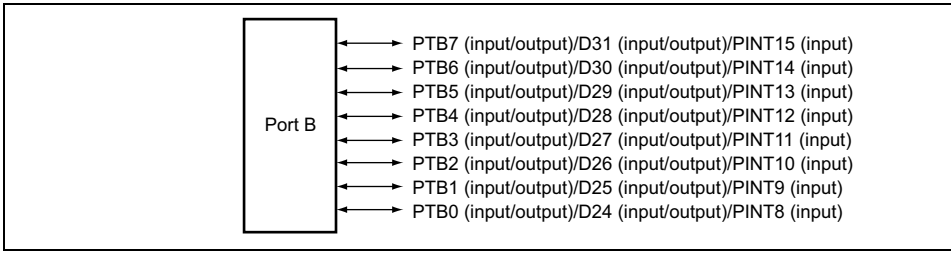


Figure 20.2 Port B

PBDR is an 8-bit readable/writable register that stores data for pins PTB7 to PTB0. Bi
PB0DT correspond to pins PTB7 to PTB0. When the pin function is general output po
is read the value of the corresponding PBDR bit is returned directly. When the functio
input port, if the port is read, the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	PB7DT to PB0DT	0	R/W	Table 20.2 shows the function of PBDR.

Table 20.2 Port B Data Register (PBDR) Read/Write Operations

PBCR State				
PBnMD1	**PBnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PBDR value	Data can be written to PBDR bu pin state.
	1	Output	PBDR value	Written data is output from the p
1	0	Input (Pull-up MOS on)	Pin state	Data can be written to PBDR bu pin state.
	1	Input (Pull-up MOS off)	Pin state	Data can be written to PBDR bu pin state.

Note: n = 0 to 7

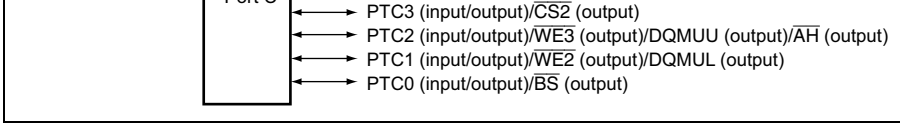


Figure 20.3 Port C

20.3.1 Register Description

Port C has the following register. For details on the register address and access size, see 24, List of Registers.

- Port C data register (PCDR)

20.3.2 Port C Data Register (PCDR)

PCDR is an 8-bit readable/writable register that stores data for pins PTC7 to PTC0. Bit PC0DT correspond to pins PTC7 to PTC0. When the pin function is general output por is read, the value of the corresponding PCDR bit is returned directly. When the functio input port, if the port is read, the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	PC7DT to PC0DT	0	R/W	Table 20.3 shows the function of PCDR.

1	Input (Pull-up MOS off)	Pin state	Data can be written to PCDR bu pin state.

Note: n = 0 to 7

20.4 Port D

Port D is an 8-bit input/output port with the pin configuration shown in figure 20.4. Ea an input pull-up MOS, which is controlled by the port D control register (PDCR) in th

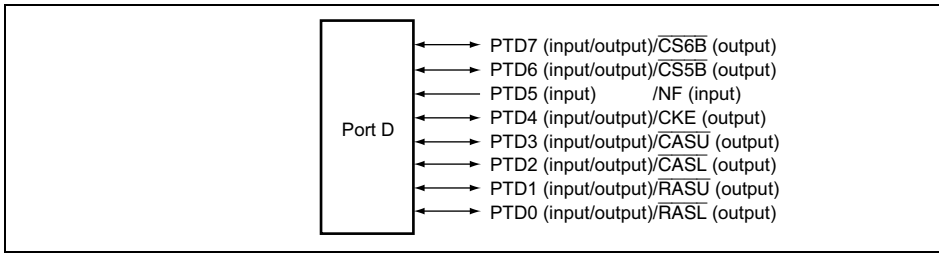


Figure 20.4 Port D

20.4.1 Register Description

Port D has the following register. For details on the register address and access size, s 24, List of Registers.

- Port D data register (PDDR)

20.4.2 Port D Data Register (PDDR)

PDDR is an 8-bit readable/writable register that stores data for pins PTD7 to PTD0. B to PD0DT correspond to pins PTD7 to PTD0. When the pin function is general output port is read, the value of the corresponding PDDR bit is returned directly. When the fu general input port, if the port is read, the corresponding pin level is read.

PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function	PDDR value	Data can be written to PDDR but on pin state.
	1	Output	PDDR value	Written data is output from the pir
1	0	Input (Pull-up MOS on)	Pin state	Data can be written to PDDR but on pin state.
	1	Input (Pull-up MOS off)	Pin state	Data can be written to PDDR but on pin state.

Note: n = 0 to 4, 6, and 7

PDCR State

PD5MD1	PD5MD0	Pin State	Read	Write
0	0	NF	PDDR value	Data can be written to PDDR but on pin state.
	1	Setting Prohibited	—	—
1	0	Input (Pull-up MOS on)	Pin state	Data can be written to PDDR but on pin state.
	1	Input (Pull-up MOS off)	Pin state	Data can be written to PDDR but on pin state.

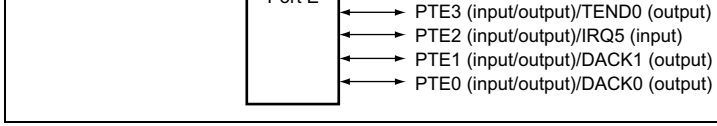


Figure 20.5 Port E

20.5.1 Register Description

Port E has the following register. For details on the register address and access size, se
24, List of Registers.

- Port E data register (PEDR)

20.5.2 Port E Data Register (PEDR)

PEDR is an 8-bit readable/writable register that stores data for pins PTE7 to PTE0. Bi
PE0DT correspond to pins PTE7 to PTE0. When the pin function is general output po
is read, the value of the corresponding PEDR bit is returned directly. When the functio
input port, if the port is read the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	PE7DT to PE0DT	0	R/W	Table 20.5 shows the function of PEDR.

1	Input (Pull-up MOS off)	Pin state	Data can be written to PEDR but pin state.

Note: n = 0 to 7

20.6 Port F

Port F is an 8-bit input port with the pin configuration shown in figure 20.6. Each pin h pull-up MOS, which is controlled by the port F control register (PFCR) in the PFC.

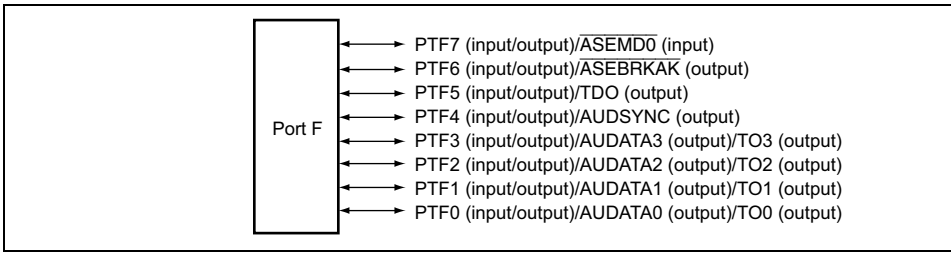


Figure 20.6 Port F

20.6.1 Register Description

Port F has the following register. For details on the register address and access size, see List of Registers.

- Port F data register (PFDR)

20.6.2 Port F Data Register (PFDR)

PFDR is an 8-bit readable/writable register that stores data for pins PTF7 to PTF0. Bits PF0DT correspond to pins PTF7 to PTF0. When the pin function is general output port is read, the value of the corresponding PFDR bit is returned directly. When the function input port, if the port is read the corresponding pin level is read.

PFnMD1	PFnMD0	Pin State	Read	Write
0	0	Other function	PFDR value	Data can be written to PFDR but pin state.
	1	Output	PFDR value	Written data is output from the p
1	0	Input (Pull-up MOS on)	Pin state	Data can be written to PFDR but pin state.
	1	Input (Pull-up MOS off)	Pin state	Data can be written to PFDR but pin state.

Note: n = 0 to 7

20.7 Port G

Port G is an 8-bit input port with the pin configuration shown in figure 20.7. Each pin pull-up MOS, which is controlled by the port G control register (PGCR) in the PFC.

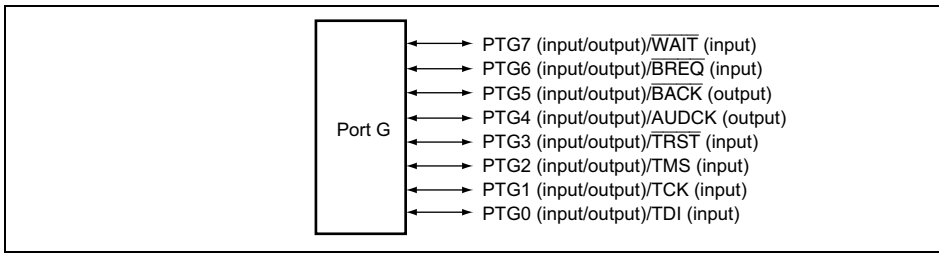


Figure 20.7 Port G

20.7.1 Register Description

Port G has the following register. For details on the register address and access size, s 24, List of Registers.

- Port G data register (PGDR)

7 to 0	PG7DT to PG0DT	0	R/W	Table 20.7 shows the function of PGDR.

Table 20.7 Port G Data Register (PGDR) Read/Write Operations

PGCR State				
PGnMD1	**PGnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PGDR value	Data can be written to PGDR but pin state.
	1	Output	PGDR value	Written data is output from the pin
1	0	Input (Pull-up MOS on)	Pin state	Data can be written to PGDR but pin state.
	1	Input (Pull-up MOS off)	Pin state	Data can be written to PGDR but pin state.

Note: n = 0 to 7

20.8 Port H

Port H is a 7-bit input/output port with the pin configuration shown in figure 20.8. Each input pull-up MOS, which is controlled by the port H control register (PHCR) in the PF

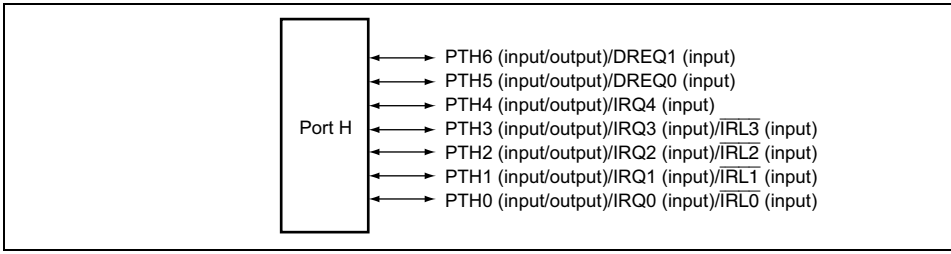


Figure 20.8 Port H

PHDR is an 8-bit readable/writable register that stores data for pins PTH6 to PTH0. B
to PH0DT correspond to pins PTH6 to PTH0. When the pin function is general output
port is read, the value of the corresponding PHDR bit is returned directly. When the fu
general input port, if the port is read, the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
6 to 0	PH6DT to PH0DT	0	R/W	Table 20.8 shows the function of PHDR.

Table 20.8 Port H Data Register (PHDR) Read/Write Operations

PHCR State				
PHnMD1	**PHnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PHDR value	Data can be written to PHDR bu on pin state.
	1	Output	PHDR value	Written data is output from the p
1	0	Input (Pull-up MOS on)	Pin state	Data can be written to PHDR bu on pin state.
	1	Input (Pull-up MOS off)	Pin state	Data can be written to PHDR bu on pin state.

Note: n = 0 to 6

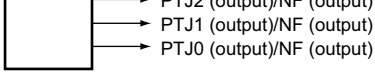


Figure 20.9 Port J

20.9.1 Register Description

Port J has the following register. For details on the register address and access size, see List of Registers.

- Port J data register (PJDR)

20.9.2 Port J Data Register (PJDR)

PJDR is an 8-bit readable/writable register that stores data for pins PTJ7 to PTJ0. Bits PJ0DT correspond to pins PTJ7 to PTJ0. When the pin function is general output port, is read, the value of the corresponding PJDR bit is returned directly.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	PJ7DT to PJ0DT	0	R/W	Table 20.9 shows the function of PJDR.

1	Setting Prohibited	—	—

Note: n = 0 to 7

20.10 Port K

Port K is an 8-bit input/output port with the pin configuration shown in figure 20.10. F
an input pull-up MOS, which is controlled by the port K control register (PKCR) in th

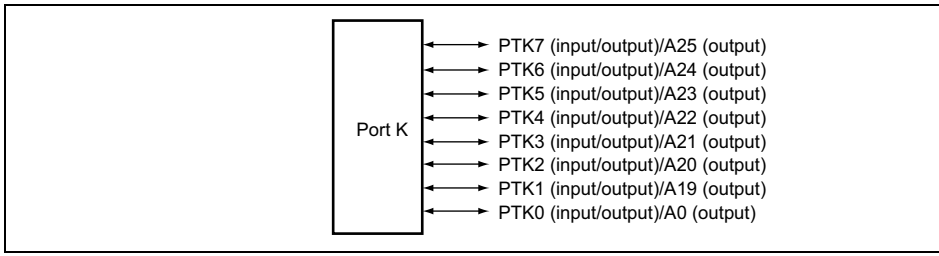


Figure 20.10 Port K

20.10.1 Register Description

Port K has the following register. For details on the register address and access size, s
24, List of Registers.

- Port K data register (PKDR)

20.10.2 Port K Data Register (PKDR)

PKDR is an 8-bit readable/writable register that stores data for pins PTK7 to PTK0. B
to PK0DT correspond to pins PTK7 to PTK0. When the pin function is general output
port is read, the value of the corresponding PKDR bit is returned directly. When the fu
general input port, if the port is read, the corresponding pin level is read.

PKnMD1	PKnMD0	Pin State	Read	Write
0	0	Other function	PKDR value	Data can be written to PKDR but on pin state.
	1	Output	PKDR value	Written data is output from the pin
1	0	Input (Pull-up MOS on)	Pin state	Data can be written to PKDR but on pin state.
	1	Input (Pull-up MOS off)	Pin state	Data can be written to PKDR but on pin state.

Note: n = 0 to 7

20.11 Port L

Port L is a 4-bit input port with the pin configuration shown in figure 20.11.

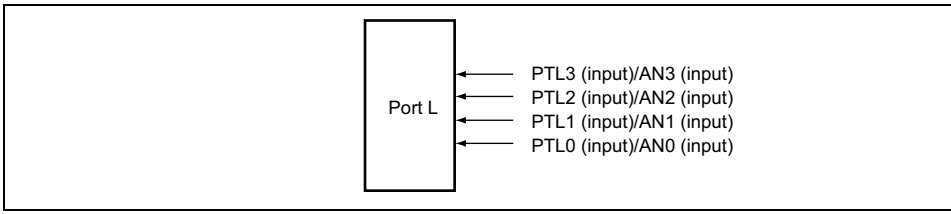


Figure 20.11 Port L

20.11.1 Register Description

Port L has the following register. For details on the register address and access size, see 24, List of Registers.

- Port L data register (PLDR)

3 to 0	PL3DT to PL0DT	0	R	Table 20.11 shows the function of PLDR.

Table 20.11 Port L Data Register (PLDR) Read/Write Operation

PLCR State				
PLnMD1	**PLnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	Read as 0	Invalid (no effect on pin state)
	1	Setting prohibited	—	—
1	0	Setting prohibited	—	—
	1	Input (Pull-up MOS off)	Pin state	Invalid (no effect on pin state)

Note: n = 0 to 3

20.12 Port M

Port M is a 6-bit input/output port with the pin configuration shown in figure 20.12. E
an input pull-up MOS, which is controlled by the port M control register (PMCR) in t

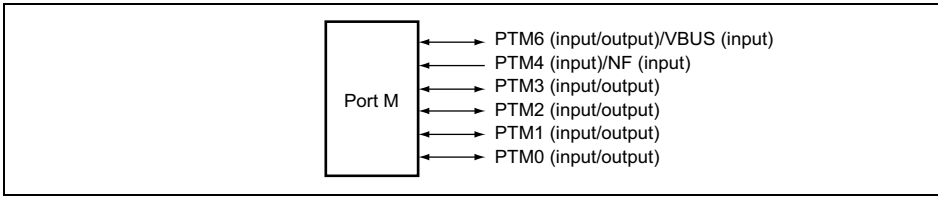


Figure 20.12 Port M

PMDR is an 8-bit readable/writable register that stores data for pins PTM6 and PTM4 t
Bits PM6DT and PM4DT to PM0DT correspond to pins PTM6 and PTM4 to PTM0. W
function is general output port, if the port is read, the value of the corresponding PMDR
returned directly. When the function is general input port, if the port is read, the corresp
level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
6	PM6DT	0	R/W	Table 20.12 shows the function of PMDR.
5	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
4 to 0	PM4DT to PM0DT	0	R/W	Table 20.12 shows the function of PMDR.

Table 20.12 Port M Data Register (PMDR) Read/Write Operations

PMCR State				
PMnMD1	**PMnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PMDR value	Data can be written to PMDR but on pin state.
	1	Output	PMDR value	Written data is output from the pir
1	0	Input (Pull-up MOS on)	Pin state	Data can be written to PMDR but on pin state.
	1	Input (Pull-up MOS off)	Pin state	Data can be written to PMDR but on pin state.

Note: n = 0 to 3 and 6

	1	Input (Pull-up MOS off)	Pin state	Data can be written to PMDR but on pin state.

20.13 Port N

Port N is an 8-bit input/output port with the pin configuration shown in figure 20.13. E an input pull-up MOS, which is controlled by the port N control register (PNCR) in th

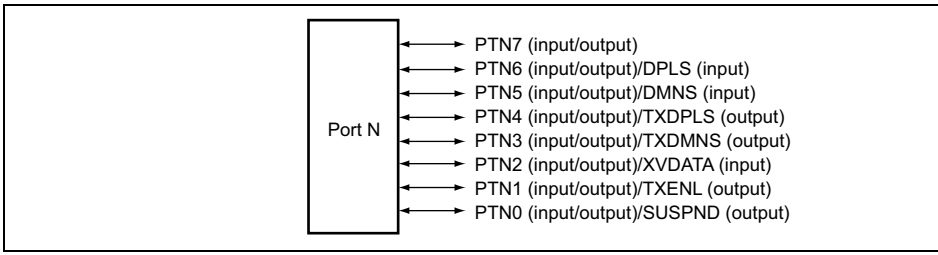


Figure 20.13 Port N

20.13.1 Register Description

Port N has the following register. For details on the register address and access size, s 24, List of Registers.

- Port N data register (PNDR)

20.13.2 Port N Data Register (PNDR)

PNDR is an 8-bit readable/writable register that stores data for pins PTN7 to PTN0. B to PN0DT correspond to pins PTN7 to PTN0. When the pin function is general output port is read, the value of the corresponding PNDR bit is returned directly. When the fu general input port, if the port is read, the corresponding pin level is read.

PNnMD1	PNnMD0	Pin State	Read	Write
0	0	Other function	PNDR value	Data can be written to PNDR but on pin state.
	1	Output	PNDR value	Written data is output from the pin
1	0	Input (Pull-up MOS on)	Pin state	Data can be written to PNDR but on pin state.
	1	Input (Pull-up MOS off)	Pin state	Data can be written to PNDR but on pin state.

Note: n = 0 to 7

20.14 SC Port

The SC port is an 8-bit input/output port with the pin configuration shown in figure 20. pin has an input pull-up MOS, which is controlled by the SC port control register (SCP PFC.

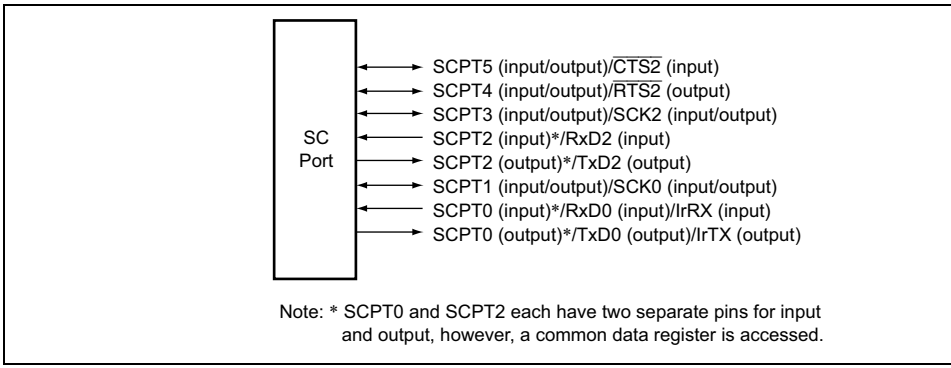


Figure 20.14 SC Port

SCPDR is an 8-bit readable/writable that stores data for pins SCPT5 to SCPT0. Bits S
SCP0DT correspond to pins SCPT5 to SCPT0. When the pin function is general outpu
port is read, the value of the corresponding SCPDR bit is returned directly. When the
general input port, if the port is read, the corresponding pin level is read.

When the RE bit of SCSCR_2 or SCSCR_0 in the serial communication interface with
(SCIF) is set to 1, the RxD2 and RxD0 pins become input pins, and their states can be
regardless of the setting of SCPCR.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	0	R	Reserved These bits are always read as 0. The write va always be 0.
5 to 0	SCP5DT to SCP0DT	0	R/W	Table 20.14 shows the function of SCPDR.

Table 20.14 SC Port Data Register (SCPDR) Read/Write Operations

- SCP1DR and SCP3DR to SCP5DR

SCPCR State				
SCPnMD1	**SCPnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	SCPDR value	Data can be written but no effect on pin s
	1	Output	SCPDR value	Written data is outpu pin.
1	0	Input (Pull-up MOS on)	Pin state	Data can be written but no effect on pin s
	1	Input (Pull-up MOS off)	Pin state	Data can be written but no effect on pin s

Note: n= 1 and 3 to 5

	impedance		no effect on pin state.
	RxD: Input (Pull-up MOS on)		
1	TxD: Output high impedance	RxD pin state	Data can be written to S no effect on pin state.
	RxD: Input (Pull-up MOS off)		

Note: n = 0 and 2

The operations are not guaranteed when read and write operations are prohibite

- Four input channels
- Minimum conversion time: 8.5 µs per channel (Pϕ = 33 MHz operation)
- Three conversion modes
  - Single mode: A/D conversion on one channel
  - Multi mode: A/D conversion on one to four channels
  - Scan mode: Continuous A/D conversion on one to four channels
- Four 16-bit data registers
  - A/D conversion results are transferred for storage into 16-bit data registers corr to the channels.
- Sample-and-hold function
- Interrupt source
  - At the end of A/D conversion, an A/D conversion end interrupt (ADI) can be r
- Module standby mode can be set

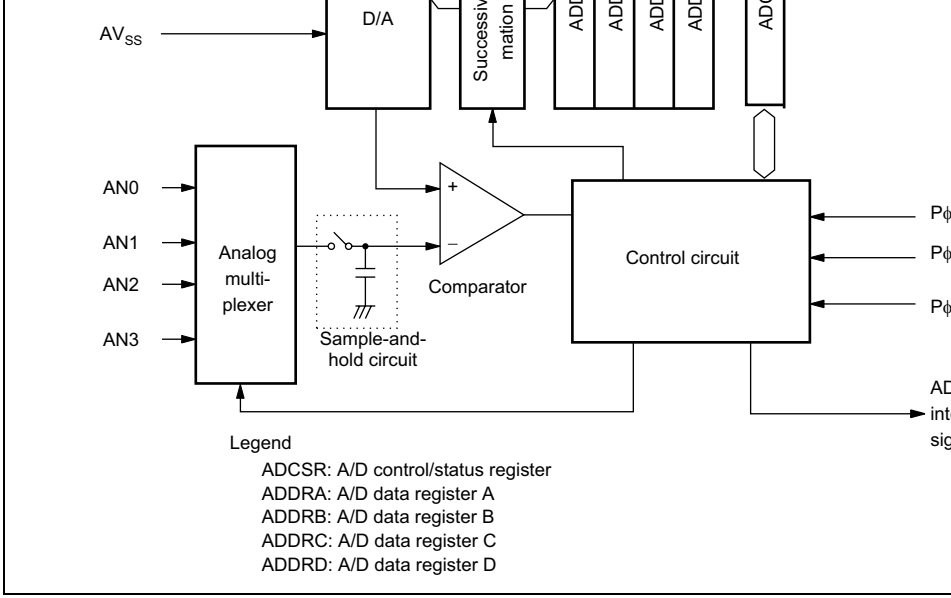


Figure 21.1 Block Diagram of A/D Converter

Pin Name	Abbreviation	Output	Function
Analog power supply	AVcc	Input	Analog power supply and referenc A/D conversion
Analog ground	AVss	Input	Analog ground and reference volta conversion
Analog input 0	AN0	Input	Analog input 0
Analog input 1	AN1	Input	Analog input 1
Analog input 2	AN2	Input	Analog input 2
Analog input 3	AN3	Input	Analog input 3

21.3 Register Descriptions

The A/D converter has the following registers. For more information on addresses of r register states in the processing, see section 24, List of Registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D control/status register (ADCSR)

The A/D data registers are initialized to H'0000.

Table 21.2 Analog Input Channels and A/D Data Registers

Analog Input Channel	A/D Data Register that Store Results of A/D Conversion
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD

21.3.2 A/D Control/Status Registers (ADCSR)

ADCSR is a 16-bit readable/writable register that selects the mode and controls the A/D

Bit	Bit Name	Initial Value	R/W	Description
15	ADF	0	R/(W)*	A/D End Flag Indicates the end of A/D conversion. [Setting conditions] Single mode: A/D conversion ends Multi mode: A/D conversion ends cycling throug selected channels Scan mode: A/D conversion ends cycling throug selected channels [Clearing conditions] (1) Reading ADF while ADF = 1, then writing 0 t (2) DMAC is activated by ADI interrupt and ADD read Note: * Clear this bit by writing 0. Writing 1 is ign

				Starts or stops A/D conversion. The ADST bit r 1 during A/D conversion. 0: A/D conversion is stopped. 1: Single mode: A/D conversion starts; ADST is automaticall when conversion ends on selected channels Multi mode: A/D conversion starts; when conversion is c cycling through the selected channels, ADS automatically cleared. Scan mode: A/D conversion starts and continues, A/D co continuously performed until ADST is cleare software, by a reset, or by a transition to sta
12	DMASL	0	R/W	DMAC Select Selects an interrupt due to ADF or activation of Set the DMASL bit while the ADST bit is 0. 0: An interrupt by ADF is selected. 1: Activation of the DMAC by ADF is selected.
11 to 8	—	0	R	Reserved These bits are always read as 0. The write valu always be 0.

				11: Setting prohibited Note: If the minimum conversion time is not lack of accuracy or abnormal operatio occur.
5 4	MULTI1 MULTI0	0 0	R/W R/W	Mode Select Selects single mode, multi mode, or scan mo 00: Single mode 01: Setting prohibited 10: Multi mode 11: Scan mode
3 2	— —	0 0	R R	Reserved These bits are always read as 0. The write va always be 0.
1 0	CH1 CH0	0 0	R/W R/W	Channel Select These bits and the MULTI bit select the analo channels. Clear the ADST bit to 0 before cha channel selection. Single mode / Multi mode or scan mode 00: AN0 / AN0 01: AN1 / AN0, AN1 10: AN2 / AN0 to AN2 11: AN3 / AN0 to AN3

Single mode should be selected when only one A/D conversion on one channel is requ

1. A/D conversion of the selected channel starts when the ADST bit of ADCSR is se software.

2. When conversion ends, the conversion results are transmitted to the A/D data regi corresponds to the channel.

3. When conversion ends, the ADF bit of ADCSR is set to 1. If the ADIE bit is also ADI interrupt is requested at this time.

4. The ADST bit holds 1 during A/D conversion. When A/D conversion is complete bit is cleared to 0 and the A/D converter becomes idle. When the ADST bit is clea during A/D conversion, the conversion is halted and the A/D converter becomes i To clear the ADF flag to 0, first read ADF, then write 0 to ADF.

21.4.2 Multi Mode

Multi mode should be selected when performing A/D conversions on one or more cha

1. When the ADST bit is set to 1 by software, A/D conversion starts with the smalle the analog input channel in the group (for instance, AN0, and AN1 to AN3).

2. When conversion of each channel ends, the conversion results are transmitted to t register that corresponds to the channel.

3. When conversion of all selected channels ends, the ADF bit of ADCSR is set to 1 ADIE bit is also set to 1, an ADI interrupt is requested at this time.

4. When A/D conversion is completed, the ADST bit is cleared to 0 and the A/D con becomes idle. When the ADST bit is cleared to 0 during A/D conversion, the con halted and the A/D converter becomes idle.
   To clear the ADF flag to 0, first read ADF, then write 0 to ADF.

register that corresponds to the channel.

3. When conversion of all selected channels ends, the ADF bit of ADCSR is set to 1. ADIE bit is also set to 1, an ADI interrupt is requested at this time. A/D conversion with the smaller number of the analog input channel.

4. The ADST bit is not automatically cleared to 0. When the ADST bit is set to 1, step above are repeated. When the ADST bit is cleared to 0, the conversion is halted and converter becomes idle.
   To clear the ADF flag to 0, first read ADF, then write 0 to ADF.

21.4.4 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples t input at an A/D conversion start delay time t_D after the ADST bit is set to 1, then starts Figure 21.2 shows the A/D conversion timing. Table 21.3 indicates the A/D conversion

As indicated in figure 21.2, the A/D conversion time (t_{CONV}) includes t_D and the input sa time (t_{SPL}). The length of t_D varies depending on the timing of the write access to ADCS total conversion time therefore varies within the ranges indicated in table 21.3.

In multi mode and scan mode, the values given in table 21.3 apply to the first conversio second and subsequent conversions, the values given in table 21.4 apply to the first con

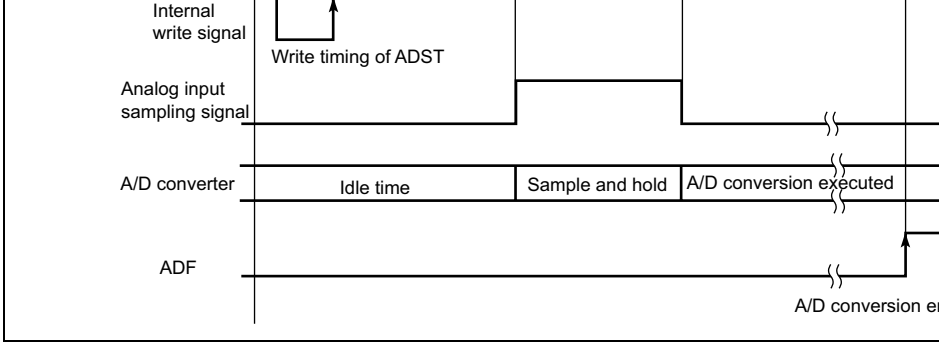


Figure 21.2 A/D Conversion Timing

Table 21.3 A/D Conversion Time (Single Mode)

		CKS1 = 1, CKS0 = 0			CKS1 = 0, CKS0 = 1			C C
	Symbol	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**
A/D conversion start delay	t_D	18	—	21	10	—	13	6
Input sampling time	t_{SPL}	—	129	—	—	65	—	—
A/D conversion time	t_{CONV}	535	—	545	275	—	285	141

Note: Values in the table are numbers of states for Pϕ.

Table 21.4 A/D Conversion Time (Multi Mode and Scan Mode)

CKS1	CKS0	Conversion Time (cycles)
0	0	128 (fixed)
0	1	256 (fixed)
1	0	512 (fixed)
1	1	Unused

data transfer.

Table 21.5 A/D Converter Interrupt Source

Name	Interrupt source	Interrupt flag	DMAC acti
ADI	A/D conversion end	ADF	Yes

21.6 Definitions of A/D Conversion Accuracy

The following shows the definitions of A/D conversion accuracy. In the figure, the 10 b
A/D converter have been simplified to 3 bits.

- Resolution
  Digital output code number of the A/D converter
- Quantization error
  Intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 21.3)
- Offset error
  Deviation between analog input voltage and ideal A/D conversion characteristics wh
  digital output value changes from the minimum (zero voltage) 0000000000 (H'00; 0
  figure 21.3) to 0000000001 (H'01; 001 in figure 21.3) (figure 21.4)
- Full-scale error
  Deviation between analog input voltage and ideal A/D conversion characteristics wh
  digital output value changes from the 1111111110 (H'3EF; 110 in figure 21.3) to th
  1111111111 (H'3FF; 111 in figure 21.3) (figure 21.4).
- Nonlinearity error
  Deviation between analog input voltage and ideal A/D conversion characteristics be
  voltage and full-scale voltage (figure 21.4). Note that it does not include offset, full-
  quantization error.
- Absolute accuracy
  Deviation between analog and digital input values. Note that it includes offset, full-
  quantization, or nonlinearity error.

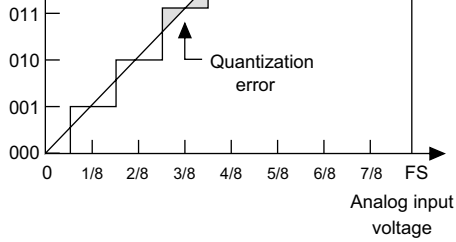


Figure 21.3 Definitions of A/D Conversion Accuracy

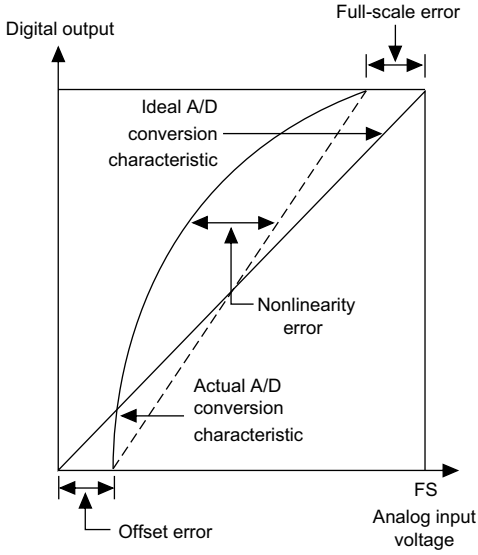


Figure 21.4 Definitions of A/D Conversion Accuracy

since input load is only internal input resistance of 3 kΩ. However, an analog signal wi
differential coefficient (5 mV/μs or greater) cannot be followed up because of a low-pa
(figure 21.5). When converting high-speed analog signals or converting in scan mode, i
low-impedance buffer.

21.7.2 Influence to Absolute Accuracy

By adding capacitance, absolute accuracy may be degraded if noise is on GND because
coupling with GND. Therefore, connect electrically stable GND such as AVcc to preve
accuracy from being degraded.

A filter circuit must not interfere with digital signals, or must not be an antenna on a m
board.

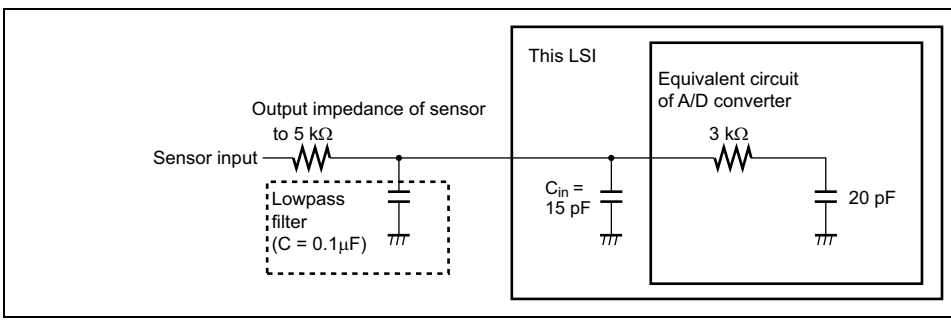


Figure 21.5 Analog Input Circuit Example

21.7.3 Setting Analog Input Voltage

Operating the chip in excess of the following voltage range may result in damage to chi
reliability.

- Analog Input Voltage Range: During A/D conversion, the voltages (VANn) input to
  input pins ANn should be in the range $AV_{SS} \leq VANn \leq AV_{CC}$ (n = 0 to 3).

ground (AVss) to one point of stable ground (Vss) on the board.

21.7.5 Notes on Countermeasures to Noise

Connect a protective circuit between AVcc and AVss, as shown in figure 21.6, to prev
of analog input pins (AN0 to AN3) due to abnormal voltage such as excessive serge. C
bypass capacitor that is connected to AVcc and a capacitor for a filter that is connecte
AN3 to AVss.

When a capacitor for a filter is connected, input currents of AN0 to AN3 are averaged
causing errors. If A/D conversion is frequently performed in scan mode, voltages of an
pins cause errors when a current that is charged/discharged for capacitance of a sampl
circuit in the A/D converter is higher than a current that is input through input impeda
Therefore, determine a circuit constant carefully.

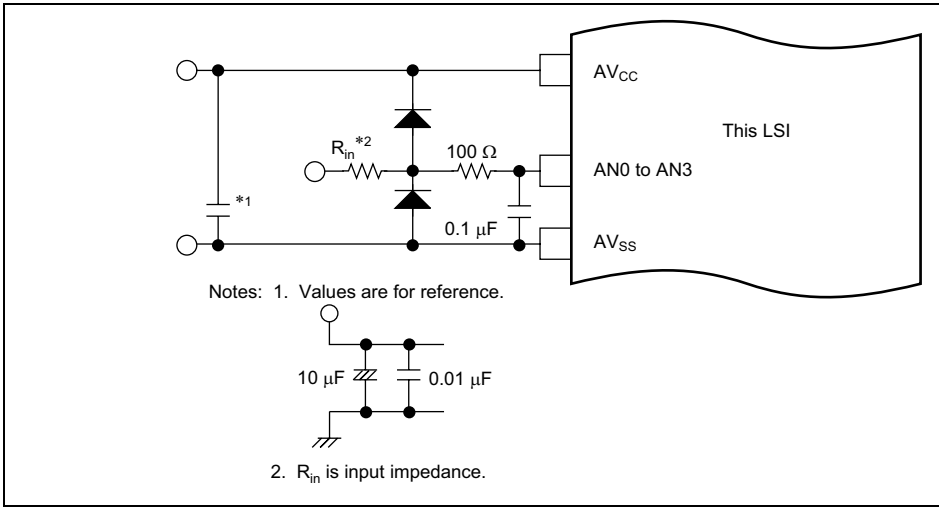


Figure 21.6 Example of Analog Input Protection Circuit

Note: Values are for reference.

Figure 21.7 Analog Input Pin Equivalent Circuit

22.1 Features

The UBC has the following features:

- The following break comparison conditions can be set.

  Number of break channels: two channels (channels A and B)
  User break can be requested as either the independent or sequential condition on ch
  and B (sequential break setting: channel A and then channel B match with break co
  but not in the same bus cycle).

  — Address (Compares 40 bits configured of the ASID and addresses 32 bits: the A
  selected either all-bit comparison or all-bit mask. Comparison bits for the addr
  maskable in 1-bit units; user can mask addresses at lower 12 bits (4 k page), lo
  (1 k page), or any size of page, etc.)

  One of the two address buses (L bus address (LAB) and I bus address (IAB)) c
  selected.

  — Data (only on channel B, 32-bit maskable)

  One of the two data buses (L bus data (LDB) and I bus data (IDB)) can be sele

  — Bus cycle: Instruction fetch or data access

  — Read/write

  — Operand size: Byte, word, or longword
- User break is generated upon satisfying break conditions. A user-designed user-bre
  condition exception processing routine can be run.
- In an instruction fetch cycle, it can be selected that a break is set before or after an
  is executed.
- Maximum repeat times for the break condition (only for channel B): $2^{12} - 1$ times.
- Eight pairs of branch source/destination buffers.

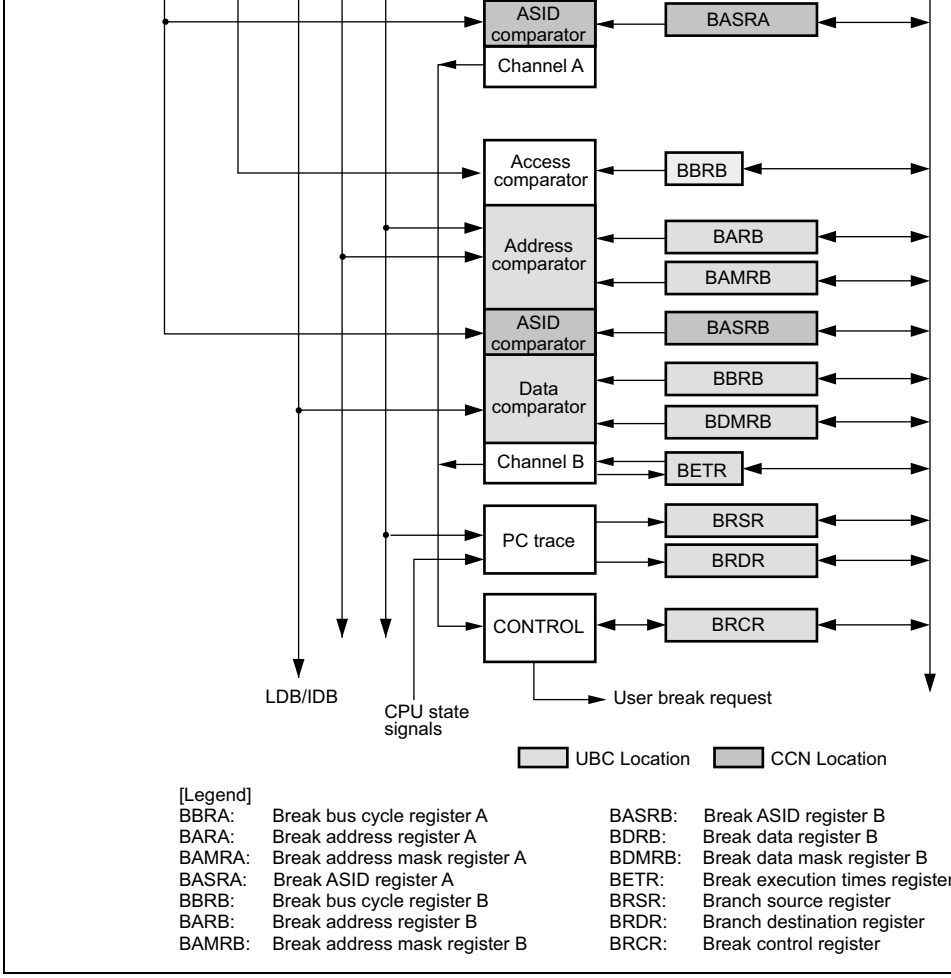


Figure 22.1 Block Diagram of User Break Controller

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

22.2.1 Break Address Register A (BARA)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAA31 to BAA0	0	R/W	Break Address A Store the address on the LAB or IAB speci conditions of channel A.

				bits specified by BARA (BAA31 to BAA0). 0: Break address bit BAAn of channel A is i the break condition 1: Break address bit BAAn of channel A is n and is not included in the break condition Note: n = 31 to 0

22.2.3 Break Bus Cycle Register A (BBRA)

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

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

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

22.2.4 Break Address Register B (BARB)

BARB is a 32-bit readable/writable register. BARB specifies the address used as a bre in channel B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAB31 to BAB0	0	R/W	Break Address B Stores an address which specifies a break co channel B. BARB specifies the break address on LAB or

specified by BARB (BAB31 to BAB0).

0: Break address BABn of channel B is includ break condition

1: Break address BABn of channel B is mask not included in the break condition

Note: n = 31 to 0

22.2.6 Break Data Register B (BDRB)

BDRB is a 32-bit readable/writable register.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDB31 to BDB0	0	R/W	Break Data Bit B Stores data which specifies a break condition B. BDRB specifies the break data on LDB or IDB

Notes: 1. Specify an operand size when including the value of the data bus in the brea

2. When the byte size is selected as a break condition, the same byte data mus bits 15 to 8 and 7 to 0 in BDRB as the break data.

				specified by BDRB (BDB31 to BDB0). 0: Break data BDBn of channel B is includ break condition 1: Break data BDBn of channel B is maske included in the break condition Note: n = 31 to 0

Notes: 1. Specify an operand size when including the value of the data bus in the bre
2. When the byte size is selected as a break condition, the same byte data mu bits 15 to 8 and 7 to 0 in BDRB as the break mask data in BDMRB.

22.2.8 Break Bus Cycle Register B (BBRB)

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	0	R	Reserved These bits are always read as 0. The write val always be 0.

5	IDB1	0	R/W	Instruction Fetch/Data Access Select B
4	IDB0	0	R/W	Select the instruction fetch cycle or data access the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch c 10: The break condition is the data access cycle 11: The break condition is the instruction fetch c access cycle
3	RWB1	0	R/W	Read/Write Select B
2	RWB0	0	R/W	Select the read cycle or write cycle as the bus c channel B break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or writ
1	SZB1	0	R/W	Operand Size Select B
0	SZB0	0	R/W	Select the operand size of the bus cycle for the break condition. 00: The break condition does not include operar 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

conditions.

4. Specifies whether to include data bus on channel B in comparison conditions.
5. Enables PC trace.
6. Enables ASID check.

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	0	R	Reserved These bits are always read as 0. The write v always be 0.
21	BASMA	0	R/W	Break ASID Mask A Specifies whether bits in channel A break AS ASID0 (BASA7 to BASA0) which are set in B masked or not. 0: All BASRA bits are included in the break c and the ASID is checked 1: All BASRA bits are not included in the bre and the ASID is not checked
20	BASMB	0	R/W	Break ASID Mask B Specifies whether bits in channel B break AS ASID0 (BASB7 to BASB0) which are set in B masked or not. 0: All BASRB bits are included in the break c and the ASID is checked 1: All BASRB bits are not included in the bre and the ASID is not checked
19 to 16	—	0	R	Reserved These bits are always read as 0. The write v always be 0.

				When the L bus cycle condition in the break cond for channel B is satisfied, this flag is set to 1. In c clear this flag, write 0 into this bit. 0: The L bus cycle condition for channel B does 1: The L bus cycle condition for channel B match
13	SCMFDA	0	R/W	I Bus Cycle Condition Match Flag A When the I bus cycle condition in the break cond channel A is satisfied, this flag is set to 1. In orde this flag, write 0 into this bit. 0: The I bus cycle condition for channel A does n 1: The I bus cycle condition for channel A matche
12	SCMFDB	0	R/W	I Bus Cycle Condition Match Flag B When the I bus cycle condition in the break cond channel B is satisfied, this flag is set to 1. In orde this flag, write 0 into this bit. 0: The I bus cycle condition for channel B does n 1: The I bus cycle condition for channel B matche
11	PCTE	0	R/W	PC Trace Enable 0: Disables PC trace 1: Enables PC trace
10	PCBA	0	R/W	PC Break Select A Selects the break timing of the instruction fetch c channel A as before or after instruction executio 0: PC break of channel A is set before instructio 1: PC break of channel A is set after instruction e

				B 1: The data bus condition is included in the condi channel B
6	PCBB	0	R/W	PC Break Select B Selects the break timing of the instruction fetch cy channel B as before or after instruction execution 0: PC break of channel B is set before instruction 1: PC break of channel B is set after instruction e
5, 4	—	0	R	Reserved These bits are always read as 0. The write value always be 0.
3	SEQ	0	R/W	Sequence Condition Select Selects two conditions of channels A and B as in sequential conditions. 0: Channels A and B are compared under indepe conditions 1: Channels A and B are compared under sequer conditions (channel A, then channel B)
2, 1	—	0	R	Reserved These bits are always read as 0. The write value always be 0.
0	ETBE	0	R/W	Number of Execution Times Break Enable Enables the execution-times break condition only B. If this bit is 1 (break enable), a user break is is the number of satisfied break conditions matches number of execution times that is specified by BE 0: The execution-times break condition is disable B 1: The execution-times break condition is enable B

15 to 12	—	0	R	Reserved These bits are always read as 0. The write v always be 0.
11 to 0	BET11 to BET0	0	R/W	Number of Execution Times

Note: If the channel B brake condition set to during instruction fetch cycles and any of t instructions below perform breaks, BETR is not decremented when the first brea The decremented values are listed below.

Instruction	Value Decremented	Instruction	Va De
RTE	4	LDC.L @Rm+,SR	6
DMULS.L Rm,Rn	2	LDC.L @Rm+,GBR	4
DMULU.L Rm,Rn	2	LDC.L @Rm+,VBR	4
MAC.L @Rm+,@Rn+	2	LDC.L @Rm+,SSR	4
MAC.W @Rm+,@Rn+	2	LDC.L @Rm+,SPC	4
MUL.L Rm,Rn	3	LDC.L @Rm+,R0_BANK	4
AND.B #imm,@(R0,GBR)	3	LDC.L @Rm+,R1_BANK	4
OR.B #imm,@(R0,GBR)	3	LDC.L @Rm+,R2_BANK	4
TAS.B @Rn	3	LDC.L @Rm+,R3_BANK	4
TST.B #imm,@(R0,GBR)	3	LDC.L @Rm+,R4_BANK	4
XOR.B #imm,@(R0,GBR)	3	LDC.L @Rm+,R5_BANK	4
LDC Rm,SR	4	LDC.L @Rm+,R6_BANK	4
LDC Rm,GBR	4	LDC.L @Rm+,R7_BANK	4
LDC Rm,VBR	4	LDC.L @Rn+,MOD	4
LDC Rm,SSR	4	LDC.L @Rn+,RS	4
LDC Rm,SPC	4	LDC.L @Rn+,RE	4
LDC Rm,R0_BANK	4	LDC Rn,MOD	4
LDC Rm,R1_BANK	4	LDC Rn,RS	4
LDC Rm,R2_BANK	4	LDC Rn,RE	4
LDC Rm,R3_BANK	4	BSR label	2
LDC Rm,R4_BANK	4	BSRF Rm	2
LDC Rm,R5_BANK	4	JSR @Rm	2
LDC Rm,R6_BANK	4		
LDC Rm,R7_BANK	4		

Bit	Name	Value	R/W	Description
31	SVF	0	R	BRSR Valid Flag Indicates whether the branch source addre When a branch source address is fetched, set to 1. This flag is cleared to 0 by readin 0: The value of BRSR register is invalid 1: The value of BRSR register is valid
30 to 28	—	0	R	Reserved These bits are always read as 0.
27 to 0	BSA27 to BSA0	Undefined	R	Branch Source Address Store bits 27 to 0 of the branch source add

Bit	Name	Value	R/W	Description
31	DVF	0	R	BRDR Valid Flag Indicates whether a branch destination ad stored. When a branch destination addres fetched, this flag is set to 1. This flag is cl by reading BRDR. 0: The value of BRDR register is invalid 1: The value of BRDR register is valid
30 to 28	—	0	R	Reserved These bits are always read as 0.
27 to 0	BDA27 to BDA0	Undefined	R	Branch Destination Address Store bits 27 to 0 of the branch destinatio

22.2.13 Break ASID Register A (BASRA)

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

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

	BASB0	channel B.

22.3 Operation

22.3.1 Flow of the User Break Operation

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

1. The break addresses and corresponding ASID are set in the break address registers BARB) and break ASID registers (BASRA and BASRB in CNN). The masked ad set in the break address mask registers (BAMRA and BAMRB). The break data is break data register (BDRB). The masked data is set in the break data mask register The bus break conditions are set in the break bus cycle registers (BBRA and BBR groups of BBRA and BBRB (L bus cycle/I bus cycle select, instruction fetch/data select, and read/write select) are each set. No user break will be generated if even groups is set with 00. The respective conditions are set in the bits of the break cont (BRCR). Make sure to set all registers related to breaks before setting BBRA/BBR
2. When the break conditions are satisfied, the UBC sends a user break request to the sets the L bus condition match flag (SCMFCA or SCMFCB) and the I bus conditio flag (SCMFDA or SCMFDB) for the appropriate channel.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCM be used to check if the set conditions match or not. The matching of the conditions but they are not reset. 0 must first be written to them before they can be used again
4. There is a chance that the data access break and its following instruction fetch brea around the same time. There will be only one break request to the CPU, but these t channel match flags could be both set.

be cached, they are issued with the data size specified on the L bus and their add not rounded.

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

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

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

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

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

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

2. An instruction set for a break before execution breaks when it is confirmed that the has been fetched and will be executed. This means this feature cannot be used on i fetched by overrun (instructions fetched at a branch or during an interrupt transitio be executed). When this kind of break is set for the delay slot of a delayed branch the break is generated prior to execution of the delayed branch instruction.

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

3. When the condition is specified to be occurred after execution, the instruction set w break condition is executed and then the break is generated prior to the execution o instruction. As with pre-execution breaks, this cannot be used with overrun fetch in When this kind of break is set for a delayed branch instruction and its delay slot, a generated until the first instruction at the branch destination.
4. When an instruction fetch cycle is set for channel B, the break data register B (BD ignored. Therefore, break data cannot be set for the break of the instruction fetch c
5. If the I bus is set for a break of an instruction fetch cycle, the condition is determin instruction fetch cycles on the I bus. For details, see 5 in section 22.3.1, Flow of th Break Operation.

2. The relationship between the data access cycle address and the comparison conditio operand size is listed in table 22.1.

Table 22.1 Data Access Cycle Addresses and Operand Size Comparison Conditi

Access Size	Address Compared
Longword	Compares break address register bits 31 to 2 to address bus bits
Word	Compares break address register bits 31 to 1 to address bus bits
Byte	Compares break address register bits 31 to 0 to address bus bits

This means that when address H'00001003 is set in the break address register (BAR BARB), for example, the bus cycle in which the break condition is satisfied is as fo (where other conditions are met).

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

3. When the data value is included in the break conditions on channel B:

   When the data value is included in the break conditions, either longword, word, or b specified as the operand size of the break bus cycle register B (BBRB). When data included in break conditions, a break is generated when the address conditions and conditions both match. To specify byte data for this case, set the same data in two b 15 to 8 and bits 7 to 0 of the break data register B (BDRB) and break data mask reg (BDMRB). When word or byte is set, bits 31 to 16 of BDRB and BDMRB are igno

4. Access by a PREF instruction is handled as read access in longword units without a Therefore, if including the value of the data bus when a PREF instruction is specifie break condition, a break will not occur.

5. If the L bus is selected, a break occurs on ending execution of the instruction that m break condition, and immediately before the next instruction is executed. However, is also specified as the break condition, the break may occur on ending execution of instruction following the instruction that matches the break condition. If the I bus is the instruction at which the break will occur cannot be determined. When this kind

To clear the channel A condition match when a channel A condition match has occ
channel B condition match has not yet occurred in a sequential break specification
SEQ bit in BRCR to 0.

2. In sequential break specification, the L or I bus can be selected and the execution t
condition can be also specified. For example, when the execution times break cond
specified, the break condition is satisfied when a channel B condition matches with
H'0001 after a channel A condition has matched.

22.3.5 Value of Saved Program Counter

When a break occurs, the address of the instruction from where execution is to be resu
saved in the SPC, and the exception handling state is entered. If the L bus is specified
condition, the instruction at which the break should occur can be clearly determined (e
when data is included in the break condition). If the I bus is specified as a break condi
instruction at which the break should occur cannot be clearly determined.

slot matches the condition, these instructions are executed, and the branch destinatio
is saved in the SPC.

3. When data access (address only) is specified as a break condition:

   The address of the instruction immediately after the instruction that matched the bre
   condition is saved in the SPC. The instruction that matches the condition is executed
   break occurs before the next instruction is executed. However when a delay slot ins
   matches the condition, the branch destination address is saved in the SPC.

4. When data access (address + data) is specified as a break condition:

   When a data value is added to the break conditions, the address of an instruction tha
   two instructions of the instruction that matched the break condition is saved in the S
   which instruction the break occurs cannot be determined accurately.

   When a delay slot instruction matches the condition, the branch destination address
   the SPC. If the instruction following the instruction that matches the break condition
   branch instruction, the break may occur after the branch instruction or delay slot has
   In this case, the branch destination address is saved in the SPC.

— If a branch occurs due to an interrupt or exception, the value saved in SPC due
occurrence is saved in BRSR and the start address of the exception handling ro
saved in BRDR.

3. BRSR and BRDR have eight pairs of queue structures. The top of queues is read f
address stored in the PC trace register is read. BRSR and BRDR share the read poi
BRSR and BRDR in order, the queue only shifts after BRDR is read. After switchi
PCTE bit (in BRCR) off and on, the values in the queues are invalid.

Specified conditions: Channel A/channel B independent mode

- Channel A

  Address: H'00000404, Address mask: H'00000000

  Bus cycle: L bus/instruction fetch (after instruction execution)/read (op
  is not included in the condition)

  The ASID check is not included.

- Channel B

  Address: H'00008010, Address mask: H'00000006

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/read (d
  is not included in the condition)

  The ASID check is not included.

A user break occurs after an instruction of address H'00000404 is executed or be
instructions of addresses H'00008010 to H'00008016 are executed.

2. Register specifications

   BARA = H'00037226, BAMRA = H'00000000, BBRA = H'0056, BARB = H'0003
   BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'000
   BRCR = H'00000008, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B sequential mode

   - Channel A

     Address: H'00037226, Address mask: H'00000000, ASID = H'80

     Bus cycle: L bus/instruction fetch (before instruction execution)/read/w

   - Channel B

     Address: H'0003722E, Address mask: H'00000000, ASID = H'70

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: L bus/instruction fetch (before instruction execution)/read/w

   After an instruction with ASID = H'80 and address H'00037226 is executed, a u
   occurs before an instruction with ASID = H'70 and address H'0003722E is exec

The ASID check is not included.

- Channel B

  Address: H'00031415, Address mask: H'00000000

  Data: H'00000000, Data mask: H'00000000

  The ASID check is not included.

  Bus cycle: L bus/instruction fetch (before instruction execution)/read (
  is not included in the condition)

On channel A, no user break occurs since instruction fetch is not a write cycle.
B, no user break occurs since instruction fetch is performed for an even addres

4. Register specifications

   BARA = H'00037226, BAMRA = H'00000000, BBRA = H'005A, BARB = H'000
   BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00
   BRCR = H'00000008, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B sequential mode

   - Channel A

     Address: H'00037226, Address mask: H'00000000, ASID = H'80

     Bus cycle: L bus/instruction fetch (before instruction execution)/write

   - Channel B

     Address: H'0003722E, Address mask: H'00000000, ASID = H'70

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: L bus/instruction fetch (before instruction execution)/read/

   Since instruction fetch is not a write cycle on channel A, a sequential condition
   match. Therefore, no user break occurs.

The ASID check is not included.

- Channel B

  Address: H'00001000, Address mask: H'00000000

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/lo

  The number of execution-times break enable (5 times)

  The ASID check is not included.

On channel A, a user break occurs before an instruction of address H'00000500
On channel B, a user break occurs after the instruction of address H'00001000 a
four times and before the fifth time.

6. Register specifications

   BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BARB = H'0000
   BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'000
   BRCR = H'00000400, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00008404, Address mask: H'00000FFF, ASID = H'80

     Bus cycle: L bus/instruction fetch (after instruction execution)/read (op
     is not included in the condition)

   - Channel B

     Address: H'00008010, Address mask: H'00000006, ASID = H'70

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: L bus/instruction fetch (before instruction execution)/read (o
     is not included in the condition)

   A user break occurs after an instruction with ASID = H'80 and addresses H'0000
   H'00008FFE is executed or before an instruction with ASID = H'70 and address
   H'00008010 to H'00008016 are executed.

Bus cycle: L bus/data access/read (operand size is not included in the

- Channel B

  Address: H'000ABCDE, Address mask: H'000000FF, ASID = H'70

  Data: H'0000A512, Data mask: H'00000000

  Bus cycle: L bus/data access/write/word

On channel A, a user break occurs with longword read from ASID = H'80 and H'00123454, word read from address H'00123456, or byte read from address H
On channel B, a user break occurs when word H'A512 is written in ASID = H'
addresses H'000ABC00 to H'000ABCFE.

Break Condition Specified for an I Bus Data Access Cycle

Register specifications:

BARA = H'00314156, BAMRA = H'00000000, BBRA = H'0094, BARB = H'0005
BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00007878, BDMRB = H'00
BRCR = H'00000080, BASRA = H'80, BASRB = H'70

Specified conditions: Channel A/channel B independent mode

- Channel A

  Address: H'00314156, Address mask: H'00000000, ASID = H'80

  Bus cycle: I bus/instruction fetch/read (operand size is not included in condition)

- Channel B

  Address: H'00055555, Address mask: H'00000000, ASID = H'70

  Data: H'00000078, Data mask: H'0000000F

  Bus cycle: I bus/data access/write/byte

On channel A, a user break occurs when instruction fetch is performed for ASI and address H'00314156 in the memory space.

On channel B, a user break occurs when byte data H'7* is written in address H' with ASID = H'70 on the I bus.

3. Note on specification of sequential break:

   A condition match occurs when a B-channel match occurs in a bus cycle after an A- match occurs in another bus cycle in sequential break setting. Therefore, no break o if a bus cycle, in which an A-channel match and a channel B match occur simultane set.

4. When a user break and another exception occur at the same instruction, which has h priority is determined according to the priority levels defined in table 5.1 in section Exception Handling. If an exception with higher priority occurs, the user break is no generated.

   — Pre-execution break has the highest priority.

   — When a post-execution break or data access break occurs simultaneously with a execution-type exception (including pre-execution break) that has higher priorit execution-type exception is accepted, and the condition match flag is not set (se exception in the following note). The break will occur and the condition match f set only after the exception source of the re-execution-type exception has been c the exception handling routine and re-execution of the same instruction has ende

   — When a post-execution break or data access break occurs simultaneously with a completion-type exception (TRAPA) that has higher priority, though a break do occur, the condition match flag is set.

5. Note the following exception for the above note.

   If a post-execution break or data access break is satisfied by an instruction that gene CPU address error (or TLB related exception) by data access, the CPU address erro related exception) is given priority to the break. Note that the UBC condition match in this case.

6. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at the delay slot instruction of the RTE instruction, th does not occur until the branch destination of the RTE instruction.

7. User breaks are disabled during USB module standby mode. Do not read from or w UBC registers during USB module standby mode; the values are not guaranteed.

23.1 Features

The UDI (User Debugging Interface) is a serial I/O interface which supports with JTA
Test Action Group, IEEE Standard 1149.1 and IEEE Standard Test Access Port and B
Scan Architecture) specifications.

The UDI in this LSI supports a boundary scan mode, and is also used for emulator co

When using an emulator, UDI functions should not be used. Refer to the emulator ma
method of connecting the emulator.

Figure 23.1 shows a block diagram of the UDI.

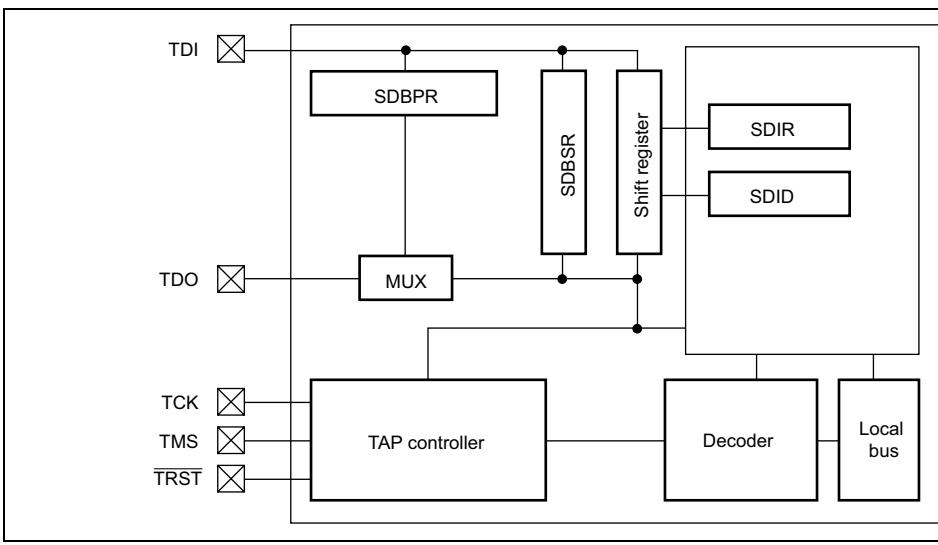


Figure 23.1 Block Diagram of UDI

		(TDI), and output from the data output pin (TDO), in synchronization with this clock.
TMS*	Input	Mode Select Input Pin The state of the TAP control circuit is determined by this signal in synchronization with TCK. The protocc supported to the JTAG standard (IEEE Std.1149.1).
$\overline{\text{TRST}}$*	Input	Reset Input Pin Input is accepted asynchronously with respect to TC when low, the UDI is reset. $\overline{\text{TRST}}$ must be held low constant period when power is turned on regardless the UDI function. As the same as the $\overline{\text{RESETP}}$ pin, pin should be driven low at the power-on reset state high after the power-on reset state is released. This from the JTAG standard. See section 23.4.2, Reset Configuration, for more in
TDI*	Input	Serial Data Input Pin Data transfer to the UDI is executed by changing th synchronization with TCK.
TDO	Output	Serial Data Output Pin Data read from the UDI is executed by reading this synchronization with TCK. The data output timing d the command type set in the SDIR. See section 23. Instruction Register (SDIR), for more information.
$\overline{\text{ASEMD0}}$*	Input	ASE Mode Select Pin If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{R}}$ is asserted, ASE mode is entered; if a high level is i mode is entered. In ASE mode, dedicated emulator can be used. The input level at the $\overline{\text{ASEMD0}}$ pin sh for at least one cycle after $\overline{\text{RESETP}}$ negation. See s 23.4.2, Reset Configuration, for more information.

23.3 Register Descriptions

The UDI has the following registers. For details on register addresses and register stat
processing state, see section 24, List of Registers.

- Bypass register (SDBPR)
- Instruction register (SDIR)
- Boundary scan register (SDBSR)
- ID register (SDID)

23.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to th
mode, SDBPR is connected between UDI pins TDI and TDO. The initial value is und
SDBPR is initialized when the TAP enters the Capture-DR state.

23.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. The register is in JTAG IDCODE in its initial stat
initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state, and can be written to
irrespective of the CPU mode. Operation is not guaranteed if a reserved command is s
register.

				This bit is always read as 0.
0	—	1	R	Reserved This bit is always read as 1.

Table 23.2 UDI Commands

Bits 15 to 8								
TI7	**TI6**	**TI5**	**TI4**	**TI3**	**TI2**	**TI1**	**TI0**	**Description**
0	0	0	0	—	—	—	—	JTAG EXTEST
0	0	1	0	—	—	—	—	JTAG CLAMP
0	0	1	1	—	—	—	—	JTAG HIGHZ
0	1	0	0	—	—	—	—	JTAG SAMPLE/PR
0	1	1	0	—	—	—	—	UDI reset negate
0	1	1	1	—	—	—	—	UDI reset assert
1	0	1	—	—	—	—	—	UDI interrupt
1	1	1	0	—	—	—	—	JTAG IDCODE (In
1	1	1	1	—	—	—	—	JTAG BYPASS
Other than the above								Reserved

23.3.3 Boundary Scan Register (SDBSR)

SDBSR is a 385-bit shift register, located on the PAD, for controlling the input/output p LSI. The initial value is undefined. SDBSR cannot be accessed by the CPU.

Using the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands, a bounda test which supports the JTAG standard can be carried out. Table 23.3 shows the corresp between this LSI's pins and boundary scan register bits.

380	D29/PTB5/PINT13	IN	348	D30/PTB6/PINT14
379	D28/PTB4/PINT12	IN	347	D29/PTB5/PINT13
378	D27/PTB3/PINT11	IN	346	D28/PTB4/PINT12
377	D26/PTB2/PINT10	IN	345	D27/PTB3/PINT11
376	D25/PTB1/PINT9	IN	344	D26/PTB2/PINT10
375	D24/PTB0/PINT8	IN	343	D25/PTB1/PINT9
374	D23/PTA7/PINT7	IN	342	D24/PTB0/PINT8
373	D22/PTA6/PINT6	IN	341	D23/PTA7/PINT7
372	D21/PTA5/PINT5	IN	340	D22/PTA6/PINT6
371	D20/PTA4/PINT4	IN	339	D21/PTA5/PINT5
370	D19/PTA3/PINT3	IN	338	D20/PTA4/PINT4
369	D18/PTA2/PINT2	IN	337	D19/PTA3/PINT3
368	D17/PTA1/PINT1	IN	336	D18/PTA2/PINT2
367	D16/PTA0/PINT0	IN	335	D17/PTA1/PINT1
366	D15	IN	334	D16/PTA0/PINT0
365	D14	IN	333	D15
364	D13	IN	332	D14
363	D12	IN	331	D13
362	D11	IN	330	D12
361	D10	IN	329	D11
360	D9	IN	328	D10
359	D8	IN	327	D9
358	D7	IN	326	D8
357	D6	IN	325	D7
356	D5	IN	324	D6
355	D4	IN	323	D5
354	D3	IN	322	D4

314	D29/PTB5/PINT13	Control	281	A21/PTK3
313	D28/PTB4/PINT12	Control	280	A22/PTK4
312	D27/PTB3/PINT11	Control	279	A23/PTK5
311	D26/PTB2/PINT10	Control	278	A24/PTK6
310	D25/PTB1/PINT9	Control	277	A25/PTK7
309	D24/PTB0/PINT8	Control	276	$\overline{\text{BS}}$/PTC0
308	D23/PTA7/PINT7	Control	275	$\overline{\text{WE2}}$/DQMUL/PTC1
307	D22/PTA6/PINT6	Control	274	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{AH}}$/PTC2
306	D21/PTA5/PINT5	Control	273	$\overline{\text{CS2}}$/PTC3
305	D20/PTA4/PINT4	Control	272	$\overline{\text{CS3}}$/PTC4
304	D19/PTA3/PINT3	Control	271	$\overline{\text{CS4}}$/PTC5
303	D18/PTA2/PINT2	Control	270	$\overline{\text{CS5A}}$/PTC6
302	D17/PTA1/PINT1	Control	269	$\overline{\text{CS5B}}$/PTD6
301	D16/PTA0/PINT0	Control	268	$\overline{\text{CS6A}}$/PTC7
300	D15	Control	267	$\overline{\text{CS6B}}$/PTD7
299	D14	Control	266	$\overline{\text{RASL}}$/PTD0
298	D13	Control	265	$\overline{\text{RASU}}$/PTD1
297	D12	Control	264	$\overline{\text{CASL}}$/PTD2
296	D11	Control	263	A0/PTK0
295	D10	Control	262	A1
294	D9	Control	261	A2
293	D8	Control	260	A3
292	D7	Control	259	A4
291	D6	Control	258	A5
290	D5	Control	257	A6
289	D4	Control	256	A7

248	A15	OUT	215	A4
247	A16	OUT	214	A5
246	A17	OUT	213	A6
245	A18	OUT	212	A7
244	A19/PTK1	OUT	211	A8
243	A20/PTK2	OUT	210	A9
242	A21/PTK3	OUT	209	A10
241	A22/PTK4	OUT	208	A11
240	A23/PTK5	OUT	207	A12
239	A24/PTK6	OUT	206	A13
238	A25/PTK7	OUT	205	A14
237	$\overline{BS}$/PTC0	OUT	204	A15
236	$\overline{RD}$	OUT	203	A16
235	$\overline{WE0}$/DQMLL	OUT	202	A17
234	$\overline{WE1}$/DQMLU	OUT	201	A18
233	$\overline{WE2}$/DQMUL/PTC1	OUT	200	A19/PTK1
232	$\overline{WE3}$/DQMUU/$\overline{AH}$/PTC2	OUT	199	A20/PTK2
231	RD/$\overline{WR}$	OUT	198	A21/PTK3
230	$\overline{CS0}$	OUT	197	A22/PTK4
229	$\overline{CS2}$/PTC3	OUT	196	A23/PTK5
228	$\overline{CS3}$/PTC4	OUT	195	A24/PTK6
227	$\overline{CS4}$/PTC5	OUT	194	A25/PTK7
226	$\overline{CS5A}$/PTC6	OUT	193	$\overline{BS}$/PTC0
225	$\overline{CS5B}$/PTD6	OUT	192	$\overline{RD}$
224	$\overline{CS6A}$/PTC7	OUT	191	$\overline{WE0}$/DQMLL
223	$\overline{CS6B}$/PTD7	OUT	190	$\overline{WE1}$/DQMLU

182	$\overline{CS5A}$/PTC6	Control	149	PTM3
181	$\overline{CS5B}$/PTD6	Control	148	$\overline{ASEBRKAK}$/PTF6
180	$\overline{CS6A}$/PTC7	Control	147	MD0
179	$\overline{CS6B}$/PTD7	Control	146	MD1
178	$\overline{RASL}$/PTD0	Control	145	MD2
177	$\overline{RASU}$/PTD1	Control	144	MD5
176	$\overline{CASL}$/PTD2	Control	143	$\overline{CASU}$/PTD3
175	$\overline{CASU}$/PTD3	IN	142	CKE/PTD4
174	CKE/PTD4	IN	141	PTD5/NF
173	PTD5/NF	IN	140	$\overline{BACK}$/PTG5
172	$\overline{BACK}$/PTG5	IN	139	$\overline{BREQ}$/PTG6
171	$\overline{BREQ}$/PTG6	IN	138	$\overline{WAIT}$/PTG7
170	$\overline{WAIT}$/PTG7	IN	137	DACK0/PTE0
169	DACK0/PTE0	IN	136	DACK1/PTE1
168	DACK1/PTE1	IN	135	TEND0/PTE3
167	TEND0/PTE3	IN	134	AUDSYNC/PTF4
166	AUDSYNC/PTF4	IN	133	AUDATA0/PTF0/TO0
165	AUDATA0/PTF0/TO0	IN	132	AUDATA1/PTF1/TO1
164	AUDATA1/PTF1/TO1	IN	131	AUDATA2/PTF2/TO2
163	AUDATA2/PTF2/TO2	IN	130	AUDATA3/PTF3/TO3
162	AUDATA3/PTF3/TO3	IN	129	NF/PTJ0
161	NF/PTJ0	IN	128	NF/PTJ1
160	NF/PTJ1	IN	127	NF/PTJ2
159	NF/PTJ2	IN	126	NF/PTJ3
158	NF/PTJ3	IN	125	NF/PTJ4
157	NF/PTJ4	IN	124	NF/PTJ5

116	$\overline{\text{ASEBRKAK}}$/PTF6	OUT	83	PTN2/XVDATA
115	$\overline{\text{CASU}}$/PTD3	Control	82	PTN3/TXDMNS
114	CKE/PTD4	Control	81	PTN4/TXDPLS
113	PTD5/NF	Control	80	PTN5/DMNS
112	$\overline{\text{BACK}}$/PTG5	Control	79	PTN6/DPLS
111	$\overline{\text{BREQ}}$/PTG6	Control	78	PTN7
110	$\overline{\text{WAIT}}$/PTG7	Control	77	TCLK/PTE6
109	DACK0/PTE0	Control	76	PTE7
108	DACK1/PTE1	Control	75	SCK0/SCPT1
107	TEND0/PTE3	Control	74	SCK2/SCPT3
106	AUDSYNC/PTF4	Control	73	$\overline{\text{RTS2}}$/SCPT4
105	AUDATA0/PTF0/TO0	Control	72	RXD0/SCPT0/IrRX
104	AUDATA1/PTF1/TO1	Control	71	RXD2/SCPT2
103	AUDATA2/PTF2/TO2	Control	70	$\overline{\text{CTS2}}$/SCPT5
102	AUDATA3/PTF3/TO3	Control	69	IRQ0/$\overline{\text{IRL0}}$/PTH0
101	NF/PTJ0	Control	68	IRQ1/$\overline{\text{IRL1}}$/PTH1
100	NAF/PTJ1	Control	67	IRQ2/$\overline{\text{IRL2}}$/PTH2
99	NF/PTJ2	Control	66	IRQ3/$\overline{\text{IRL3}}$/PTH3
98	NF/PTJ3	Control	65	IRQ4/PTH4
97	NF/PTJ4	Control	64	IRQ5/PTE2
96	NF/PTJ5	Control	63	AUDCK/PTG4
95	NF/PTJ6	Control	62	NMI
94	NF/PTJ7	Control	61	DREQ0/PTH5
93	NF/PTM4	Control	60	DREQ1/PTH6
92	PTM0	Control	59	MD3
91	PTM1	Control	58	MD4

50	PTN1/TXENL	OUT	20	PTN4/TXDPLS
49	PTN2/XVDATA	OUT	19	PTN5/DMNS
48	PTN3/TXDMNS	OUT	18	PTN6/DPLS
47	PTN4/TXDPLS	OUT	17	PTN7
46	PTN5/DMNS	OUT	16	TCLK/PTE6
45	PTN6/DPLS	OUT	15	PTE7
44	PTN7	OUT	14	TXD0/SCPT0/IrTX
43	TCLK/PTE6	OUT	13	SCK0/SCPT1
42	PTE7	OUT	12	TxD2/SCPT2
41	TXD0/SCPT0/IrTX	OUT	11	SCK2/SCPT3
40	SCK0/SCPT1	OUT	10	$\overline{\text{RTS2}}$/SCPT4
39	TXD2/SCPT2	OUT	9	$\overline{\text{CTS2}}$/SCPT5
38	SCK2/SCPT3	OUT	8	IRQ0/$\overline{\text{IRL0}}$/PTH0
37	$\overline{\text{RTS2}}$/SCPT4	OUT	7	IRQ1/$\overline{\text{IRL1}}$/PTH1
36	$\overline{\text{CTS2}}$/SCPT5	OUT	6	IRQ2/$\overline{\text{IRL2}}$/PTH2
35	IRQ0/$\overline{\text{IRL0}}$/PTH0	OUT	5	IRQ3/$\overline{\text{IRL3}}$/PTH3
34	IRQ1/$\overline{\text{IRL1}}$/PTH1	OUT	4	IRQ4/PTH4
33	IRQ2/$\overline{\text{IRL2}}$/PTH2	OUT	3	IRQ5/PTE2
32	IRQ3/$\overline{\text{IRL3}}$/PTH3	OUT	2	AUDCK/PTG4
31	IRQ4/PTH4	OUT	1	DREQ0/PTH5
30	IRQ5/PTE2	OUT	0	DREQ1/PTH6
29	AUDCK/PTG4	OUT		to TDO
28	DREQ0/PTH5	OUT		

Note: Control is an active-low signal.
When Control is driven low, the corresponding pin is driven by the value of OUT.

31 to 0	DID31 to DID0	Refer to description	R	Device ID31 to 0 Device ID register that is stipulat JTAG. H'001A200F (initial value LSI. Upper four bits may be cha chip version. SDIDH corresponds to bits 31 to SDIDL corresponds to bits 15 to

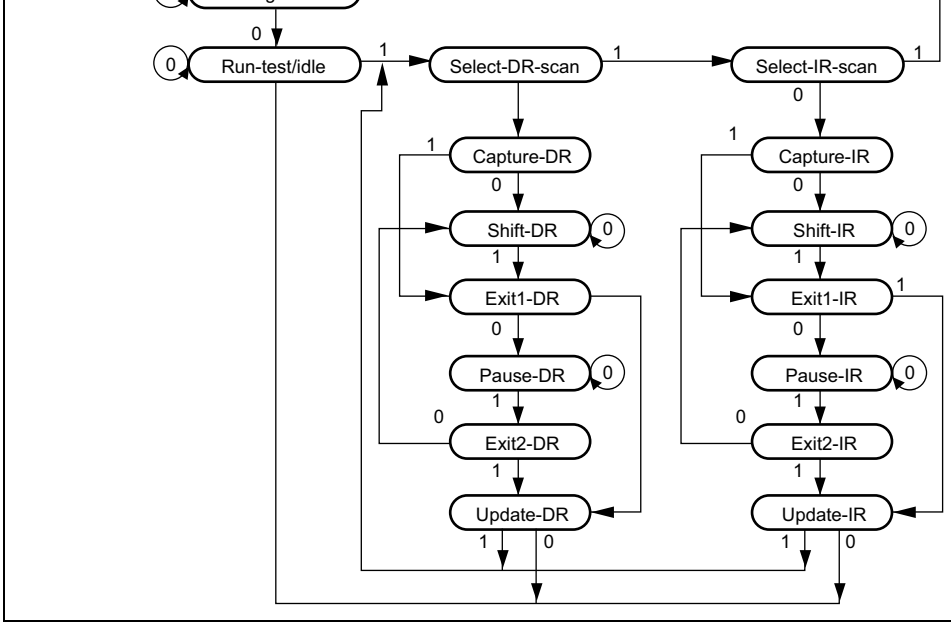


Figure 23.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI v
sampled at the rising edge of TCK; shifting occurs at the falling edge of TCK. I
on change timing of the TDO value, see section 23.4.3, TDO Output Timing. T
at high impedance, except with shift-DR and shift-IR states. During the change
0, there is a transition to test-logic-reset asynchronously with TCK.

L	L	L	Reset hold*2
		H	In ASE user mode*3: Normal rese In ASE break mode*3: $\overline{\text{RESETP}}$ a masked
	H	L	UDI reset only
		H	Normal operation

Notes: 1. Performs normal mode and ASE mode settings
$\overline{\text{ASEMD0}}$ = H, normal mode
$\overline{\text{ASEMD0}}$ = L, ASE mode
2. In ASE mode, reset hold is enabled by driving the $\overline{\text{RESETP}}$ and $\overline{\text{TRST}}$ pins constant cycle. In this state, the CPU does not start up, even if $\overline{\text{RESETP}}$ is When $\overline{\text{TRST}}$ is driven high, UDI operation is enabled, but the CPU does no The reset hold state is canceled by the following:
• Another $\overline{\text{RESETP}}$ assert (power-on reset)
• $\overline{\text{TRST}}$ reassert
3. ASE mode is classified into two modes; ASE break mode to execute the fir of an emulator and ASE user mode to execute the user program.
4. Make sure the $\overline{\text{TRST}}$ pin is low when the power is turned on.

23.4.3 TDO Output Timing

The timing of data output from the TDO is switched by the command type set in the S timing changes at the TCK falling edge when JTAG commands (EXTEST, CLAMP, I SAMPLE/PRELOAD, IDCODE, and BYPASS) are set. This is a timing of the JTAG When the UDI commands (UDI reset negate, UDI reset assert, and UDI interrupt) are output at the TCK rising edge earlier than the JTAG standard by a half cycle.

23.4.4 UDI Reset

An UDI reset is executed by setting an UDI reset assert command in SDIR. An UDI res
same kind as a power-on reset. An UDI reset is released by inputting an UDI reset nega
command. The required time between the UDI reset assert command and UDI reset neg
command is the same as time for keeping the $\overline{\text{RESETP}}$ pin low to apply a power-on res

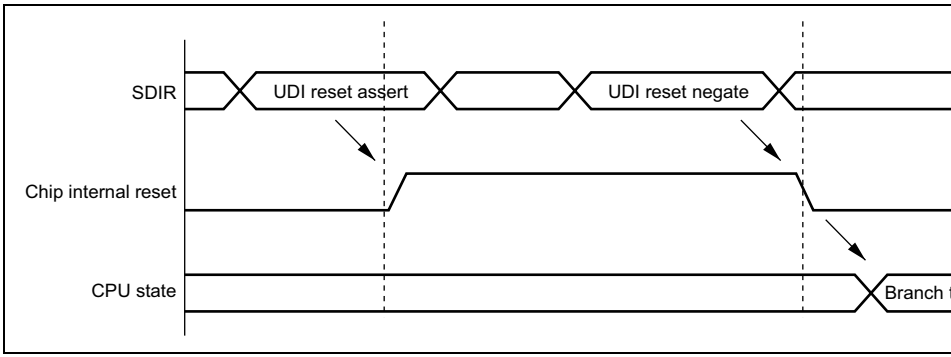


Figure 23.4 UDI Reset

23.4.5 UDI Interrupt

The UDI interrupt function generates an interrupt by setting a command from the UDI
SDIR. An UDI interrupt is a general exception/interrupt operation, resulting in a branch
address based on the VBR value plus offset, and with return by the RTE instruction. Th
request has a fixed priority level of 15.

UDI interrupts are accepted in sleep mode, but not in standby mode.

SAMPLE/PRELOAD, and EXTEST) and three option instructions (IDCODE, CLAM
HIGHZ).

1. BYPASS:

   The BYPASS instruction is an essential standard instruction that operates the bypa
   This instruction shortens the shift path to speed up serial data transfer involving ot
   the printed circuit board. While this instruction is executing, the test circuit has no
   system circuits. The upper four bits of the instruction code are 1111.

2. SAMPLE/PRELOAD:

   The SAMPLE/PRELOAD instruction inputs values from this LSI's internal circuit
   boundary scan register, outputs values from the scan path, and loads data onto the
   When this instruction is executing, this LSI's input pin signals are transmitted dire
   internal circuitry, and internal circuit values are directly output externally from the
   This LSI's system circuits are not affected by execution of this instruction. The up
   of the instruction code are 0100.

   In a SAMPLE operation, a snapshot of a value to be transferred from an input pin
   internal circuitry, or a value to be transferred from the internal circuitry to an outp
   latched into the boundary scan register and read from the scan path. Snapshot latch
   performed in synchronization with the rise of TCK in the Capture-DR state. Snaps
   does not affect normal operation of this LSI.

   In a PRELOAD operation, an initial value is set in the parallel output latch of the b
   scan register from the scan path prior to the EXTEST instruction. Without a PREL
   operation, when the EXTEST instruction was executed an undefined value would
   from the output pin until completion of the initial scan sequence (transfer to the ou
   (with the EXTEST instruction, the parallel output latch value is constantly output t
   pin).

The upper four bits of the instruction code are 0000.

4. IDCODE:

   A command can be set in SDIR by the UDI pins to place the UDI pins in the IDCO stipulated by JTAG. When the UDI is initialized ($\overline{\text{TRST}}$ is asserted or TAP is in the Logic-Reset state), the IDCODE mode is entered.

5. CLAMP, HIGHZ:

   A command can be set in SDIR by the UDI pins to place the UDI pins in the CLAM HIGHZ mode stipulated by JTAG.

23.5.2 Points for Attention

1. Boundary scan mode does not cover clock-related signals (EXTAL, EXTAL2, XTA XTAL2, EXTAL_USB, XTAL_USB, and CKIO).
2. Boundary scan mode does not cover reset-related signals ($\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, and
3. Boundary scan mode does not cover UDI-related signals (TCK, TDI, TDO, TMS, a
4. Fix the $\overline{\text{RESETP}}$ pin low during boundary scan.
5. Fix the CA pin high during boundary scan.
6. Fix the $\overline{\text{ASEMD0}}$ pin high during boundary scan.
7. The CKIO cock should operate during boundary scan. The MD[2:0] pin should be s clock mode used during normal operation, and EXTAL and CKIO should be set wit frequency range specified in the Clock Pulse Generator (CPG) section.

   As during normal operation, the boundary scan test should be performed after allow sufficient settling time for the crystal oscillator, PLL1, and PLL2.

3. The UDI is used for emulator connection. Therefore, UDI functions cannot be used
an emulator.

23.7 Advanced User Debugger (AUD)

The AUD is a function only for an emulator. For details on the AUD, refer to each em
User's Manual.

- Access to reserved addresses which are not described in this list is prohibited.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given, on th presumption of a big-endian system.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register A (by functional module, in order of the corresponding section numbers).
- Reserved bits are indicated by — in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a for holding data.
- When registers consist of 16 or 32 bits, bits are described from the MSB side. The order in which bytes are described is on the presumption of a big-endian syste

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (by funct module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the corresp section.
- The register states described are for the basic operating modes. If there is a specifi an on-chip module, refer to the section on that on-chip module.

access.

Register Name	Abbreviation	Number of Bits	Address	Modul
MMU control register	MMUCR	32	H'FFFF FFE0	MMU
Page table entry register high	PTEH	32	H'FFFF FFF0	
Page table entry register low	PTEL	32	H'FFFF FFF4	
Translation table base register	TTB	32	H'FFFF FFF8	
—	—	—	—	—
Cache control register 1	CCR1	32	H'FFFF FFEC	Cache
Cache control register 2	CCR2	32	H'A400 00B0	
Cache control register 3	CCR3	32	H'A400 00B4	
—	—	—	—	—
Interrupt event register 2	INTEVT2	32	H'A400 0000	Except
TRAPA exception register	TRA	32	H'FFFF FFD0	handli
Exception event register	EXPEVT	32	H'FFFF FFD4	
Interrupt event register	INTEVT	32	H'FFFF FFD8	
TLB exception address register	TEA	32	H'FFFF FFFC	
—	—	—	—	—
Interrupt priority level setting register A	IPRA	16	H'FFFF FEE2	INTC
Interrupt priority level setting register B	IPRB	16	H'FFFF FEE4	
Interrupt priority level setting register C	IPRC	16	H'A400 0016	
Interrupt priority level setting register D	IPRD	16	H'A400 0018	
Interrupt priority level setting register E	IPRE	16	H'A400 001A	
Interrupt priority level setting register F	IPRF	16	H'A408 0000	
Interrupt priority level setting register G	IPRG	16	H'A408 0002	
Interrupt priority level setting register H	IPRH	16	H'A408 0004	

PINT interrupt enable register	PINTER	16	H'A400 0014	
—	—	—	—	—
Common control register	CMNCR	32	H'A4FD 0000	BSC
Bus control register for CS0 space	CS0BCR	32	H'A4FD 0004	
Bus control register for CS2 space	CS2BCR	32	H'A4FD 0008	
Bus control register for CS3 space	CS3BCR	32	H'A4FD 000C	
Bus control register for CS4 space	CS4BCR	32	H'A4FD 0010	
Bus control register for CS5A space	CS5ABCR	32	H'A4FD 0014	
Bus control register for CS5B space	CS5BBCR	32	H'A4FD 0018	
Bus control register for CS6A space	CS6ABCR	32	H'A4FD 001C	
Bus control register for CS6B space	CS6BBCR	32	H'A4FD 0020	
Wait control register for CS0 space	CS0WCR	32	H'A4FD 0024	
Wait control register for CS2 space	CS2 WCR	32	H'A4FD 0028	
Wait control register for CS3 space	CS3 WCR	32	H'A4FD 002C	
Wait control register for CS4 space	CS4 WCR	32	H'A4FD 0030	
Wait control register for CS5A space	CS5A WCR	32	H'A4FD 0034	
Wait control register for CS5B space	CS5B WCR	32	H'A4FD 0038	
Wait control register for CS6A space	CS6A WCR	32	H'A4FD 003C	
Wait control register for CS6B space	CS6B WCR	32	H'A4FD 0040	
SDRAM control register	SDCR	32	H'A4FD 0044	
Refresh timer control/status register	RTCSR	32	H'A4FD 0048	
Refresh timer counter	RTCNT	32	H'A4FD 004C	
Refresh time constant register	RTCOR	32	H'A4FD 0050	
SDRAM mode register for CS2 space	SDMR2	—	H'A4FD 4xxx*2	
SDRAM mode register for CS3 space	SDMR3	—	H'A4FD 5xxx*2	
—	—	—	—	—

DMA transfer count register_1	DMATCR_1	32	H'A400 0038	
DMA channel control register_1	CHCR_1	32	H'A400 003C	
DMA source address register_2	SAR_2	32	H'A400 0040	
DMA destination address register_2	DAR_2	32	H'A400 0044	
DMA transfer count register_2	DMATCR_2	32	H'A400 0048	
DMA channel control register_2	CHCR_2	32	H'A400 004C	
DMA source address register_3	SAR_3	32	H'A400 0050	
DMA destination address register_3	DAR_3	32	H'A400 0054	
DMA transfer count register_3	DMATCR_3	32	H'A400 0058	
DMA channel control register_3	CHCR_3	32	H'A400 005C	
DMA operation register	DMAOR	16	H'A400 0060	
DMA extended resource selector 0	DMARS0	16	H'A409 0000	
DMA extended resource selector 1	DMARS1	16	H'A409 0004	
—	—	—	—	—
USB clock control register	UCLKCR	8	H'A40A 0008	CPG
Frequency control register	FRQCR	16	H'FFFF FF80	
—	—	—	—	—
Watchdog timer counter	WTCNT	8	H'FFFF FF84	WDT
Watchdog timer control/status register	WTCSR	8	H'FFFF FF86	
—	—	—	—	—
Standby control register	STBCR	8	H'FFFF FF82	Power- modes
Standby control register 2	STBCR2	8	H'FFFF FF88	
Standby control register 3	STBCR3	8	H'A40A 0000	
—	—	—	—	—

Timer control register_1	TCR_1	16	H'FFFF FEA8	
Timer constant register_2	TCOR_2	32	H'FFFF FEAC	
Timer counter_2	TCNT_2	32	H'FFFF FEB0	
Timer control register_2	TCR_2	16	H'FFFF FEB4	
Input capture register_2	TCPR_2	32	H'FFFF FEB8	
—	—	—	—	—
Compare match timer start register	CMSTR	16	H'A400 0070	CMT
Compare match timer control/status register	CMCSR	16	H'A400 0074	
Compare match timer counter	CMCNT	16	H'A400 0078	
Compare match timer constant register	CMCOR	16	H'A400 007C	
—	—	—	—	—
Timer start register	TSTR	16	H'A449 0000	TPU
Timer control register_0	TCR_0	16	H'A449 0010	
Timer mode register_0	TMDR_0	16	H'A449 0014	
Timer I/O control register_0	TIOR_0	16	H'A449 0018	
Timer interrupt enable register_0	TIER_0	16	H'A449 001C	
Timer status register_0	TSR_0	16	H'A449 0020	
Timer counter_0	TCNT_0	16	H'A449 0024	
Timer general register A_0	TGRA_0	16	H'A449 0028	
Timer general register B_0	TGRB_0	16	H'A449 002C	
Timer general register C_0	TGRC_0	16	H'A449 0030	
Timer general register D_0	TGRD_0	16	H'A449 0034	
Timer control register_1	TCR_1	16	H'A449 0050	
Timer mode register_1	TMDR_1	16	H'A449 0054	
Timer I/O control register_1	TIOR_1	16	H'A449 0058	

Timer general register D_1	TGRD_1	16	H'A449 0074	
Timer control register_2	TCR_2	16	H'A449 0090	
Timer mode register_2	TMDR_2	16	H'A449 0094	
Timer I/O control register_2	TIOR_2	16	H'A449 0098	
Timer interrupt enable register_2	TIER_2	16	H'A449 009C	
Timer status register_2	TSR_2	16	H'A449 00A0	
Timer counter_2	TCNT_2	16	H'A449 00A4	
Timer general register A_2	TGRA_2	16	H'A449 00A8	
Timer general register B_2	TGRB_2	16	H'A449 00AC	
Timer general register C_2	TGRC_2	16	H'A449 00B0	
Timer general register D_2	TGRD_2	16	H'A449 00B4	
Timer control register_3	TCR_3	16	H'A449 00D0	
Timer mode register_3	TMDR_3	16	H'A449 00D4	
Timer I/O control register_3	TIOR_3	16	H'A449 00D8	
Timer interrupt enable register_3	TIER_3	16	H'A449 00DC	
Timer status register_3	TSR_3	16	H'A449 00E0	
Timer counter_3	TCNT_3	16	H'A449 00E4	
Timer general register A_3	TGRA_3	16	H'A449 00E8	
Timer general register B_3	TGRB_3	16	H'A449 00EC	
Timer general register C_3	TGRC_3	16	H'A449 00F0	
Timer general register D_3	TGRD_3	16	H'A449 00F4	
—	—	—	—	—

Month counter	RMONCNT	8	H'FFFF FECC	
Year counter	RYRCNT	16	H'FFFF FECE	
Second alarm register	RSECAR	8	H'FFFF FED0	
Minute alarm register	RMINAR	8	H'FFFF FED2	
Hour alarm register	RHRAR	8	H'FFFF FED4	
Day of week alarm register	RWKAR	8	H'FFFF FED6	
Date alarm register	RDAYAR	8	H'FFFF FED8	
Month alarm register	RMONAR	8	H'FFFF FEDA	
RTC control register 1	RCR1	8	H'FFFF FEDC	
RTC control register 2	RCR2	8	H'FFFF FEDE	
Year alarm register	RYRAR	16	H'A413 FEE0	
RTC control register 3	RCR3	8	H'A413 FEE4	
—	—	—	—	—
Serial mode register_0	SCSMR_0	16	H'A440 0000	SCIF_0
Bit rate register_0	SCBRR_0	8	H'A440 0004	(Channe
Serial control register_0	SCSCR_0	16	H'A440 0008	
Transmit data stop register_0	SCTDSR_0	8	H'A440 000C	
FIFO error count register_0	SCFER_0	16	H'A440 0010	
Serial status register_0	SCSSR_0	16	H'A440 0014	
FIFO control register_0	SCFCR_0	16	H'A440 0018	
FIFO data count register_0	SCFDR_0	16	H'A440 001C	
Transmit FIFO data register_0	SCFTDR_0	8	H'A440 0020	
Receive FIFO data register_0	SCFRDR_0	8	H'A440 0024	
—	—	—	—	—

FIFO control register_2	SCFCR_2	16	H'A441 0018	
FIFO data count register_2	SCFDR_2	16	H'A441 001C	
Transmit FIFO data register_2	SCFTDR_2	8	H'A441 0020	
Receive FIFO data register_2	SCFRDR_2	8	H'A441 0024	
—	—	—	—	—
IrDA mode register	SCSMR_Ir	16	H'A44A 0000	IrDA
—	—	—	—	—
EP0i data register	EPDR0i	8B	H'A448 0000	USB
EP0o data register	EPDR0o	8B	H'A448 0004	
EP0s data register	EPDR0s	8B	H'A448 0008	
EP1 data register	EPDR1	128B	H'A448 000C	
EP2 data register	EPDR2	128B	H'A448 0010	
EP3 data register	EPDR3	8B	H'A448 0014	
Interrupt flag register 0	IFR0	8	H'A448 0018	
Interrupt flag register 1	IFR1	8	H'A448 001C	
Trigger register	TRG	8	H'A448 0020	
FIFO clear register	FCLR	8	H'A448 0024	
EP0o receive data size register	EPSZ0o	8	H'A448 0028	
Data status register	DASTS	8	H'A448 002C	
Endpoint stall register	EPSTL	8	H'A448 0030	
Interrupt enable register 0	IER0	8	H'A448 0034	
Interrupt enable register 1	IER1	8	H'A448 0038	
EP1 receive data size register	EPSZ1	8	H'A448 003C	
DMA transfer setting register	DMAR	8	H'A448 0040	
Interrupt select register 0	ISR0	8	H'A448 0044	
Interrupt select register 1	ISR1	8	H'A448 0048	
Transceiver control register	XVERCR	8	H'A448 0060	

Port E control register 2	PECR2	8	H'A405 0148	
Port F control register	PFCR	16	H'A400 010A	
Port F control register 2	PFCR2	8	H'A405 014A	
Port G control register	PGCR	16	H'A400 010C	
Port H control register	PHCR	16	H'A400 010E	
Port J control register	PJCR	16	H'A400 0110	
Port K control register	PKCR	16	H'A400 0112	
Port L control register	PLCR	16	H'A400 0114	
Port SC control register	SCPCR	16	H'A400 0116	
Port M control register	PMCR	16	H'A400 0118	
Port N control register	PNCR	16	H'A400 011A	
Port N control register 2	PNCR2	8	H'A405 015A	
—	—	—	—	—
Port A data register	PADR	8	H'A400 0120	Port
Port B data register	PBDR	8	H'A400 0122	
Port C data register	PCDR	8	H'A400 0124	
Port D data register	PDDR	8	H'A400 0126	
Port E data register	PEDR	8	H'A400 0128	
Port F data register	PFDR	8	H'A400 012A	
Port G data register	PGDR	8	H'A400 012C	
Port H data register	PHDR	8	H'A400 012E	
Port J data register	PJDR	8	H'A400 0130	
Port K data register	PKDR	8	H'A405 0132	
Port L data register	PLDR	8	H'A400 0134	
SC port data register	SCPDR	8	H'A400 0136	
Port M data register	PMDR	8	H'A400 0138	
Port N data register	PNDR	8	H'A400 013A	

—	—	—	—	—
Break data register B	BDRB	32	H'FFFF FF90	UBC
Break data mask register B	BDMRB	32	H'FFFF FF94	
Break control register	BRCR	32	H'FFFF FF98	
Execution count break register	BETR	16	H'FFFF FF9C	
Break address register B	BARB	32	H'FFFF FFA0	
Break address mask register B	BAMRB	32	H'FFFF FFA4	
Break bus cycle register B	BBRB	16	H'FFFF FFA8	
Branch source register	BRSR	32	H'FFFF FFAC	
Break address register A	BARA	32	H'FFFF FFB0	
Break address mask register A	BAMRA	32	H'FFFF FFB4	
Break bus cycle register A	BBRA	16	H'FFFF FFB8	
Branch destination register	BRDR	32	H'FFFF FFBC	
Break ASID register A	BASRA	8	H'FFFF FFE4	
Break ASID register B	BASRB	8	H'FFFF FFE8	
—	—	—	—	—
Instruction register	SDIR	16	H'A400 0200	UDI
ID register	SDID/SDIDH	16	H'A400 0214	
ID register	SDIDL	16	H'A400 0216	

Notes: 1. 8 bits when reading and 16 bits when writing.
2. The value of xxx depends on the setting value because of access control SDRAM mode register.

MMUCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	SV
	—	—	RC1	RC0	—	TF	IX	AT
PTEH	VPN	VPN	VPN	VPN	VPN	VPN	VPN	VPN
	VPN	VPN	VPN	VPN	VPN	VPN	VPN	VPN
	VPN	VPN	VPN	VPN	VPN	VPN	—	—
	ASID7	ASID6	ASID5	ASID4	ASID3	ASID2	ASID1	ASID0
PTEL	—	—	—	PPN	PPN	PPN	PPN	PPN
	PPN	PPN	PPN	PPN	PPN	PPN	PPN	PPN
	PPN	PPN	PPN	PPN	PPN	PPN	—	V
	—	PR1	PR0	SZ	C	D	SH	—
TTB								

	—	—	—	—	—	—	W3LOAD	W3L
	—	—	—	—	—	—	W2LOAD	W2L
CCR3	—	—	—	—	—	—	—	—
	CSIZE7	CSIZE6	CSIZE5	CSIZE4	CSIZE3	CSIZE2	CSIZE1	CSIZ
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
INTEVT2	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—				
TRA	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	imm	imm
	imm	imm	imm	imm	imm	imm	—	—
EXPEVT	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—				
INTEVT	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—				

IPRB	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0
IPRC	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0
IPRD	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0
IPRE	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0
IPRF	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0
IPRG	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0
IPRH	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0
ICR0	NMIL	—	—	—	—	—	—	NMIE
	—	—	—	—	—	—	—	—
ICR1	MAI	IRQLVL	BLMSK	—	IRQ51S	IRQ50S	IRQ41S	IRQ40
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00
ICR2	PINT15S	PINT14S	PINT13S	PINT12S	PINT11S	PINT10S	PINT9S	PINT8
	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0
IRR0	PINT0R	PINT1R	IRQ5R	IRQ4R	IRQ3R	IRQ2R	IRQ1R	IRQ0R
IRR1	TXI0R	—	RXI0R	ERI0R	DEI3R	DEI2R	DEI1R	DEI0R
IRR2	—	—	—	ADIR	TXI2R	—	RXI2R	ERI2R
PINTER	PINT15E	PINT14E	PINT13E	PINT12E	PINT11E	PINT10E	PINT9E	PINT8
	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0

	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS2BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRI
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS3BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRI
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS4BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRI
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS5ABCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRI
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS5BBCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRI
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS6ABCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRI
	—	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—

ROM)	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0
CS0WCR (burst ROM)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	BW1	BW0
	—	—	—	—	—	W3	W2	W1
	W0	WM	—	—	—	—	—	—
CS2 WCR (except SDRAM)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	WR3	WR2	WR1
	WR0	WM	—	—	—	—	—	—
CS2 WCR (SDRAM)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	A2CL1
	A2CL0	—	—	—	—	—	—	—
CS3 WCR (except SDRAM)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	WR3	WR2	WR1
	WR0	WM	—	—	—	—	—	—
CS3 WCR (SDRAM)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	TRP1	TRP0	—	TRCD1	TRCD0	—	A3CL1
	A3CL0	—	—	—	TRWL1	TRWL0	TRC1	TRC0
CS4 WCR (except burst ROM)	—	—	—	—	—	—	—	—
	—	—	—	—	—	WW2	WW1	WW0
	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0

	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0
CS5B WCR	—	—	—	—	—	—	—	—
	—	—	—	MPXW	—	WW2	WW1	WW0
	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0
CS6A WCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0
CS6B WCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0
SDCR	—	—	—	—	—	—	—	—
	—	—	—	A2ROW1	A2ROW0	—	A2COL1	A2C0
	—	—	—	SLOW	RFSH	RMODE	—	BACT
	—	—	—	A3ROW1	A3ROW0	—	A3COL1	A3C0
RTCSR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	CMF	CMIE	CKS2	CKS1	CKS0	RRC2	RRC1	RRC
RTCNT	—	—	—	—	—	—	—	—
RTCOR	—	—	—	—	—	—	—	—

DAR_0								
DMATCR_0	—	—	—	—	—	—	—	—
CHCR_0	—	—	—	—	—	—	—	—
	DO	TL	—	—	—	—	AM	AL
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	DL	DS	TB	TS1	TS0	IE	TE	DE
SAR_1								
DAR_1								
DMATCR_1	—	—	—	—	—	—	—	—

DAR_2								
DMATCR_2	—	—	—	—	—	—	—	—
CHCR_2	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	—	—	TB	TS1	TS0	IE	TE	DE
SAR_3								
DAR_3								
DMATCR_3	—	—	—	—	—	—	—	—

DMARS0	C1MID5	C1MID4	C1MID3	C1MID2	C1MID1	C1MID0	C1RID1	C1RID
	C0MID5	C0MID4	C0MID3	C0MID2	C0MID1	C0MID0	C0RID1	C0RID
DMARS1	C3MID5	C3MID4	C3MID3	C3MID2	C3MID1	C3MID0	C3RID1	C3RID
	C2MID5	C2MID4	C2MID3	C2MID2	C2MID1	C2MID0	C2RID1	C2RID
UCLKCR	USSCS1	USSCS0	USBEN	—	—	—	—	—
FRQCR	—	—	—	CKOEN	—	—	STC1	STC0
	—	—	IFC1	IFC0	—	—	PFC1	PFC0
WTCNT								
WTCSR	TME	WT/IT	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0
STBCR	STBY	—	—	STBXTL	—	MSTP2	MSTP1	—
STBCR2	MSTP10	MSTP9	MSTP8	—	MSTP6	MSTP5	—	—
STBCR3	MSTP37	—	MSTP35	MSTP34	MSTP33	MSTP32	MSTP31	MSTP
TSTR	—	—	—	—	—	STR2	STR1	STR0
TCOR_0								
TCNT_0								
TCR_0	—	—	—	—	—	—	—	UNF
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC

TCR_1	—	—	—	—	—	—	—	UNF
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPS
TCOR_2								
TCNT_2								
TCR_2	—	—	—	—	—	—	ICPF	UNF
	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPS
TCPR_2								
CMSTR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	STR
CMCSR	—	—	—	—	—	—	—	—
	CMF	—	—	CMR	—	—	CKS1	CKS
CMCNT								
CMCOR								

TIOR_0	—	—	—	—	—	—	—	—
	—	—	—	—	—	IOA2	IOA1	IOA0
TIER_0	—	—	—	—	—	—	—	—
	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
TSR_0	—	—	—	—	—	—	—	—
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_0								
TGRA_0								
TGRB_0								
TGRC_0								
TGRD_0								
TCR_1	—	—	—	—	—	—	—	—
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC
TMDR_1	—	—	—	—	—	—	—	—
	—	BFWT	BFB	BFA	—	MD2	MD1	MD0
TIOR_1	—	—	—	—	—	—	—	—
	—	—	—	—	—	IOA2	IOA1	IOA0
TIER_1	—	—	—	—	—	—	—	—
	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
TSR_1	—	—	—	—	—	—	—	—
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA

TGRC_1								
TGRD_1								
TCR_2	—	—	—	—	—	—	—	—
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPS
TMDR_2	—	—	—	—	—	—	—	—
	—	BFWT	BFB	BFA	—	MD2	MD1	MD0
TIOR_2	—	—	—	—	—	—	—	—
	—	—	—	—	—	IOA2	IOA1	IOA0
TIER_2	—	—	—	—	—	—	—	—
	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIE
TSR_2	—	—	—	—	—	—	—	—
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGF
TCNT_2								
TGRA_2								
TGRB_2								
TGRC_2								
TGRD_2								
TCR_3	—	—	—	—	—	—	—	—
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPS

TSR_3	—	—	—	—	—	—	—	—
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_3								
TGRA_3								
TGRB_3								
TGRC_3								
TGRD_3								
R64CNT	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
RSECCNT	—							
RMINCNT	—							
RHRCNT	—	—						
RWKCNT	—	—	—	—	—			
RDAYCNT	—	—						
RMONCNT	—	—	—					
RYRCNT								
RSECAR	ENB							
RMINAR	ENB							
RHRAR	ENB	—						
RWKAR	ENB	—	—	—	—			
RDAYAR	ENB	—						
RMONAR	ENB	—	—					

	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS
SCBRR_0	SCBRD7	SCBRD6	SCBRD5	SCBRD4	SCBRD3	SCBRD2	SCBRD1	SCB
SCSCR_0	—	—	—	—	TSIE	ERIE	BRIE	DRIE
	TIE	RIE	TE	RE	—	—	CKE1	CKE
SCTDSR_0								
SCFER_0	—	—	PER5	PER4	PER3	PER2	PER1	PER
	—	—	FER5	FER4	FER3	FER2	FER1	FER
SCSSR_0	—	—	—	—	—	—	ORER	TSF
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
SCFCR_0	TSE	TCRST	—	—	—	RSTRG2	RSTRG1	RST
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOO
SCFDR_0	—	T6	T5	T4	T3	T2	T1	T0
	—	R6	R5	R4	R3	R2	R1	R0
SCFTDR_0	SCFTD7	SCFTD6	SCFTD5	SCFTD4	SCFTD3	SCFTD2	SCFTD1	SCF
SCFRDR_0	SCFRD7	SCFRD6	SCFRD5	SCFRD4	SCFRD3	SCFRD2	SCFRD1	SCF
SCSMR_2	—	—	—	—	—	SRC2	SRC1	SRC
	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS
SCBRR_2	SCBRD7	SCBRD6	SCBRD5	SCBRD4	SCBRD3	SCBRD2	SCBRD1	SCB
SCSCR_2	—	—	—	—	TSIE	ERIE	BRIE	DRIE
	TIE	RIE	TE	RE	—	—	CKE1	CKE
SCTDSR_2								
SCFER_2	—	—	PER5	PER4	PER3	PER2	PER1	PER
	—	—	FER5	FER4	FER3	FER2	FER1	FER
SCSSR_2	—	—	—	—	—	—	ORER	TSF
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
SCFCR_2	TSE	TCRST	—	—	—	RSTRG2	RSTRG1	RST
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOO

EPDR0i	D7	D6	D5	D4	D3	D2	D1	D0
EPDR0o	D7	D6	D5	D4	D3	D2	D1	D0
EPDR0s	D7	D6	D5	D4	D3	D2	D1	D0
EPDR1	D7	D6	D5	D4	D3	D2	D1	D0
EPDR2	D7	D6	D5	D4	D3	D2	D1	D0
EPDR3	D7	D6	D5	D4	D3	D2	D1	D0
IFR0	BRST	EP1FULL	EP2TR	EP2EMPTY	SETUPTS	EP0oTS	EP0iTR	EP0i
IFR1	—	—	—	—	VBUSMN	EP3TR	EP3TS	VBUS
TRG	—	EP3PKTE	EP1RDFN	EP2PKTE	—	EP0sRDFN	EP0oRDFN	EP0iF
FCLR	—	EP3CLR	EP1CLR	EP2CLR	—	—	EP0oCLR	EP0i
EPSZ0o	D7	D6	D5	D4	D3	D2	D1	D0
DASTS	—	—	EP3DE	EP2DE	—	—	—	EP0i
EPSTL	—	—	—	—	EP3STL	EP2STL	EP1STL	EP0S
IER0	BRST	EP1FULL	EP2TR	EP2EMPTY	SETUPTS	EP0oTS	EP0iTR	EP0i
IER1	—	—	—	—	—	EP3TR	EP3TS	VBUS
EPSZ1	D7	D6	D5	D4	D3	D2	D1	D0
DMAR	—	—	—	—	—	—	EP2DMAE	EP1D
ISR0	BRST	EP1FULL	EP2TR	EP2EMPTY	SETUPTS	EP0oTS	EP0iTR	EP0i
ISR1	—	—	—	—	—	EP3TR	EP3TS	VBUS
XVERCR	—	—	—	—	—	—	—	XVER
PACR	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4
	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0
PBCR	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0
PCCR	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4
	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0

	PF3MD1	PF3MD0	PF2MD1	PF2MD0	PF1MD1	PF1MD0	PF0MD1	PF(
PFCR2	—	—	—	—	PF3MD2	PF2MD2	PF1MD2	PF(
PGCR	PG7MD1	PG7MD0	PG6MD1	PG6MD0	PG5MD1	PG5MD0	PG4MD1	PG
	PG3MD1	PG3MD0	PG2MD1	PG2MD0	PG1MD1	PG1MD0	PG0MD1	PG
PHCR	—	—	PH6MD1	PH6MD0	PH5MD1	PH5MD0	PH4MD1	PH
	PH3MD1	PH3MD0	PH2MD1	PH2MD0	PH1MD1	PH1MD0	PH0MD1	PH
PJCR	PJ7MD1	PJ7MD0	PJ6MD1	PJ6MD0	PJ5MD1	PJ5MD0	PJ4MD1	PJ4
	PJ3MD1	PJ3MD0	PJ2MD1	PJ2MD0	PJ1MD1	PJ1MD0	PJ0MD1	PJ(
PKCR	PK7MD1	PK7MD0	PK6MD1	PK6MD0	PK5MD1	PK5MD0	PK4MD1	PK
	PK3MD1	PK3MD0	PK2MD1	PK2MD0	PK1MD1	PK1MD0	PK0MD1	PK(
PLCR	—	—	—	—	—	—	—	—
	PL3MD1	PL3MD0	PL2MD1	PL2MD0	PL1MD1	PL1MD0	PL0MD1	PL(
SCPCR	—	—	—	—	SCP5MD1	SCP5MD0	SCP4MD1	SCF
	SCP3MD1	SCP3MD0	SCP2MD1	SCP2MD0	SCP1MD1	SCP1MD0	SCP0MD1	SCF
PMCR	—	—	PM6MD1	PM6MD0	—	—	PM4MD1	PM
	PM3MD1	PM3MD0	PM2MD1	PM2MD0	PM1MD1	PM1MD0	PM0MD1	PM
PNCR	PN7MD1	PN7MD0	PN6MD1	PN6MD0	PN5MD1	PN5MD0	PN4MD1	PN
	PN3MD1	PN3MD0	PN2MD1	PN2MD0	PN1MD1	PN1MD0	PN0MD1	PN
PNCR2	—	PN6MD2	PN5MD2	PN4MD2	PN3MD2	PN2MD2	PN1MD2	PN
PADR	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA(
PBDR	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB(
PCDR	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC
PDDR	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD
PEDR	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE(
PFDR	PF7DT	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF(
PGDR	PG7DT	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG

PNDR	PN7DT	PN6DT	PN5DT	PN4DT	PN3DT	PN2DT	PN1DT	PN0I
ADDRA								
			—	—	—	—	—	—
ADDRB								
			—	—	—	—	—	—
ADDRC								
			—	—	—	—	—	—
ADDRD								
			—	—	—	—	—	—
ADCSR	ADF	ADIE	ADST	DMASL	—	—	—	—
	CKS1	CKS0	MULTI1	MULTI0	—	—	CH1	CH0
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDM
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDM
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDM
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDM
BRCR	—	—	—	—	—	—	—	—
	—	—	BASMA	BASMB	—	—	—	—
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—
	DBEB	PCBB	—	—	SEQ	—	—	ETBI
BETR	—	—	—	—	BET11	BET10	BET9	BET8
	BET7	BET6	BET5	BET4	BET3	BET2	BET1	BET0

	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BA
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BA
BBRB	—	—	—	—	—	—	—	—
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZ
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BS
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BS
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BS
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BS
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BA
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BA
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BA
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BA
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BA
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BA
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BA
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BA
BBRA	—	—	—	—	—	—	—	—
	CDA1	CDA0	IDA1	IDA0	RWA1	RWA0	SZA1	SZ
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BD
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BD
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BD
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BD
BASRA	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA1	BA
BASRB	BASB7	BASB6	BASB5	BASB4	BASB3	BASB2	BASB1	BA

CCR1	Initialized	Initialized	Retained	Retained	Retained
CCR2	Initialized	Initialized	Retained	Retained	Retained
CCR3	Initialized	Initialized	Retained	Retained	Retained
INTEVT2	Undefined	Undefined	Retained	Retained	Retained
TRA	Undefined	Undefined	Retained	Retained	Retained
EXPEVT	Initialized*7	Initialized*7	Retained	Retained	Retained
INTEVT	Undefined	Undefined	Retained	Retained	Retained
TEA	Undefined	Undefined	Retained	Retained	Retained
IPRA	Initialized	Initialized	Retained	Retained	Retained
IPRB	Initialized	Initialized	Retained	Retained	Retained
IPRC	Initialized	Initialized	Retained	Retained	Retained
IPRD	Initialized	Initialized	Retained	Retained	Retained
IPRE	Initialized	Initialized	Retained	Retained	Retained
IPRF	Initialized	Initialized	Retained	Retained	Retained
IPRG	Initialized	Initialized	Retained	Retained	Retained
IPRH	Initialized	Initialized	Retained	Retained	Retained
ICR0	Initialized*8	Initialized*8	Retained	Retained	Retained
ICR1	Initialized	Initialized	Retained	Retained	Retained
ICR2	Initialized	Initialized	Retained	Retained	Retained
IRR0	Initialized	Initialized	Retained	Retained	Retained
IRR1	Initialized	Initialized	Retained	Retained	Retained
IRR2	Initialized	Initialized	Retained	Retained	Retained
PINTER	Initialized	Initialized	Retained	Retained	Retained
CMNCR	Initialized*9	Retained	Retained	Retained	Retained
CS0BCR	Initialized	Retained	Retained	Retained	Retained
CS2BCR	Initialized	Retained	Retained	Retained	Retained
CS3BCR	Initialized	Retained	Retained	Retained	Retained
CS4BCR	Initialized	Retained	Retained	Retained	Retained

CS3 WCR	Initialized	Retained	Retained	Retained	Retained	
CS4 WCR	Initialized	Retained	Retained	Retained	Retained	
CS5A WCR	Initialized	Re4ained	Retained	Retained	Retained	
CS5B WCR	Initialized	Retained	Retained	Retained	Retained	
CS6A WCR	Initialized	Retained	Retained	Retained	Retained	
CS6B WCR	Initialized	Retained	Retained	Retained	Retained	
SDCR	Initialized	Retained	Retained	Retained	Retained	
RTCSR	Initialized	Retained	Retained	Retained	Retained	
RTCNT	Initialized	Retained	Retained	Retained	Retained	
RTCOR	Initialized	Retained	Retained	Retained	Retained	
SDMR2	—	—	—	—	—	
SDMR3	—	—	—	—	—	
SAR_0	Undefined	Undefined	Retained	Retained	Retained	D
DAR_0	Undefined	Undefined	Retained	Retained	Retained	
DMATCR_0	Undefined	Undefined	Retained	Retained	Retained	
CHCR_0	Initialized	Initialized	Retained	Retained	Retained	
SAR_1	Undefined	Undefined	Retained	Retained	Retained	
DAR_1	Undefined	Undefined	Retained	Retained	Retained	
DMATCR_1	Undefined	Undefined	Retained	Retained	Retained	
CHCR_1	Initialized	Initialized	Retained	Retained	Retained	
SAR_2	Undefined	Undefined	Retained	Retained	Retained	
DAR_2	Undefined	Undefined	Retained	Retained	Retained	
DMATCR_2	Undefined	Undefined	Retained	Retained	Retained	
CHCR_2	Initialized	Initialized	Retained	Retained	Retained	
SAR_3	Undefined	Undefined	Retained	Retained	Retained	
DAR_3	Undefined	Undefined	Retained	Retained	Retained	

FRQCR	Initialized[*5]	Retained	Retained	Retained	Retained
WTCNT	Initialized[*5]	Retained	Retained	Retained	Retained
WTCSR	Initialized[*5]	Retained	Retained	Retained	Retained
STBCR	Initialized	Retained	Retained	Retained	Retained
STBCR2	Initialized	Retained	Retained	Retained	Retained
STBCR3	Initialized	Retained	Retained	Retained	Retained
TSTR	Initialized	Initialized	Initialized	Initialized	Retained
TCOR_0	Initialized	Initialized	Retained	Retained	Retained
TCNT_0	Initialized	Initialized	Retained	Retained	Retained
TCR_0	Initialized	Initialized	Retained	Retained	Retained
TCOR_1	Initialized	Initialized	Retained	Retained	Retained
TCNT_1	Initialized	Initialized	Retained	Retained	Retained
TCR_1	Initialized	Initialized	Retained	Retained	Retained
TCOR_2	Initialized	Initialized	Retained	Retained	Retained
TCNT_2	Initialized	Initialized	Retained	Retained	Retained
TCR_2	Initialized	Initialized	Retained	Retained	Retained
TCPR_2	Undefined	Undefined	Retained	Retained	Retained
CMSTR	Initialized	Initialized	Retained	Retained	Retained
CMCSR	Initialized	Initialized	Retained	Retained	Retained
CMCNT	Initialized	Initialized	Retained	Retained	Retained
CMCOR	Initialized	Initialized	Retained	Retained	Retained

TCNT_0	Initialized	Initialized	Retained	Retained	Retained
TGRA_0	Initialized	Initialized	Retained	Retained	Retained
TGRB_0	Initialized	Initialized	Retained	Retained	Retained
TGRC_0	Initialized	Initialized	Retained	Retained	Retained
TGRD_0	Initialized	Initialized	Retained	Retained	Retained
TCR_1	Initialized	Initialized	Retained	Retained	Retained
TMDR_1	Initialized	Initialized	Retained	Retained	Retained
TIOR_1	Initialized	Initialized	Retained	Retained	Retained
TIER_1	Initialized	Initialized	Retained	Retained	Retained
TSR_1	Initialized	Initialized	Retained	Retained	Retained
TCNT_1	Initialized	Initialized	Retained	Retained	Retained
TGRA_1	Initialized	Initialized	Retained	Retained	Retained
TGRB_1	Initialized	Initialized	Retained	Retained	Retained
TGRC_1	Initialized	Initialized	Retained	Retained	Retained
TGRD_1	Initialized	Initialized	Retained	Retained	Retained
TCR_2	Initialized	Initialized	Retained	Retained	Retained
TMDR_2	Initialized	Initialized	Retained	Retained	Retained
TIOR_2	Initialized	Initialized	Retained	Retained	Retained
TIER_2	Initialized	Initialized	Retained	Retained	Retained
TSR_2	Initialized	Initialized	Retained	Retained	Retained
TCNT_2	Initialized	Initialized	Retained	Retained	Retained
TGRA_2	Initialized	Initialized	Retained	Retained	Retained
TGRB_2	Initialized	Initialized	Retained	Retained	Retained
TGRC_2	Initialized	Initialized	Retained	Retained	Retained
TGRD_2	Initialized	Initialized	Retained	Retained	Retained

TGRA_3	Initialized	Initialized	Retained	Retained	Retained
TGRB_3	Initialized	Initialized	Retained	Retained	Retained
TGRC_3	Initialized	Initialized	Retained	Retained	Retained
TGRD_3	Initialized	Initialized	Retained	Retained	Retained
R64CNT	Operation continued	Operation continued	Operation continued	Retained	Operation continued
RSECCNT	Operation continued	Operation continued	Operation continued	Retained	Operation continued
RMINCNT	Operation continued	Operation continued	Operation continued	Retained	Operation continued
RHRCNT	Operation continued	Operation continued	Operation continued	Retained	Operation continued
RWKCNT	Operation continued	Operation continued	Operation continued	Retained	Operation continued
RDAYCNT	Operation continued	Operation continued	Operation continued	Retained	Operation continued
RMONCNT	Operation continued	Operation continued	Operation continued	Retained	Operation continued
RYRCNT	Operation continued	Operation continued	Operation continued	Retained	Operation continued
RSECAR	Retained*1	Retained	Retained	Retained	Retained
RMINAR	Retained*1	Retained	Retained	Retained	Retained
RHRAR	Retained*1	Retained	Retained	Retained	Retained
RWKAR	Retained*1	Retained	Retained	Retained	Retained
RDAYAR	Retained*1	Retained	Retained	Retained	Retained
RMONAR	Retained*1	Retained	Retained	Retained	Retained
RCR1	Initialized*2	Initialized*2	Retained	Retained	Retained
RCR2	Initialized*10	Initialized*10	Retained	Retained	Retained
RYRAR	Retained	Retained	Retained	Retained	Retained
RCR3	Initialized	Retained	Retained	Retained	Retained

SCFCR_0	Initialized	Initialized	Retained	Retained	Retained	
SCFDR_0	Initialized	Initialized	Retained	Retained	Retained	
SCFTDR_0	Undefined	Undefined	Retained	Retained	Retained	
SCFRDR_0	Undefined	Undefined	Retained	Retained	Retained	
SCSMR_2	Initialized	Initialized	Retained	Retained	Retained	S
SCBRR_2	Initialized	Initialized	Retained	Retained	Retained	
SCSCR_2	Initialized	Initialized	Retained	Retained	Retained	
SCTDSR_2	Initialized	Initialized	Retained	Retained	Retained	
SCFER_2	Initialized	Initialized	Retained	Retained	Retained	
SCSSR_2	Initialized	Initialized	Retained	Retained	Retained	
SCFCR_2	Initialized	Initialized	Retained	Retained	Retained	
SCFDR_2	Initialized	Initialized	Retained	Retained	Retained	
SCFTDR_2	Undefined	Undefined	Retained	Retained	Retained	
SCFRDR_2	Undefined	Undefined	Retained	Retained	Retained	
SCSMR_Ir	Initialized	Initialized	Retained	Retained	Retained	I
EPDR0i	Undefined	Undefined	Retained	Retained	Retained	U
EPDR0o	Undefined	Undefined	Retained	Retained	Retained	
EPDR0s	Undefined	Undefined	Retained	Retained	Retained	
EPDR1	Undefined	Undefined	Retained	Retained	Retained	
EPDR2	Undefined	Undefined	Retained	Retained	Retained	
EPDR3	Undefined	Undefined	Retained	Retained	Retained	
IFR0	Initialized	Initialized	Retained	Retained	Retained	
IFR1	Initialized	Initialized	Retained	Retained	Retained	
TRG	Undefined	Undefined	Retained	Retained	Retained	
FCLR	Undefined	Undefined	Retained	Retained	Retained	
EPSZ0o	Initialized	Initialized	Retained	Retained	Retained	
DASTS	Initialized	Initialized	Retained	Retained	Retained	

ISR1	Initialized	Initialized	Retained	Retained	Retained
XVERCR	Initialized	Initialized	Retained	Retained	Retained
PACR	Initialized	Retained	Retained	Retained	Retained
PBCR	Initialized	Retained	Retained	Retained	Retained
PCCR	Initialized	Retained	Retained	Retained	Retained
PDCR	Initialized	Retained	Retained	Retained	Retained
PECR	Initialized	Retained	Retained	Retained	Retained
PECR2	Initialized	Retained	Retained	Retained	Retained
PFCR	Initialized	Retained	Retained	Retained	Retained
PFCR2	Initialized	Retained	Retained	Retained	Retained
PGCR	Initialized	Retained	Retained	Retained	Retained
PHCR	Initialized	Retained	Retained	Retained	Retained
PJCR	Initialized	Retained	Retained	Retained	Retained
PKCR	Initialized	Retained	Retained	Retained	Retained
PLCR	Initialized	Retained	Retained	Retained	Retained
SCPCR	Initialized	Retained	Retained	Retained	Retained
PMCR	Initialized	Retained	Retained	Retained	Retained
PNCR	Initialized	Retained	Retained	Retained	Retained
PNCR2	Initialized	Retained	Retained	Retained	Retained
PADR	Initialized	Retained	Retained	Retained	Retained
PBDR	Initialized	Retained	Retained	Retained	Retained
PCDR	Initialized	Retained	Retained	Retained	Retained
PDDR	Initialized	Retained	Retained	Retained	Retained
PEDR	Initialized	Retained	Retained	Retained	Retained
PFDR	Initialized	Retained	Retained	Retained	Retained
PGDR	Initialized	Retained	Retained	Retained	Retained

PNDR	Initialized	Retained	Retained	Retained	Retained
ADDRA	Initialized	Initialized	Initialized	Initialized	Retained
ADDRB	Initialized	Initialized	Initialized	Initialized	Retained
ADDRC	Initialized	Initialized	Initialized	Initialized	Retained
ADDRD	Initialized	Initialized	Initialized	Initialized	Retained
ADCSR	Initialized	Initialized	Initialized	Initialized	Retained
BDRB	Initialized	Retained	Retained	Retained	Retained
BDMRB	Initialized	Retained	Retained	Retained	Retained
BRCR	Initialized	Retained	Retained	Retained	Retained
BETR	Initialized	Retained	Retained	Retained	Retained
BARB	Initialized	Retained	Retained	Retained	Retained
BAMRB	Initialized	Retained	Retained	Retained	Retained
BBRB	Initialized	Retained	Retained	Retained	Retained
BRSR	Initialized*3	Retained	Retained	Retained	Retained
BARA	Initialized	Retained	Retained	Retained	Retained
BAMRA	Initialized	Retained	Retained	Retained	Retained
BBRA	Initialized	Retained	Retained	Retained	Retained
BRDR	Initialized*3	Retained	Retained	Retained	Retained
BASRA	Undefined	Retained	Retained	Retained	Retained
BASRB	Undefined	Retained	Retained	Retained	Retained
SDIR	Retained	Retained	Retained	Retained	Retained
SDID/SDIDH*4	—	—	—	—	—
SDIDL*4	—	—	—	—	—

Notes: 1. The ENB bit is initialized. Other bits are retained.
2. The CF bit is undefined.
3. Although the flag is initialized, other bits are not initialized (except the res bits).
4. Values of these resisters are fixed.
5. These registers are not initialized by a power-on reset caused by the WD

Power supply voltage (I/O)	$V_{CC}Q$ V_{CC}-RTC	–0.3 to 4.2	V
Power supply voltage (internal)	V_{CC} V_{CC}-PLL1 V_{CC}-PLL2	–0.3 to 2.1	V
Input voltage (except port L)	V_{in}	–0.3 to $V_{CC}Q$ + 0.3	V
Input voltage (port L)	V_{in}	–0.3 to AV_{CC} + 0.3	V
Analog power supply voltage (AD)	AV_{CC}	–0.3 to 4.2	V
Analog input voltage (AD)	V_{AN}	–0.3 to AV_{CC} + 0.3	V
Analog power supply voltage (USB)	V_{CC}-USB	–0.3 to 4.2	V
Analog input voltage (USB)	V_{IN}	–0.3 to (V_{CC}-USB) + 0.3	V
Operating temperature	T_{opr}	–20 to 75	°C
Storage temperature	T_{stg}	–55 to 125	°C

Caution:

- Operating the chip in excess of the absolute maximum rating may result in perman
- Order of turning on 1.5 V power (Vcc, Vcc-PLL1, Vcc-PLL2) and 3.3 V power (V RTC, AVcc, Vcc-USB):
  1. The 3.3 V power and the 1.5 V power should be turned on simultaneously or th power should be tuned on first. When the 3.3 V is turned on first, turn on the 1 within 1 ms. It is recommended that this interval will be as short as possible.
  2. Until voltage is applied to all power supplies and a low level is input at the $\overline{RE}$ internal circuits remain unsettled, and so pin states are also undefined. The syst must ensure that these undefined states do not cause erroneous system operatio
  3. When the power is turned on, make sure that the voltage of the 1.5 V power is that of the 3.3 V power.

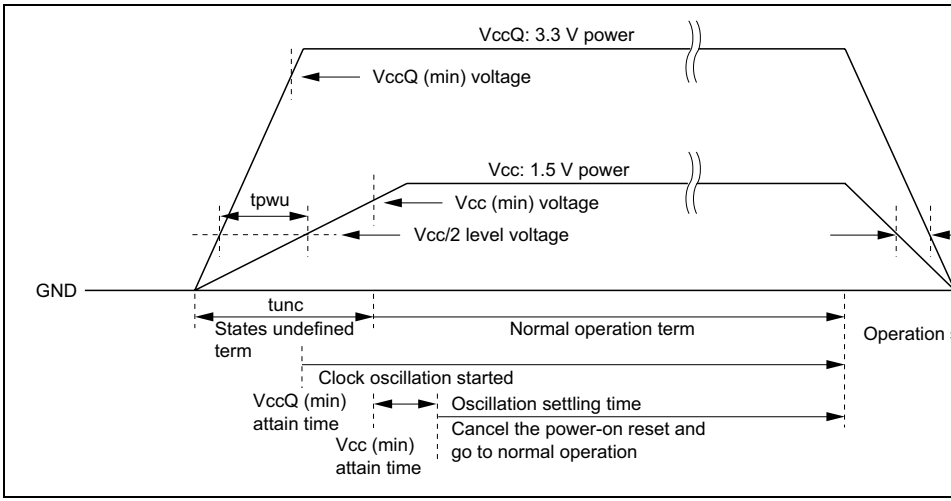


Figure 25.1 Power On/Off Sequence

Recommended Power On/Off Times

Item	Symbol	Max. Permitted Value	U
VccQ to Vcc power-on time interval	tpwu	1	r
VccQ to Vcc power-off time interval	tpwd	1	r
State undefined term	tunc	10	r

The recommended times shown above do not require strict settings.

The state undefined term indicates that pins are at the power rising stage. The pin state at VccQ (min) attain time. However, a power-on reset ($\overline{\text{RESETP}}$) is accepted successfu after VccQ (min) attain time and clock oscillation settling time. Set the state undefined than 10 ms.

Power supply voltage		$V_{CC}Q$, V_{CC}-RTC	3.0	3.3	3.6	V	
		V_{CC}, V_{CC}-PLL1 V_{CC}-PLL2	1.4	1.5	1.6		
Current consumption	Normal operation	I_{CC}	—	133	200	mA	V_{CC} =
							Iϕ = 13
			—	105	150		V_{CC} =
							Iϕ = 10
		$I_{CC}Q$	—	20	40		$V_{CC}Q$ =
							Bϕ = 3
	In sleep mode*	I_{CC}	—	25	40	mA	Bϕ = 3
		$I_{CC}Q$	—	10	20		
	In standby mode	I_{CC}	—	150	500	µA	T_a = 2
		$I_{CC}Q$	—	10	30		on)
							$V_{CC}Q$ =
							V_{CC} =
		I_{CC}	—	150	500	µA	T_a = 2
		$I_{CC}Q$	—	10	30		off)
							$V_{CC}Q$ =
							V_{CC} =
Input leak current	All input pins	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0
							$V_{CC}Q$ =
Three-state leak current	I/O, all output pins (off condition)	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	Vin = 0
							$V_{CC}Q$ =
Pull-up resistance	Port pin	R_{pull}	30	60	120	kΩ	

	PTM3 to PTM0					
Analog power-supply voltage (AD)		AV_{CC}	3.0	3.3	3.6	V
Analog power-supply voltage (USB)		V_{CC}-USB	3.0	3.3	3.6	V
Analog power-supply current (AD)	During A/D conversion	AI_{CC}	—	0.8	2	mA
	Idle		—	0.01	5.0	µA

Note: * No external bus cycles except refresh cycles.

	MD6 to MD0, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$, EXTAL, CKIO, CA					
	EXTAL2		—	—	—	Wh is co the res pin co the (pu
	Port L		2.0	—	AV_{CC} + 0.3	
	Other input pins		2.0	—	$V_{CC}Q$ + 0.3	
Input low voltage	$\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI IRQ5 to IRQ0, PINT15 to PINT0, RXD0, MD6 to MD0, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$, EXTAL, CKIO, CA	V_{IL}	–0.3	—	$V_{CC}Q \times 0.1$	V
	EXTAL2		—	—	—	Wh is co the res pin co the (pu
	Port L		–0.3	—	$AV_{CC} \times 0.2$	

Output low voltage	All output pins	V_{OL}			0.55	V	V_{CC} I_{OL} =

Notes: 1. Even when the RTC is not used, power must be supplied between V_{CC}-RTC RTC.

2. AV_{CC} must satisfy the condition: $V_{CC}Q - 0.2\ V \le AV_{CC} \le V_{CC}Q + 0.2\ V$. Do no AV_{CC} and AV_{SS} pins open if the A/D converter is not used; connect AV_{CC} to V AV_{SS} to $V_{SS}Q$.

3. Current consumption values are for V_{IH}min = $V_{CC}Q - 0.5$ V and V_{IL}max = 0.5 output pins unloaded.

Table 25.2 DC Characteristics (2-b) [USB-Related Pins*]

(Condition: Ta = –

Item	Symbol	Min	Typ	Max	Unit	Mea Con
Power supply voltage	$V_{CC}Q$	3.0	3.3	3.6	V	
Input high voltage	V_{IH}	2.0	—	$V_{CC}Q + 0.3$	V	
Input low voltage	V_{IL}	–0.3	—	$V_{CC}Q \times 0.2$	V	
Input high voltage (EXTAL_USB)	V_{IH} (EXTAL_USB)	$V_{CC}Q - 0.3$	—	$V_{CC}Q + 0.3$	V	
Input low voltage (EXTAL_USB)	V_{IL} (EXTAL_USB)	–0.3	—	$V_{CC}Q \times 0.2$	V	
Output high voltage	V_{OH}	2.4	—	—	V	V_{CC} I_{OH} =
		2.0	—	—		V_{CC} I_{OH} =
Output low voltage	V_{OL}	—	—	0.55	V	V_{CC} I_{OL} =

Note: * XVDATA, DPLS, DMNS, TXDPLS, TXDMNS, TXENL, VBUS, SUSPND, and EX pins

Single ended receiver threshold voltage	V_{SE}	0.8	—	2.0	V	
Output high voltage	V_{OH}	2.8	—	V_{CC}-USB	V	
Output low voltage	V_{OL}	—	—	0.3	V	
Tri-state leak current	I_{LO}	–10	—	10	µA	0V

Notes: 1. D+ and D– pins

2. V_{CC}-USB must satisfy the condition: $V_{CC}Q \leq V_{CC}$-USB. Even when the USB power must be supplied between V_{CC}-USB and V_{SS}-USB.

Table 25.3 Permitted Output Current Values

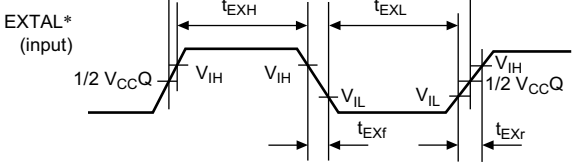
(Conditions: $V_{CC}Q = V_{CC}$-RTC = V_{CC}-USB = 3.0 to 3.6 V, $V_{CC} = V_{CC}$-PLL1 = V_{CC}-
1.6 V, AV_{CC} = 3.0 to 3.6 V, $V_{SS}Q = V_{SS} = V_{SS}$-RTC = V_{SS}-USB = V_{SS}-PLL1 =
AV_{SS} = 0 V, T_a =

Item	Symbol	Min	Typ	Ma
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0
Output low-level permissible current (total)	$\sum I_{OL}$	—	—	12
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0
Output high-level permissible current (total)	$\sum (-I_{OH})$	—	—	40

Caution: To ensure LSI reliability, do not exceed the value for output current given in ta

Item		Symbol	Min	Typ	Max	Unit	R
Operating frequency	CPU, cache (Iϕ)	f	20	—	133.34	MHz	13 pr
					100		10 pr
	External bus (Bϕ)		20	—	66.67		
	Peripheral module (Pϕ)		5	—	33.34		

EXTAL clock input low pulse width	t_{EXL}	1.5	—	ns	
EXTAL clock input high pulse width	t_{EXH}	1.5	—	ns	
EXTAL clock input rise time	t_{EXr}	—	6	ns	
EXTAL clock input fall time	t_{EXf}	—	6	ns	
CKIO clock input frequency	f_{CKI}	20	66.67	MHz	25.
CKIO clock input cycle time	t_{CKICYC}	15	50	ns	
CKIO clock input low pulse width	t_{CKIL}	3	—	ns	
CKIO clock input high pulse width	t_{CKIH}	3	—	ns	
CKIO clock input rise time	t_{CKIR}	—	4	ns	
CKIO clock input fall time	t_{CKIF}	—	4	ns	
CKIO clock output frequency	t_{OP}	20	66.67	MHz	25.
CKIO clock output cycle time	t_{cyc}	15	50	ns	
CKIO clock output low pulse width	t_{CKOL}	3	—	ns	
CKIO clock output high pulse width	t_{CKOH}	3	—	ns	
CKIO clock output rise time	t_{CKOr}	—	5	ns	
CKIO clock output fall time	t_{CKOf}	—	5	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	25.
$\overline{\text{RESETP}}$ setup time	t_{RESPS}	20	—	ns	25.
$\overline{\text{RESETP}}$ assert time	t_{RESPW}	20	—	t_{cyc}	25.
$\overline{\text{RESETM}}$ assert time	t_{RESMW}	20	—	t_{cyc}	25.
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	25.
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	25.
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	25.



Note: * The clock input from the EXTAL pin.

Figure 25.2 EXTAL Clock Input Timing

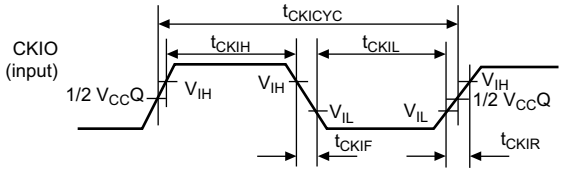


Figure 25.3 CKIO Clock Input Timing

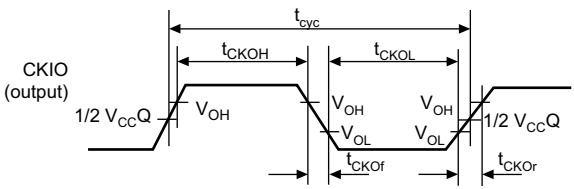


Figure 25.4 CKIO Clock Output Timing

Figure 25.5 Power-On Oscillation Settling Time

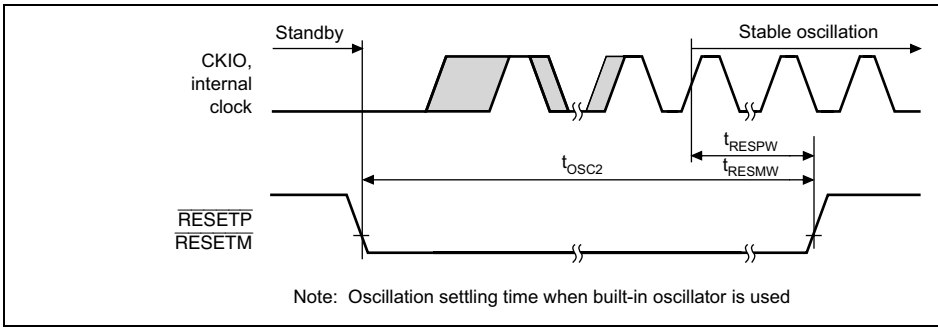


Figure 25.6 Oscillation Settling Time at Standby Return (Return by Re

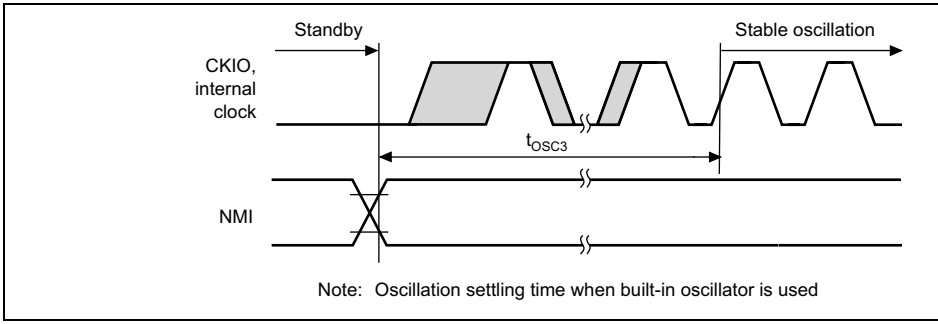


Figure 25.7 Oscillation Settling Time at Standby Return (Return by NM

Figure 25.8 Oscillation Settling Time at Standby Return (Return by IRQ5 to IRQ0, PINT15 to PINT0, and $\overline{IRL3}$ to $\overline{IRL0}$)

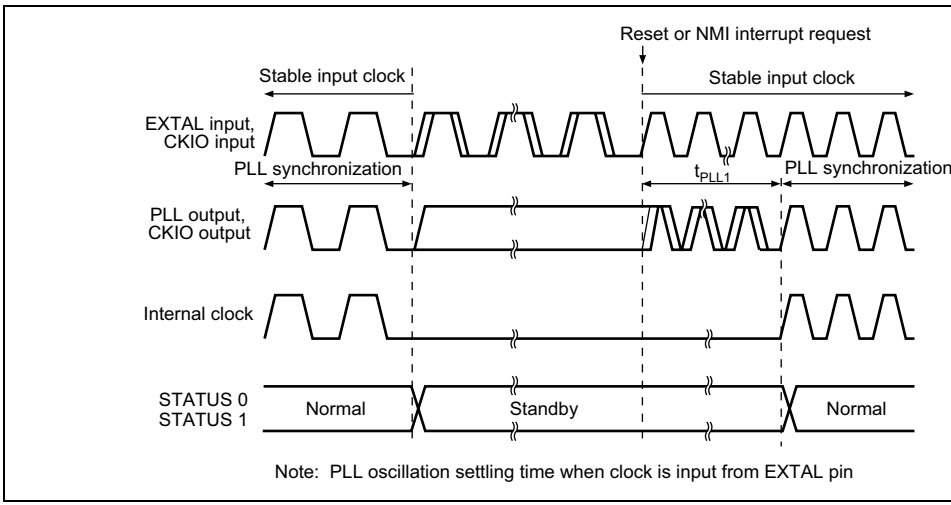
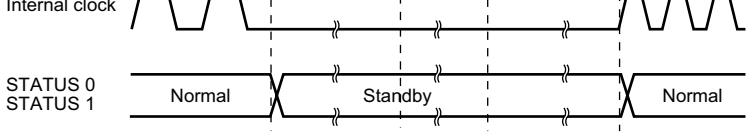
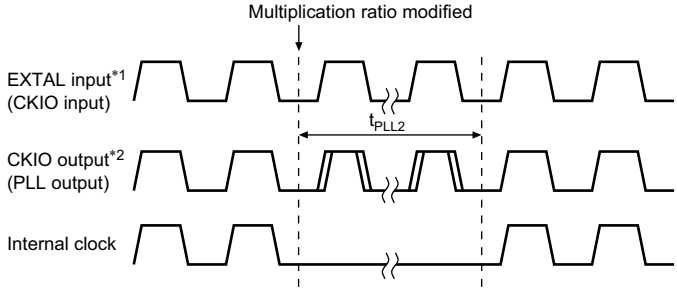


Figure 25.9 PLL Synchronization Settling Time by Reset or NMI



Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin in oscillation continuous mode.

Figure 25.10 PLL Synchronization Settling Time by IRQ/IRL, PINT Inte



Notes: 1. CKIO input in clock mode 7
2. PLL output except in clock mode 7

Figure 25.11 PLL Synchronization Settling Time when Frequency Multipl Ratio Modified

$\overline{\text{RESETP}}$ pulse width	t_{RESPW}	20	—	t_{cyc}
$\overline{\text{RESETP}}$ setup time*1	t_{RESPS}	20	—	ns
$\overline{\text{RESETM}}$ pulse width	t_{RESMW}	20*4	—	t_{cyc}
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	10	—	ns
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	1/2 t_{cyc}+10	—	ns
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	1/2 t_{cyc}+3	—	ns
NMI setup time*1	t_{NMIS}	10	—	ns
NMI hold time	t_{NMIH}	3	—	ns
IRQ5 to IRQ0 setup time*1	t_{IRQS}	10	—	ns
IRQ5 to IRQ0 hold time	t_{IRQH}	3	—	ns
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	1/2 t_{cyc}+13	ns
STATUS1, STATUS0 delay time	t_{STD}	—	18	ns
Bus tri-state delay time 1	t_{BOFF1}	0	30	ns
Bus tri-state delay time 2	t_{BOFF2}	0	30	ns
Bus buffer-on time 1	t_{BON1}	0	30	ns
Bus buffer-on time 2	t_{BON2}	0	30	ns

Notes: t_{cyc} is the external bus clock cycle (B clock cycle).

1. $\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI, and IRQ5 to IRQ0 are asynchronous. Changes ar at the clock rise when the setup shown is kept. When the setup cannot be ke detection can be delayed until the next clock rises.
2. The upper limit of the external bus clock is 66.67 MHz.
3. In standby mode, $t_{RESPW} = t_{OSC2}$ (10 ms). When the crystal oscillation continu clock multiplication ratio is changed in standby mode, $t_{RESPW} = t_{PLL1}$ (100 µs)
4. In standby mode, $t_{RESMW} = t_{OSC2}$ (10 ms). When the crystal oscillation continu clock multiplication ratio is changed in standby mode, $\overline{\text{RESETM}}$ must be kep STATUS (0-1) changes to reset (HH).

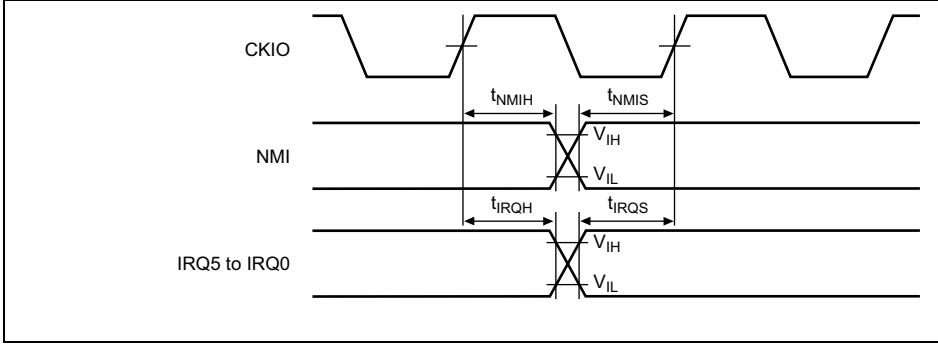


Figure 25.13 Interrupt Signal Input Timing

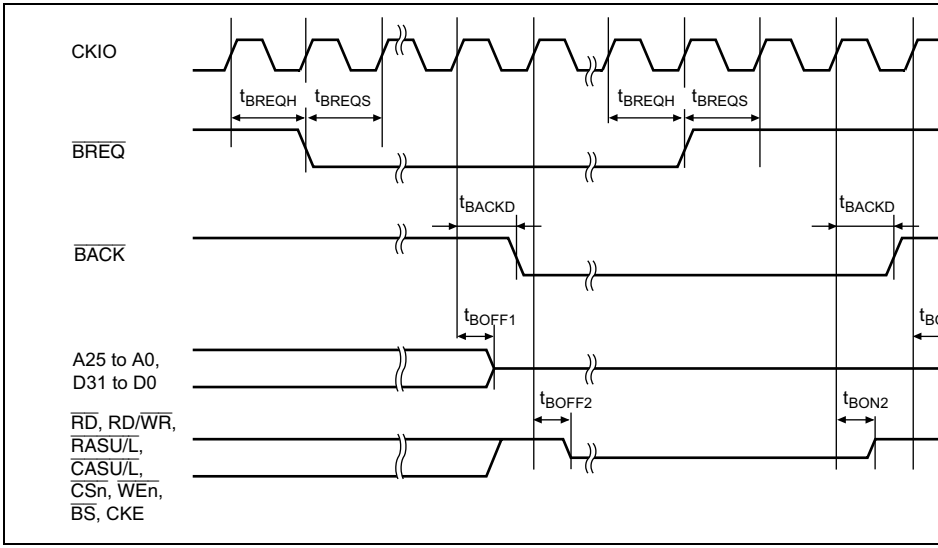


Figure 25.14 Bus Release Timing

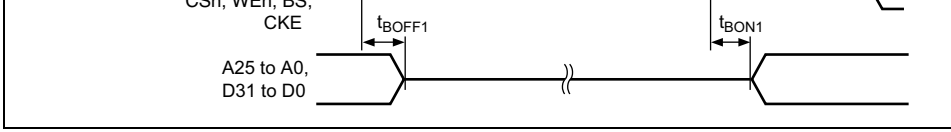


Figure 25.15 Pin Drive Timing at Standby

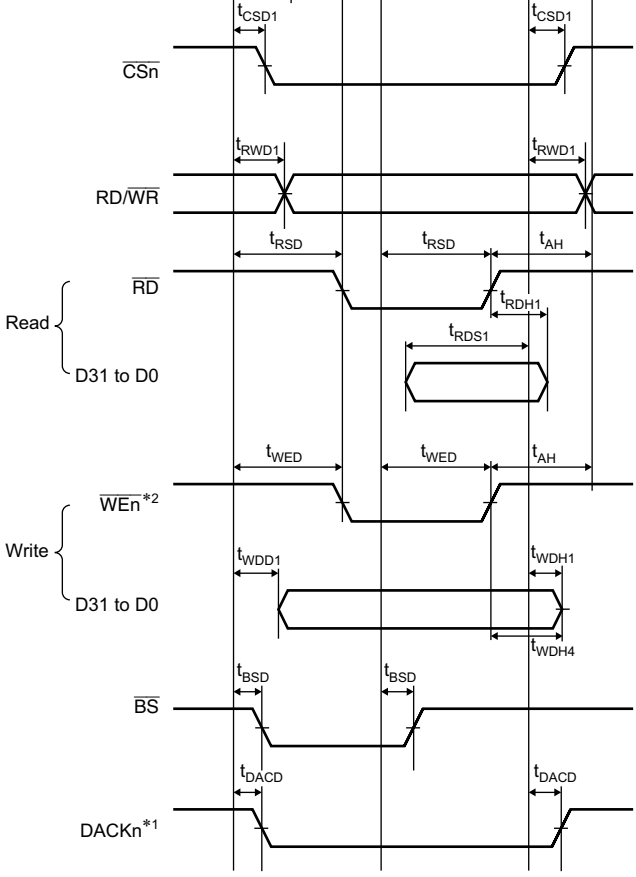
25.3.3 AC Bus Timing

Table 25.7 Bus Timing (1)

(Conditions: $V_{CC}Q$ = V_{CC}-RTC = V_{CC}-USB = 3.0 to 3.6 V, V_{CC} = V_{CC}-PLL1 = V_{CC}-P
1.6 V, AV_{CC} = 3.0 to 3.6 V, $V_{SS}Q$ = V_{SS} = V_{SS}-RTC = V_{SS}-USB = V_{SS}-PLL1 =
AV_{SS} = 0 V, T_a = –20 to 75°C, Clock mode 0/

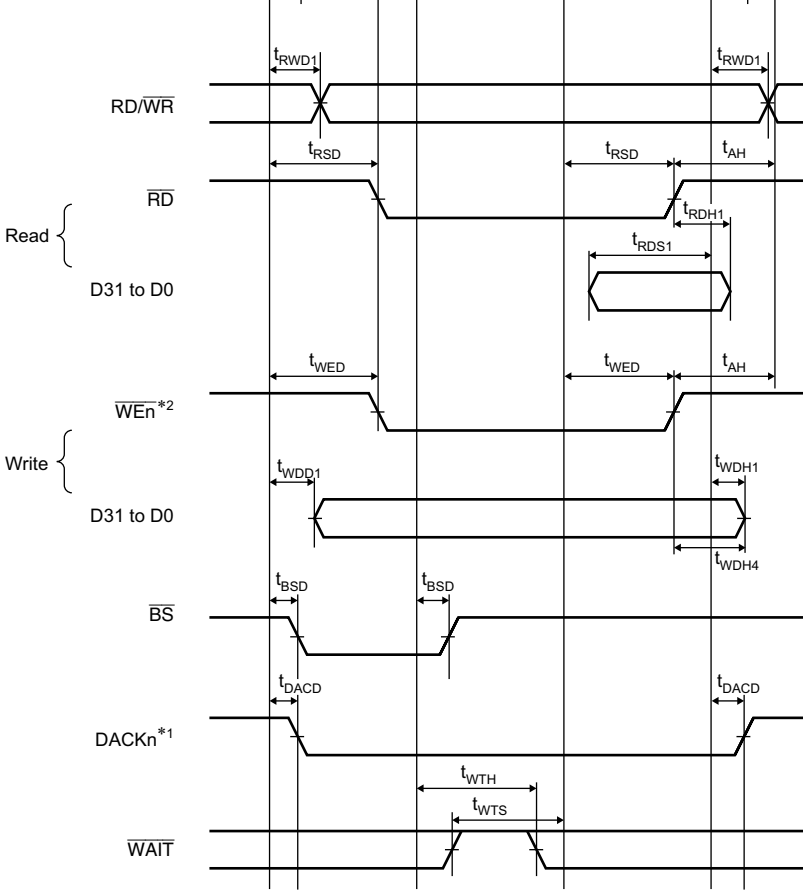
		66.67 MHz			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
Address delay time 1	t_{AD1}	1	12	ns	25.16 to 25.38
Address delay time 2	t_{AD2}	—	1/2 t_{cyc}+12	ns	25.21
Address setup time	t_{AS}	0	—	ns	25.16 to 25.19
Address hold time	t_{AH}	0	—	ns	25.16 to 25.19
$\overline{BS}$ delay time	t_{BSD}	—	10	ns	25.16 to 25.35
$\overline{CS}$ delay time 1	t_{CSD1}	1	10	ns	25.16 to 25.38
Read/write delay time 1	t_{RWD1}	1	10	ns	25.16 to 25.38
Read strobe delay time	t_{RSD}	—	1/2 t_{cyc}+10	ns	25.16 to 25.21
Read data setup time 1	t_{RDS1}	1/2 t_{cyc}+6	—	ns	25.16 to 25.20
Read data setup time 2	t_{RDS2}	6	—	ns	25.22 to 25.25, 25
Read data setup time 3	t_{RDS3}	1/2 t_{cyc}+6	—	ns	25.21
Read data hold time 1	t_{RDH1}	0	—	ns	25.16 to 25.20
Read data hold time 2	t_{RDH2}	2	—	ns	25.22 to 25.25, 25
Read data hold time 3	t_{RDH3}	0	—	ns	25.21

$\overline{WAIT}$ setup time	t_{WTS}	1/2 t_{cyc}+6	—	ns	25.17 to 25.21
$\overline{WAIT}$ hold time	t_{WTH}	1/2 t_{cyc}+2	—	ns	25.17 to 25.21
$\overline{RAS}$ delay time 1	t_{RASD1}	1	10	ns	25.22 to 25.38
$\overline{CAS}$ delay time 1	t_{CASD1}	1	10	ns	25.22 to 25.38
DQM delay time 1	t_{DQMD1}	1	10	ns	25.22 to 25.35
CKE delay time 1	t_{CKED1}	1	10	ns	25.37
$\overline{AH}$ delay time	t_{AHD}	1/2 t_{cyc}	1/2 t_{cyc}+10	ns	25.20
Multiplex address delay time	t_{MAD}	—	12	ns	25.20
Multiplex address hold time	t_{MAH}	0	—	ns	25.20
DACK delay time	t_{DACD}	—	10	ns	25.16 to 25.35



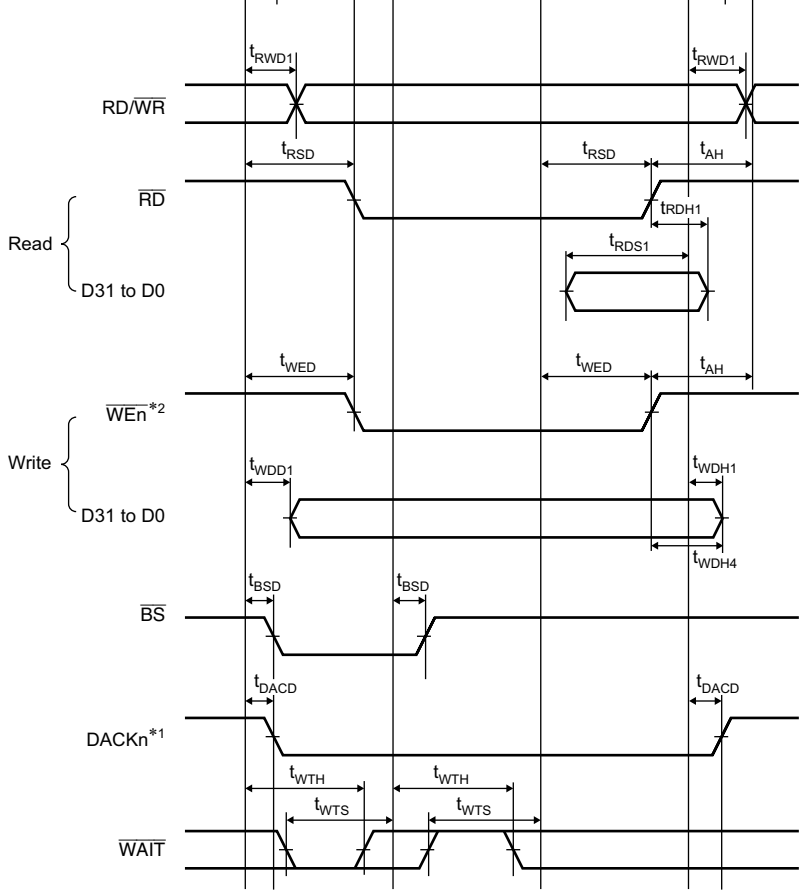
Notes: 1. DACKn is a waveform when active-low is specified.
2. Output timing is the same when reading byte-selection SRAM.

Figure 25.16 Basic Bus Cycle (No Wait)



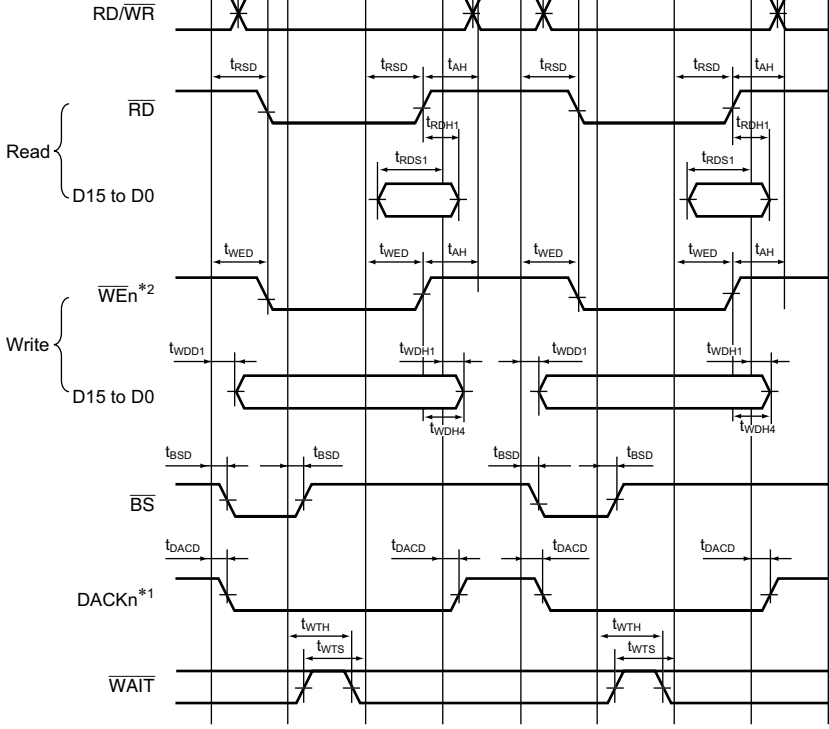
Notes: 1. DACKn is a waveform when active-low is specified.
2. Output timing is the same when reading byte-selection SRAM.

Figure 25.17 Basic Bus Cycle (One Software Wait)



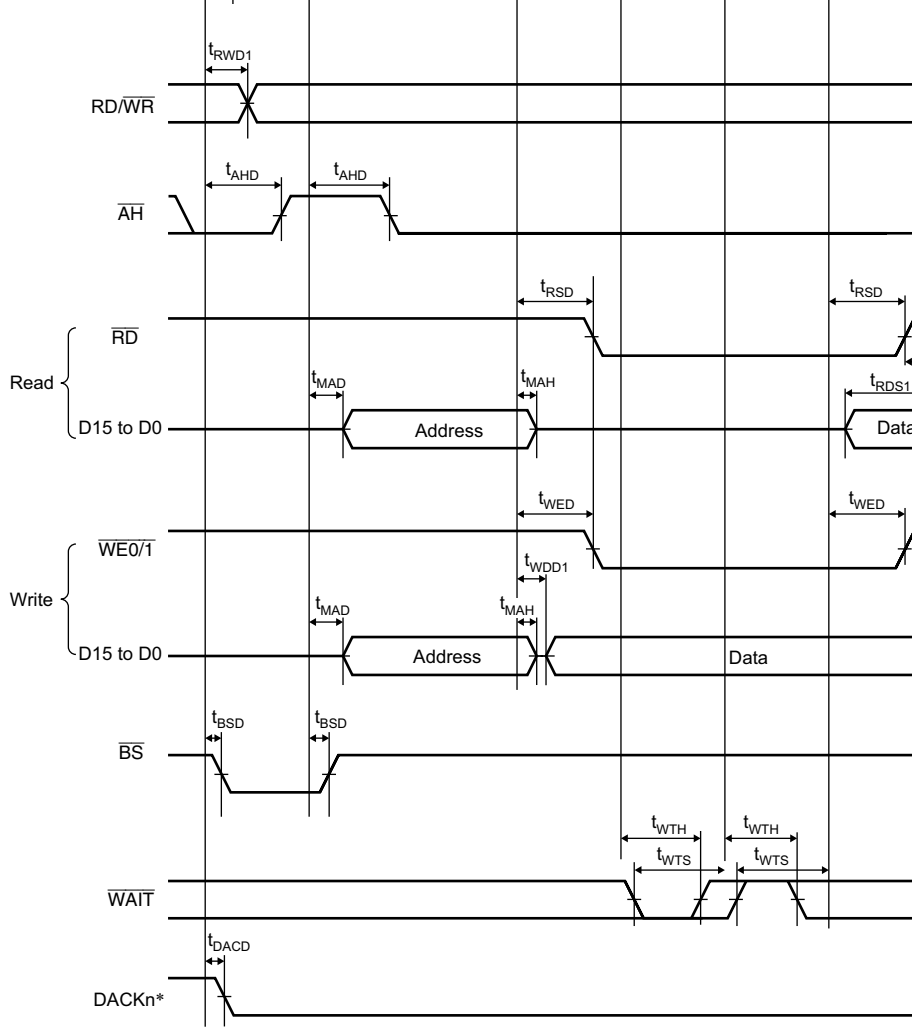
Notes: 1. DACKn is a waveform when active-low is specified.
2. Output timing is the same when reading byte-selection SRAM.

Figure 25.18 Basic Bus Cycle (One External Wait)



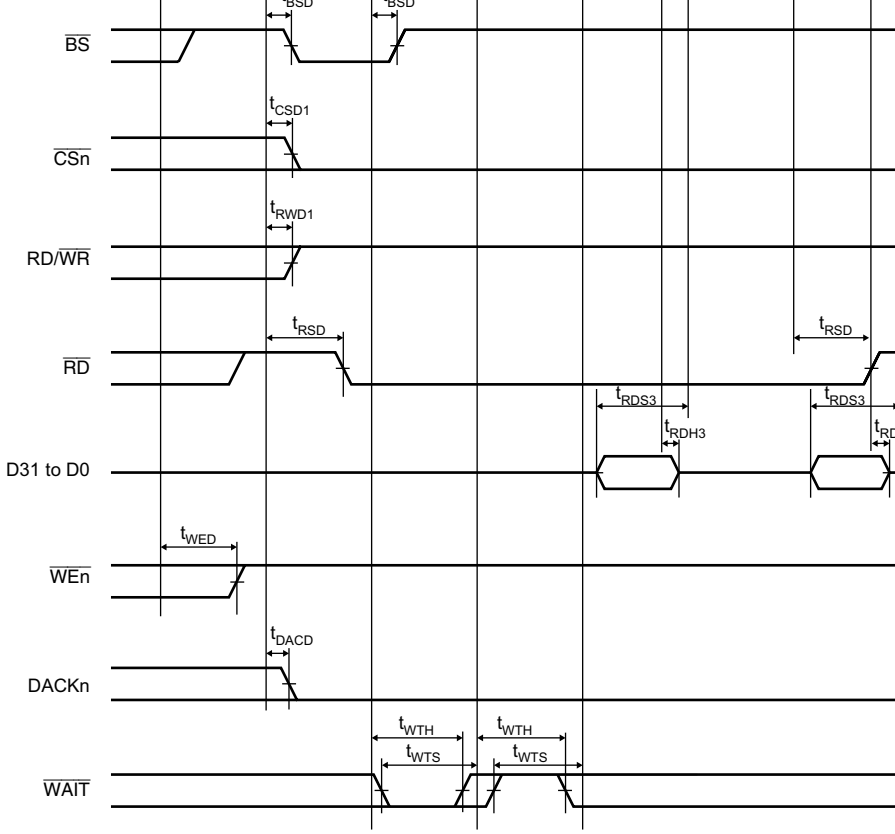
Notes: 1. DACKn is a waveform when active-low is specified.
2. Output timing is the same when reading byte-selection SRAM.

Figure 25.19 Basic Bus Cycle (One Software Wait, External Wait Enab (WM Bit = 0), No Idle Cycle Setting)



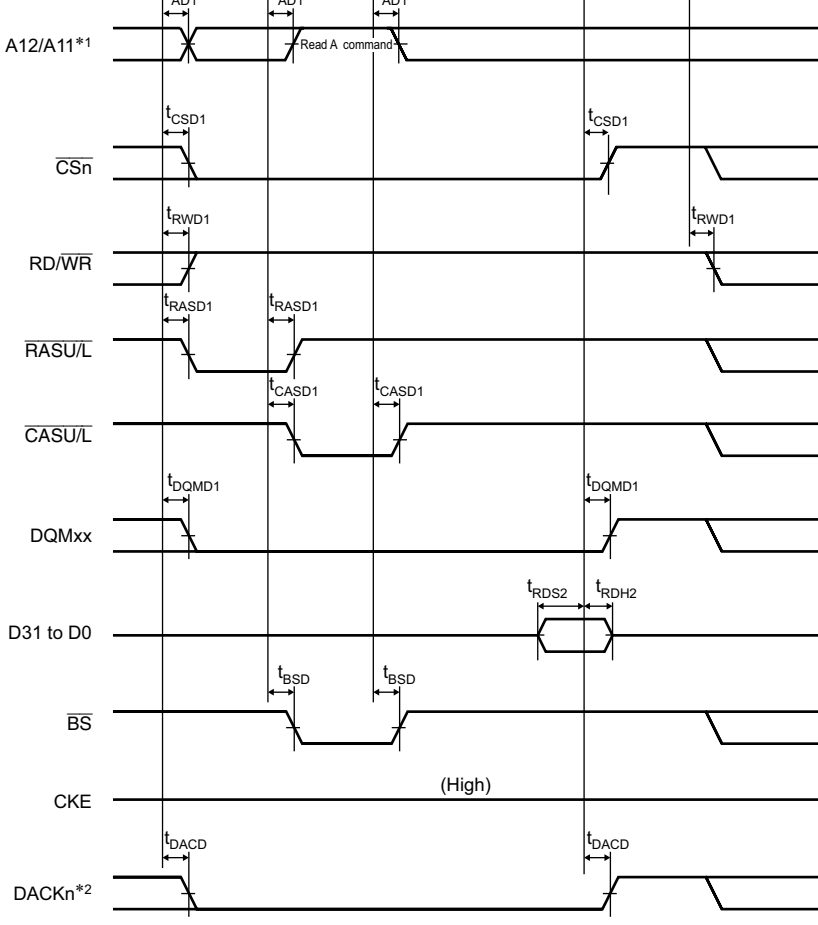
Note: * DACKn is a waveform when active-low is specified.

Figure 25.20 Address/Data Multiplex I/O Bus Cycle (Three Address Cycles, One Software Wait, One External Wait)



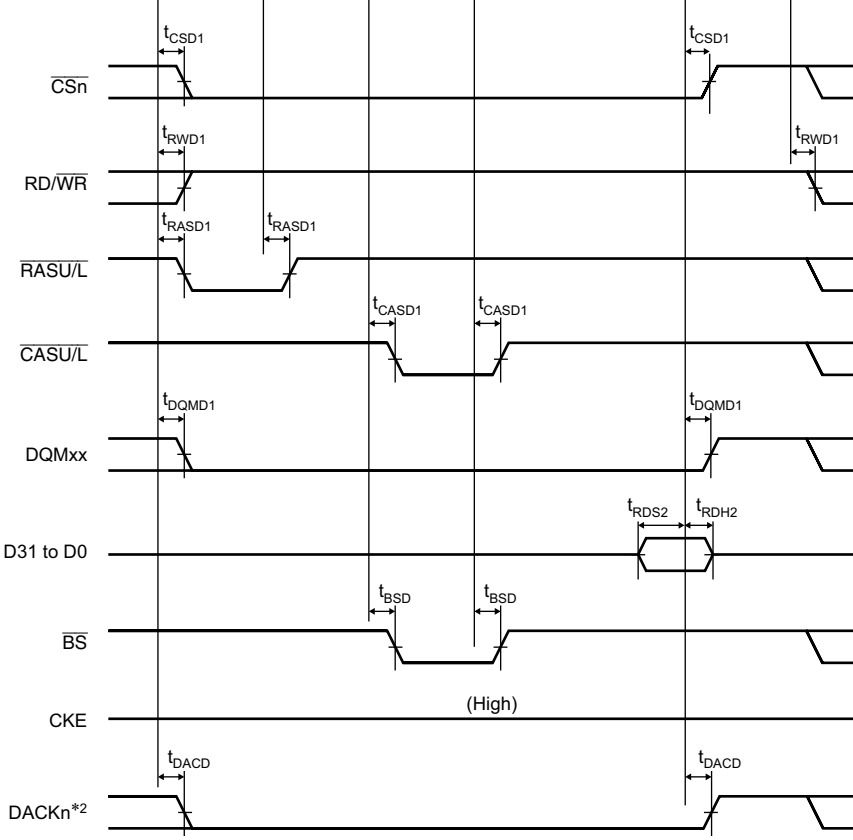
Notes: 1. t_{RDH3} is specified by earlier one of change of A25 to A0 or the $\overline{RD}$ rising edge.
2. DACKn is a waveform when active-low is specified.

Figure 25.21 Burst ROM Read Cycle (One Access Wait, One External W
One Burst Wait, Two Bursts)



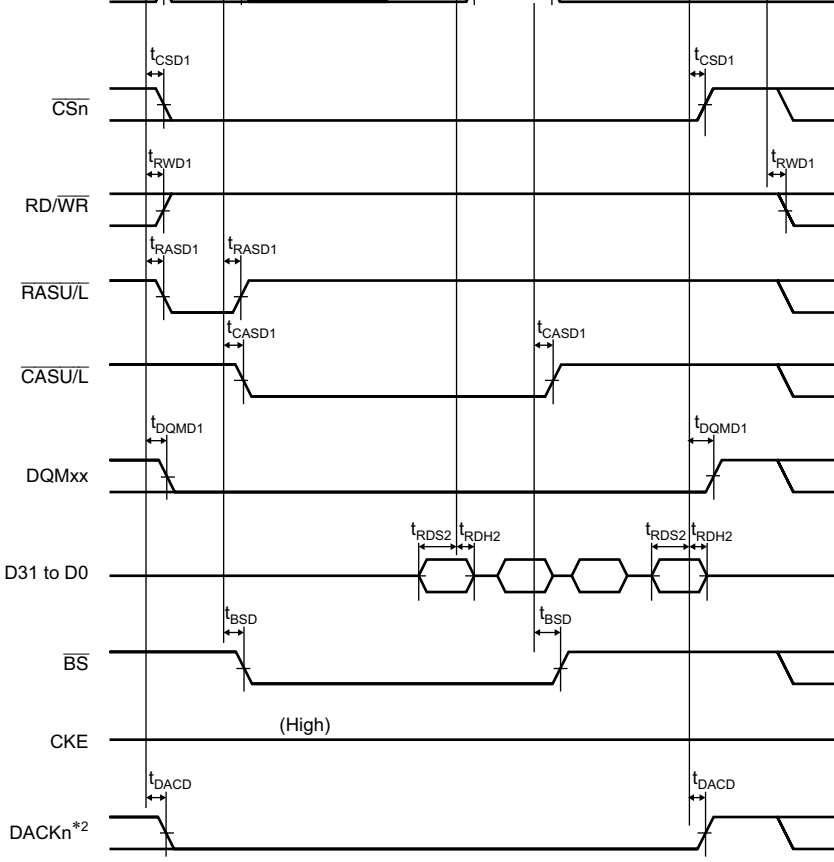
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.22 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency = 2, TRCD = 1 Cycle, TRP = 1 Cycle)



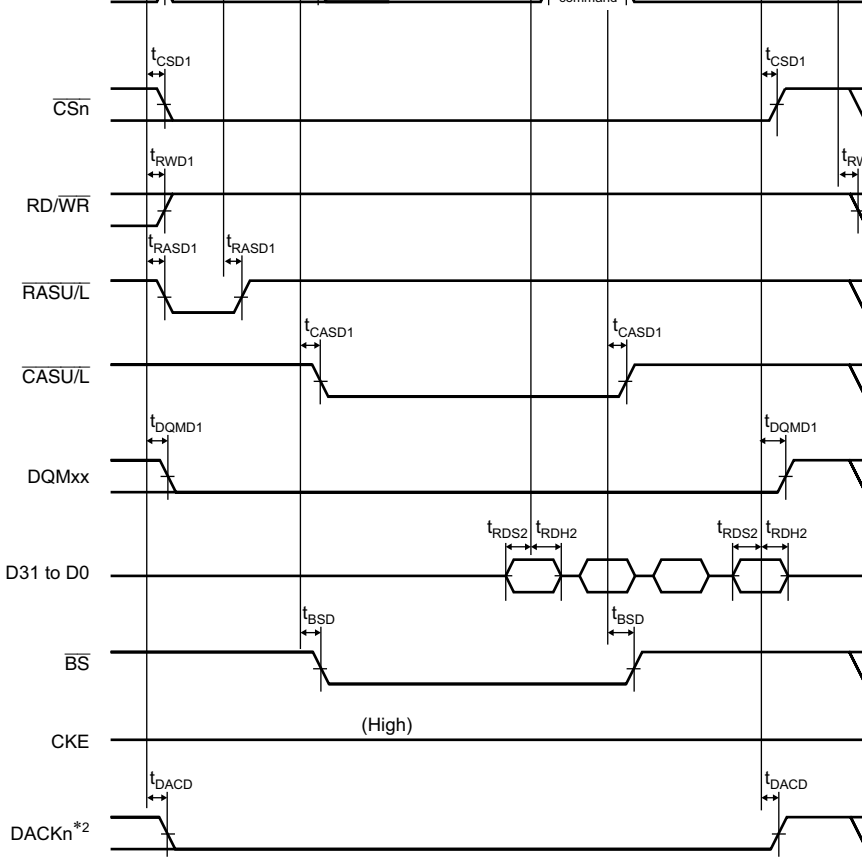
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.23 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency = 2, TRCD = 2 Cycle, TRP = 2 Cycle)



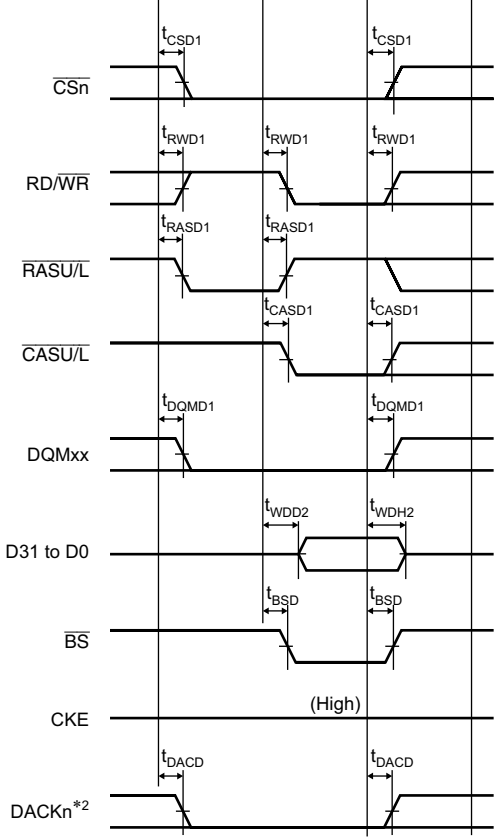
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.24 Synchronous DRAM Burst Read Bus Cycle (Single Read × (Auto Precharge, CAS Latency = 2, TRCD = 1 Cycle, TRP = 2 Cycle)



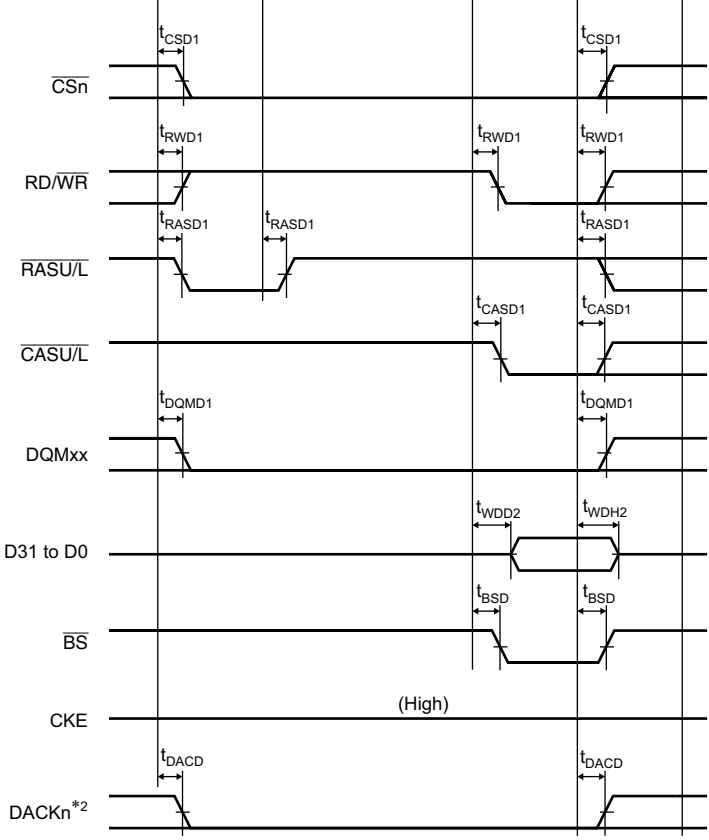
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

**Figure 25.25 Synchronous DRAM Burst Read Bus Cycle (Single Read ×
(Auto Precharge, CAS Latency = 2, TRCD = 2 Cycle, TRP = 1 Cycle)**



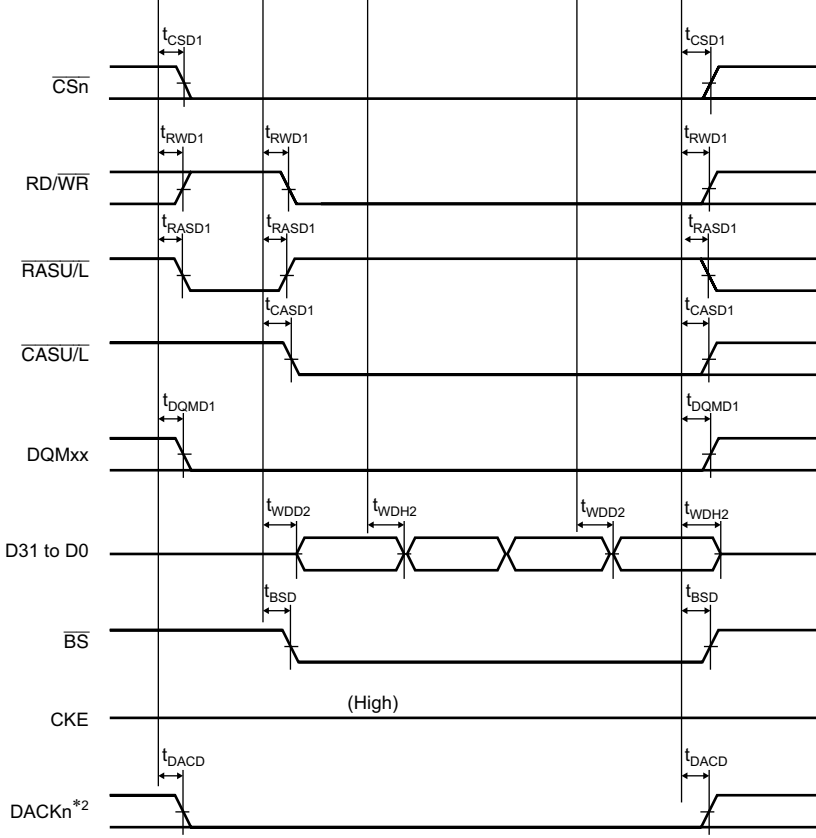
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.26 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 2 Cycle)



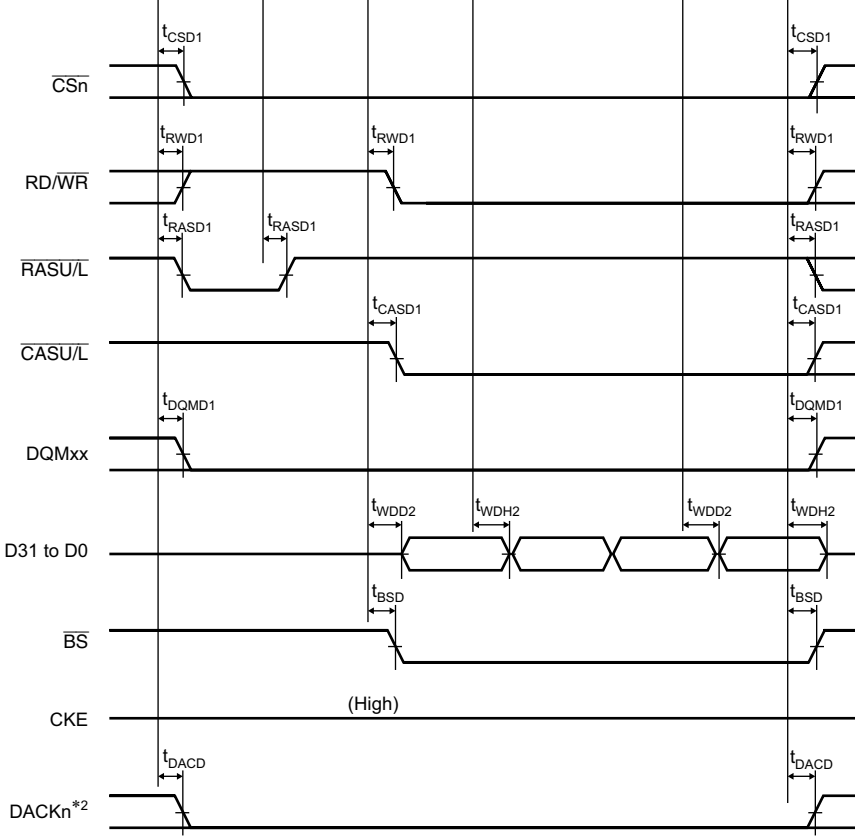
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.27 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRCD = 3 Cycle, TRWL = 2 Cycle)



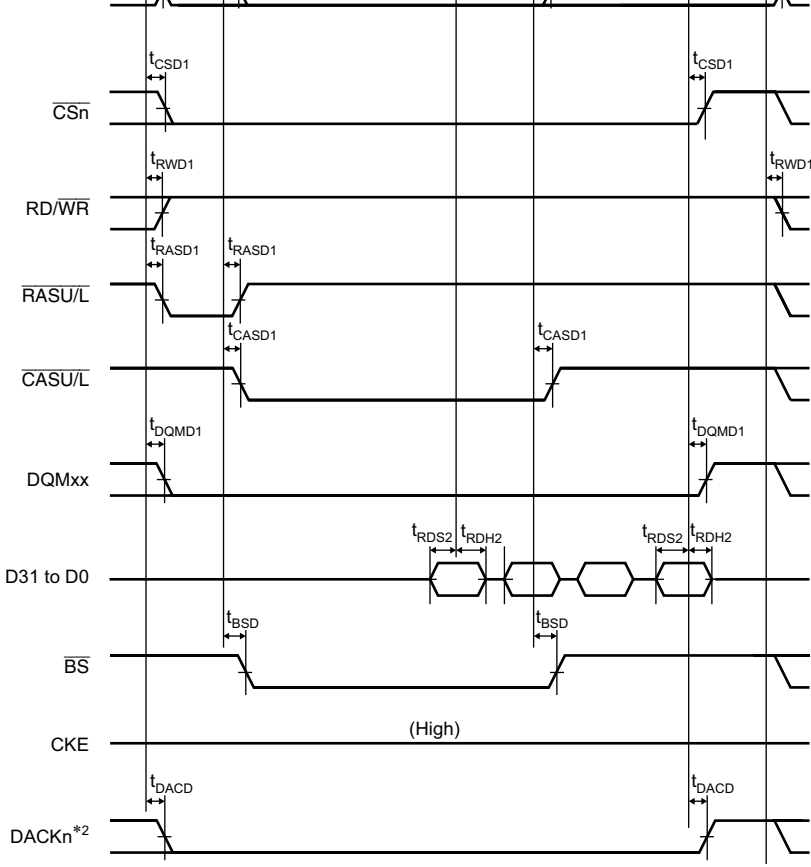
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

**Figure 25.28 Synchronous DRAM Burst Write Bus Cycle (Single Write ×
(Auto Precharge, TRCD = 1 Cycle, TRWL = 2 Cycle)**



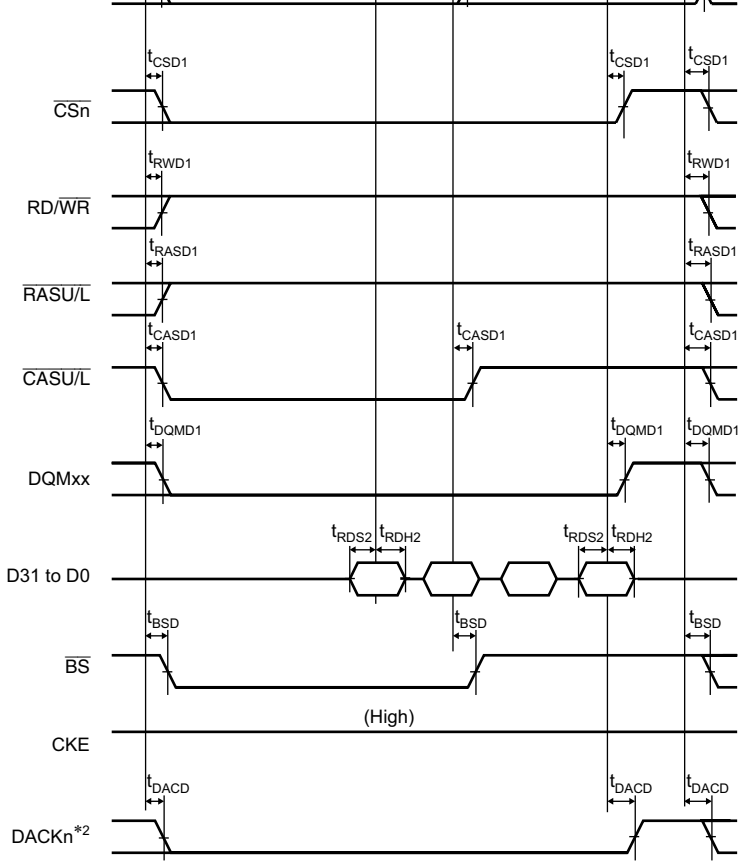
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.29 Synchronous DRAM Burst Write Bus Cycle (Single Write (Auto Precharge, TRCD = 2 Cycle, TRWL = 2 Cycle)



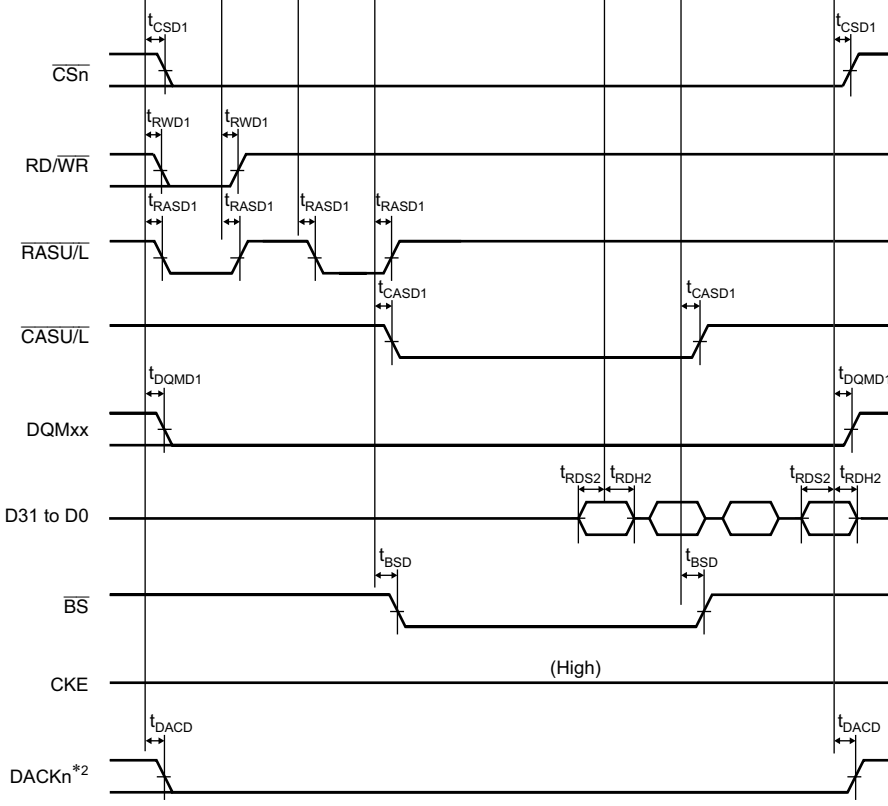
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.30 Synchronous DRAM Burst Read Bus Cycle (Single Read ×
(Bank Active Mode: ACTV + READ Commands, CAS Latency = 2, TRCD = 1



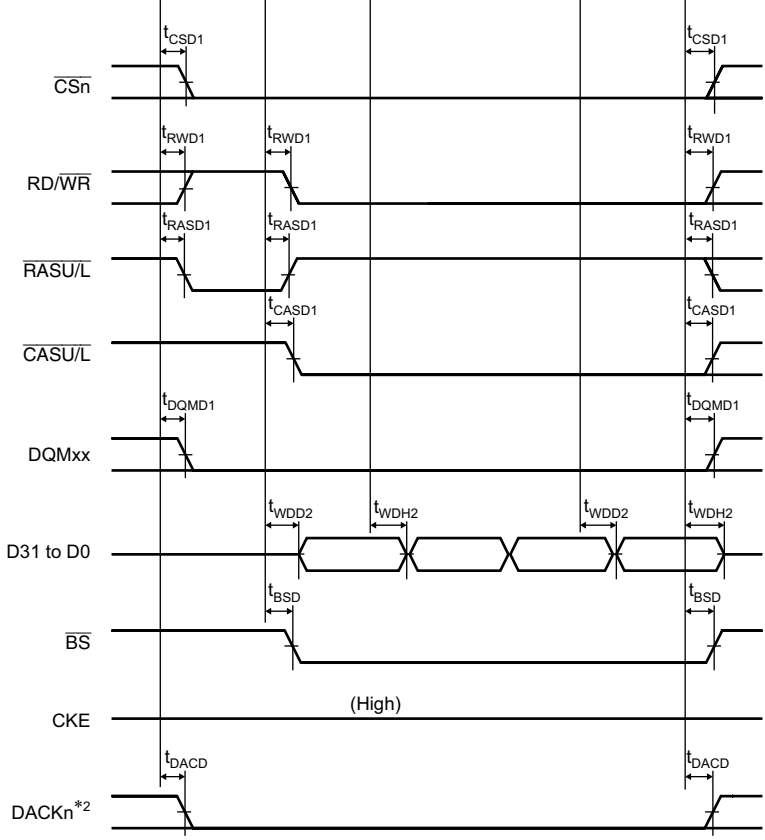
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.31 Synchronous DRAM Burst Read Bus Cycle (Single Read x
(Bank Active Mode: READ Command, Same Row Address,
CAS Latency = 2, TRCD = 1 Cycle)



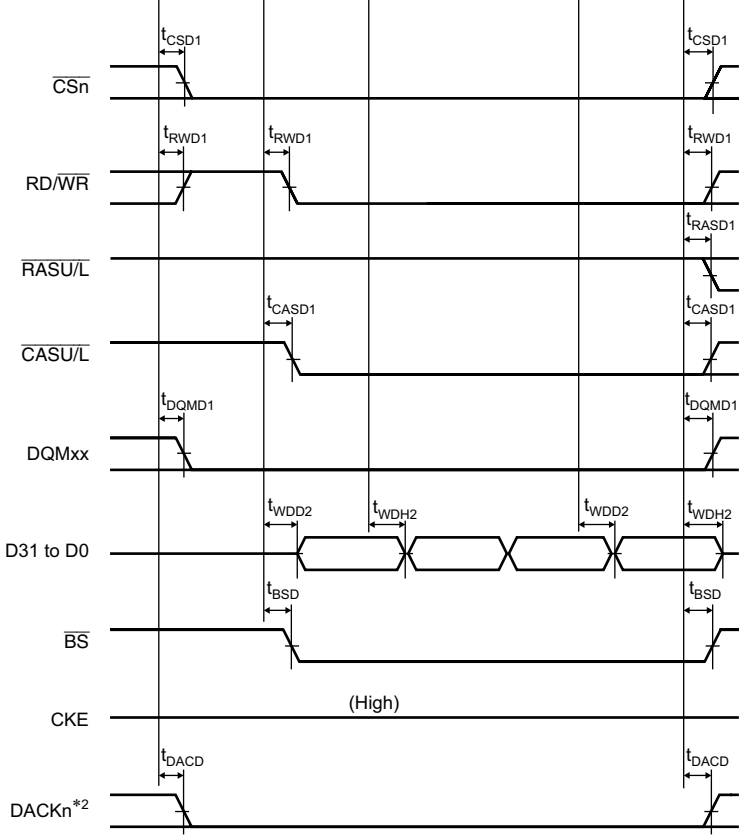
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.32 Synchronous DRAM Burst Read Bus Cycle (Single Read × (Bank Active Mode: PRE + ACTV + READ Commands, Different Row Address, CAS Latency = 2, TRCD = 1 Cycle)



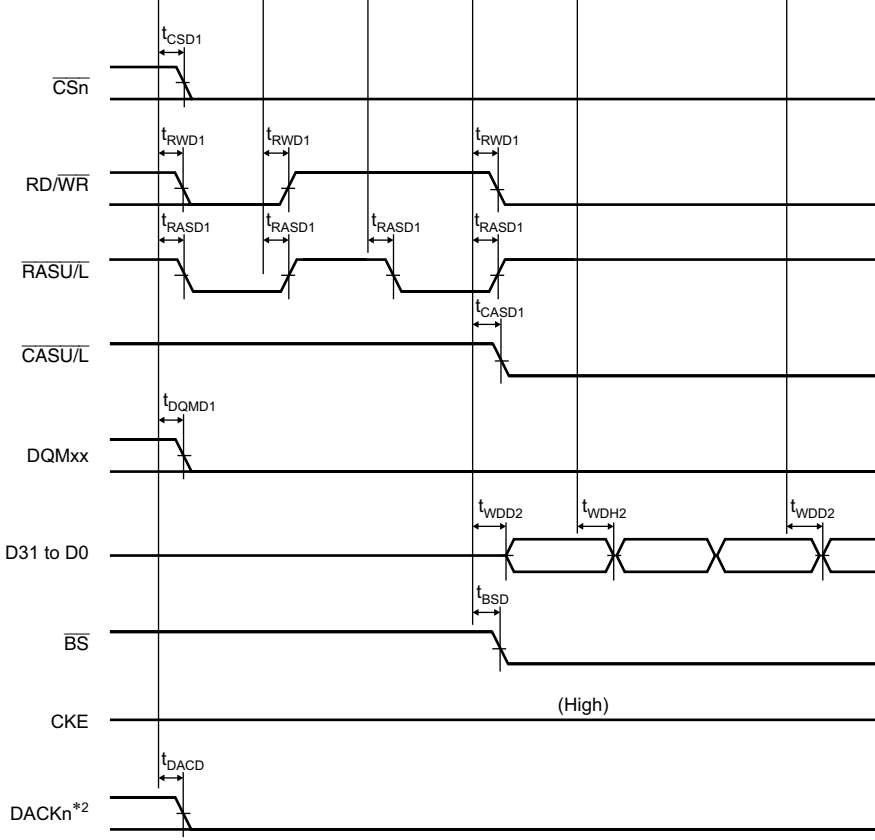
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.33 Synchronous DRAM Burst Write Bus Cycle (Single Write
(Bank Active Mode: ACTV + WRITE Commands, TRCD = 1 Cycle, TRWL =



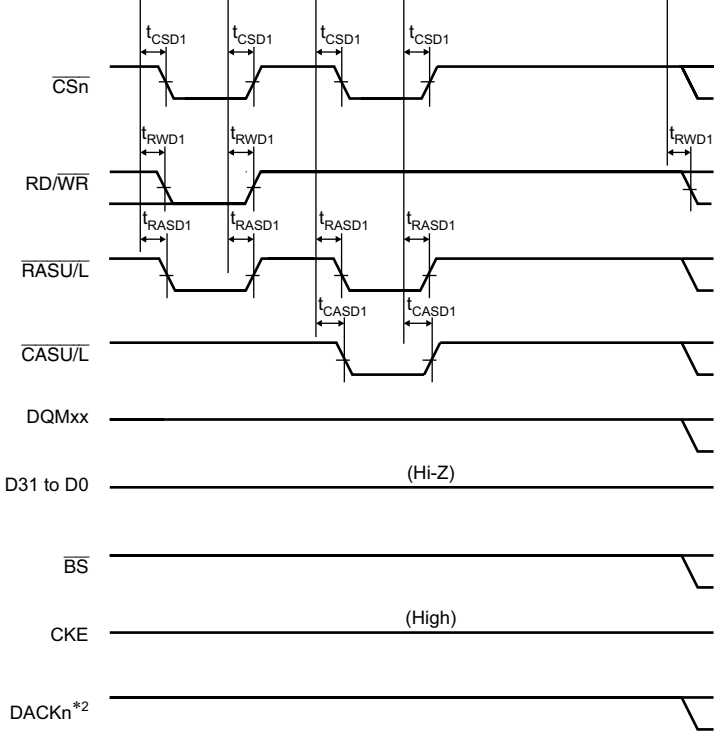
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.34 Synchronous DRAM Burst Write Bus Cycle (Single Write x
(Bank Active Mode: WRITE Command, Same Row Address,
TRCD = 1 Cycle, TRWL = 1 Cycle)



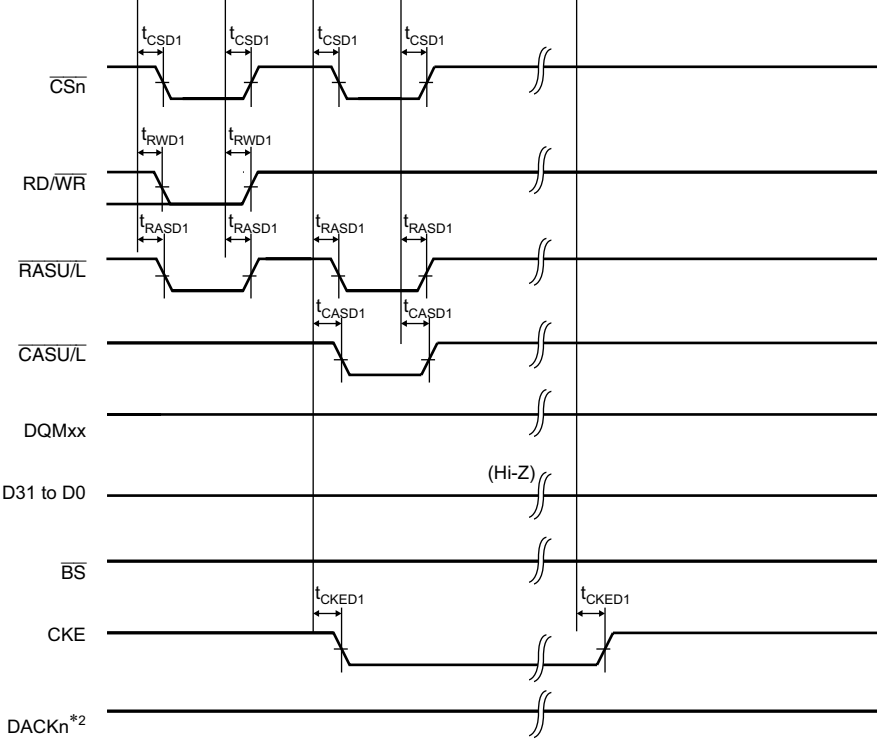
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.35 Synchronous DRAM Burst Write Bus Cycle (Single Write (Bank Active Mode: PRE + ACTV + WRITE Commands, Different Row Address, TRCD = 1 Cycle, TRWL = 1 Cycle)



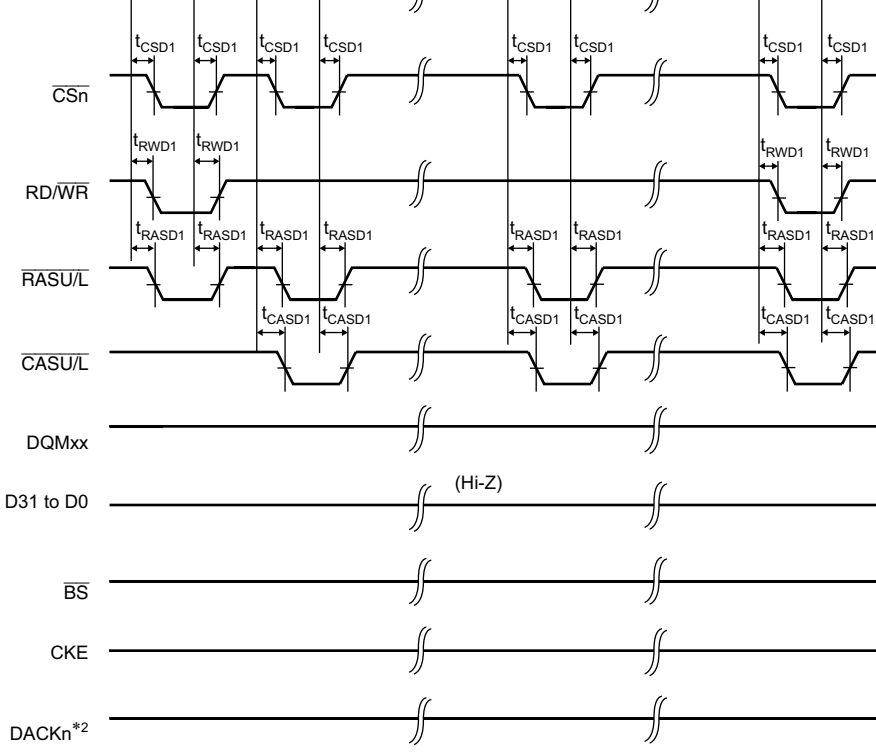
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.36 Synchronous DRAM Auto-Refresh Timing (TRP = 2 Cycl



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

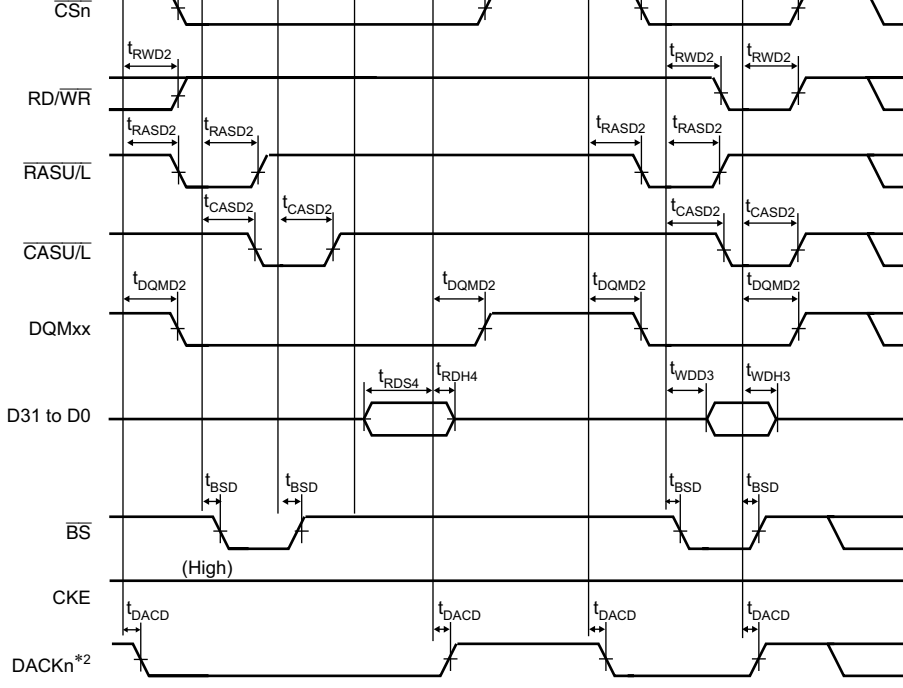
Figure 25.37 Synchronous DRAM Self-Refresh Timing (TRP = 2 Cycl



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

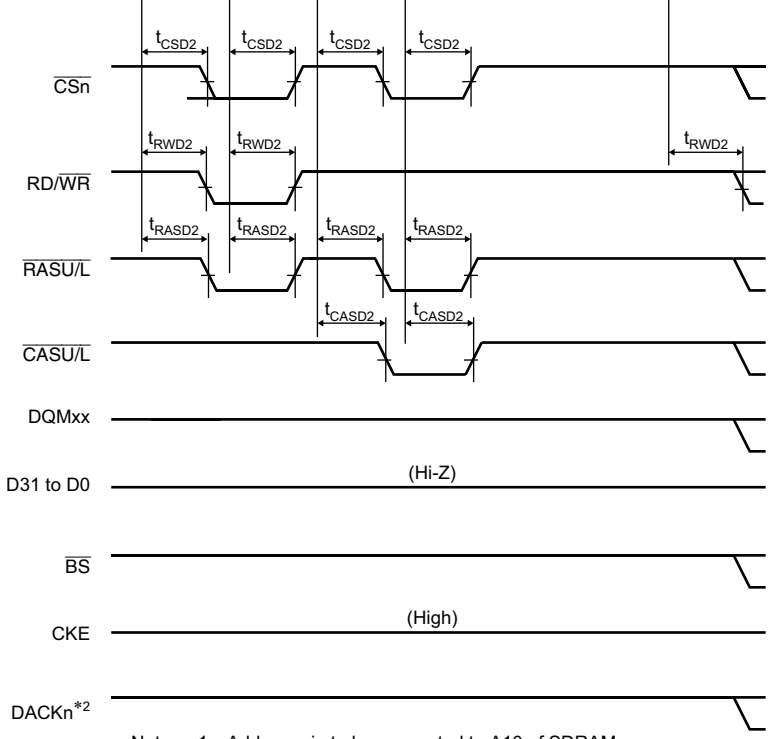
Figure 25.38 Synchronous DRAM Mode Register Write Timing (TRP = 2 C

Read data setup time 4	t_{RDS4}	1/2 t_{cyc}+6	—	ns	25.39
Read data hold time 4	t_{RDH4}	0	—	ns	25.39
Write data delay time 3	t_{WDD3}	—	1/2 t_{cyc}+12	ns	25.39
Write data hold time 3	t_{WDH3}	1/2 t_{cyc}	—	ns	25.39
$\overline{RAS}$ delay time 2	t_{RASD2}	1/2 t_{cyc}	1/2 t_{cyc}+10	ns	25.39 to 25.42
$\overline{CAS}$ delay time 2	t_{CASD2}	1/2 t_{cyc}	1/2 t_{cyc}+10	ns	25.39 to 25.42
DQM delay time 2	t_{DQMD2}	1/2 t_{cyc}	1/2 t_{cyc}+10	ns	25.39
CKE delay time 2	t_{CKED2}	1/2 t_{cyc}	1/2 t_{cyc}+10	ns	25.41



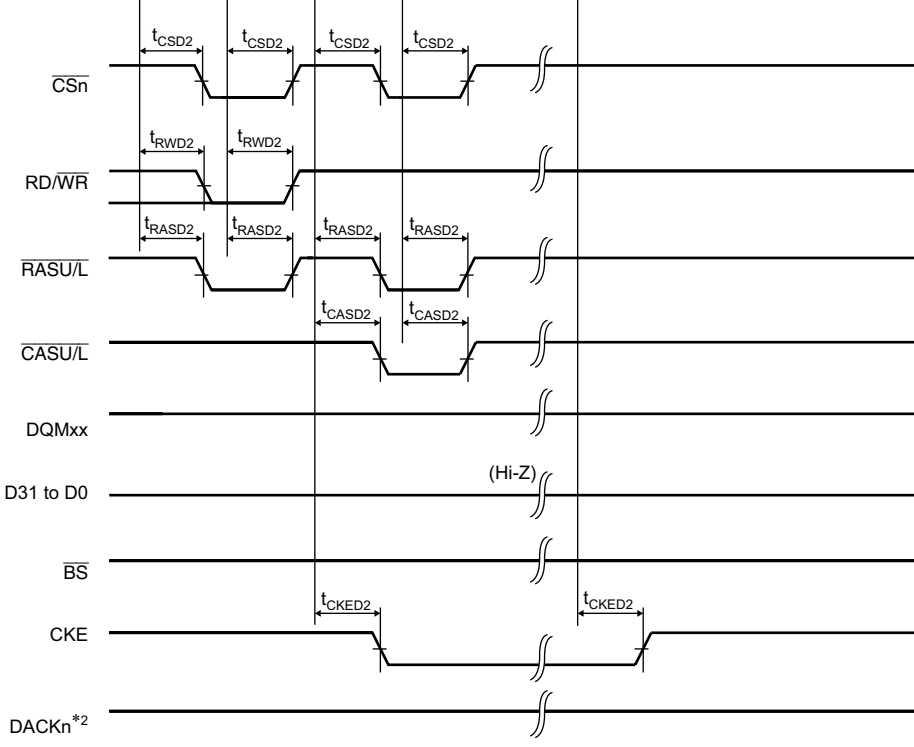
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.39 Access Timing in Low-Frequency Mode (Auto Precharge



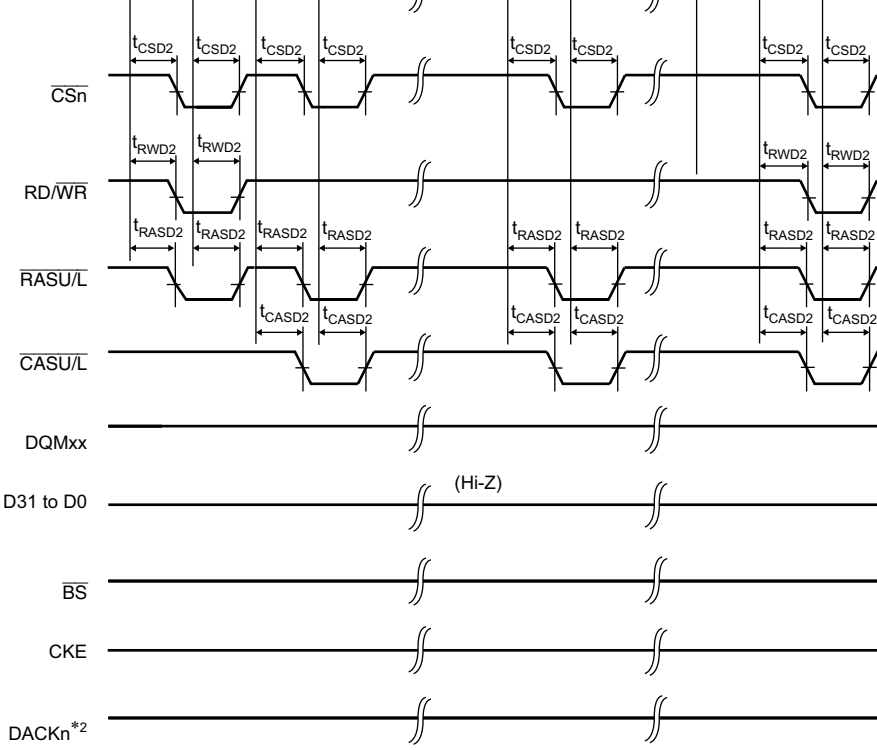
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.40 Synchronous DRAM Auto-Refresh Timing (TRP = 2 Cycle, Low-Frequency Mode)



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.41 Synchronous DRAM Self-Refresh Timing (TRP = 2 Cycle, Low-Frequency Mode)



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.42 Synchronous DRAM Mode Register Write Timing (TRP = 2 Cycle, Low-Frequency Mode)

DREQ hold time	t_{DRQH}	3	—
DACK, TEND delay time	t_{DACD}	—	10

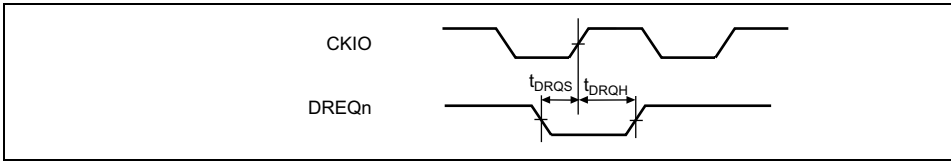


Figure 25.43 DREQ Input Timing

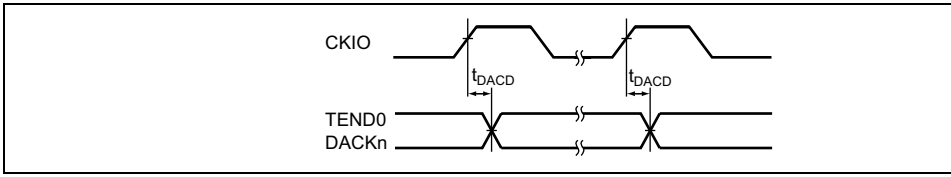


Figure 25.44 DACK, TEND Output Timing

	B:P clock ratio = 2:1		t_{cyc}+15	—	
	B:P clock ratio = 4:1		3 × t_{cyc}+15	—	
Timer clock input setup time		t_{TCKS}	15	—	
Timer clock pulse width	Edge specification	$t_{TCKWH/L}$	2.0	—	t_{pcyc}*
	Both edge specification	$t_{TCKWH/L}$	3.0	—	

Note: * t_{pcyc} indicates a peripheral clock (Pϕ) cycle.

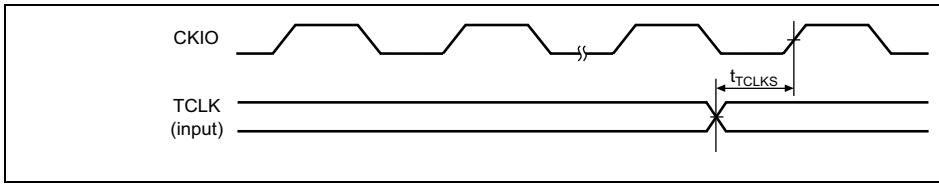


Figure 25.45 TCLK Input Timing

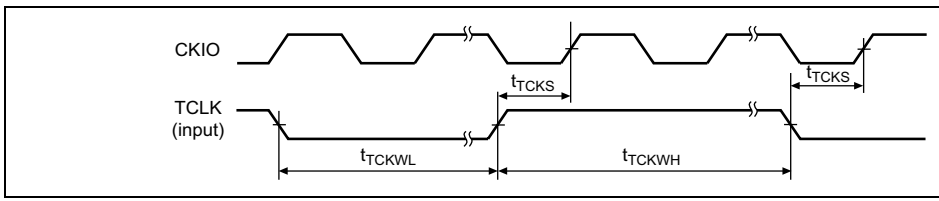


Figure 25.46 TCLK Clock Input Timing

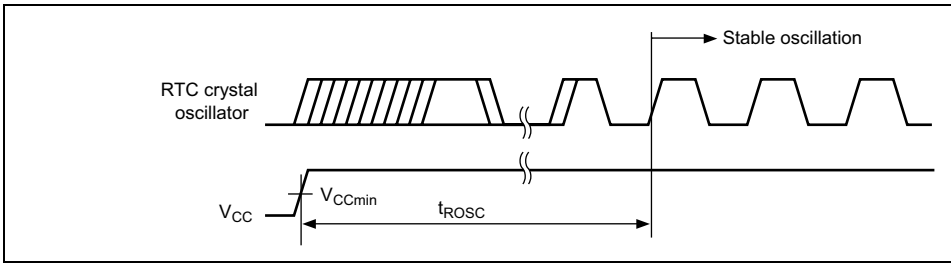


Figure 25.47 Oscillation Settling Time when RTC Crystal Oscillator Is Turn

25.3.10 16-Bit Timer Pulse Unit (TPU) Signal Timing

Table 25.12 16-Bit Timer Pulse Unit (TPU) Signal Timing

(Conditions: $V_{CC}Q = V_{CC}$-RTC = V_{CC}-USB = 3.0 to 3.6 V, $V_{CC} = V_{CC}$-PLL1 = V_{CC}-P
1.6 V, AV_{CC} = 3.0 to 3.6 V, $V_{SS}Q = V_{SS} = V_{SS}$-RTC = V_{SS}-USB = V_{SS}-PLL1 =
AV_{SS} = 0 V, T_a = –

Item	Symbol	Min	Max	Unit
Timer output delay time	t_{TOD}	—	15	ns

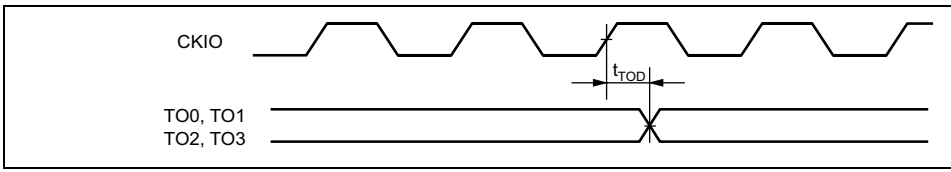


Figure 25.48 TPU Output Timing

	Asynchronization		4	—
Input clock rise time	t_{SCKr}	—	1.5	
Input clock fall time	t_{SCKf}	—	1.5	
Input clock pulse width	t_{SCKW}	0.4	0.6	t_{scyc}
Transmission data delay time (clock synchronization)	t_{TXD}	—	3 t_{pcyc}* + 50	ns
Receive data setup time (clock synchronization)	t_{RXS}	2 t_{pcyc}*	—	
Receive data hold time (clock synchronization)	t_{RXH}	2 t_{pcyc}*	—	
$\overline{RTS}$ delay time (clock synchronization)	t_{RTSD}	—	100	
$\overline{CTS}$ setup time (clock synchronization)	t_{CTSS}	100	—	
$\overline{CTS}$ hold time (clock synchronization)	t_{CTSH}	100	—	

Note: * t_{pcyc} indicates a peripheral clock (Pϕ) cycle.

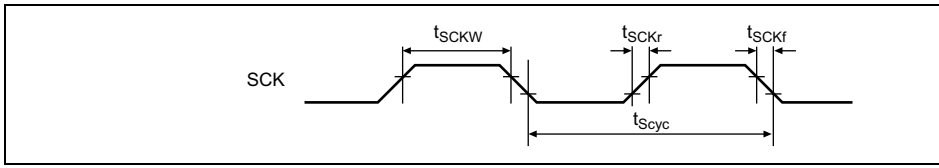


Figure 25.49 SCK Input Clock Timing

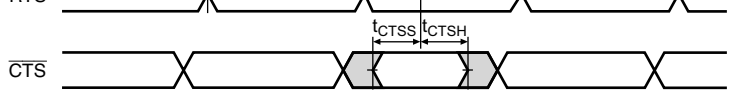


Figure 25.50 SCIF Input/Output Timing in Clock Synchronous Mode

25.3.12 USB Module Signal Timing

Table 25.14 USB Module Clock Timing

(Conditions: $V_{CC}Q = V_{CC}$-RTC = V_{CC}-USB = 3.0 to 3.6 V, $V_{CC} = V_{CC}$-PLL1 = V_{CC}-P
1.6 V, AV_{CC} = 3.0 to 3.6 V, $V_{SS}Q = V_{SS} = V_{SS}$-RTC = V_{SS}-USB = V_{SS}-PLL1 =
AV_{SS} = 0 V, T_a = –

Item	Symbol	Min	Max	Unit
Frequency (48 MHz)*	t_{FREQ}	47.9	48.1	MHz
Clock rise time*	t_{R48}	—	4	ns
Clock fall time*	t_{F48}	—	4	ns
Duty (t_{HIGH}/t_{LOW})*	t_{DUTY}	90	110	%
Oscillation settling time	t_{UOSC}	10	—	ms

Note: * When the USB is operated by supplying a clock to the EXTAL_USB pin from off-
supplied clock must satisfy the above clock specifications.

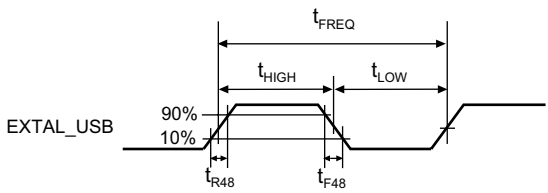


Figure 25.51 USB Clock Timing

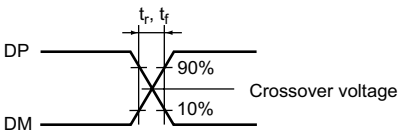
25.3.13 USB Transceiver Timing

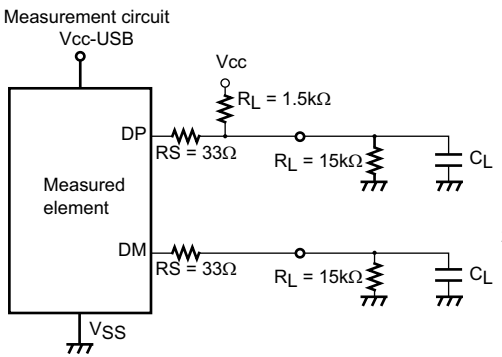
Table 25.15 USB Transceiver Timing

(Conditions: $V_{CC}Q = V_{CC}$-RTC = V_{CC}-USB = 3.0 to 3.6 V, $V_{CC} = V_{CC}$-PLL1 = V_{CC}-
1.6 V, AV_{CC} = 3.0 to 3.6 V, $V_{SS}Q = V_{SS} = V_{SS}$-RTC = V_{SS}-USB = V_{SS}-PLL1 =
AV_S = 0 V, T_a =

Item	Symbol	Min	Typ	Max	Unit	Measu Condi
Rising time	t_r	4	—	20	ns	C_L = 5
Falling time	t_f	4	—	20	ns	C_L = 5
Rising/falling time ratio	t_r / t_f	90	—	110	%	
Output signal crossover voltage	V_{CRS}	1.3	—	2.0	V	C_L = 5

Note: This transceiver complies with the full-speed specifications.





1. t_r and t_f are judged by the time taken for transitions between 10% and 90% ampl
2. The electrostatic capacitance, C_L, includ floating capacitance of the wiring conne and the input capacitance of the probe.

Input data setup time	B:P clock ratio = 1:1	t_{PORTS}	15	—
	B:P clock ratio = 2:1		t_{cyc}+15	—
	B:P clock ratio = 4:1		3 × t_{cyc}+15	—
Input data hold time		t_{PORTH}	8	—

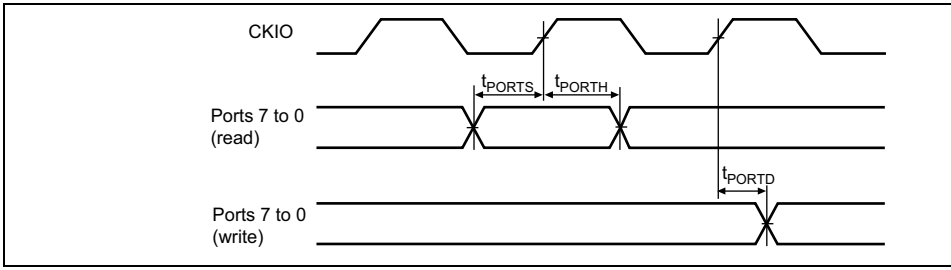


Figure 25.53 I/O Port Timing

TCK high-pulse width	t_{TCKH}	12	—	ns
TCK low-pulse width	t_{TCKL}	12	—	ns
TCK rise/fall time	t_{TCKf}	—	4	ns
$\overline{\text{TRST}}$ setup time	t_{TRSTS}	12	—	ns
$\overline{\text{TRST}}$ hold time	t_{TRSTH}	50	—	t_{cyc}
TDI setup time	t_{TDIS}	10	—	ns
TDI hold time	t_{TDIH}	10	—	ns
TMS setup time	t_{TMSS}	10	—	ns
TMS hold time	t_{TMSH}	10	—	ns
TDO delay time	t_{TDOD}	—	15	ns
$\overline{\text{ASEMD0}}$ setup time	$t_{ASEMD0S}$	12	—	ns
$\overline{\text{ASEMD0}}$ hold time	$t_{ASEMD0H}$	12	—	ns

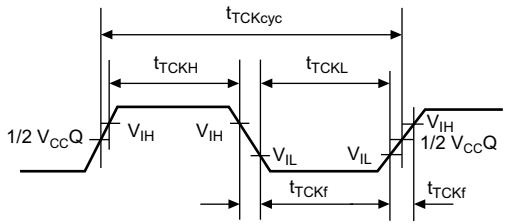


Figure 25.54 TCK Input Timing

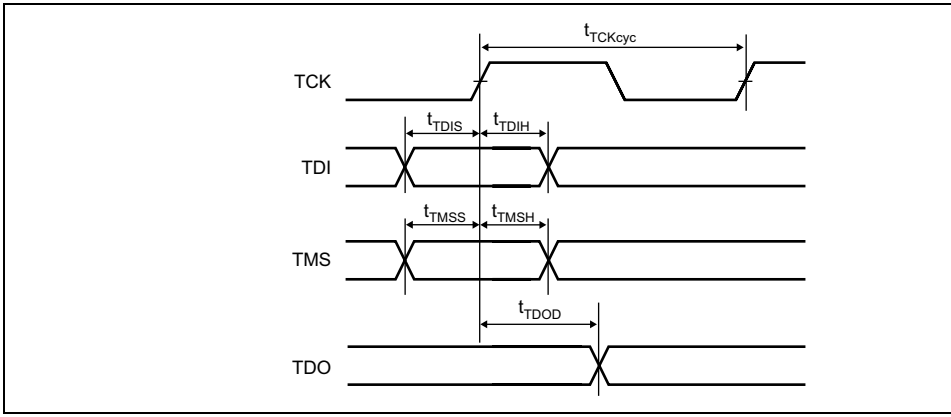


Figure 25.56 UDI Data Transfer Timing

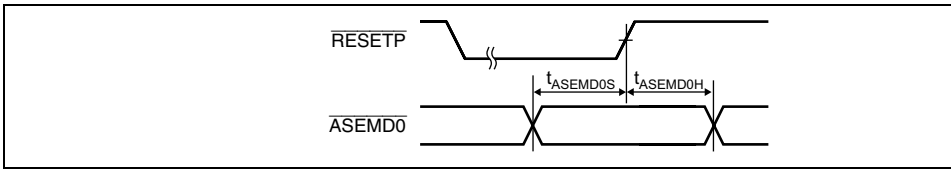
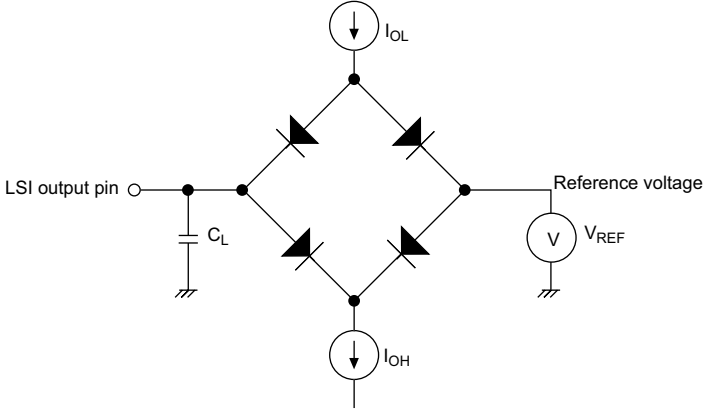


Figure 25.57 $\overline{\text{ASEMD0}}$ Input Timing



Notes: 1. C_L is the total value that includes the capacitance of measurement instruments, etc., and is set as follows for each pin.
30 pF: CKIO, $\overline{\text{RASU/L}}$, $\overline{\text{CASU/L}}$, $\overline{\text{CS0}}$, $\overline{\text{CS2}}$ to $\overline{\text{CS6B}}$, $\overline{\text{BACK}}$
50 pF: All other pins
2. I_{OL} =1.6 mA, I_{OH} = –200µA

Figure 25.58 Output Load Circuit

Item	Min	Typ	Max	
Resolution	10	10	10	
Conversion time	8.5	—	—	
Analog input capacitance	—	—	20	
Permissible signal-source impedance (single-source)	—	—	5	
Nonlinearity error	—	—	±3.0	
Offset error	—	—	±2.0	
Full-scale error	—	—	±2.0	
Quantization error	—	—	±0.5	
Absolute accuracy	—	—	±4.0	

Category	Pin	Reset	Reset	Standby	Sleep	Released	I/O
Clock	EXTAL	I	I	I	I	I	I
	XTAL	O	O	O	O	O	O
	EXTAL2	I	I	I	I	I	I
	XTAL2	O	O	O	O	O	O
	CKIO	I O[*1]	I O[*1]	I O[*1]	I O[*1]	I O[*1]	I O
System control	$\overline{\text{RESETP}}$	I[*11]	I[*11]	I[*11]	I[*11]	I[*11]	I
	$\overline{\text{RESETM}}$	I	I	I	I	I	I
	$\overline{\text{BREQ}}$/PTG[6]	Z	i P[*2]	i K[*3]	I P[*2]	I	I/IO
	$\overline{\text{BACK}}$/PTG[5]	O	O P[*2]	O K[*3]	O P[*2]	L P[*2]	O/IO
	MD6	I	i	Z	i	i	I
	MD[2:0]	I	i	i	i	i	I
	MD[5:3]	I	i	Z	i	i	I
	CA	I	I	I	I	I	I
	STATUS0/ PTE[4]/$\overline{\text{RTS0}}$	H	H P[*2] Z[*6]	H K[*3] Z[*6]	L P[*2] O	L P[*2] O	O/IO/ O
	STATUS1/ PTE[5]/$\overline{\text{CTS0}}$	H	H P[*2] Z[*6]	L K[*3] Z[*7]	H P[*2] I	L P[*2] I	O/IO/ I
Interrupt	IRQ[3:0]/ $\overline{\text{IRL}}$[3:0]/ PTH[3:0]	Z	I P[*2]	I K[*3]	I P[*2]	I P[*2]	I/I/IO
	IRQ4/PTH[4]	Z	I P[*2]	I K[*3]	I P[*2]	I P[*2]	I/IO
	IRQ5/PTE[2]	Z	I P[*2]	I K[*3]	I P[*2]	I P[*2]	I/IO
	NMI	I	I	I	I	I	I
Address	A[25:19,0]/ PTK[7:0]	O	O P[*2]	ZO[*8] K[*3]	O P[*2]	Z P[*2]	O/IO
	A[18:1]	O	O	ZO[*8]	O	Z	O

	PINT[15:8]						
Bus control	$\overline{CS0}$	H	O	ZH*6	O	Z	O
	$\overline{CS2}$/PTC[3]	H	O P*2	ZH*6 K*3	O P*2	Z P*2	O/IO
	$\overline{CS3}$/PTC[4]	H	O P*2	ZH*6 K*3	O P*2	Z P*2	O/IO
	$\overline{CS4}$/PTC[5]	H	O P*2	ZH*6 K*3	O P*2	Z P*2	O/IO
	$\overline{CS5A}$/PTC[6]	Z	O P*2	ZH*6 K*3	O P*2	Z P*2	O/IO
	$\overline{CS5B}$/PTD[6]	Z	O P*2	ZH*6 K*3	O P*2	Z P*2	O/IO
	$\overline{CS6A}$/PTC[7]	Z	O P*2	ZH*6 K*3	O P*2	Z P*2	O/IO
	$\overline{CS6B}$/PTD[7]	Z	O P*2	ZH*6 K*3	O P*2	Z P*2	O/IO
	$\overline{BS}$/PTC[0]	H	O P*2	ZH*6 K*3	O P*2	Z P*2	O/IO
	$\overline{RASL}$/PTD[0]	H	O P*2	ZH*6 K*3	O P*2	ZH*6 P*2	O/IO
	$\overline{RASU}$/PTD[1]	Z	O P*2	ZH*6 K*3	O P*2	ZH*6 P*2	O/IO
	$\overline{CASL}$/PTD[2]	H	O P*2	ZH*6 K*3	O P*2	ZH*6 P*2	O/IO
	$\overline{CASU}$/PTD[3]	Z	O P*2	ZH*6 K*3	O P*2	ZH*6 P*2	O/IO
	$\overline{WE0}$/DQMLL	H	O	ZH*6	O	Z	O/O
	$\overline{WE1}$/DQMLU	H	O	ZH*6	O	Z	O/O
	$\overline{WE2}$/DQMUL/ PTC[1]	H	O P*2	ZH*6 K*3	O P*2	Z P*2	O/O/IO
	$\overline{WE3}$/DQMUU/ $\overline{AH}$/PTC[2]	H	O P*2	ZH*6 K*3	O P*2	Z P*2	O/O/ O/IO
	RD/$\overline{WR}$	H	O	ZH*6	O	Z	O
	$\overline{RD}$	H	O	ZH*6	O	Z	O
	CKE/PTD[4]	H	O P*2	OK*3	O P*2	OP*2	O/IO
	$\overline{WAIT}$/PTG[7]	I	I P*2	I K*3	I P*2	I P*2	I/IO

	DREQ1/ PTH[6]	Z	Z P*2	Z K*3	I P*2	I P*2	I/IO
	DACK1/ PTE[1]	V	O P*2	Z K*3	O P*2	O P*2	O/IO
Timer	TCLK/PTE[6]	V	I P*2	Z K*3	I P*2	I P*2	I/IO
SCIF	RxD0/ SCPT[0]/IrRX	Z	Z I*4	Z	I	I	I/I/I
	TxD0/ SCPT[0]/IrTX	Z	Z O*5	Z O*5	O	O	O/O/ O
	SCK0/ SCPT[1]	Z	Z P*2	Z K*3	IO P*2	IO P*2	IO/IO
	RxD2/ SCPT[2]	Z	Z I*4	Z	I	I	I/I
	TxD2/ SCPT[2]	Z	Z O*5	Z O*5	O	O	O/O
	SCK2/ SCPT[3]	Z	Z P*2	Z K*3	IO P*2	IO P*2	IO/IO
	$\overline{\text{RTS2}}$/ SCPT[4]	V	Z P*2	Z K*3	O P	O P	O/IO
	$\overline{\text{CTS2}}$/ SCPT[5]	Z	Z P*2	Z K*3	I P*2	I P*2	I/IO
Analog	AN[3:0]/ PTL[3:0]	i	Z I*4	i	I	I	I/I
USB	VBUS/ PTM[6]	V	I P*2	Z K*2	I P*2	I P*2	I/IO
	SUSPND/ PTN[0]	V	O P*2	O K*3	O P*2	O P*2	O/IO
	TXENL/ PTN[1]	V	O P*2	O K*3	O P*2	O P*2	O/IO
	XVDATA/ PTN[2]	V	I P*2	V K*3	I P*2	I P*2	I/IO

	DPLS/PTN[6]	V	I P[*2]	V K[*3]	I P[*2]	I P[*2]	I/IO
	EXTAL_USB	I	I	i	I	I	I
	XTAL_USB	O	O	O	O	O	O
	D+	Z	IO[*9]	Z	IO[*9]	IO[*9]	IO
	D-	Z	IO[*9]	Z	IO[*9]	IO[*9]	IO
Port	NF/PTD[5]	I	I	Z	I	I	I/I
	PTE[7]	V	P	K	P	P	IO
	NF/PTJ[7]	L	O	O	O	O	O/O
	NF/PTJ[6:0]	H[*13]	O	O	O	O	O/O
	NF/PTM[4]	I	I	Z	I	I	I/I
	PTM[3:0]	V	P	K	P	P	IO
	PTN[7]	V	P	K	P	P	IO
Advanced user debugger	AUDSYNC/ PTF[4]	V/V[*10]	O P[*2]	O K[*3]	O P[*2]	O P[*2]	O/IO
	AUDATA[3:0]/ PTF[3:0]/ TO[3:0]	V/V[*10]	O P[*2] Z[*8]	O K[*3] Z[*8]	O P[*2]	O P[*2]	O/IO/ O
	AUDCK/ PTG[4]	O/V[*10]	O P[*2]	O K[*3]	O P[*2]	O P[*2]	O/IO
User debugg-ing interface	TDI/PTG[0]	I[*11]	I[*11] P[*2]	i[*11] K[*3]	I[*11] P[*2]	I[*11] P[*2]	I/IO
	TCK/PTG[1]	I[*11]	I[*11] P[*2]	i[*11] K[*3]	I[*11] P[*2]	I[*11] P[*2]	I/IO
	TMS/PTG[2]	I[*11]	I[*11] P[*2]	i[*11] K[*3]	I[*11] P[*2]	I[*11] P[*2]	I/IO
	$\overline{\text{TRST}}$/PTG[3]	I[*11]	I[*11] P[*2]	i[*11] K[*3]	I[*11] P[*2]	I[*11] P[*2]	I/IO
	TDO/PTF[5]	OZ	O P[*2]	Z K[*3]	O P[*2]	O P[*2]	O/IO
	$\overline{\text{ASEBRKAK}}$/ PTF[6]	V/V[*10]	O P[*2]	O K[*3]	O P[*2]	O P[*2]	O/IO
	$\overline{\text{ASEMD0}}$/ PTF[7]	I[*11]	I[*11] P[*2]	V K[*3]	I[*11] P[*2]	I[*11] P[*2]	I/IO

AVss	—	—	—	—	—	—
VccQ	—	—	—	—	—	—
VssQ	—	—	—	—	—	—
Vcc-PLL1	—	—	—	—	—	—
Vss-PLL1	—	—	—	—	—	—
Vcc-PLL2	—	—	—	—	—	—
Vss-PLL2	—	—	—	—	—	—
Vcc	—	—	—	—	—	—
Vss	—	—	—	—	—	—

Legend:

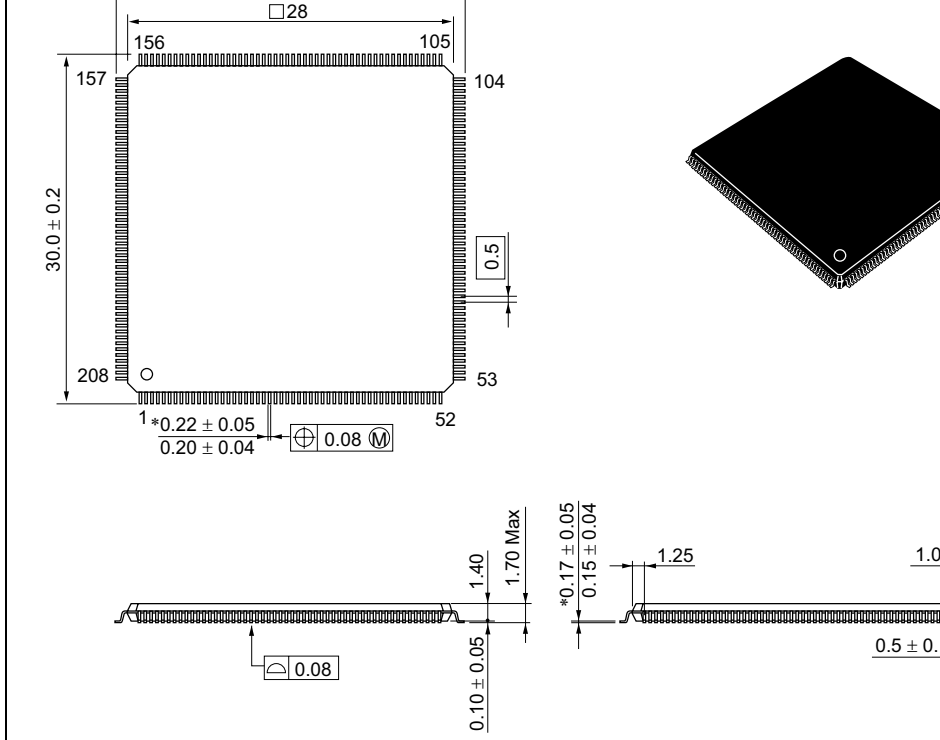
I : Input state
i : Input state (however, input is fixed by the internal logic.)
O : Output state (high or low, undefined)
L : Low-level output
H : High-level output
Z : High impedance (input/output buffer off)
V : Input/output buffer off, pull-up on
K : The high-level output or low-level output/input becomes high impedance.
P : Input or output depending on the register settings.

Notes:

1. Depends on clock mode.
2. The state is P when the port function is used.
3. The state is K when the port function is used.
4. The state is I when the port function is used.
5. The state is O when the port function is used.
6. The state is Z or H depending on the register settings.
7. The state is Z or L depending on the register settings.
8. The state is Z or O depending on the register settings.
9. The state is i when the USB is not used.
10. The initial value (power-on reset) changes depending on the input level o $\overline{\text{ASEMD0}}$ pin. First, this list shows the value when the $\overline{\text{ASEMD0}}$ pin is 0, t value when the $\overline{\text{ASEMD0}}$ pin is 1.

PTJ6/NF	1	0	1
PTJ1/NF	1	1	0
PTJ0/NF	1	0	1

Connect the pull-up pins shown in table A.1 to 3.3 V power through the pull-up resistor



Package Code	FP-208C
JEDEC	—
JEITA	Conform
Mass (reference value)	2.7 g

*Dimension including the plating thickness
Base material dimension

Figure B.1 Package Dimensions (FP-208C)

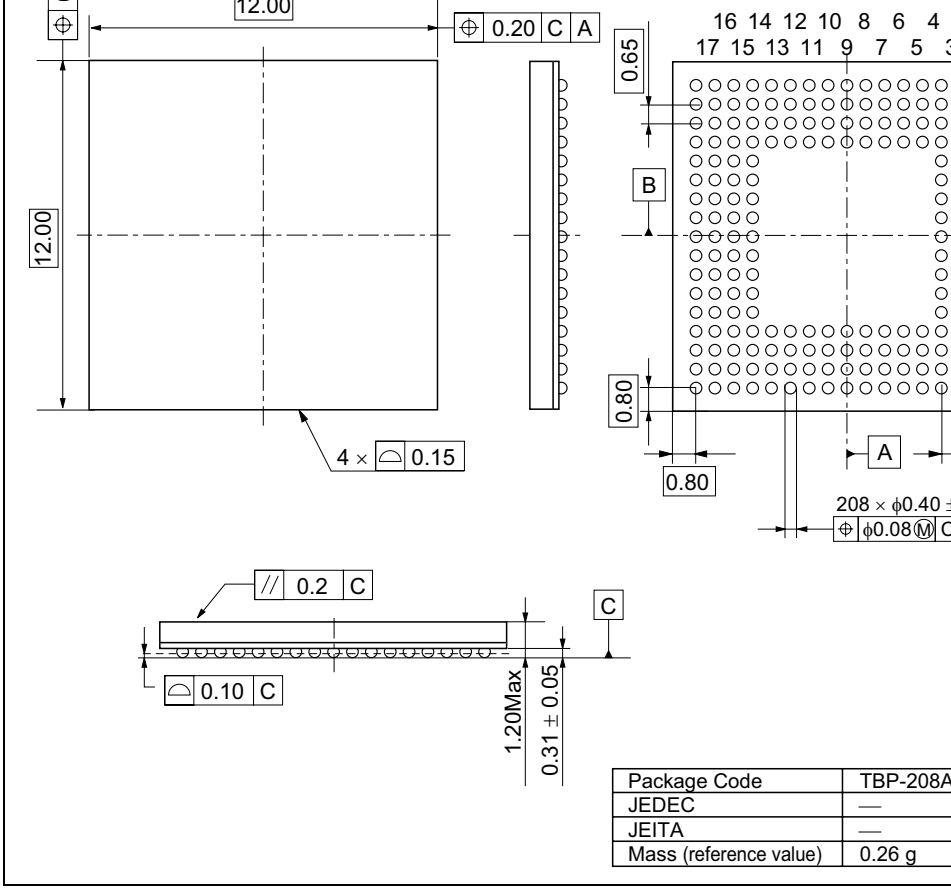


Figure B.2 Package Dimensions (TBP-208A)

SH7705 Group Hardware Manual

Publication Date:	1st Edition, November 2002
	Rev.2.00, September 19, 2003
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.

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, United Kingdom
Tel: <44> (1628) 585 100, Fax: <44> (1628) 585 900

Renesas Technology Europe GmbH
Dornacher Str. 3, D-85622 Feldkirchen, Germany
Tel: <49> (89) 380 70 0, Fax: <49> (89) 929 30 11

Renesas Technology Hong Kong Ltd.
7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2375-6836

Renesas Technology Taiwan Co., Ltd.
FL 10, #99, Fu-Hsing N. Rd., Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
26/F., Ruijin 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

SH7705 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0082-0200O

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Microprocessors - MPU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MC68302EH20C MC7457RX1000LC MC7457RX1267LC MC7457VG1267LC A2C00010998 A A2C52004004 R5F117BCGNA#20 R5F52106BDLA#U0 R5S72690W266BG#U0 ADJ3400IAA5DOE MPC8245TVV266D MPC8245TZU300D MPC8260ACVVMHBB MPC8323ECVRAFDCA MPC8323VRADDCA MPC8536ECVJAVLA BOXNUC5PGYH0AJ 20-668-0024 P1010NSN5DFB P2010NSN2MHC P2020NXE2HHC P5020NSE7QMB P5020NSE7TNB P5020NSE7VNB LS1020ASN7KQB LS1020AXN7HNB LS1020AXN7KQB A2C00010729 A A2C00039344 T1022NSE7MQB T1022NXN7PQB T1023NSE7MQA T1024NXE7PQA T1042NSE7MQB T1042NSN7MQB T1042NXN7WQB T2080NSE8TTB T2080NSN8PTB T2080NXE8TTB T2081NXN8TTB R5F101AFASP#V0 MC68302CEH20C TS68040MF33A MPC8260ACVVMIBB MPC8280CZUUPEA MPC8313ECVRAFFC MPC8313ECVRAGDC MPC8313EVRADDC MPC8313EVRAFFC MPC8313VRADDC