To all our customers

Regarding the change of names mentioned in the document, such as Hitachi Electric and Hitachi XX, to Renesas Technology Corp.

The semiconductor operations of Mitsubishi Electric and Hitachi were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.)

Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp.

Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: www.renesas.com

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

RENESAS

Renesas Technology Corp.

Hitachi SuperH™ RISC engine

SH7706

Hardware Manual

HITACHI

ADE-602-235

Rev. 1.0
09/21/01
Hitachi, Ltd.

Hitachi
semiconductor

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi’s or any third party’s patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party’s rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi’s sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi’s sales office for any questions regarding this document or Hitachi semiconductor products.

General Precautions on the Handling of Products

1. Treatment of NC Pins

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

2. Treatment of Unused Input Pins

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

3. Processing before Initialization

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

4. Prohibition of access to undefined or reserved addresses

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

Configuration of this Manual

This manual comprises the following items:

1. Precautions in Relation to this Product
2. Configuration of this Manual
3. Overview
4. Table of Contents
5. Summary
6. Description of Functional Modules
    - CPU and System-Control Modules
    - On-chip Peripheral Modules

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

      i) Features

      ii) I/O pins

      iii) Description of Registers

      iv) Description of Operation

      v) Usage: Points for Caution

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

7. List of Registers
8. Electrical Characteristics

Preface

The SH7706 is a RISC microprocessor that integrates a Hitachi-original RISC-type CPU as its core that has peripheral functions required for system configuration.

Reader: This manual is written for a reader who designs application system using this LSI. The reader is expected to have a general knowledge about electric circuit, logic circuit, and microcomputer.

Objective: This manual is written to make the reader to understand the hardware function and electric characteristics of this LSI.
Refer to “SH-3, SH-3E, SH3-DSP Programming Manual” for more details about the instructions.

How to read:

- Target products and their abbreviation.
  This manual describes about the following products.

Product lineup and abbreviation	Model Name
Basic classification	
SH7706 (176-pin plastic LQFP)	HD6417706F133
SH7706 (208-pin plastic TFBGA)	HD6417706BP133V

- To understand the whole functions of the LSI:
  Read this manual according to the index. This manual has following major categories:
  — CPU
  — System Control Functions
  — Peripheral Functions
  — Electric Characteristics
- To understand the details of the CPU functions:
  Refer to “SH-3, SH-3E, SH3-DSP Programming Manual”.

Legend:

Register: The following expression is used when the same function or similar function exists in several channels, such as serial communication function.
XXX-N (XXX is a basic register name, N is a channel number)

Bit: Left side is an upper bit and right is lower bit.

Figures: Binary number is expressed with B’xxxx.
Hexadecimal number is expressed with B’xxxx.
Decimal number is expressed with B’xxxx.

Signals: The overbar (‾‾) is attached with the signal that is active low.

Related documents

Refer to the following web-site for more hot information. And conferm that the documents is the newest one. (http://www.hitachisemiconductor.com/)

- Users manual for SH7706

Document Title	**Document Number**
SH7706 Hardware Manual	This Manual
SH-3/SH-3E/SH3-DSP Programming Manual	ADE-602-096B

- Users manual for development tools

Document Title	**Document Number**
SuperH RISK engine C/C++ Compiler Assembler Optimizing Linkage Editor	ADE-702-246
SuperH RISK engine Simulator/Debugger	ADE-702-186B
Hitachi Embedded Workshop	ADE-702-201A
SuperH RISK engine Hitachi Embedded Workshop, Hitachi Debugging Interface Tutorial	ADE-702-230
SH7706 E10A Emulator	ADE-702-253

Abbreviated words

ACIA	Asynchronous Communication Interface Adapter
ADC	Analog to Digital Converter
AUD	Advanced User Debugger
BSC	Bus State Controller
CPG	Clock Pulse Generator
CMT	Compare Match Timer
DAC	Digital to Analog Converter
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
DRAM	Dynamic Random Access Memory
ETU	Elementary Time Unit
FIFO	First-In First-Out
H-UDI	Hitachi User Debug Interface

INTC	Interrupt Controller
JEIDA	Japan Electronic Industry Development Association
JTAG	Joint Test Action Group
LRU	Least Recently Used
LSB	Least Significant Bit
MMU	Memory Management Unit
MSB	Most Significant Bit
PCMCIA	Personal Computer Memory Card International Association
PFC	Pin Function Controller
PLL	Phase Locked Loop
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
RTC	Real Time Clock
SCI	Serial Communication Interface
SCIF	Serial Communication Interface with FIFO
SRAM	Static Random Access Memory
TLB	Translation Lookaside Buffer
TMU	Timer Unit
UART	Universal Asynchronous Receiver/Transmitter
UBC	User Break Controller
WDT	Watch Dog Timer

Contents

Figures of Contents

Section 1 Overview

Section 2 CPU

Section 3 Memory Management Unit (MMU)

Section 4 Exception Processing

Section 5 Cache

Section 6 Interrupt Controller (INTC)

Section 7 User Break Controller

Section 8 Bus State Controller (BSC)

Section 9 Direct Memory Access Controller (DMAC)

Section 10 Clock Pulse Generator (CPG)

Section 11 Watchdog Timer (WDT)

Section 12 Timer Unit(TMU)

Section 13 Realtime Clock (RTC)

Section 14 Serial Communication Interface (SCI)

Section 15 Smart Card Interface

Section 16 Serial Communication Interface with FIFO (SCIF)

Section 18 I/O Ports

Section 19 A/D Converter (ADC)

Section 20 D/A Converter (DAC)

Section 21 Hitachi User-Debugging Interface (H-UDI)

Section 22 Power-Down Modes

Section 24 Electrical Characteristics

Appendix

Tables of Contents

Section 2 CPU

Section 3 Memory Management Unit (MMU)

Section 4 Exception Processing

Section 5 Cache

Section 6 Interrupt Controller (INTC)

Section 7 User Break Controller

Section 8 Bus State Controller (BSC)

Section 9 Direct Memory Access Controller (DMAC)

Section 10 Clock Pulse Generator (CPG)

Section 12 Timer Unit(TMU)

Section 13 Realtime Clock (RTC)

Section 14 Serial Communication Interface (SCI)

Section 19 A/D Converter (ADC)

Section 20 D/A Converter (DAC)

Section 21 Hitachi User-Debugging Interface (H-UDI)

Section 22 Power-Down Modes

Section 24 Electrical Characteristics

Appendix

Section 1 Overview

The SH7706 is a RISC microprocessor that integrates a Hitachi-original RISC-type SuperH™ architecture SH-3 CPU as its core that has peripheral functions required for system configuration. The CPU of this LSI has upper compatibility with the SH-1 and SH-2 at object code level. This LSI incorporates a memory management unit (MMU) that has a 128-entry 4-way set associative translation lookaside buffer (TLB).

The LSI incorporates the following peripheral functions: an on-chip direct memory access controller (DMAC) that enables high-speed data transfer and a bus state controller (BSC) that enables direct connection to different types of memory. The LSI also incorporates a serial communication interface, an A/D converter, a D/A converter, a timer, and a realtime clock that enable system configuration at low cost.

A built-in power management function enables dynamic control of power consumption. Thus, this LSI is optimum for portable electronic devices such as PDAs that require both high performance and low power.

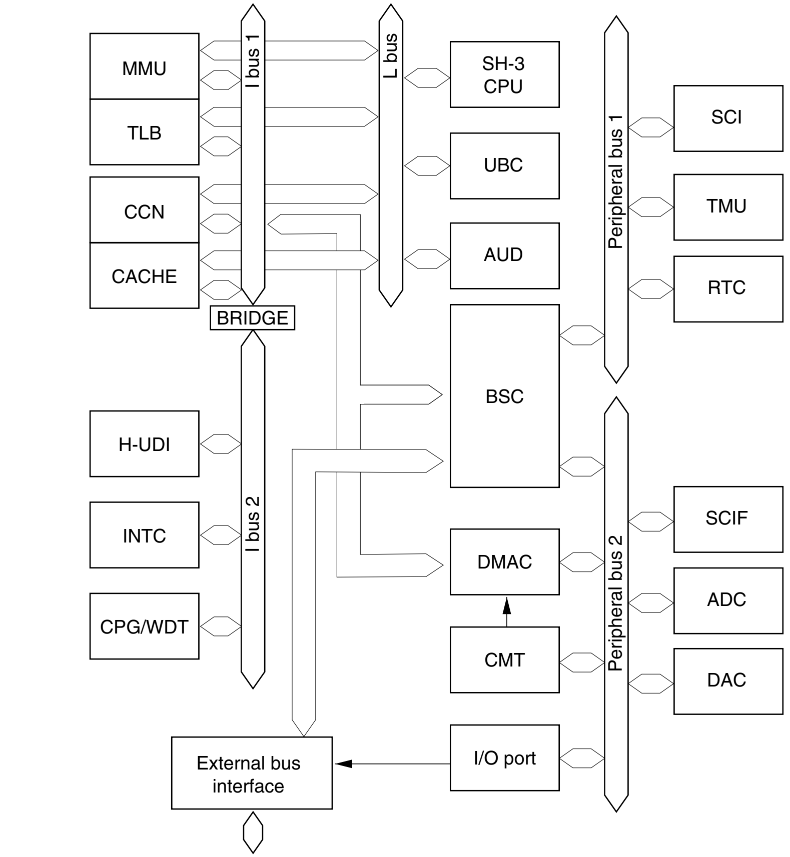
The SH7706 incorporates a Hitachi user debug interface (H-UDI) and an advanced user debugger (AUD) to support emulator functions such as E10A. This LSI also incorporates a user break controller (UBC) for self debugging.

1.1 Features

- Original Hitachi SuperH architecture
- Object code level compatible with SH-1, SH-2 and SH-3
- 32-bit RISC-type instruction set
  - Instruction length: 16-bit fixed length
  - Improved code efficiency
  - Load-store architecture
  - Delayed branch instructions
  - Instruction set oriented for C language
- Five-stage pipeline
- Instruction execution time: one instruction/cycle for basic instructions
- General-register: Sixteen 32-bit general registers
- Control-register: Eight 32-bit control registers
- System-register: Four 32-bit system registers
- 32-bit internal data bus
- Logical address space: 4 Gbytes
- Space identifier ASID: 8 bits, 256 logical address space

- Abundant Peripheral Functions
  - Memory Management Unit (MMU)
  - User Break Controller (UBC)
  - Bus state Controller (BSC)
  - Direct Memory Access Controller (DMAC)
  - Clock Pulse Generator (CPG)
  - Watchdog Timer (WDT)
  - Timer Unit (TMU)
  - Realtime Clock (RTC)
  - Serial Communication Interface (SCI)
  - Smartcard Interface
  - Serial Communication Interface with FIFO (SCIF)
  - 10-bitits A/D converter (ADC)
  - 8-bitit D/A converter (DAC)
  - Hitachi User Debugging Interface (H-UDI)
  - Advanced User Debugger (AUD)

1.2 Block Diagram



Legend:

ADC:	A/D converter
AUD:	Advanced user debugger
BSC:	Bus state controller
CACHE:	Cache memory
CCN:	Cache memory controller
CMT:	Compare match timer
CPG/WDT:	Clock pulse generator/watchdog timer
CPU:	Central processing unit
DAC:	D/A converter
DMAC:	Direct memory access controller
H-UDI:	Hitachi user-debugging interface
INTC:	Interrupt controller
MMU:	Memory management unit
RTC:	Realtime clock
SCI:	Serial communication interface (with smart card interface)
SCIF:	Serial communication interface (with FIFO)
TLB:	Address translation buffer
TMU:	Timer unit
UBC:	User break controller

Figure 1.1 SH7706 Block Diagram

1.3 Pin Assignment

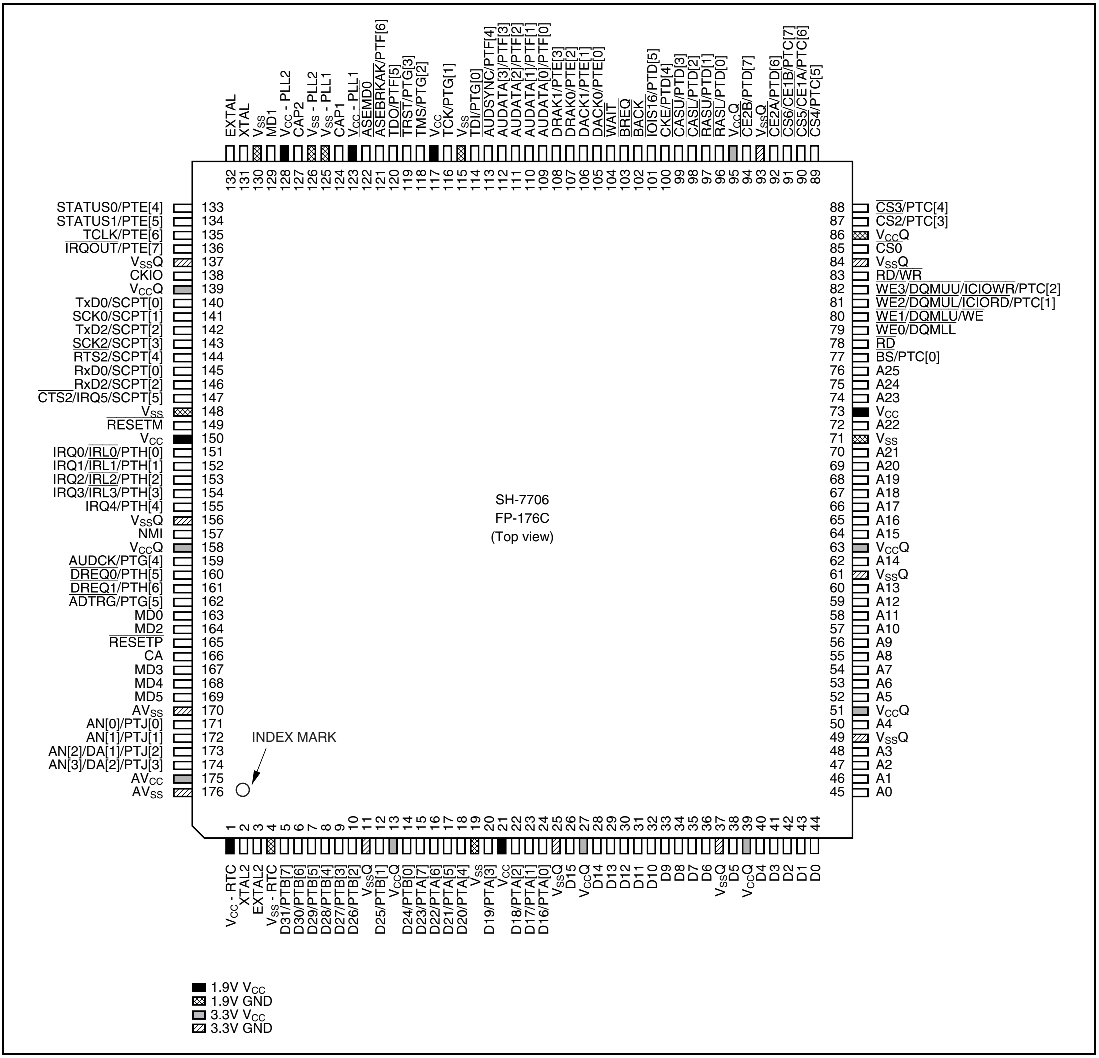
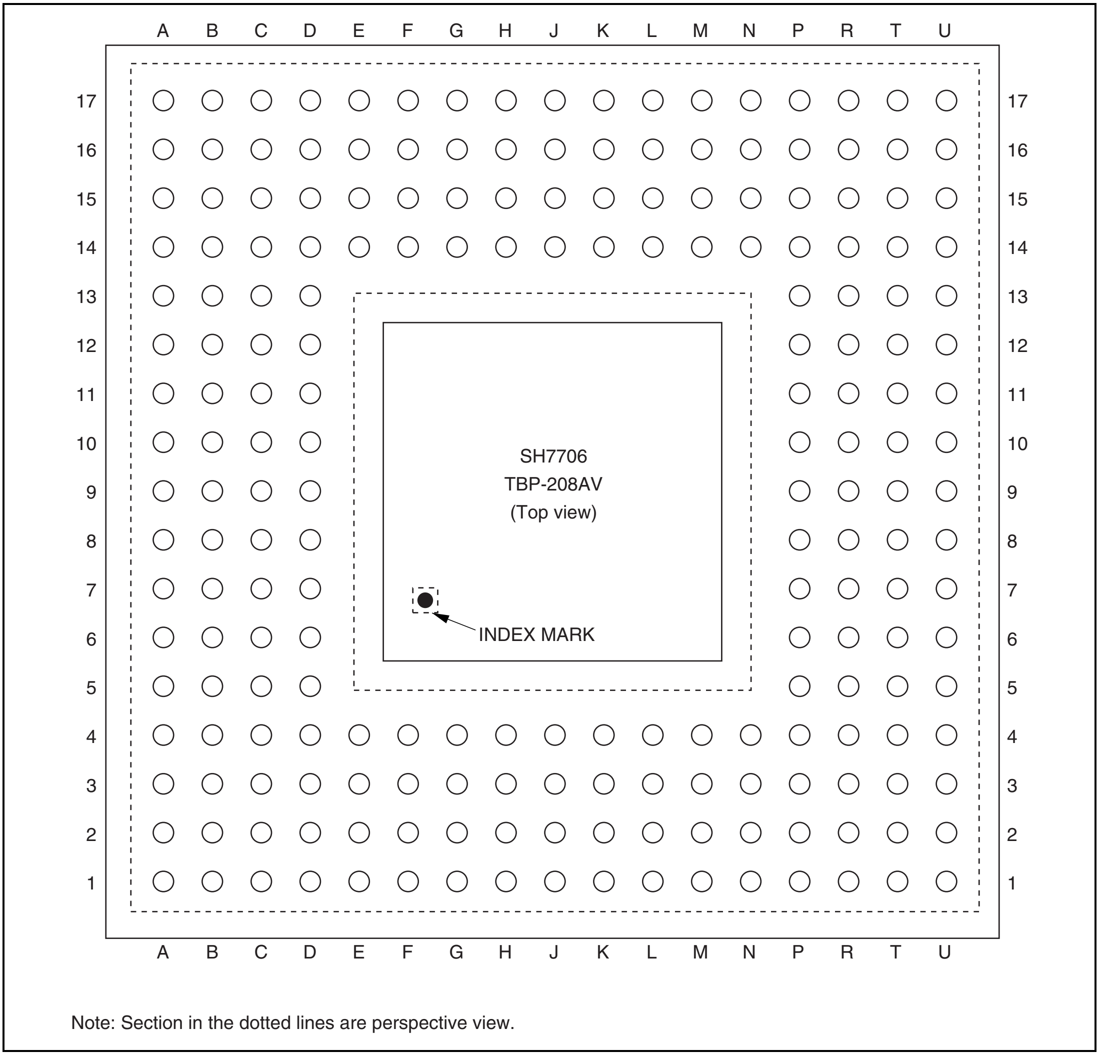


Figure 1.2 Pin Assignment (FP-176C)



Note: Section in the dotted lines are perspective view.

Figure 1.3 Pin Assignment (TBP-208AV)

1.4 Pin Function

Number of Pins				
FP-176C	**TBP-208AV**	**Pin Name**	**I/O**	**Description**
1	C3	V_{CC}-RTC*[1]	—	RTC power supply (1.9 V)
2	C2	XTAL2	O	On-chip RTC crystal oscillator pin
3	C1	EXTAL2	I	On-chip RTC crystal oscillator pin
4	D3	V_{SS}-RTC*[1]	—	RTC power supply (0 V)
5	F4	D31/PTB[7]	I/O	Data bus / input/output port B
6	F3	D30/PTB[6]	I/O	Data bus / input/output port B
7	F2	D29/PTB[5]	I/O	Data bus / input/output port B
8	F1	D28/PTB[4]	I/O	Data bus / input/output port B
9	G4	D27/PTB[3]	I/O	Data bus / input/output port B
10	G3	D26/PTB[2]	I/O	Data bus / input/output port B
11	G2	$V_{SS}Q$	—	Input/output power supply (0 V)
12	G1	D25/PTB[1]	I/O	Data bus / input/output port B
13	H4	$V_{CC}Q$	—	Input/output power supply (3.3 V)
14	H3	D24/PTB[0]	I/O	Data bus / input/output port B
15	H2	D23/PTA[7]	I/O	Data bus / input/output port A
16	H1	D22/PTA[6]	I/O	Data bus / input/output port A
17	J4	D21/PTA[5]	I/O	Data bus / input/output port A
18	J2	D20/PTA[4]	I/O	Data bus / input/output port A
19	J1	V_{SS}	—	Internal power supply (0 V)
20	J3	D19/PTA[3]	I/O	Data bus / input/output port A
21	K1	V_{CC}	—	Internal power supply (1.9 V)
22	K2	D18/PTA[2]	I/O	Data bus / input/output port A
23	K3	D17/PTA[1]	I/O	Data bus / input/output port A
24	K4	D16/PTA[0]	I/O	Data bus / input/output port A
25	L1	$V_{SS}Q$	—	Input/output power supply (0 V)
26	L2	D15	I/O	Data bus
27	L3	$V_{CC}Q$	—	Input/output power supply (3.3 V)
28	L4	D14	I/O	Data bus
29	M1	D13	I/O	Data bus

Number of Pins				
FP-176C	TBP-208AV	Pin Name	I/O	Description
30	M2	D12	I/O	Data bus
31	M3	D11	I/O	Data bus
32	M4	D10	I/O	Data bus
33	N1	D9	I/O	Data bus
34	N2	D8	I/O	Data bus
35	N3	D7	I/O	Data bus
36	N4	D6	I/O	Data bus
37	P1	$V_{SS}Q$	—	Input/output power supply (0 V)
38	P2	D5	I/O	Data bus
39	P3	$V_{CC}Q$	—	Input/output power supply (3.3 V)
40	R1	D4	I/O	Data bus
41	R2	D3	I/O	Data bus
42	P4	D2	I/O	Data bus
43	T1	D1	I/O	Data bus
44	T2	D0	I/O	Data bus
45	U1	A0	O	Address bus
46	U2	A1	O	Address bus
47	R3	A2	O	Address bus
48	T3	A3	O	Address bus
49	U3	$V_{SS}Q$	—	Input/output power supply (0 V)
50	R4	A4	O	Address bus
51	T4	$V_{CC}Q$	—	Input/output power supply (3.3 V)
52	U4	A5	O	Address bus
53	P5	A6	O	Address bus
54	R5	A7	O	Address bus
55	T5	A8	O	Address bus
56	U5	A9	O	Address bus
57	P6	A10	O	Address bus
58	R6	A11	O	Address bus
59	T6	A12	O	Address bus

Number of Pins				
FP-176C	TBP-208AV	Pin Name	I/O	Description
60	U6	A13	O	Address bus
61	P7	$V_{ss}Q$	—	Input/output power supply (0 V)
62	R7	A14	O	Address bus
63	T7	$V_{cc}Q$	—	Input/output power supply (3.3 V)
64	U7	A15	O	Address bus
65	P8	A16	O	Address bus
66	R8	A17	O	Address bus
67	T8	A18	O	Address bus
68	U8	A19	O	Address bus
69	P9	A20	O	Address bus
70	T9	A21	O	Address bus
71	U9	V_{ss}	—	Internal power supply (0 V)
72	R9	A22	O	Address bus
73	U10	V_{cc}	—	Internal power supply (1.9 V)
74	T10	A23	O	Address bus
75	P10	A24	O	Address bus
76	T11	A25	O	Address bus
77	R11	$\overline{\text{BS}}$/PTC[0]	O / I/O	Bus cycle start signal / input/output port C
78	P11	$\overline{\text{RD}}$	O	Read strobe
79	U12	$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	O	D7–D0 select signal / DQM (SDRAM)
80	T12	$\overline{\text{WE1}}$/$\overline{\text{DQMLU}}$/$\overline{\text{WE}}$	O	D15–D8 select signal / DQM (SDRAM) / write strobe (PCMCIA)
81	R12	$\overline{\text{WE2}}$/$\overline{\text{DQMUL}}$/ $\overline{\text{ICIORD}}$/PTC[1]	O / O / O / I/O	D23–D16 select signal / DQM (SDRAM) / PCMCIA input/output read / input/output port C
82	P12	$\overline{\text{WE3}}$/$\overline{\text{DQMUU}}$/ $\overline{\text{ICIOWR}}$/PTC[2]	O / O / O / I/O	D31–D24 select signal / DQM (SDRAM) / PCMCIA input/output write / input/output port C
83	U13	RD/$\overline{\text{WR}}$	O	Read/write

Number of Pins				
FP-176C	TBP-208AV	Pin Name	I/O	Description
84	R13	$V_{SS}Q$	—	Input/output power supply (0 V)
85	P13	$\overline{\text{CS0}}$	O	Chip select
86	U14	$V_{CC}Q$	—	Input/output power supply (3.3 V)
87	T14	$\overline{\text{CS2}}$/PTC[3]	O / I/O	Chip select 2 / input/output port C
88	R14	$\overline{\text{CS3}}$/PTC[4]	O / I/O	Chip select 3 / input/output port C
89	U17	$\overline{\text{CS4}}$/PTC[5]	O / I/O	Chip select 4 / input/output port C
90	T17	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/PTC[6]	O / O / I/O	Chip select 5 / CE1 (area 5 PCMCIA) / input/output port C
91	R15	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$/PTC[7]	O / O / I/O	Chip select 6 / CE1 (area 6 PCMCIA) / input/output port C
92	R16	$\overline{\text{CE2A}}$/PTD[6]	O / I/O	Area 5 PCMCIA CE2 / input/output port D
93	R17	$V_{SS}Q$	—	Input/output power supply (0 V)
94	P15	$\overline{\text{CE2B}}$/PTD[7]	O / I/O	Area 6 PCMCIA CE2 / input/output port D
95	P16	$V_{CC}Q$	—	Input/output power supply (3.3 V)
96	P17	$\overline{\text{RASL}}$/PTD[0]	O / I/O	Lower 32 Mbytes address RAS (SDRAM) / input/output port D
97	N14	$\overline{\text{RASU}}$/PTD[1]	O / I/O	Upper 32 Mbytes address RAS (SDRAM) / input/output port D
98	N15	$\overline{\text{CASL}}$/PTD[2]	O / I/O	Lower 32 Mbytes address CAS (SDRAM) / input/output port D
99	N16	$\overline{\text{CASU}}$/PTD[3]	O / I/O	Upper 32 Mbytes address CAS (SDRAM) / input/output port D
100	N17	CKE/PTD[4]	O / I/O	CK enable (SDRAM) / input/output port D
101	M14	$\overline{\text{IOIS16}}$/PTD[5]	I / I/O	IOIS16 (PCMCIA) / input port D
102	M15	$\overline{\text{BACK}}$	O	Bus acknowledge
103	M16	$\overline{\text{BREQ}}$	I	Bus request
104	M17	$\overline{\text{WAIT}}$	I	Hardware wait request
105	L14	DACK0/PTE[0]	O / I/O	DMA acknowledge 0 / input/output port E
106	L15	DACK1/PTE[1]	O / I/O	DMA acknowledge 1 / input/output port E

Number of Pins				
FP-176C	TBP-208AV	Pin Name	I/O	Description
107	L16	DRAK0/PTE[2]	O / I/O	DMA request acknowledge / input/output port E
108	L17	DRAK1/PTE[3]	O / I/O	DMA request acknowledge / input/output port E
109	K15	AUDATA[0]/PTF[0]	I/O	AUD data / input/output port F
110	K16	AUDATA[1]/PTF[1]	I/O	AUD data / input/output port F
111	K17	AUDATA[2]/PTF[2]	I/O	AUD data / input/output port F
112	J14	AUDATA[3]/PTF[3]	I/O	AUD data / input/output port F
113	J16	$\overline{\text{AUDSYNC}}$/PTF[4]	O / I/O	AUD synchronous / input/output port F
114	J17	TDI/PTG[0]	I	Data input (H-UDI) / input port G
115	J15	V_{ss}	—	Internal power supply (0 V)
116	H17	TCK/PTG[1]	I	Clock (H-UDI) / input port G
117	H16	V_{cc}	—	Internal power supply (1.9 V)
118	G16	TMS/PTG[2]	I	Mode select (H-UDI) / input port G
119	G15	$\overline{\text{TRST}}$/PTG[3]	I	Reset (H-UDI) / input port G
120	G14	TDO/PTF[5]	O / I/O	Data output (H-UDI) / input/output port F
121	F16	$\overline{\text{ASEBRKAK}}$/PTF[6]	O / I/O	ASE break acknowledge (H-UDI) / input/output port F
122	F15	$\overline{\text{ASEMDO}}$*3	I	ASE mode (H-UDI)
123	E17	V_{cc}-PLL1*2	—	PLL1 power supply (1.9 V)
124	E16	CAP1	—	PLL1 external capacitance pin
125	E15	V_{ss}-PLL1*2	—	PLL1 power supply (0 V)
126	E14	V_{ss}-PLL2*2	—	PLL2 power supply (0 V)
127	D17	CAP2	—	PLL2 external capacitance pin
128	D16	V_{cc}-PLL2*2	—	PLL2 power supply (1.9 V)
129	C17	MD1	I	Clock mode setting
130	C16	V_{ss}	—	Internal power supply (0 V)
131	B17	XTAL	O	Clock oscillator pin
132	B16	EXTAL	I	External clock / crystal oscillator pin

Number of Pins				
FP-176C	**TBP-208AV**	**Pin Name**	**I/O**	**Description**
133	A17	STATUS0/PTE[4]	O / I/O	Processor status / input/output port E
134	A16	STATUS1/PTE[5]	O / I/O	Processor status / input/output port E
135	C15	TCLK/PTE[6]	I/O	TMU or RTC clock input/output / input/output port E
136	B15	$\overline{\text{IRQOUT}}$/PTE[7]	O / I/O	Interrupt request notification / input/output port E
137	A15	$V_{SS}Q$	—	Input/output power supply (0 V)
138	C14	CKIO	I/O	System clock input/output
139	B14	$V_{CC}Q$	—	Input/output power supply (3.3 V)
140	A14	TxD0/SCPT[0]	O	SCI transmit data 0 / SC port
141	D13	SCK0/SCPT[1]	I/O	SCI clock 0 / SC port
142	C13	TxD2/SCPT[2]	O	SCIF transmit data 2 / SC port
143	B13	SCK2/SCPT[3]	I/O	SCIF clock 2 / SC port
144	A13	$\overline{\text{RTS2}}$/SCPT[4]	O / I/O	SCIF transmit request 2 / SC port
145	D12	RxD0/SCPT[0]	I	SCI receive data 0 / SC port
146	C12	RxD2/SCPT[2]	I	SCIF receive data 2 / SC port
147	B12	$\overline{\text{CTS2}}$/IRQ5/SCPT[5]	I	SCIF transmit clear / external interruption request / SC port
148	D11	V_{SS}	—	Internal power supply (0 V)
149	C11	$\overline{\text{RESETM}}$	I	Manual reset request
150	B11	V_{CC}	—	Internal power supply (1.9 V)
151	A11	IRQ0/$\overline{\text{IRL0}}$/PTH[0]	I / I / I/O	External interrupt request / input/output port H
152	D10	IRQ1/$\overline{\text{IRL1}}$/PTH[1]	I / I / I/O	External interrupt request / input/output port H
153	C10	IRQ2/$\overline{\text{IRL2}}$/PTH[2]	I / I / I/O	External interrupt request / input/output port H
154	B10	IRQ3/$\overline{\text{IRL3}}$/PTH[3]	I / I / I/O	External interrupt request / input/output port H
155	A10	IRQ4/PTH[4]	I / I/O	External interrupt request / input/output port H

Number of Pins				
FP-176C	TBP-208AV	Pin Name	I/O	Description
156	D9	$V_{ss}Q$	—	Input/output power supply (0 V)
157	B9	NMI	I	Nonmaskable interrupt request
158	A9	$V_{cc}Q$	—	Input/output power supply (3.3 V)
159	C9	AUDCK/PTG[4]	I	AUD clock / input port G
160	A8	$\overline{\text{DREQ0}}$/PTH[5]	I / I/O	DMA request / input/output port H
161	B8	$\overline{\text{DREQ1}}$/PTH[6]	I / I/O	DMA request / input/output port H
162	C8	$\overline{\text{ADTRG}}$/PTG[5]	I	Analog trigger / input port G
163	D8	MD0	I	Clock mode setting
164	B7	MD2	I	Clock mode setting
165	A6	$\overline{\text{RESETP}}$	I	Power-on reset request
166	B6	CA	I	Chip activate / hardware standby request
167	C6	MD3	I	Area 0 bus width setting
168	D6	MD4	I	Area 0 bus width setting
169	A5	MD5	I	Endian setting
170	B5	AV_{ss}	—	Analog power supply (0 V)
171	C5	AN[0]/PTJ[0]	I	A/D converter input / input port J
172	D5	AN[1]/PTJ[1]	I	A/D converter input / input port J
173	A4	AN2[2]/DA[1]/PTJ[2]	I / O / I	A/D converter input / D/A converter output / input port J
174	B4	AN3[3]/DA[0]/PTJ[3]	I / O / I	A/D converter input / D/A converter output / input port J
175	B3	AV_{cc}	—	Analog power supply (3.3 V)
176	B2	AV_{ss}	—	Analog power supply (0 V)

Notes: *1 Must be connected to the power supply even when the RTC is not used.

*2. Must be connected to the power supply even when the on-chip PLL circuits are not used (except in hardware standby mode).

*3 Must be high level when the user system is used independently without using the emulator or H-UDI.

1. Except in hardware standby mode, all V_{cc}/V_{ss} pins must be connected to the system power supply. (Supply power constantly.) In hardware standby mode, power must be supplied at least to V_{cc}–RTC and V_{ss}–RTC. If power is not supplied to V_{cc} and V_{ss} pins other than V_{cc}–RTC and V_{ss}–RTC, hold the CA pin low.
2. A1, A2, A3, A7, A12, B1, C4, C7, D1, D2, D4, D7, D14, D15, E1, E2, E3, E4, F14, F17, G17, H14, H15, K14, P14, R10, T13, T15, T16, U11, U15, and U16 must be connected to V_{ss}.

Section 2 CPU

2.1 Description of Registers

2.1.1 Privileged Mode and Banks

Processor Modes: There are two processor modes: user mode and privileged mode. The SH7706 normally operates in user mode, and enters privileged mode when an exception occurs or an interrupt is accepted. There are three kinds of registers—general registers, system registers, and control registers—and the registers that can be accessed differ in the two processor modes.

General Registers: There are 16 general registers, designated R0 to R15. General registers R0 to R7 are banked registers which are switched by a processor mode change. In privileged mode, the register bank bit (RB) in the status register (SR) defines which banked register set is accessed as general registers, and which set is accessed only through the load control register (LDC) and store control register (STC) instructions.

When the RB bit is 1, BANK1 general registers R0_BANK1–R7_BANK1 and non-banked general registers R8–R15 function as the general register set, with BANK0 general registers R0_BANK0–R7_BANK0 accessed only by the LDC/STC instructions.

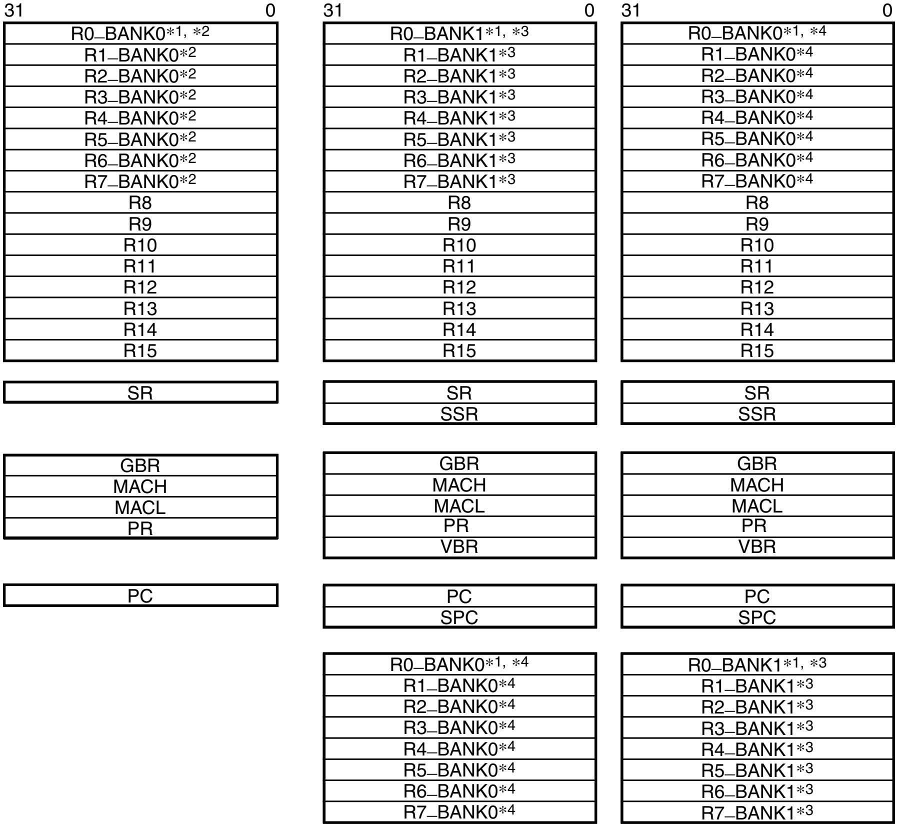
When the RB bit is 0, BANK0 general registers R0_BANK0–R7_BANK0 and nonbanked general registers R8–R15 function as the general register set, with BANK1 general registers R0_BANK1–R7_BANK1 accessed only by the LDC/STC instructions. In user mode, the 16 registers comprising bank 0 general registers R0_BANK0–R7_BANK0 and non-banked registers R8–R15 can be accessed as general registers R0–R15, and bank 1 general registers R0_BANK1–R7_BANK1 cannot be accessed.

Control Registers: Control registers comprise the global base register (GBR) and status register (SR) which can be accessed in both processor modes, and the saved status register (SSR), saved program counter (SPC), and vector base register (VBR) which can only be accessed in privileged mode. Some bits of the status register (such as the RB bit) can only be accessed in privileged mode.

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

The register configuration in each mode is shown in figures 2.1.

Switching between user mode and privileged mode is controlled by the processor mode bit (MD) in the status register.



a. User mode register configuration

b. Privileged mode register configuration (RB = 1)

c. Privileged mode register configuration (RB = 0)

Notes: *1 R0 functions as an index register in the indexed register-indirect addressing mode and indexed GBR-indirect addressing mode.

*2 Banked register

*3 Banked register
When the RB bit of the SR register is 1, the register can be accessed for general use. When the RB bit is 0, it can only be accessed with the LDC/STC instruction.

*4 Banked register
When the RB bit of the SR register is 0, the register can be accessed for general use. When the RB bit is 1, it can only be accessed with the LDC/STC instruction.

Figure 2.1 Register Configuration

Register values after a reset are shown in table 2.1.

Table 2.1 Initial Register Values

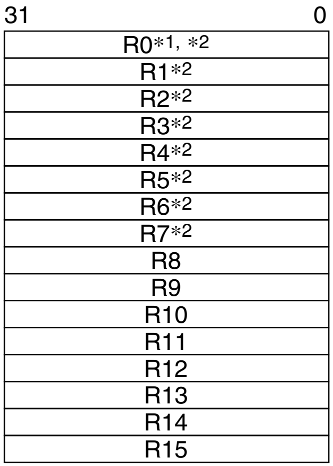
Type	Registers	Initial Value*
General registers	R0 to R15	Undefined
Control registers	SR	MD bit = 1, RB bit = 1, BL bit = 1, I3–I0 = 1111 (H'F), reserved bits = 0, others undefined
	GBR, SSR, SPC	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	H'A0000000

Note: * Initial value is set at power-on-reset or manual-reset.

2.1.2 General Registers

There are 16 general registers, designated R0 to R15. General registers R0 to R7 are banked registers, with a different R0–R7 register bank (R0_BANK0–R7_BANK0 or R0_BANK1–R7_BANK1) being accessed according to the processor mode. For details, see figure 2.1.

The general register configuration is shown in figure 2.2.



General Registers

Initialized to undefined by a reset.

Notes:

*1 R0 functions as an index register in the indexed register-indirect addressing mode and indexed GBR-indirect addressing mode. In some instructions, only R0 can be used as the source register or destination register.

*2 R0–R7 are banked registers.
In privileged mode, SR.RB specifies which banked registers are accessed as general registers (R0_BANK0–R7_BANK0 or R0_BANK1–R7_BANK1).

Figure 2.2 General Registers

2.1.3 System Registers

System registers can be accessed by the LDS and STS instructions. When an exception occurs, the contents of the program counter (PC) are saved in the saved program counter (SPC). The SPC contents are restored to the PC by the RTE instruction used at the end of the exception handling. There are three system registers, as follows.

- Multiply and accumulate register (MAC)
- Procedure register (PR)
- Program counter (PC)

The system register configuration is shown in figure 2.3.

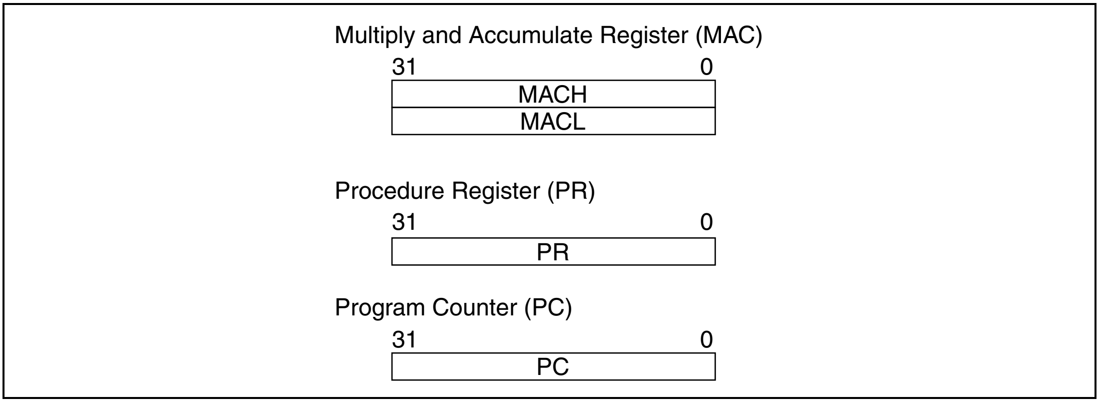


Figure 2.3 System Registers

1. Multiply and Accumulate Register (MAC)

   Multiply and Accumulate register is consist of Higher part register (MACH) and Lower part register (MACL).
   Store the results of multiply-and-accumulate operations.

   Initialized to undefined by a reset.

2. Procedure Register (PR)

   Stores the return address for exiting a subroutine procedure.

   Initialized to undefined by a reset.

3. Program Counter (PC)

   Indicates the address four addresses (two instructions) ahead of the currently executing instruction. Initialized to H'A0000000 by a reset.

2.1.4 Control Registers

Control registers can be accessed in privileged mode using the LDC and STC instructions. The GBR register can also be accessed in user mode. There are five control registers, as follows:

- Status register (SR)
- Saved status register (SSR)
- Saved program counter (SPC)
- Global base register (GBR)
- Vector base register (VBR)

The control register configuration is shown in figure 2.4.

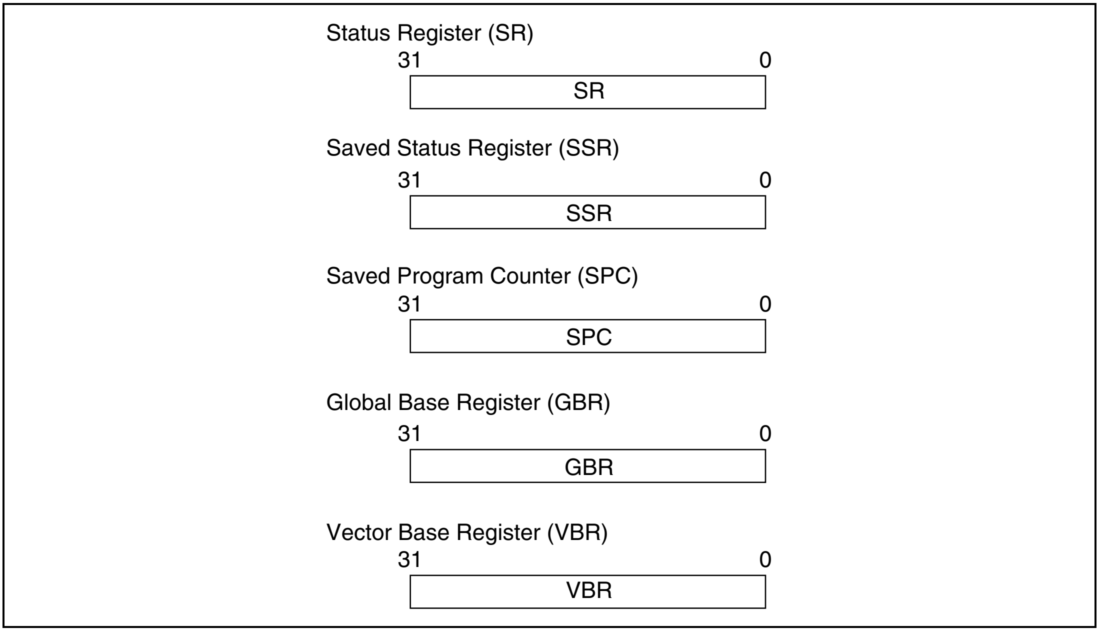

Status Register (SR)
31
0
SR
Saved Status Register (SSR)
31
0
SSR
Saved Program Counter (SPC)
31
0
SPC
Global Base Register (GBR)
31
0
GBR
Vector Base Register (VBR)
31
0
VBR

Figure 2.4 Control Registers

- **Status Register (SR)**

The information of system status are set in this register.

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved These bits always read as 0, and the write value should always be 0.
30	MD	1	R/W	Processor operation mode bit Indicates the processor operation mode. 0: User mode 1: Privileged mode MD is set to 1 when an exception or interruption is occurred.
29	RB	1	R/W	Register bank bit Determines the bank of general registers R0–R7 used in processing mode. 1: R0_BANK1–R7_BANK1 and R8–R15 are general registers, and R0_BANK0–R7_BANK0 can be accessed by LDC/STC instructions. 0: R0_BANK0–R7_BANK0 and R8–R15 are general registers, and R0_BANK1–R7_BANK1 can be accessed by LDC/STC instructions. RB is set to 1 when an exception or interruption is occured.
28	BL	1	R/W	Block bit 0: Exceptions and interrupts are accepted. 1: Exceptions and interrupts are suppressed. See section 4, Exception Handling, for details. BL is set to 1 when an exception or interruption is occured.
27 to 13	—	0	R	Reserved These bits always read as 0, and the write value should always be 0.
12	CL	0	R/W	Cache lock bit 0: Cache look function is disabled. 1: Cache look function is enabled.

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved
10	—	0	R	These bits always read as 0, and the write value should always be 0.
9	M	—	R/W	M bit
8	Q	—	R/W	Q bit
				Used by the DIV0S/U and DIV1 instructions.
7	I3	1	R/W	Interrupt mask bits
6	I2	1	R/W	4-bit field indicating the interrupt request mask level.
5	I1	1	R/W	
4	I0	1	R/W	I3–I0 do not change to the interrupt acceptance level when an interrupt is occured.
3	—	0	R	Reserved
2	—	0	R	These bits always read as 0, and the write value should always be 0.
1	S	—	R/W	S bit Used by the MAC instruction.
0	T	—	R/W	T bit Used by the MOVT, CMP/cond, TAS, TST, BT, BF, SETT, CLRT, and DT instructions to indicate true (1) or false (0). Used by the ADDV/C, SUBV/C, DIV0U/S, DIV1, NEGC, SHAR/L, SHLR/L, ROTR/L, and ROTCR/L instructions to indicate a carry, borrow, overflow, or underflow.

Note: The M, Q, S and T bits can be set or cleared by special instructions in user mode. Their values are undifined after a reset. All other bits can be read or written in privileged mode.

- Saved Status Register (SSR)
  Stores current SR value at time of exception to indicate processor status in return to instruction stream from exception handler.
  Initialized to undefined by a reset.
- Saved Program Counter (SPC)
  Stores current PC value at time of exception to indicate return address at completion of exception
  handling.
  Initialized to undefined by a reset.
- Global Base Register (GBR)
  Stores base address of GBR-indirect addressing mode. The GBR-indirect addressing mode is

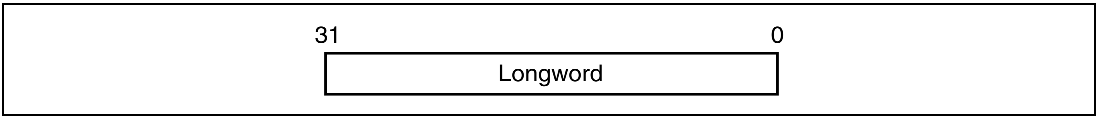
used for on-chip supporting module register area data transfers and logic operations.
The GBR register can also be accessed in user mode.
Initialized to undefined by a reset.

- **Vector Base Register (VBR)**
  Stores base address of exception handling vector area.
  Initialized to H'0000000 by a reset.

2.2 Data Formats

2.2.1 Data Format in Registers

Register operands are always longwords (32 bits). When a memory operand is only a byte (8 bits) or a word (16 bits), the sign is extended to the lognword, and stores into the register.



2.2.2 Data Format in Memory

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

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

Big-endian or little-endian byte order can be selected for the data format. The endian mode should be set with the MD5 external pin in a power-on reset. Big-endian mode is selected when the MD5 pin is low, and little-endian when high. The endian mode cannot be changed dynamically. Bit positions are numbered left to right from most-significant to least-significant. Thus, in a 32-bit longword, the leftmost bit, bit 31, is the most significant bit and the rightmost bit, bit 0, is the least significant bit.

The data format in memory is shown in figure 2.5.

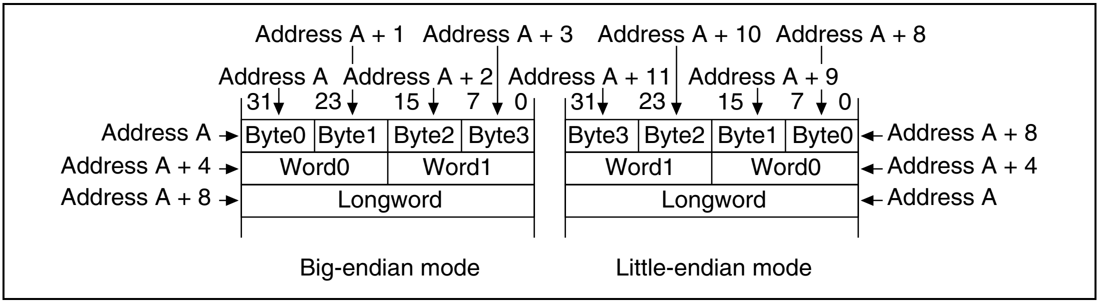


Figure 2.5 Data Format in Memory

2.3 Instruction Features

2.3.1 Execution Environment

Data Length: The instruction set is implemented with fixed-length 16-bit wide instructions executed in a pipelined sequence with single-cycle execution for most instructions. All operations are executed in 32-bit longword units. Memory can be accessed in 8-bit byte, 16-bit word, or 32-bit longword units, with byte or word units sign-extended into 32-bit longwords. Literals are sign-extended in arithmetic operations (MOV, ADD, and CMP/EQ instructions) and zero-extended in logical operations (TST, AND, OR, and XOR instructions).

Load/Store Architecture: The load-store architecture is used, so basic operations are executed by the registers. Operations requiring memory access are executed in registers following register loading, except for bit-manipulation operations such as logical AND functions, which are executed directly in memory.

Delayed Branching: Unconditional branching is implemented as delayed branch operations. Pipeline disruptions due to branching are minimized by the execution of the instruction following the delayed branch instruction prior to branching. Conditional branch instructions are of two kinds, delayed and normal.

```
BRA      TRGET
ADD      R1, R0     ;ADD is executed prior to branching to TRGET
```

T bit: The T bit in the status register (SR) is used to indicate the result of compare operations, and is read as a TRUE/FALSE condition determining if a conditional branch is taken or not. To improve processing speed, the T bit logic state is modified only by specific operations. An example of how the T bit may be used in a sequence of operations is shown below.

```
ADD      #1, R0     ;T bit not modified by ADD operation
CMP/EQ   R1, R0     ;T bit set to 1 when R0 = 0
BT       TRGET      ;branch taken to TRGET when T bit = 1 (R0 = 0)
```

Literals: Byte-length literals are inserted directly into the instruction code as immediate data. To maintain the 16-bit fixed-length instruction code, word or longword literals are stored in a table in main memory rather than inserted directly into the instruction code. The memory table is accessed by the MOV instruction using PC-relative addressing with displacement, as follows:

```
MOV.W    @(disp, PC), R0
```

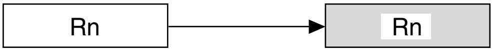
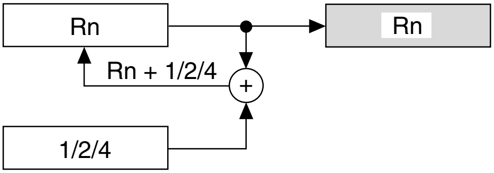
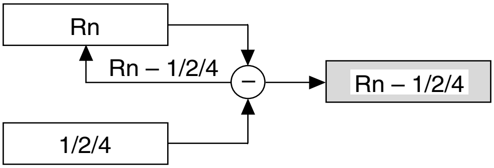
Absolute Addresses: As with word and longword literals, absolute addresses must also be stored in a table in main memory. The value of the absolute address is transferred to a register and the operand access is specified by indexed register-indirect addressing, with the absolute address loaded (like word and longword immediate data) during instruction execution.

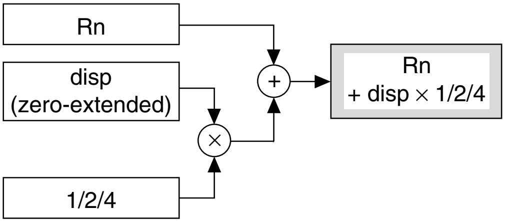
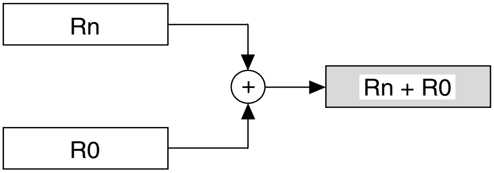
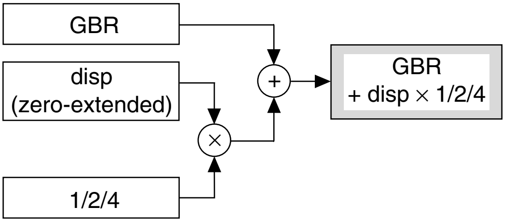
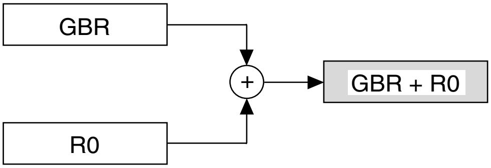
16-Bit and 32-Bit Displacements: In the same way, 16-bit and 32-bit displacements also must be stored in a table in main memory. Exactly like absolute addresses, the displacement value is transferred to a register and the operand access is specified by indexed register-indirect addressing, loading the displacement (like word and longword immediate data) during instruction execution.

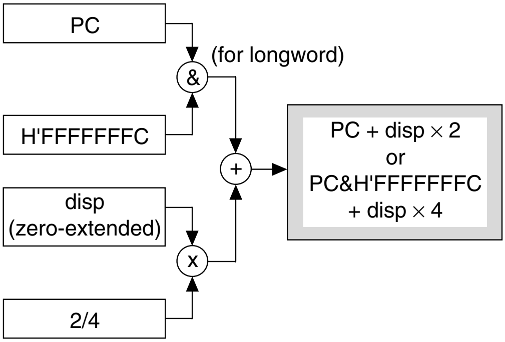
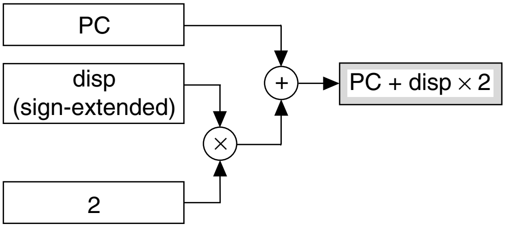
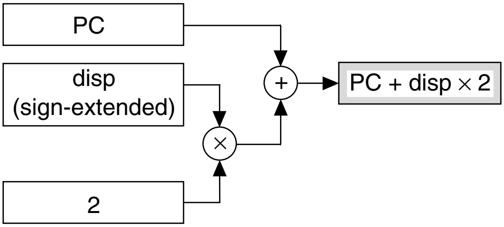
2.3.2 Addressing Modes

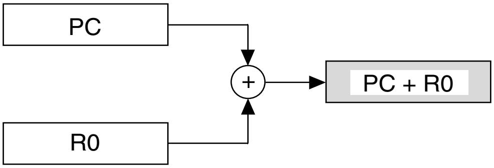
Addressing modes and effective address calculation methods are shown in table 2.2.

Table 2.2 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn
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 instruction execution Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → 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 – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction executed with Rn after calculation)

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
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.  Rn disp (zero-extended) 1/2/4 + × Rn + disp × 1/2/4	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Indexed register indirect	@(R0, Rn)	Effective address is sum of register Rn and R0 contents.  	Rn + R0
GBR indirect with displacement	@(disp:8, GBR)	Effective address is register GBR contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size.  GBR disp (zero-extended) 1/2/4 + × GBR + disp × 1/2/4	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4
Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents.  	GBR + R0

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative with displacement	@(disp:8, PC)	Effective address is register PC contents 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.  PC (for longword) & H'FFFFFFFC disp (zero-extended) x 2/4 + PC + disp × 2 or PC&H'FFFFFFFC + disp × 4	Word: PC + disp × 2 Longword: PC & H'FFFF FFFC + disp × 4
PC-relative	disp:8	Effective address is register PC contents with 8-bit displacement disp added after being sign-extended and multiplied by 2.  	PC + disp × 2
	disp:12	Effective address is register PC contents with 12-bit displacement disp added after being sign-extended and multiplied by 2.  	PC + disp × 2

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

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

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

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

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

disp:8, disp:12; PC-relative

2.3.3 Instruction Formats

Table 2.3 explains the meaning of instruction formats and source and destination operands. The meaning of the operands depends on the operation code. The following symbols are used.

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

Table 2.3 Instruction Formats

Instruction Format	Source Operand	Destination Operand	Instruction Example
0 format 15 0 xxxx xxxx xxxx xxxx	—	—	NOP
n format 15 0 xxxx nnnn xxxx xxxx	—	nnnn: register direct	MOVT Rn
	Control register or system register	nnnn: register direct	STS MACH,Rn
	Control register or system register	nnnn: register indirect with pre-decrement	STC.L SR,@–Rn
m format 15 0 xxxx mmmm xxxx xxxx	mmmm: register direct	Control register or system register	LDC Rm,SR
	mmmm: register indirect with post-increment	Control register or system register	LDC.L @Rm+,SR
	mmmm: register indirect	—	JMP @Rm
	mmmm: PC-relative using Rm	—	BRAF Rm

Instruction Format	Source Operand	Destination Operand	Instruction Example
nm format 15 0 xxxx nnnn mmmm xxxx	mmmm: register direct	nnnn: register direct	ADD Rm,Rn
	mmmm: register indirect	nnnn: register indirect	MOV.L Rm,@Rn
	mmmm: register indirect with post-increment (multiply-and-accumulate operation) nnnn: * register indirect with post-increment (multiply-and-accumulate operation)	MACH,MACL	MAC.W @Rm+,@Rn+
	mmmm: register indirect with post-increment	nnnn: register direct	MOV.L @Rm+,Rn
	mmmm: register direct	nnnn: register indirect with pre-decrement	MOV.L Rm,@–Rn
	mmmm: register direct	nnnn: indexed register indirect	MOV.L Rm,@(R0,Rn)
md format 15 0 xxxx xxxx mmmm dddd	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp,Rm),R0
nd4 format 15 0 xxxx xxxx nnnn dddd	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(disp,Rn)

Instruction Format		Source Operand	Destination Operand	Instruction Example
nmd format	15 0 xxxx nnnn mmmm dddd	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@(disp,Rn)
		mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L @(disp,Rm),Rn
d format	15 0 xxxx xxxx dddd dddd	dddddddd: GBR indirect with displacement	R0 (register direct)	MOV.L @(disp,GBR),R0
		R0 (register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(disp,GBR)
		dddddddd: PC-relative with displacement	R0 (register direct)	MOVA @(disp,PC),R0
		dddddddd: PC-relative	—	BF label
d12 format	15 0 xxxx dddd dddd dddd	dddddddddddd: PC-relative	—	BRA label (label = disp + PC)
nd8 format	15 0 xxxx nnnn dddd dddd	dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn
i format	15 0 xxxx xxxx iiii iiii	iiiiiiii: immediate	Indexed GBR indirect	AND.B #imm, @(R0,GBR)
		iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
		iiiiiiii: immediate	—	TRAPA #imm
ni format	15 0 xxxx nnnn iiii iiii	iiiiiiii: immediate	nnnn: register direct	ADD #imm,Rn

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

.

2.4 Instruction Set

2.4.1 Instruction Set Classified by Function

The SH7706 instruction set includes 68 basic instruction types, as listed in table 2.4.

Table 2.4 Classification of Instructions

Classification	Types	Operation Code	Function	No. of Instructions
Data transfer	5	MOV	Data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operations	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Initialization of signed division	
		DIV0U	Initialization of unsigned division	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate operation, double-precision multiply-and-accumulate operation	

Classification	Types	Operation Code	Function	No. of Instructions
Arithmetic operations (cont)	21	MUL	Double-precision multiplication (32 × 32 bits)	33
		MULS	Signed multiplication (16 × 16 bits)	
		MULU	Unsigned multiplication (16 × 16 bits)	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow check	
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	16
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
		SHAD	Dynamic arithmetic shift	
		SHLD	Dynamic logical shift	

Classification	Types	Operation Code	Function	No. of Instructions
Branch	9	BF	Conditional branch, delayed conditional branch (T = 0)	11
		BT	Conditional branch, delayed conditional branch (T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control	15	CLRMAC	MAC register clear	75
		CLRT	Clear T bit	
		CLRS	Clear S bit	
		LDC	Load to control register	
		LDS	Load to system register	
		LDTLB	Load PTE to TLB	
		NOP	No operation	
		PREF	Prefetch data to cache	
		RTE	Return from exception handling	
		SETS	Set S bit	
		SETT	Set T bit	
		SLEEP	Shift to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	68			188

The instruction codes are listed from table 2.5 to 2.10. Those table are described according to the following itemes.

Item	Format	Explanation
Instruction mnemonic	OP.Sz SRC,DEST	OP: Operation code Sz: Size SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement
Instruction code	MSB ↔ LSB	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 .... 1111: R15 iiii: Immediate data dddd: Displacement*
Operation summary	→, ← (xx) M/Q/T & | ^ ~ <<n, >>n	Direction of transfer Memory operand Flag bits in SR Logical AND of each bit Logical OR of each bit Exclusive OR of each bit Logical NOT of each bit n-bit shift
Privileged mode		Indicates whether privileged mode applies
Execution cycles		Value when no wait states are inserted The execution cycles listed in the table are minimums. The actual number of cycles may be increased in cases such as the followsing: 1. When contention occurs between instruction fetches and data access 2. When the destination register of the load instruction (memory → register) and the register used by the next instruction are the same
T bit		Value of T bit after instruction is executed —: No change

Note: * Scaling (×1, ×2, ×4) is performed according to the instruction operand size.

Table 2.5 lists the data transfer instructions

Table 2.5 Data Transfer Instructions

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
MOV	#imm,Rn	imm → Sign extension → Rn	1110nnnniiiiiiii	—	1	—
MOV.W	@(disp,PC),Rn	(disp × 2 + PC) → Sign extension → Rn	1001nnnndddddddd	—	1	—
MOV.L	@(disp,PC),Rn	(disp × 4 + PC) → Rn	1101nnnndddddddd	—	1	—
MOV	Rm,Rn	Rm → Rn	0110nnnnmmmm0011	—	1	—
MOV.B	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0000	—	1	—
MOV.W	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0001	—	1	—
MOV.L	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0010	—	1	—
MOV.B	@Rm,Rn	(Rm) → Sign extension → Rn	0110nnnnmmmm0000	—	1	—
MOV.W	@Rm,Rn	(Rm) → Sign extension → Rn	0110nnnnmmmm0001	—	1	—
MOV.L	@Rm,Rn	(Rm) → Rn	0110nnnnmmmm0010	—	1	—
MOV.B	Rm,@–Rn	Rn–1 → Rn, Rm → (Rn)	0010nnnnmmmm0100	—	1	—
MOV.W	Rm,@–Rn	Rn–2 → Rn, Rm → (Rn)	0010nnnnmmmm0101	—	1	—
MOV.L	Rm,@–Rn	Rn–4 → Rn, Rm → (Rn)	0010nnnnmmmm0110	—	1	—
MOV.B	@Rm+,Rn	(Rm) → Sign extension → Rn, Rm + 1 → Rm	0110nnnnmmmm0100	—	1	—
MOV.W	@Rm+,Rn	(Rm) → Sign extension → Rn, Rm + 2 → Rm	0110nnnnmmmm0101	—	1	—
MOV.L	@Rm+,Rn	(Rm) → Rn,Rm + 4 → Rm	0110nnnnmmmm0110	—	1	—
MOV.B	R0,@(disp,Rn)	R0 → (disp + Rn)	10000000nnnndddd	—	1	—
MOV.W	R0,@(disp,Rn)	R0 → (disp × 2 + Rn)	10000001nnnndddd	—	1	—
MOV.L	Rm,@(disp,Rn)	Rm → (disp × 4 + Rn)	0001nnnnmmmmdddd	—	1	—
MOV.B	@(disp,Rm),R0	(disp + Rm) → Sign extension → R0	10000100mmmmdddd	—	1	—
MOV.W	@(disp,Rm),R0	(disp × 2 + Rm) → Sign extension → R0	10000101mmmmdddd	—	1	—
MOV.L	@(disp,Rm),Rn	(disp × 4 + Rm) → Rn	0101nnnnmmmmdddd	—	1	—
MOV.B	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0100	—	1	—

Instruction	Operation	Code	Privileged Mode	Cycles	T Bit
MOV.W Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0101	—	1	—
MOV.L Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	—	1	—
MOV.B @(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1100	—	1	—
MOV.W @(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1101	—	1	—
MOV.L @(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	1	—
MOV.B R0,@(disp,GBR)	R0 → (disp + GBR)	11000000dddddddd	—	1	—
MOV.W R0,@(disp,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	1	—
MOV.L R0,@(disp,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	1	—
MOV.B @(disp,GBR),R0	(disp + GBR) → Sign extension → R0	11000100dddddddd	—	1	—
MOV.W @(disp,GBR),R0	(disp × 2 + GBR) → Sign extension → R0	11000101dddddddd	—	1	—
MOV.L @(disp,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	1	—
MOVA @(disp,PC),R0	disp × 4 + PC → R0	11000111dddddddd	—	1	—
MOVT Rn	T → Rn	0000nnnn00101001	—	1	—
SWAP.B Rm,Rn	Rm → Swap the bottom two bytes → REG	0110nnnnmmmm1000	—	1	—
SWAP.W Rm,Rn	Rm → Swap two consecutive words → Rn	0110nnnnmmmm1001	—	1	—
XTRCT Rm,Rn	Rm: Middle 32 bits of Rn → Rn	0010nnnnmmmm1101	—	1	—

Table 2.6 lists the arithmetic instructions.

Table 2.6 Arithmetic Instructions

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
ADD	Rm,Rn	Rn + Rm → Rn	0011nnnnmmmm1100	—	1	—
ADD	#imm,Rn	Rn + imm → Rn	0111nnnniiiiiiii	—	1	—
ADDC	Rm,Rn	Rn + Rm + T → Rn, Carry → T	0011nnnnmmmm1110	—	1	Carry
ADDV	Rm,Rn	Rn + Rm → Rn, Overflow → T	0011nnnnmmmm1111	—	1	Overflow
CMP/EQ	#imm,R0	If R0 = imm, 1 → T	10001000iiiiiiii	—	1	Comparison result
CMP/EQ	Rm,Rn	If Rn = Rm, 1 → T	0011nnnnmmmm0000	—	1	Comparison result
CMP/HS	Rm,Rn	If Rn ≥ Rm with unsigned data, 1 → T	0011nnnnmmmm0010	—	1	Comparison result
CMP/GE	Rm,Rn	If Rn ≥ Rm with signed data, 1 → T	0011nnnnmmmm0011	—	1	Comparison result
CMP/HI	Rm,Rn	If Rn > Rm with unsigned data, 1 → T	0011nnnnmmmm0110	—	1	Comparison result
CMP/GT	Rm,Rn	If Rn > Rm with signed data, 1 → T	0011nnnnmmmm0111	—	1	Comparison result
CMP/PZ	Rn	If Rn ≥ 0, 1 → T	0100nnnn00010001	—	1	Comparison result
CMP/PL	Rn	If Rn > 0, 1 → T	0100nnnn00010101	—	1	Comparison result
CMP/STR	Rm,Rn	If Rn and Rm have an equivalent byte, 1 → T	0010nnnnmmmm1100	—	1	Comparison result
DIV1	Rm,Rn	Single-step division (Rn/Rm)	0011nnnnmmmm0100	—	1	Calculation result
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	0010nnnnmmmm0111	—	1	Calculation result
DIV0U		0 → M/Q/T	0000000000011001	—	1	0

Instruction	Operation	Code	Privileged Mode	Cycles	T Bit
`DMULS.L Rm,Rn`	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	`0011nnnnmmmm1101`	—	2(5)*	—
`DMULU.L Rm,Rn`	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	`0011nnnnmmmm0101`	—	2(5)*	—
`DT Rn`	Rn – 1 → Rn, if Rn = 0, 1 → T, else 0 → T	`0100nnnn00010000`	—	1	Comparison result
`EXTS.B Rm,Rn`	A byte in Rm is sign-extended → Rn	`0110nnnnmmmm1110`	—	1	—
`EXTS.W Rm,Rn`	A word in Rm is sign-extended → Rn	`0110nnnnmmmm1111`	—	1	—
`EXTU.B Rm,Rn`	A byte in Rm is zero-extended → Rn	`0110nnnnmmmm1100`	—	1	—
`EXTU.W Rm,Rn`	A word in Rm is zero-extended → Rn	`0110nnnnmmmm1101`	—	1	—
`MAC.L @Rm+,@Rn+`	Signed operation of (Rn) × (Rm) + MAC → MAC, Rn + 4 → Rn, Rm + 4 → Rm 32 × 32 + 64 → 64 bits	`0000nnnnmmmm1111`	—	2(5)*	—
`MAC.W @Rm+,@Rn+`	Signed operation of (Rn) × (Rm) + MAC → MAC, Rn + 2 → Rn, Rm + 2 → Rm 16 × 16 + 64 → 64 bits	`0100nnnnmmmm1111`	—	2(5)*	—
`MUL.L Rm,Rn`	Rn × Rm → MACL 32 × 32 → 32 bits	`0000nnnnmmmm0111`	—	2(5)*	—
`MULS.W Rm,Rn`	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	`0010nnnnmmmm1111`	—	1(3)*	—
`MULU.W Rm,Rn`	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	`0010nnnnmmmm1110`	—	1(3)*	—
`NEG Rm,Rn`	0–Rm → Rn	`0110nnnnmmmm1011`	—	1	—
`NEGC Rm,Rn`	0–Rm–T → Rn, Borrow → T	`0110nnnnmmmm1010`	—	1	Borrow

			Privileged		
Instruction	**Operation**	**Code**	**Mode**	**Cycles**	**T Bit**
SUB Rm,Rn	Rn–Rm → Rn	0011nnnnmmmm1000	—	1	—
SUBC Rm,Rn	Rn–Rm–T → Rn, Borrow → T	0011nnnnmmmm1010	—	1	Borrow
SUBV Rm,Rn	Rn–Rm → Rn, Underflow → T	0011nnnnmmmm1011	—	1	Underflow

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

Table 2.7 lists the logic operation instructions.

Table 2.7 Logic Operation Instructions

			Privileged		
Instruction	**Operation**	**Code**	**Mode**	**Cycles**	**T Bit**
AND Rm,Rn	Rn & Rm → Rn	0010nnnnmmmm1001	—	1	—
AND #imm,R0	R0 & imm → R0	11001001iiiiiiii	—	1	—
AND.B #imm,@(R0,GBR)	(R0 + GBR) & imm → (R0 + GBR)	11001101iiiiiiii	—	3	—
NOT Rm,Rn	~Rm → Rn	0110nnnnmmmm0111	—	1	—
OR Rm,Rn	Rn | Rm → Rn	0010nnnnmmmm1011	—	1	—
OR #imm,R0	R0 | imm → R0	11001011iiiiiiii	—	1	—
OR.B #imm,@(R0,GBR)	(R0 + GBR) | imm → (R0 + GBR)	11001111iiiiiiii	—	3	—
TAS.B @Rn*	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)*	0100nnnn00011011	—	3	Test result
TST Rm,Rn	Rn & Rm; if the result is 0, 1 → T	0010nnnnmmmm1000	—	1	Test result
TST #imm,R0	R0 & imm; if the result is 0, 1 → T	11001000iiiiiiii	—	1	Test result
TST.B #imm,@(R0,GBR)	(R0 + GBR) & imm; if the result is 0, 1 → T	11001100iiiiiiii	—	3	Test result
XOR Rm,Rn	Rn ^ Rm → Rn	0010nnnnmmmm1010	—	1	—
XOR #imm,R0	R0 ^ imm → R0	11001010iiiiiiii	—	1	—
XOR.B #imm,@(R0,GBR)	(R0 + GBR) ^ imm → (R0 + GBR)	11001110iiiiiiii	—	3	—

Note:* The on-chip DMAC's bus cycle is not inserted between the read and write cycles of the TAS instruction. The bus authority is not released by the BREQ.

Table 2.8 lists the shift instructions.

Table 2.8 Shift Instructions

Instruction	Operation	Code	Privileged Mode	Cycles	T Bit
`ROTL Rn`	T ← Rn ← MSB	`0100nnnn00000100`	—	1	MSB
`ROTR Rn`	LSB → Rn → T	`0100nnnn00000101`	—	1	LSB
`ROTCL Rn`	T ← Rn ← T	`0100nnnn00100100`	—	1	MSB
`ROTCR Rn`	T → Rn → T	`0100nnnn00100101`	—	1	LSB
`SHAD Rm,Rn`	Rn ≥ 0: Rn << Rm → Rn Rn < 0: Rn >> Rm → [MSB → Rn]	`0100nnnnmmmm1100`	—	1	—
`SHAL Rn`	T ← Rn ← 0	`0100nnnn00100000`	—	1	MSB
`SHAR Rn`	MSB → Rn → T	`0100nnnn00100001`	—	1	LSB
`SHLD Rm,Rn`	Rn ≥ 0: Rn << Rm → Rn Rn < 0: Rn >> Rm → [0 → Rn]	`0100nnnnmmmm1101`	—	1	—
`SHLL Rn`	T ← Rn ← 0	`0100nnnn00000000`	—	1	MSB
`SHLR Rn`	0 → Rn → T	`0100nnnn00000001`	—	1	LSB
`SHLL2 Rn`	Rn << 2 → Rn	`0100nnnn00001000`	—	1	—
`SHLR2 Rn`	Rn >> 2 → Rn	`0100nnnn00001001`	—	1	—
`SHLL8 Rn`	Rn << 8 → Rn	`0100nnnn00011000`	—	1	—
`SHLR8 Rn`	Rn >> 8 → Rn	`0100nnnn00011001`	—	1	—
`SHLL16 Rn`	Rn << 16 → Rn	`0100nnnn00101000`	—	1	—
`SHLR16 Rn`	Rn >> 16 → Rn	`0100nnnn00101001`	—	1	—

Table 2.9 lists the branch instructions.

Table 2.9 Branch Instructions

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
BF	label	If T = 0, disp × 2 + PC → PC; if T = 1, nop (where label is disp + PC)	10001011dddddddd	—	3/1*	—
BF/S	label	Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop	10001111dddddddd	—	2/1*	—
BT	label	Delayed branch, if T = 1, disp × 2 + PC → PC; if T = 0, nop	10001001dddddddd	—	3/1*	—
BT/S	label	If T = 1, disp × 2 + PC → PC; if T = 0, nop	10001101dddddddd	—	2/1*	—
BRA	label	Delayed branch, disp × 2 + PC → PC	1010dddddddddddd	—	2	—
BRAF	Rm	Delayed branch, Rm + PC → PC	0000mmmm00100011	—	2	—
BSR	label	Delayed branch, PC → PR, disp × 2 + PC → PC	1011dddddddddddd	—	2	—
BSRF	Rm	Delayed branch, PC → PR, Rm + PC → PC	0000mmmm00000011	—	2	—
JMP	@Rm	Delayed branch, Rm → PC	0100mmmm00101011	—	2	—
JSR	@Rm	Delayed branch, PC → PR, Rm → PC	0100mmmm00001011	—	2	—
RTS		Delayed branch, PR → PC	0000000000001011	—	2	—

Note:* One state when there is no branch.

Table 2.10 lists the system control instructions.

Table 2.10 System Control Instructions

Instruction	Operation	Code	Privileged Mode	Cycles	T Bit
CLRMAC	0 → MACH, MACL	0000000000101000	—	1	—
CLRS	0 → S	0000000001001000	—	1	—
CLRT	0 → T	0000000000001000	—	1	0
LDC Rm,SR	Rm → SR	0100mmmm00001110	√	5	LSB
LDC Rm,GBR	Rm → GBR	0100mmmm00011110	—	1	—
LDC Rm,VBR	Rm → VBR	0100mmmm00101110	√	1	—
LDC Rm,SSR	Rm → SSR	0100mmmm00111110	√	1	—
LDC Rm,SPC	Rm → SPC	0100mmmm01001110	√	1	—
LDC Rm,R0_BANK	Rm → R0_BANK	0100mmmm10001110	√	1	—
LDC Rm,R1_BANK	Rm → R1_BANK	0100mmmm10011110	√	1	—
LDC Rm,R2_BANK	Rm → R2_BANK	0100mmmm10101110	√	1	—
LDC Rm,R3_BANK	Rm → R3_BANK	0100mmmm10111110	√	1	—
LDC Rm,R4_BANK	Rm → R4_BANK	0100mmmm11001110	√	1	—
LDC Rm,R5_BANK	Rm → R5_BANK	0100mmmm11011110	√	1	—
LDC Rm,R6_BANK	Rm → R6_BANK	0100mmmm11101110	√	1	—
LDC Rm,R7_BANK	Rm → R7_BANK	0100mmmm11111110	√	1	—
LDC.L @Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	√	7	LSB
LDC.L @Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	—	1	—
LDC.L @Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	√	1	—
LDC.L @Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	√	1	—
LDC.L @Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	√	1	—
LDC.L @Rm+, R0_BANK	(Rm) → R0_BANK, Rm + 4 → Rm	0100mmmm10000111	√	1	—
LDC.L @Rm+, R1_BANK	(Rm) → R1_BANK, Rm + 4 → Rm	0100mmmm10010111	√	1	—
LDC.L @Rm+, R2_BANK	(Rm) → R2_BANK, Rm + 4 → Rm	0100mmmm10100111	√	1	—
LDC.L @Rm+, R3_BANK	(Rm) → R3_BANK, Rm + 4 → Rm	0100mmmm10110111	√	1	—

Instruction	Operation	Code	Privileged Mode	Cycles	T Bit
LDC.L @Rm+, R4_BANK	(Rm) → R4_BANK, Rm + 4 → Rm	0100mmmm11000111	√	1	—
LDC.L @Rm+, R5_BANK	(Rm) → R5_BANK, Rm + 4 → Rm	0100mmmm11010111	√	1	—
LDC.L @Rm+, R6_BANK	(Rm) → R6_BANK, Rm + 4 → Rm	0100mmmm11100111	√	1	—
LDC.L @Rm+, R7_BANK	(Rm) → R7_BANK, Rm + 4 → Rm	0100mmmm11110111	√	1	—
LDS Rm,MACH	Rm → MACH	0100mmmm00001010	—	1	—
LDS Rm,MACL	Rm → MACL	0100mmmm00011010	—	1	—
LDS Rm,PR	Rm → PR	0100mmmm00101010	—	1	—
LDS.L @Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	—	1	—
LDS.L @Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	—	1	—
LDS.L @Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	—	1	—
LDTLB	PTEH/PTEL → TLB	0000000000111000	√	1	—
NOP	No operation	0000000000001001	—	1	—
PREF @Rm	(Rm) → cache	0000mmmm10000011	—	1	—
RTE	Delayed branch, SSR/SPC → SR/PC	0000000000101011	√	4	—
SETS	1 → S	0000000001011000	—	1	—
SETT	1 → T	0000000000011000	—	1	1
SLEEP	Sleep	0000000000011011	√	4*	—
STC SR,Rn	SR → Rn	0000nnnn00000010	√	1	—
STC GBR,Rn	GBR → Rn	0000nnnn00010010	—	1	—
STC VBR,Rn	VBR → Rn	0000nnnn00100010	√	1	—
STC SSR,Rn	SSR → Rn	0000nnnn00110010	√	1	—
STC SPC,Rn	SPC → Rn	0000nnnn01000010	√	1	—
STC R0_BANK,Rn	R0_BANK→ Rn	0000nnnn10000010	√	1	—
STC R1_BANK,Rn	R1_BANK→ Rn	0000nnnn10010010	√	1	—
STC R2_BANK,Rn	R2_BANK→ Rn	0000nnnn10100010	√	1	—
STC R3_BANK,Rn	R3_BANK→ Rn	0000nnnn10110010	√	1	—

Instruction	Operation	Code	Privileged Mode	Cycles	T Bit
STC R4_BANK,Rn	R4_BANK→ Rn	0000nnnn11000010	√	1	—
STC R5_BANK,Rn	R5_BANK→ Rn	0000nnnn11010010	√	1	—
STC R6_BANK,Rn	R6_BANK→ Rn	0000nnnn11100010	√	1	—
STC R7_BANK,Rn	R7_BANK→ Rn	0000nnnn11110010	√	1	—
STC.L SR,@–Rn	Rn–4 → Rn, SR → (Rn)	0100nnnn00000011	√	1	—
STC.L GBR,@–Rn	Rn–4 → Rn, GBR → (Rn)	0100nnnn00010011	—	1	—
STC.L VBR,@–Rn	Rn–4 → Rn, VBR → (Rn)	0100nnnn00100011	√	1	—
STC.L SSR,@–Rn	Rn–4 → Rn, SSR → (Rn)	0100nnnn00110011	√	1	—
STC.L SPC,@–Rn	Rn–4 → Rn, SPC → (Rn)	0100nnnn01000011	√	1	—
STC.L R0_BANK, @–Rn	Rn–4 → Rn, R0_BANK → (Rn)	0100nnnn10000011	√	2	—
STC.L R1_BANK, @–Rn	Rn–4 → Rn, R1_BANK → (Rn)	0100nnnn10010011	√	2	—
STC.L R2_BANK, @–Rn	Rn–4 → Rn, R2_BANK → (Rn)	0100nnnn10100011	√	2	—
STC.L R3_BANK, @–Rn	Rn–4 → Rn, R3_BANK → (Rn)	0100nnnn10110011	√	2	—
STC.L R4_BANK, @–Rn	Rn–4 → Rn, R4_BANK → (Rn)	0100nnnn11000011	√	2	—
STC.L R5_BANK, @–Rn	Rn–4 → Rn, R5_BANK → (Rn)	0100nnnn11010011	√	2	—
STC.L R6_BANK, @–Rn	Rn–4 → Rn, R6_BANK → (Rn)	0100nnnn11100011	√	2	—
STC.L R7_BANK, @–Rn	Rn–4 → Rn, R7_BANK → (Rn)	0100nnnn11110011	√	2	—
STS MACH,Rn	MACH → Rn	0000nnnn00001010	—	1	—
STS MACL,Rn	MACL → Rn	0000nnnn00011010	—	1	—
STS PR,Rn	PR → Rn	0000nnnn00101010	—	1	—
STS.L MACH,@–Rn	Rn–4 → Rn, MACH → (Rn)	0100nnnn00000010	—	1	—
STS.L MACL,@–Rn	Rn–4 → Rn, MACL → (Rn)	0100nnnn00010010	—	1	—
STS.L PR,@–Rn	Rn–4 → Rn, PR → (Rn)	0100nnnn00100010	—	1	—
TRAPA #imm	PC → SPC, SR → SSR, imm → TRA	11000011iiiiiiii	—	6	—

Notes: * The number of cycles until the sleep state is entered.

1. The table shows the minimum number of execution cycles. The actual number of instruction execution cycles will increase in cases such as the following:

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

   b. When the destination register in a load (memory-to-register) instruction is also used by the next instruction

2. With the addressing modes using displacement (disp) listed below, the assembler descriptions in this manual show the value before scaling (×1, ×2, or ×4) is performed. This is done to clarify the operation of the chip. For the actual assembler descriptions, refer to the individual assembler notation rules.

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

2.4.2 Instruction Code Map

Table 2.11 shows the instruction code map.

Table 2.11 Instruction Code Map

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB			LSB	MD: 00	MD: 01	MD: 10	MD: 11
0000	Rn	Fx	0000				
0000	Rn	Fx	0001				
0000	Rn	00MD	0010	STC SR,Rn	STC GBR,Rn	STC VBR,Rn	STC SSR,Rn
0000	Rn	01MD	0010	STC SPC,Rn			
0000	Rn	10MD	0010	STC R0_BANK,Rn	STC R1_BANK,Rn	STC R2_BANK,Rn	STC R3_BANK,Rn
0000	Rn	11MD	0010	STC R4_BANK,Rn	STC R5_BANK,Rn	STC R6_BANK,Rn	STC R7_BANK,Rn
0000	Rm	00MD	0011	BSRF Rm		BRAF Rm	
0000	Rn	10MD	0011	PREF @Rn			
0000	Rn	Rm	01MD	MOV.B Rm,@(R0,Rn)	MOV.W Rm,@(R0,Rn)	MOV.L Rm,@(R0,Rn)	MUL.L Rm,Rn
0000	0000	00MD	1000	CLRT	SETT	CLRMAC	LDTLB
0000	0000	01MD	1000	CLRS	SETS		
0000	0000	Fx	1001	NOP	DIV0U		
0000	0000	Fx	1010				
0000	0000	Fx	1011	RTS	SLEEP	RTE	
0000	Rn	Fx	1000				
0000	Rn	Fx	1001			MOVT Rn	
0000	Rn	Fx	1010	STS MACH,Rn	STS MACL,Rn	STS PR,Rn	
0000	Rn	Fx	1011				
0000	Rn	Rm	11MD	MOV.B @(R0,Rm),Rn	MOV.W @(R0,Rm),Rn	MOV.L @(R0,Rm),Rn	MAC.L @Rm+,@Rn+
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 Rm,Rn
0010	Rn	Rm	10MD	TST Rm,Rn	AND Rm,Rn	XOR Rm,Rn	OR Rm,Rn
0010	Rn	Rm	11MD	CMP/STR Rm,Rn	XTRCT Rm,Rn	MULU.W Rm,Rn	MULSW Rm,Rn
0011	Rn	Rm	00MD	CMP/EQ Rm,Rn		CMP/HS Rm,Rn	CMP/GE Rm,Rn
0011	Rn	Rm	01MD	DIV1 Rm,Rn	DMULU.LRm,Rn	CMP/HI Rm,Rn	CMP/GT Rm,Rn
0011	Rn	Rm	10MD	SUB Rm,Rn		SUBC Rm,Rn	SUBV Rm,Rn
0011	Rn	Rm	11MD	ADD Rm,Rn	DMULS.LRm,Rn	ADDC Rm,Rn	ADDV Rm,Rn

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB			LSB	MD: 00	MD: 01	MD: 10	MD: 11
0100	Rn	Fx	0000	SHLL Rn	DT Rn	SHAL Rn	
0100	Rn	Fx	0001	SHLR Rn	CMP/PZ Rn	SHAR Rn	
0100	Rn	Fx	0010	STS.L MACH,@-Rn	STS.L MACL,@-Rn	STS.L PR,@-Rn	
0100	Rn	00MD	0011	STC.L SR,@-Rn	STC.L GBR,@-Rn	STC.L VBR,@-Rn	STC.L SSR,@-Rn
0100	Rn	01MD	0011	STC.L SPC,@-Rn			
0100	Rn	10MD	0011	STC.L R0_BANK,@-Rn	STC.L R1_BANK,@-Rn	STC.L R2_BANK,@-Rn	STC.L R3_BANK,@-Rn
0100	Rn	11MD	0011	STC.L R4_BANK,@-Rn	STC.L R5_BANK,@-Rn	STC.L R6_BANK,@-Rn	STC.L R7_BANK,@-Rn
0100	Rn	Fx	0100	ROTL Rn		ROTCL Rn	
0100	Rn	Fx	0101	ROTR Rn	CMP/PL Rn	ROTCR Rn	
0100	Rm	Fx	0110	LDS.L @Rm+,MACH	LDS.L @Rm+,MACL	LDS.L @Rm+,PR	
0100	Rm	00MD	0111	LDC.L @Rm+,SR	LDC.L @Rm+,GBR	LDC.L @Rm+,VBR	LDC.L @Rm+,SSR
0100	Rm	01MD	0111	LDC.L @Rm+,SPC			
0100	Rm	10MD	0111	LDC.L @Rm+,R0_BANK	LDC.L @Rm+,R1_BANK	LDC.L @Rm+,R2_BANK	LDC.L @Rm+,R3_BANK
0100	Rm	11MD	0111	LDC.L @Rm+,R4_BANK	LDC.L @Rm+,R5_BANK	LDC.L @Rm+,R6_BANK	LDC.L @Rm+,R7_BANK
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 Rm,MACH	LDS Rm,MACL	LDS Rm,PR	
0100	Rm/ Rn	Fx	1011	JSR @Rm	TAS.B @Rn	JMP @Rm	
0100	Rn	Rm	1100	SHAD Rm,Rn			
0100	Rn	Rm	1101	SHLD Rm,Rn			
0100	Rm	00MD	1110	LDC Rm,SR	LDC Rm,GBR	LDC Rm,VBR	LDC Rm,SSR
0100	Rm	01MD	1110	LDC Rm,SPC			
0100	Rm	10MD	1110	LDC Rm,R0_BANK	LDC Rm,R1_BANK	LDC Rm,R2_BANK	LDC Rm,R3_BANK
0100	Rm	11MD	1110	LDC Rm,R4_BANK	LDC Rm,R5_BANK	LDC Rm,R6_BANK	LDC Rm,R7_BANK
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 Rm,Rn
0110	Rn	Rm	01MD	MOV.B @Rm+,Rn	MOV.W @Rm+,Rn	MOV.L @Rm+,Rn	NOT Rm,Rn
0110	Rn	Rm	10MD	SWAP.B Rm,Rn	SWAP.W Rm,Rn	NEGC Rm,Rn	NEG Rm,Rn
0110	Rn	Rm	11MD	EXTU.B Rm,Rn	EXTU.W Rm,Rn	EXTS.B Rm,Rn	EXTS.W Rm,Rn
0111	Rn	imm		ADD #imm:8,Rn			

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB			LSB	MD: 00	MD: 01	MD: 10	MD: 11
1000	00MD	Rn	disp	MOV.B R0,@(disp:4,Rn)	MOV.W R0,@(disp:4,Rn)		
1000	01MD	Rm	disp	MOV.B @(disp:4,Rm),R0	MOV.W @(disp:4,Rm),R0		
1000	10MD	imm/disp		CMP/EQ #imm:8,R0	BT label:8		BF label:8
1000	11MD	imm/disp			BT/S label:8		BF/S label:8
1001	Rn	disp		MOV.W @(DISP:8,PC),RN			
1010	disp			BRA label:12			
1011	disp			BSR label:12			
1100	00MD	imm/disp		MOV.B R0,@(disp:8,GBR)	MOV.W R0,@(disp:8,GBR)	MOV.L R0,@(disp:8,GBR)	TRAPA #imm:8
1100	01MD	disp		MOV.B @(disp:8,GBR),R0	MOV.W @(disp:8,GBR),R0	MOV.L @(disp:8,GBR),R0	MOVA @(disp:8,PC),R0
1100	10MD	imm		TST #imm:8,R0	AND #imm:8,R0	XOR #imm:8,R0	OR #imm:8,R0
1100	11MD	imm		TST.B #imm:8,@(R0,GBR)	AND.B #imm:8,@(R0,GBR)	XOR.B #imm:8,@(R0,GBR)	OR.B #imm:8,@(R0,GBR)
1101	Rn	disp		MOV.L @(disp:8,PC),Rn			
1110	Rn	imm		MOV #imm:8,Rn			
1111	************						

Note: See the SH-3/SH-3E/SH3-DSP Programming Manual for details.

2.5 Processor States and Processor Modes

2.5.1 Processor States

The SH7706 has five processor states: the reset state, exception-handling state, bus-released state, program execution state, and power-down state.

Reset State: In this state the CPU is reset. The CPU enters the power-on reset state if the $\overline{\text{RESETP}}$ pin is low, or the manual reset state if the $\overline{\text{RESETM}}$ pin is low. See section 4, Exception Handling, for more information on resets.

In the power-on reset state, the internal states of the CPU and the on-chip supporting module registers are initialized. In the manual reset state, the internal states of the CPU and registers of on-chip supporting modules other than the bus state controller (BSC) are initialized. Since the BSC is not initialized in the manual reset state, refreshing operations continue. Refer to the register configurations in the relevant sections for further details.

Exception-Handling State: This is a transient state during which the CPU's processor state flow is altered by a reset, general exception, or interrupt exception handling.

In the case of a reset, the CPU branches to address H'A0000000 and starts executing the user-coded exception handling program.

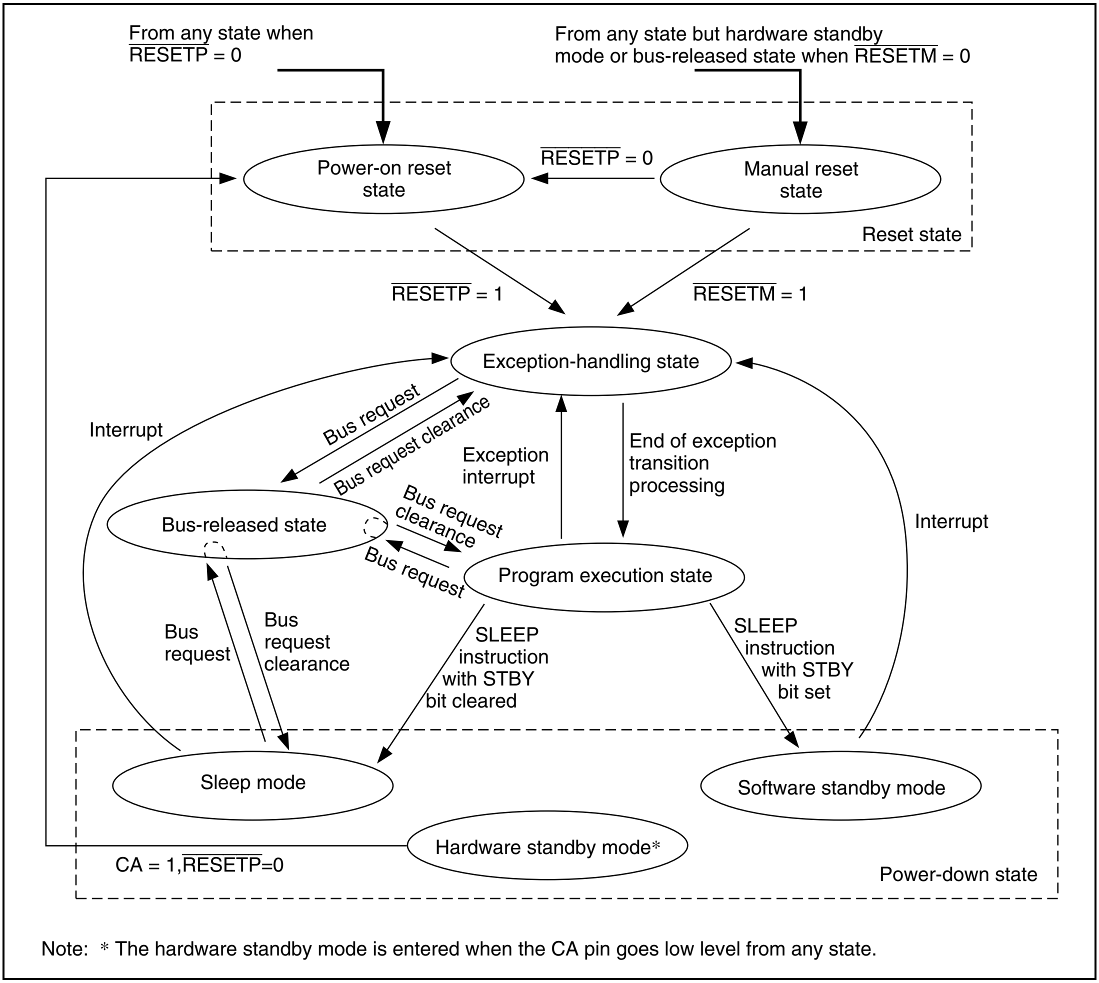
In the case of a general exception or interrupt, the program counter (PC) contents are saved in the saved program counter (SPC) and the status register (SR) contents are saved in the saved status register (SSR). The CPU branches to the start address of the user-coded exception service routine found from the sum of the contents of the vector base address and the vector offset. See section 4, Exception Processing, for more information on resets, general exceptions, and interrupts.

Program Execution State: In this state the CPU executes program instructions in sequence.

Power-Down State: In the power-down state, CPU operation halts and power consumption is reduced. There are three modes in the power-down state: sleep mode, software standby mode and hardware standby mode. The software standby mode and hardware standby mode are expressed by a generlc name, standby mode. See section 22, Power-Down Modes, for more information.

Bus-Released State: In this state the CPU has released the bus to a device that requested it.

Transitions between the states are shown in figure 2.6.


From any state when $\overline{\text{RESETP}}$ = 0
From any state but hardware standby mode or bus-released state when $\overline{\text{RESETM}}$ = 0
Power-on reset state
$\overline{\text{RESETP}}$ = 0
Manual reset state
Reset state
$\overline{\text{RESETP}}$ = 1
$\overline{\text{RESETM}}$ = 1
Exception-handling state
Interrupt
Bus request
Bus request clearance
Exception interrupt
End of exception transition processing
Interrupt
Bus-released state
Bus request clearance
Bus request
Program execution state
Bus request
Bus request clearance
SLEEP instruction with STBY bit cleared
SLEEP instruction with STBY bit set
Sleep mode
Software standby mode
Hardware standby mode*
CA = 1,$\overline{\text{RESETP}}$=0
Power-down state

Note: * The hardware standby mode is entered when the CA pin goes low level from any state.

Figure 2.6 Processor State Transitions

2.5.2 Processor Modes

There are two processor modes: privileged mode and user mode. The processor mode is determined by the processor mode bit (MD) in the status register (SR). User mode is selected when the MD bit is 0, and privileged mode when the MD bit is 1. When the reset state or exception state is entered, the MD bit is set to 1. When exception handling ends, the MD bit is cleared to 0 and user mode is entered. There are certain registers and bits which can only be accessed in privileged mode.

Section 3 Memory Management Unit (MMU)

This LSI has an on-chip memory management unit (MMU) that implements address translation. This LSI's features a resident translation look-aside buffer (TLB) that caches information for user-created address translation tables located in external memory. It enables high-speed translation of virtual addresses into physical addresses. Address translation uses the paging system and supports two page sizes (1 Kbytes and 4 Kbytes). The access right to virtual address space can be set for privileged and user modes to provide memory protection.

3.1 Role of MMU

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

The virtual memory system that came into being in this way is particularly effective in a time-sharing system (TSS) in which a number of processes are running simultaneously (figure 3.1(3)). If processes running in a TSS had to take mapping onto virtual memory into consideration while running, it would not be possible to increase efficiency. Virtual memory is thus used to reduce this load on the individual processes and so improve efficiency (figure 3.1(4)). In the virtual memory system, virtual memory is allocated to each process. The task of the MMU is to perform efficient mapping of these virtual memory areas onto physical memory. It also has a memory protection feature that prevents one process from inadvertently accessing another process's physical memory.

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

Although the functions of the MMU could also be implemented by software alone, the need for translation to be performed by software each time a process accesses physical memory would result in poor efficiency. For this reason, a buffer for address translation (translation look-aside buffer: TLB) is provided in hardware to hold frequently used address translation information. The TLB can be described as a cache for storing address translation information. Unlike cache memory, however, if address translation fails, that is, if an exception is generated, switching of address translation information is normally performed by software. This makes it possible for memory management to be performed flexibly by software.

The MMU has two methods of mapping from virtual memory to physical memory: a paging method using fixed-length address translation, and a segment method using variable-length address translation. With the paging method, the unit of translation is a fixed-size address space (usually of 1 to 64 Kbytes) called a page.

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

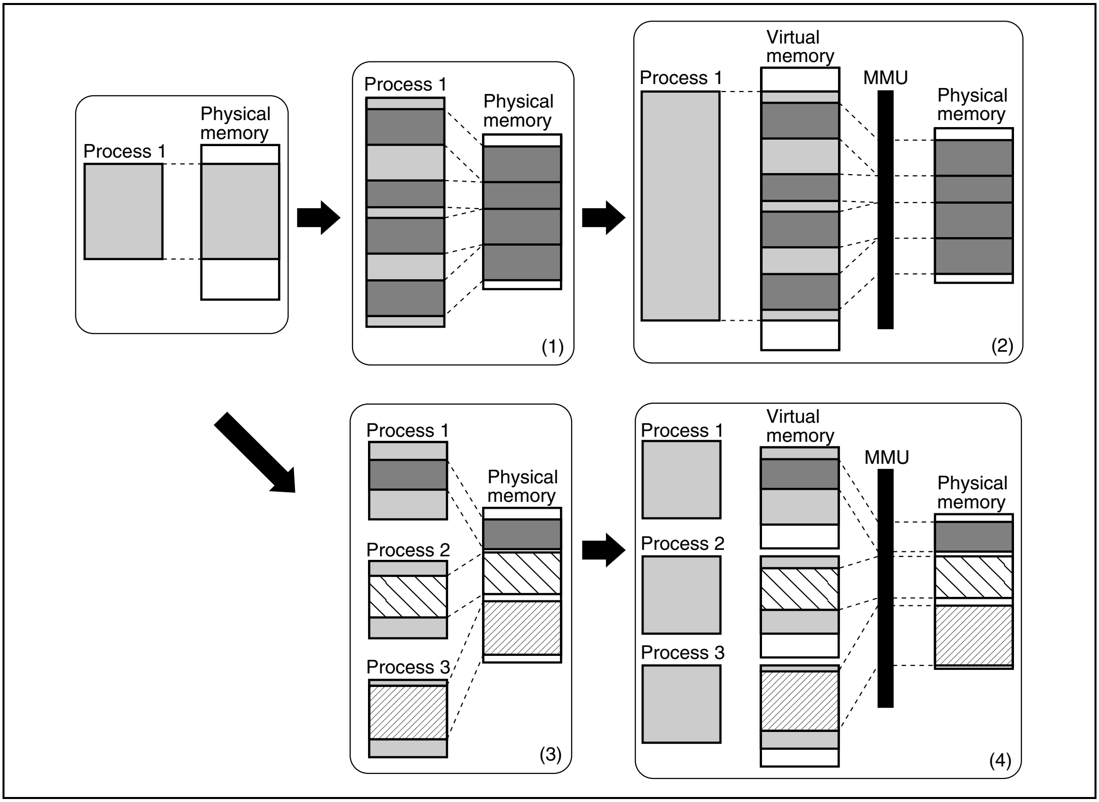


Figure 3.1 MMU Functions

3.1.1 This LSI's MMU

Virtual Address Map: This LSI uses 32-bit virtual addresses to access a 4-Gbyte virtual address space that is divided into several areas. Address space mapping is shown in figure 3.2.

In the privileged mode, the virtual address space is divided into five areas.

P0 and P3 areas are mapped to physical address spaces in page units according to the information in the address translation table. Write-back or write-through can be selected for write access by means of a cache control register (CCR) setting.

Mapping of the P1 area is fixed to physical address space (H'00000000 to H'1FFFFFFF). In the P1 area, setting a virtual address MSBs (bit 31) to 0 generates the corresponding physical address. P1 area access can be cached, write-back or write-through can be selected according to the setting of CCR whether to cache or not.

Mapping of the P2 area is fixed to physical address space (H'00000000 to H'1FFFFFFF). In the P2 area, setting the top three virtual address bits (bits 31, 30, and 29) to 0 generates the corresponding physical address. P2 area access cannot be cached.

The P1 and P2 areas are not mapped by the address translation table, so the TLB is not used and no exceptions like TLB misses occur. Initialization of MMU-related registers, exception processing, and the like are located in the P1 and P2 areas. Because the P1 area is cached, handlers that require high-speed processing are placed there.

A part of the control register in the peripheral module is allocated in P2 area.

The P4 area is used for mapping on-chip control register addresses. Address spaces from H'E0000000 to H'EFFFFFFF and from H'F4000000 to H'FBFFFFFF are reserved. An operation of this LSI is not guaranteed when these address spaces are accessed. Address space from H'F0000000 to H'F1FFFFFF is assigned to the cache, and address space from H'F2000000 to H'F3FFFFFF is assigned to the TLB. Address space from H'FC000000 to H'FFFFFFFF is a space for control registers. However, an operation of this LSI is not guaranteed when an address space that is not assigned to any control register is accessed.

In the user mode, 2 Gbytes of the virtual address space from H'00000000 to H'7FFFFFFF (area U0) can be accessed. U0 is mapped onto physical address space in page units. Write-back or write-through mode can be selected for write accesses by means of CCR setting. 2 Gbytes of the virtual address space from H'80000000 to H'FFFFFFFF cannot be accessed in the user mode. Attempting to do so creates an address error. Write-back or write-through can be selected for write access by means of the CCR setting.

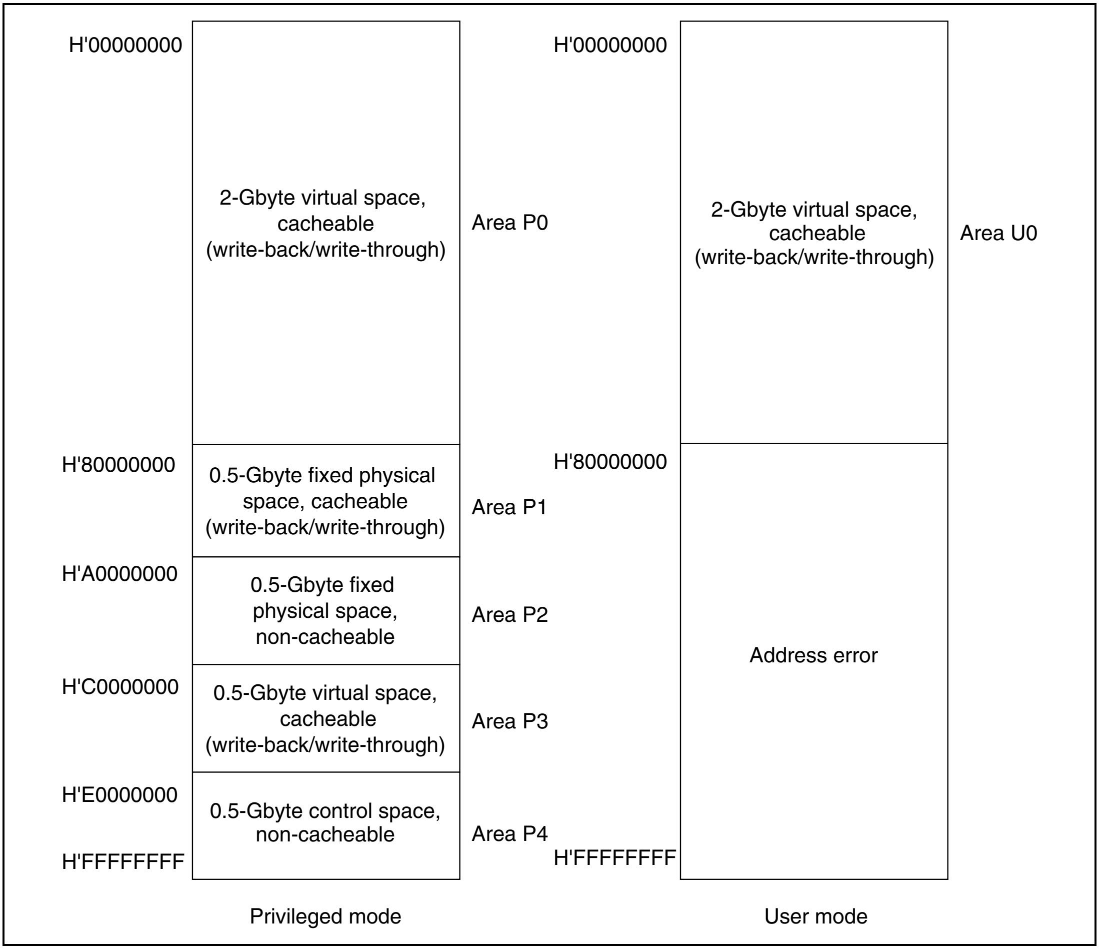

H'00000000
2-Gbyte virtual space, cacheable (write-back/write-through)
Area P0
H'80000000
0.5-Gbyte fixed physical space, cacheable (write-back/write-through)
Area P1
H'A0000000
0.5-Gbyte fixed physical space, non-cacheable
Area P2
H'C0000000
0.5-Gbyte virtual space, cacheable (write-back/write-through)
Area P3
H'E0000000
0.5-Gbyte control space, non-cacheable
Area P4
H'FFFFFFFF
Privileged mode
H'00000000
2-Gbyte virtual space, cacheable (write-back/write-through)
Area U0
H'80000000
Address error
H'FFFFFFFF
User mode

Figure 3.2 Virtual Address Space Mapping

Physical Address Space: This LSI supports a 32-bit physical address space, but the upper 3 bits are actually ignored and treated as a shadow. See section 8, Bus State Controller (BSC), for details.

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

If the virtual address is not registered in the TLB, a TLB miss exception occurs and processing will shift to the TLB miss handler. In the TLB miss handler, the TLB address translation table in external memory is searched and the corresponding physical address and the page control information are registered in the TLB. After returning from the handler, the instruction that caused the TLB miss is re-executed. When the MMU is enabled, address translation information that results in a physical address space of H'80000000–H'FFFFFFFF should not be registered in the TLB.

When the MMU is disabled, the virtual address is used directly as the physical address. As this LSI supports a 29-bit address space as the physical address space, the top 3 bits of the physical address are ignored, and constitute a shadow space. For example, addresses H'00001000 in the P0 area, H'80001000 in the P1 area, H'A0001000 in the P2 area, and H'C0001000 in the P3 area are all mapped onto the same physical address. When access to these addresses is performed with the cache enabled, an address with the top 3 bits of the physical address masked to 0 is stored in the cache address array to ensure data congruity.

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

In terms of operation, the only difference between single virtual memory mode and multiple virtual memory mode is the TLB address comparison method.

Address Space Identifier (ASID): In multiple virtual memory mode, the address space identifier (ASID) is used to differentiate between processes running in parallel and sharing virtual address space. The ASID is 8 bits in length and can be set by software setting of the ASID of the currently running process in page table entry register high (PTEH) within the MMU. When the process is switched using the ASID, the TLB does not have to be purged.

In single virtual memory mode, the ASID is used to provide memory protection for processes running simulataneously and using the virtual address space exclusively.

3.2 Description of Registers

There are five registers for MMU processing. These are located in address space area P4 and can only be accessed from privileged mode by specifying the address.

These registers for MMU processing are shown below. Refer to section 23, Control Registers Table for more details of the addresses and access sizes.

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

3.2.1 Page Table Entry Register High (PTEH)

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

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

3.2.2 Page Table Entry Register Low (PTEL)

The page table entry register low register (PTEL) is used to store the physical page number and page management information to be recorded in the TLB by the LDTLB instruction. The contents of this register are only modified by a software command.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	PPN	—	R/W	Physical page number
9	—	0	R	Page management information Refer to section 3.3 TLB Functions.
8	V	—	R/W	
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 The Translation Table Base Register (TTB)

The translation table base register (TTB) is a 32-bit register. TTB is used to store the base address of the current page table. The contents of this register are only modified in response to a software command. TTB is available to use by software for general purposes.

3.2.4 The TLB Exception Address Register (TEA)

The TLB exception address register (TEA) is a 32-bit register. TEA is used to store the virtual address corresponding to a MMU or address error exception after these exceptions has occurred. This value remains valid until the next exception or interrupt occurs.

3.2.5 MMU Control Register (MMUCR)

The MMU control register (MMUCR) makes the MMU settings. Any program that modifies MMUCR should reside in the P1 or P2 area.

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SV	—	R/W	Single virtual memory mode 0: multiple virtual memory mode 1: single virtual memory mode
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write value should always be 0.
5, 4	RC	0	R/W	Random counter A 2-bit random counter, automatically updated by hardware according to the following rules in the event of an MMU exception. When a TLB miss exception occurs, all TLB entry ways corresponding to the virtual address at which the exception occurred are checked, and if all ways are valid, 1 is added to RO; if there is one or more invalid ways, they are set by priority from way 0, in the order: way 0, way 1, way 2, way 3. In the event of an MMU exception other than a TLB miss exception, the way which caused the exception is set in RC.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	TF	0	R/W	TLB flush When 1 is set, all valid bits of TLB are cleared to 0 (flush). This bit is always reads as 0.
1	IX	0	R/W	Index mode When 0, VPN bits 16 to 12 are used as the TLB index number. When 1, the value obtained by EX-ORing ASID bits 4 to 0 in PTEH and VPN bits 16 to 12 are used as the TLB index number.
0	AT	0	R/W	Address translation Enables (valid) or disables (invalid) the MMU. 0: MMU disabled 1: MMU enabled

3.3 TLB Functions

3.3.1 Configuration of the TLB

The TLB caches address translation table information located in the external memory. The address translation table stores the physical page number translated from the virtual page number and the control information for the page, which is the unit of address translation. Figure 3.3 shows the overall TLB configuration. The TLB is 4-way set associative with 128 entries. There are 32 entries for each way. Figure 3.4 shows the configuration of virtual addresses and TLB entries.

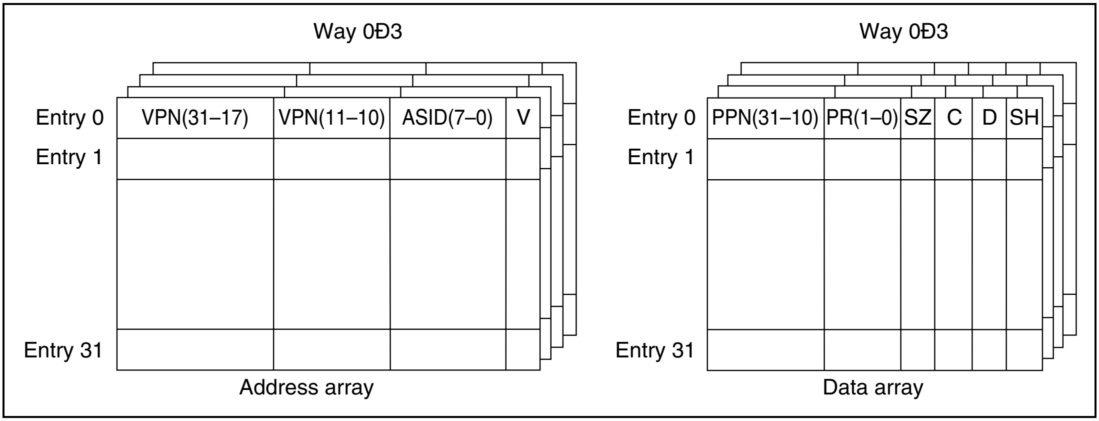
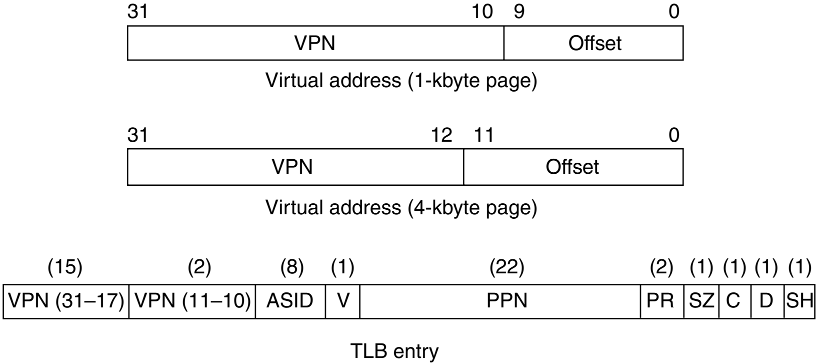


Figure 3.3 Overall Configuration of the TLB



Legend:

VPN: Virtual page number. Top 22 bits of virtual address for a 1-kbyte page, or top 20 bits of virtual address for a 4-kbyte page. Since VPN bits 16-12 are used as the index number, they are not stored in the TLB entry.

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

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

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

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

PPN: Physical page number. Top 22 bits of physical address. PPN bits 11-10 are not used in case of a 4-kbyte page. Attention must be paid to the synonym problem in case of a 1-kbyte page (see section 3.4.4 Avoiding Synonym Problems).

PR: Set the most significant bit to 0.
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 3.4 Virtual Address and TLB Structure

3.3.2 TLB Indexing

The TLB uses a 4-way set associative scheme, so entries must be selected by index. VPN bits 16 to 12 are used as the index number regardless of the page size. The index number can be generated in two different ways depending on the setting of the IX bit in MMUCR.

1. When IX = 0, VPN bits 16–12 alone are used as the index number
2. When IX = 1, VPN bits 16–12 are EX-ORed with ASID bits 4–0 to generate a 5-bit index number

The second method is used to prevent lowered TLB efficiency that results when multiple processes run simultaneously in the same virtual address space (multiple virtual memory) and a specific entry is selected by indexing of each process. Figures 3.5 and 3.6 show the indexing schemes.

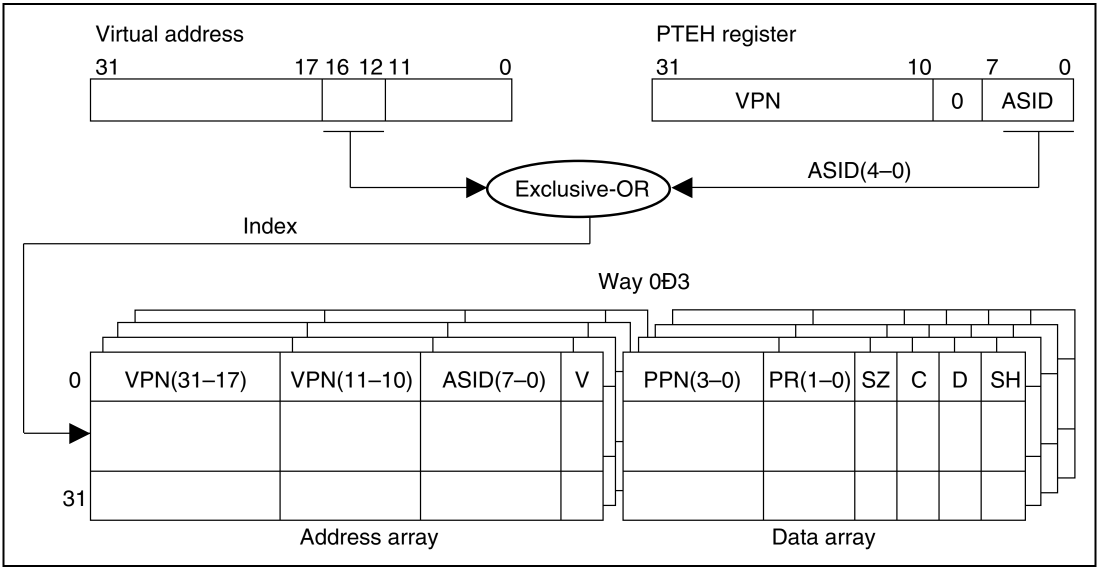


Figure 3.5 TLB Indexing (IX = 1)

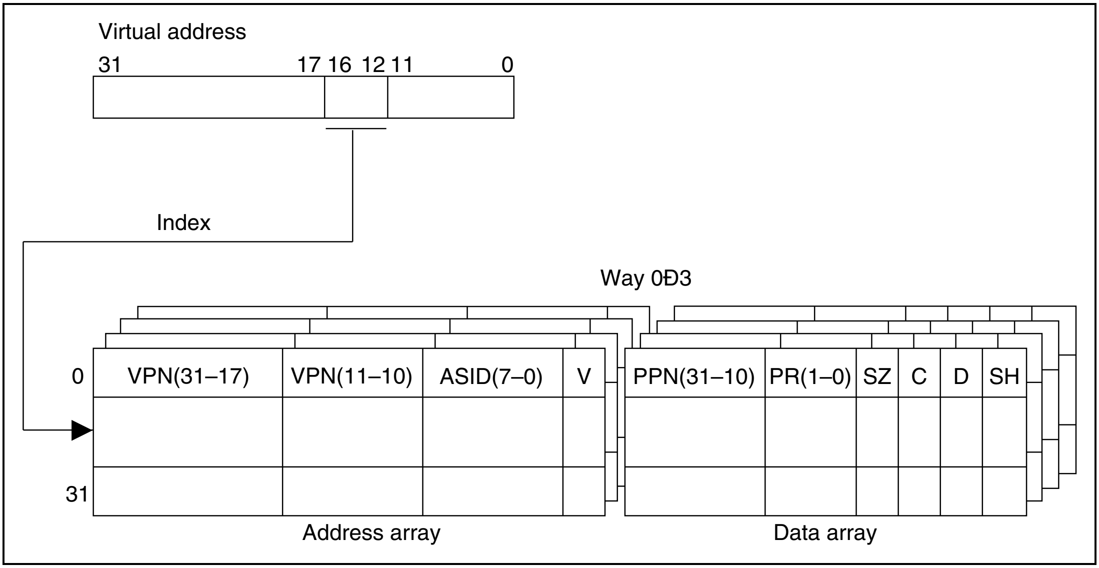


Figure 3.6 TLB Indexing (IX = 0)

3.3.3 TLB Address Comparison

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

It is necessary to have software ensure that TLB hits do not occur simultaneously in more than one way, as hardware operation is not guaranteed if this occurs. For example, if there are two identical TLB entries with the same VPN and a setting is made such that a TLB hit is made only by a process with ASID = H'FF when one is in the shared state (SH = 1) and the other in the non-shared state (SH = 0), then if the ASID in PTEH is set to H'FF, there is a possibility of simultaneous TLB hits in both these ways. It is therefore necessary to ensure that this kind of setting is not made by software.

The object compared varies depending on the page management information (SZ, SH) in the TLB entry. It also varies depending on whether the system supports multiple virtual memory or single virtual memory.

The page-size information determines whether VPN (11–10) is compared. VPN (11–10) is compared for 1-kbyte pages (SZ = 0) but not for 4-kbyte pages (SZ = 1).

The sharing information (SH) determines whether the PTEH.ASID and the ASID in the TLB entry are compared. ASIDs are compared when there is no sharing between processes (SH = 0) but not when there is sharing (SH = 1).

When single virtual memory is supported (MMUCR.SV = 1) and privileged mode is engaged (SR.MD = 1), all process resources can be accessed. This means that ASIDs are not compared when single virtual memory is supported and privileged mode is engaged. The objects of address comparison are shown in figure 3.7.

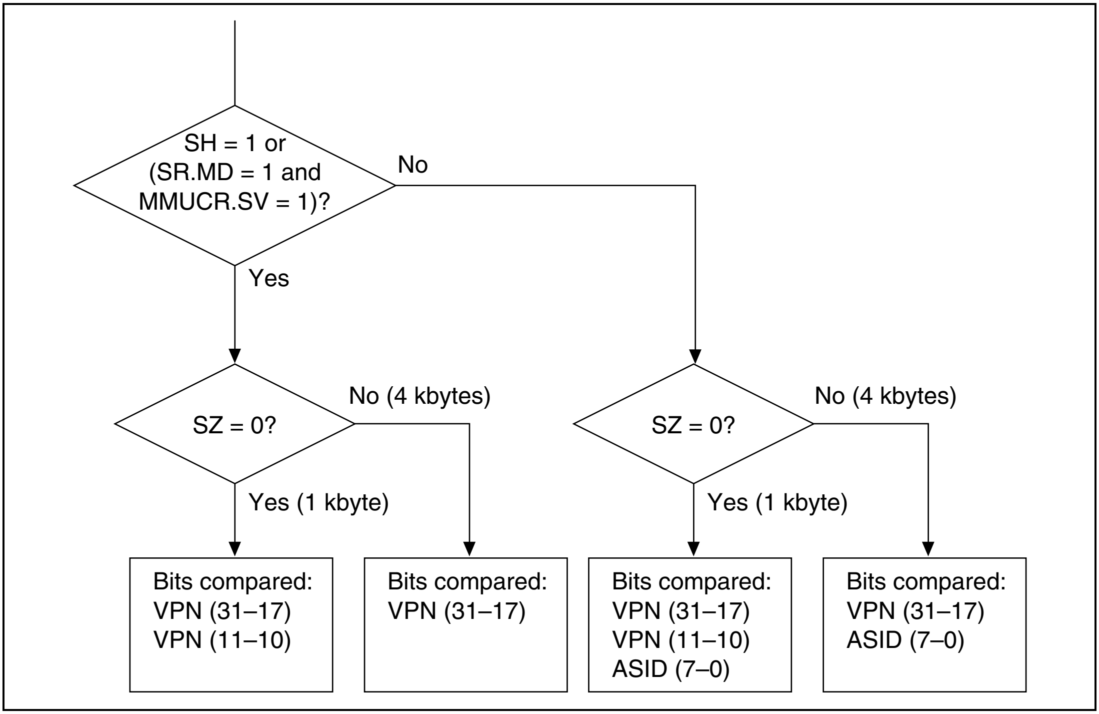


Figure 3.7 Objects of Address Comparison

3.3.4 Page Management Information

In addition to the SH and SZ bits, the page management information of TLB entries also includes D, C, and PR bits.

The D bit of a TLB entry indicates whether the page is dirty (i.e., has been written to). If the D bit is 0, an attempt to write to the page results in an initial page write exception. For physical page swapping between secondary memory and main memory, for example, pages are controlled so that a dirty page is paged out of main memory only after that page is written back to secondary memory. To record that there has been a write to a given page in the address translation table in memory, an initial page write exception is used.

The C bit in the entry indicates whether the referenced page resides in a cacheable or non-cacheable area of memory. The PR field specifies the access rights for the page in privileged and user modes and is used to protect memory. Attempts at nonpermitted accesses result in TLB protection 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 page write exception
	1	Permitted	Permitted	Permitted	Permitted
C bit	0	Permitted (no caching)	Permitted (no caching)	Permitted (no caching)	Permitted (no caching)
	1	Permitted (with caching)	Permitted (with caching)	Permitted (with caching)	Permitted (with caching)
PR bit	00	Permitted	TLB protection violation exception	TLB protection violation exception	TLB protection violation exception
	01	Permitted	Permitted	TLB protection violation exception	TLB protection violation exception
	10	Permitted	TLB protection violation exception	Permitted	TLB protection violation exception
	11	Permitted	Permitted	Permitted	Permitted

3.4 MMU Functions

3.4.1 MMU Hardware Management

There are two kinds of MMU hardware management as follows:

1. The MMU decodes the virtual address accessed by a process and performs address translation by controlling the TLB in accordance with the MMUCR settings.
2. In address translation, the MMU receives page management information from the TLB, and determines the MMU exception and whether the cache is to be accessed (using the C bit). For details of the determination method and the hardware processing, see section 3.5, MMU 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 address translation is not performed. Also, since SV and IX bit changes constitute address translation system changes, in this case, TLB flushing should be performed by simultaneously writing 1 to the TF bit also. Since MMU exceptions are not generated in the MMU disabled state with the AT bit cleared to 0, use in the disabled state must be avoided with software that does not use the MMU.
2. TLB entry recording, deletion, and reading. TLB entry recording can be done in two ways by using the LDTLB instruction, or by writing directly to the memory-mapped TLB. For TLB entry deletion and reading, the memory allocation TLB can be accessed. See section 3.4.3, MMU Instruction (LDTLB), for details of the LDTLB instruction, and section 3.6, Memory-Mapped TLB Configuration, for details of the memory-mapped TLB.
3. MMU exception processing. When an MMU exception is generated, it is handled on the basis of information set from the 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 physical memory access is enabled in the privileged mode only by clearing the share status bit (SH) to 0 to specify recording of all TLB entries. This strengthens inter-process memory protection, and enables special access levels to be created in the privileged mode only.

Recording a 1-kbyte page TLB entry may result in a synonym problem. See section 3.4.4, Avoiding Synonym Problems.

3.4.3 MMU Instruction (LDTLB)

The load TLB instruction (LDTLB) is used to record TLB entries. When the IX bit in MMUCR is 0, the LDTLB instruction changes the TLB entry in the way specified by the RC bit in MMUCR to the value specified by PTEH and PTEL, using VPN bits 16–12 specified in PTEH as the index number. When the IX bit in MMUCR is 1, the EX-OR of VPN bits 16–12 specified in PTEH and ASID bits 4–0 in PTEH are used as the index number.

Figure 3.8 shows the case where the IX bit in MMUCR is 0.

When an MMU exception occurs, the virtual page number of the virtual address that caused the exception is set in PTEH by hardware. The way is set in the RC bit of MMUCR for each exception according to the rules described in section 3.2.5 MMU Control Register. Consequently, if the LDTLB instruction is issued after setting only PTEL in the MMU exception processing routine, TLB entry recording is possible. Any TLB entry can be updated by software rewriting of PTEH and the RC bits in MMUCR.

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

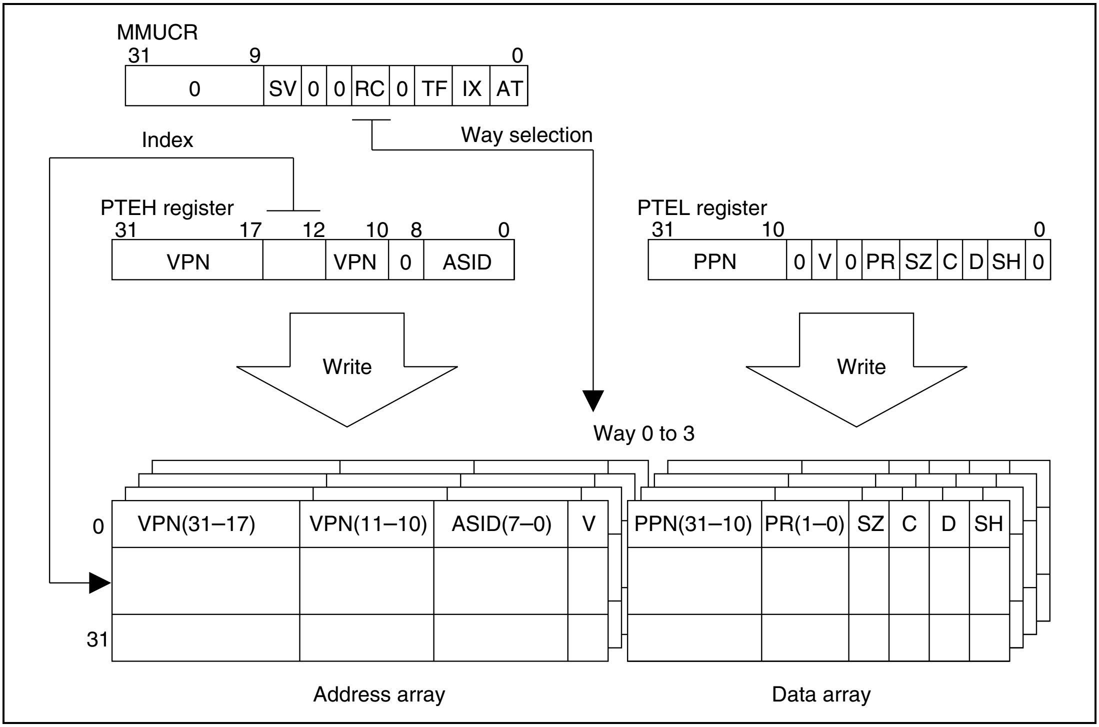


Figure 3.8 Operation of LDTLB Instruction

3.4.4 Avoiding Synonym Problems

When a 1-kbyte page is recorded in a TLB entry, a synonym problem may arise. If a number of virtual addresses are mapped onto a single physical address, the same physical address data will be recorded in a number of cache entries, and it will not be possible to guarantee data congruity. The reason why this problem only occurs when using a 1-kbyte page is explained below with reference to figure 3.9.

To achieve high-speed operation of the SH7706 cache, an index number is created using virtual address bits 11–4. When a 4-kbyte page is used, virtual address bits 11–4 are included in the offset, and since they are not subject to address translation, they are the same as physical address bits 11–4. In cache-based address comparison and recording in the address array, since the cache tag address is a physical address, physical address bits 31–10 are recorded.

When a 1-kbyte page is used, also, a cache index number is created using virtual address bits 10-4. However, in case of a 1-kbyte page, virtual address bits (11, 10) are subject to address translation and therefore may not be the same as physical address bits 11 and 10. Consequently, the physical address is recorded in a different entry from that of the index number indicated by the physical address in the cache address array.

For example, assume that, with 1-kbyte page TLB entries, TLB entries for which the following 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'00, and virtual address 2 in cache entry H'C0. Since two virtual addresses are recorded in different cache entries despite the fact that the physical addresses are the same, memory inconsistency will occur as soon as a write is performed to either virtual address. Therefore, when recording a 1-kbyte TLB entry, if the physical address is the same as a physical address already used in another TLB entry, it should be recorded in such a way that physical address bits (11, 10) ar e the same.

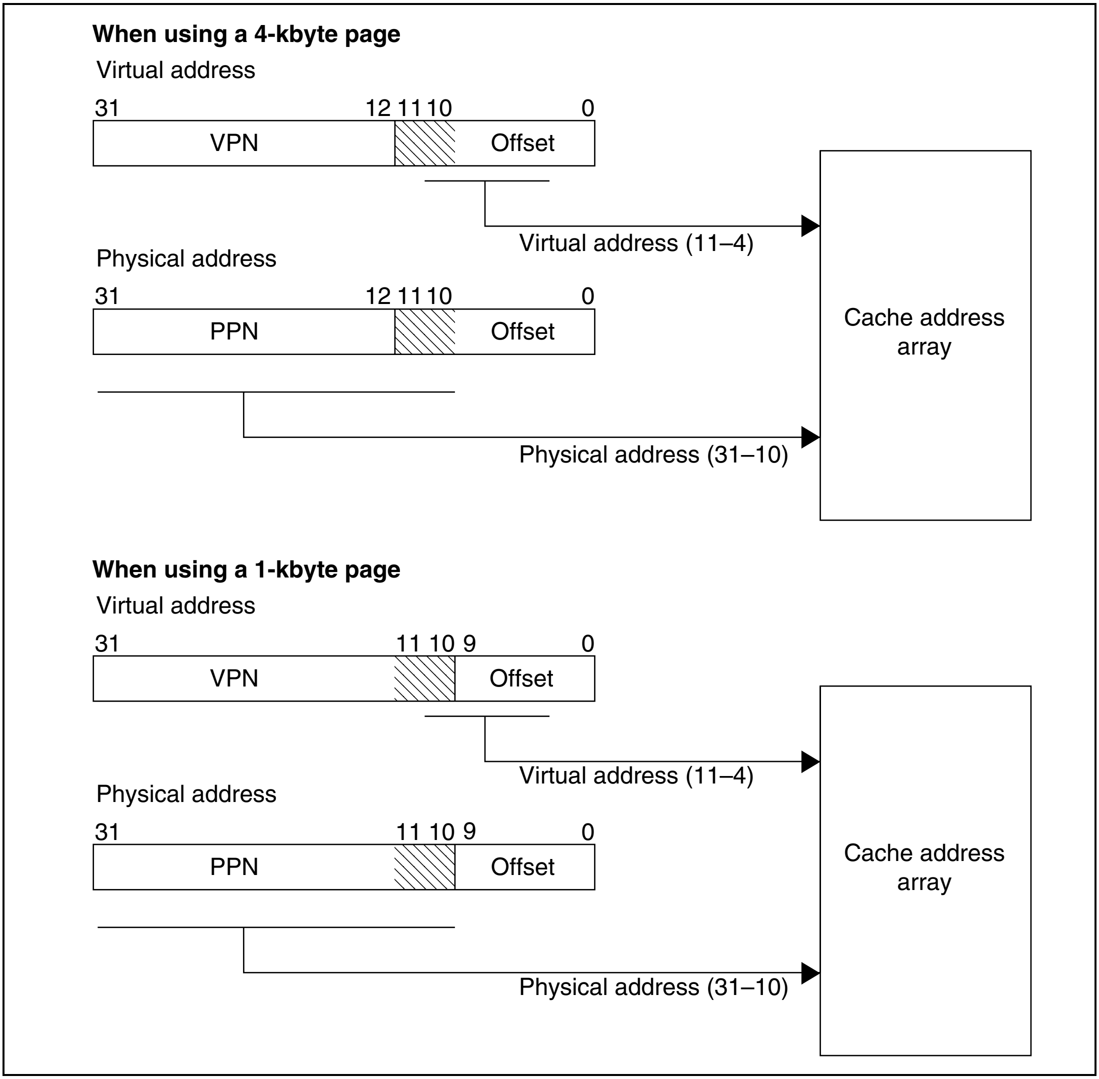


Figure 3.9 Synonym Problem

3.5 MMU Exceptions

There are four MMU exceptions: TLB miss, TLB protection violation, TLB invalid, and initial page write.

3.5.1 TLB Miss Exception

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

Hardware Operations: In a TLB miss, this LSI's hardware executes a set of prescribed operations, as follows:

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

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

1. Write the value of the physical page number (PPN) field and the protection key (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table entry recorded in the address translation table in the external memory into the PTEL register in this LSI.
2. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
3. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.

4. Issue the return from exception handler (RTE) instruction to terminate the handler routine and return to the instruction stream.

3.5.2 TLB Protection Violation Exception

A TLB protection violation exception results when the virtual address and the address array of the selected TLB entry are compared and a valid entry is found to match, but the type of access is not permitted by the access rights specified in the PR field. TLB protection violation exception processing includes both hardware and software operations.

Hardware Operations: In a TLB protection violation exception, this LSI's hardware executes a set of prescribed operations, as follows:

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

Software (TLB Protection Violation Handler) Operations: Software resolves the TLB protection violation and issues the RTE (return from exception handler) instruction to terminate the handler and return to the instruction stream.

3.5.3 TLB Invalid Exception

A TLB invalid exception results when the virtual address is compared to a selected TLB entry address array and a match is found but the entry is not valid (the V bit is 0). TLB invalid exception processing includes both hardware and software operations.

Hardware Operations: In a TLB invalid exception, this LSI's hardware executes a set of prescribed operations, as follows:

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

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

1. Write the values of the PPN, PR, SZ, C, D, SH, and V of the page table entry recorded in the external memory to the PTEL register.
2. If using software for way selection for entry replacement, write the desired value to the RC 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 stream. The RTE instruction should be issued after two LDTLB instructions.

3.5.4 Initial Page Write Exception

An initial page write exception results in a write access when the virtual address and the address array of the selected TLB entry are compared and a valid entry with the appropriate access rights is found to match, but the D (dirty) bit of the entry is 0 (the page has not been written to). Initial page write exception processing includes both hardware and software operations.

Hardware Operations: In an initial page write exception, this LSI's hardware executes a set of prescribed operations, as follows:

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

Software (Initial Page Write Handler) Operations: The software must execute the following 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 table entry in the external memory to the PTEL register.
4. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
5. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
6. Issue the RTE instruction to terminate the handler and return to the instruction stream. The RTE instruction should be issued after two LDTLB instructions.

Figure 3.10 shows the flowchart for MMU exceptions.

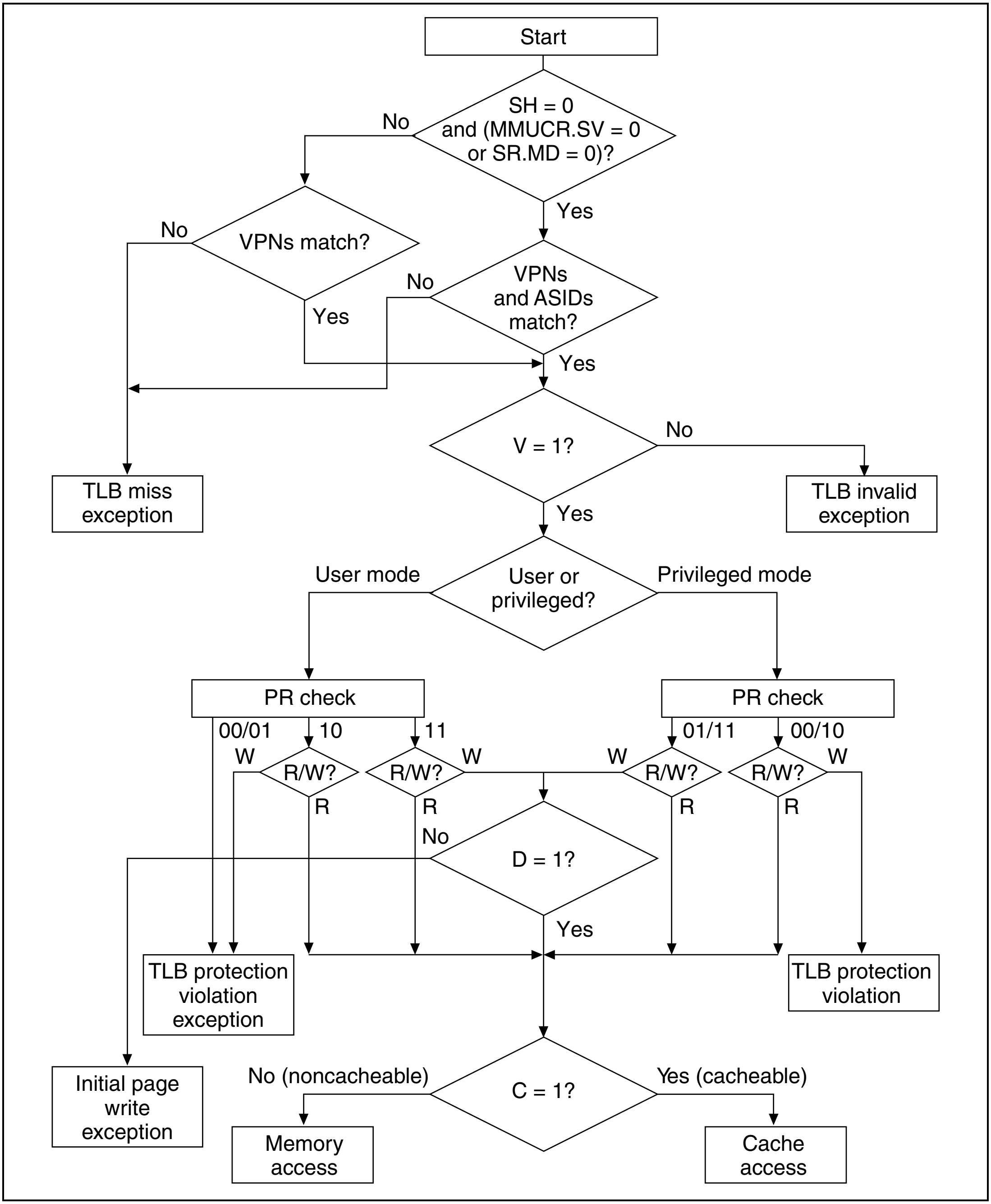
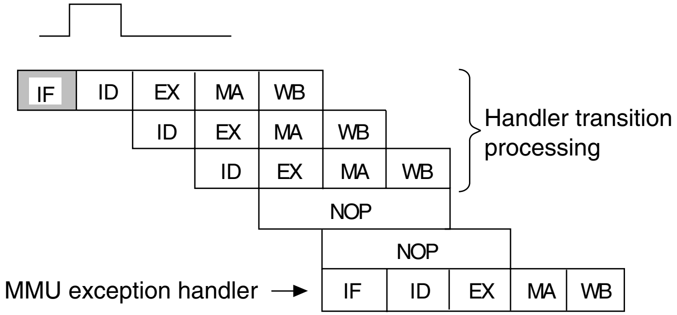


Figure 3.10 MMU Exception Generation Flowchart

3.5.5 Processing Flow in Event of MMU Exception (Same Processing Flow for Address Error)

Figure 3.11 shows the MMU exception signals in the instruction fetch mode.



▨ : Exception source stage

IF = Instruction fetch
ID = Instruction decode
EX = Instruction execution
MA = Memory access
WB = Write back
NOP = No operation

Figure 3.11 MMU Exception Signals in Instruction Fetch

Figure 3.12 shows the MMU exception signals in the data access mode.

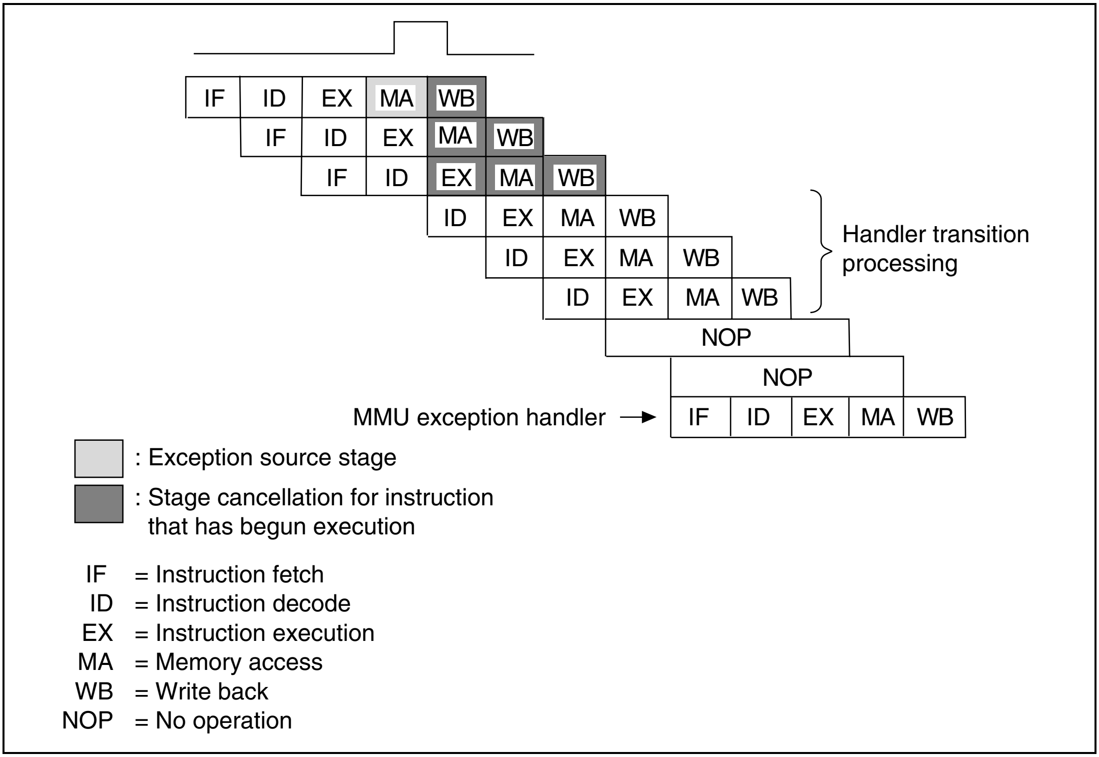


Figure 3.12 MMU Exception Signals in Data Access

3.6 Configuration of the Memory-Mapped TLB

In order for TLB operations to be managed by software, TLB contents can be read or written to in the privileged mode using the MOV instruction. The TLB is assigned to the P4 area in the virtual address space. The TLB address array (VPN, V bit, and ASID) is assigned to H'F2000000–H'F2FFFFFF, and the data array (PPN, PR, SZ, C, D, and SH bits) to H'F3000000–H'F3FFFFFF. The V bit in the address array can also be accessed from the data array. Only longword access is possible for both the address array and the data array.

3.6.1 Address Array

The address array is assigned to H'F2000000 to H'F2FFFFFF. To access an address array, the 32-bit address field (for read/write operations) and 32-bit data field (for write operations) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the VPN, V bit and ASID to be written to the address array (figure 3.13 (1)).

In the address field, specify the entry address for selecting the entry (bits 16–12), W for selecting the way (bits 9–8: 00 is way 0, 01 is way 1, 10 is way 2, 11 is way 3) and H'F2 to indicate address array access (bits 31–24). The IX bit in MMUCR indicates whether an EX-OR is taken of the entry address and ASID.

When writing, the write is performed to the entry selected with the index address and way.

When reading, the VPN, V bit, and ASID of the entry selected with the index address and way in the format of the data field in figure 3.13 without comparing addresses. 0 is written to data field bits 16-12.

To invalidate a specific entry, specify the entry and way, and write 0 to the corresponding V bit.

3.6.2 Data Array

The data array is assigned to H'F3000000 to H'F3FFFFFF. To access a data array, the 32-bit address field (for read/write operations), and 32-bit data field (for write operations) must be specified. These are specified in the general register. The address section specifies information for selecting the entry to be accessed; the data section specifies the longword data to be written to the data array (figure 3.13 (2)).

In the address section, specify the entry address for selecting the entry (bits 16–12), W for selecting the way (bits 9–8: 00 is way 0, 01 is way 1, 10 is way 2, 11 is way 3), and H'F3 to indicate data array access (bits 31–24). The IX bit in 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 address and way number. The access size of the data array is fixed at longword.

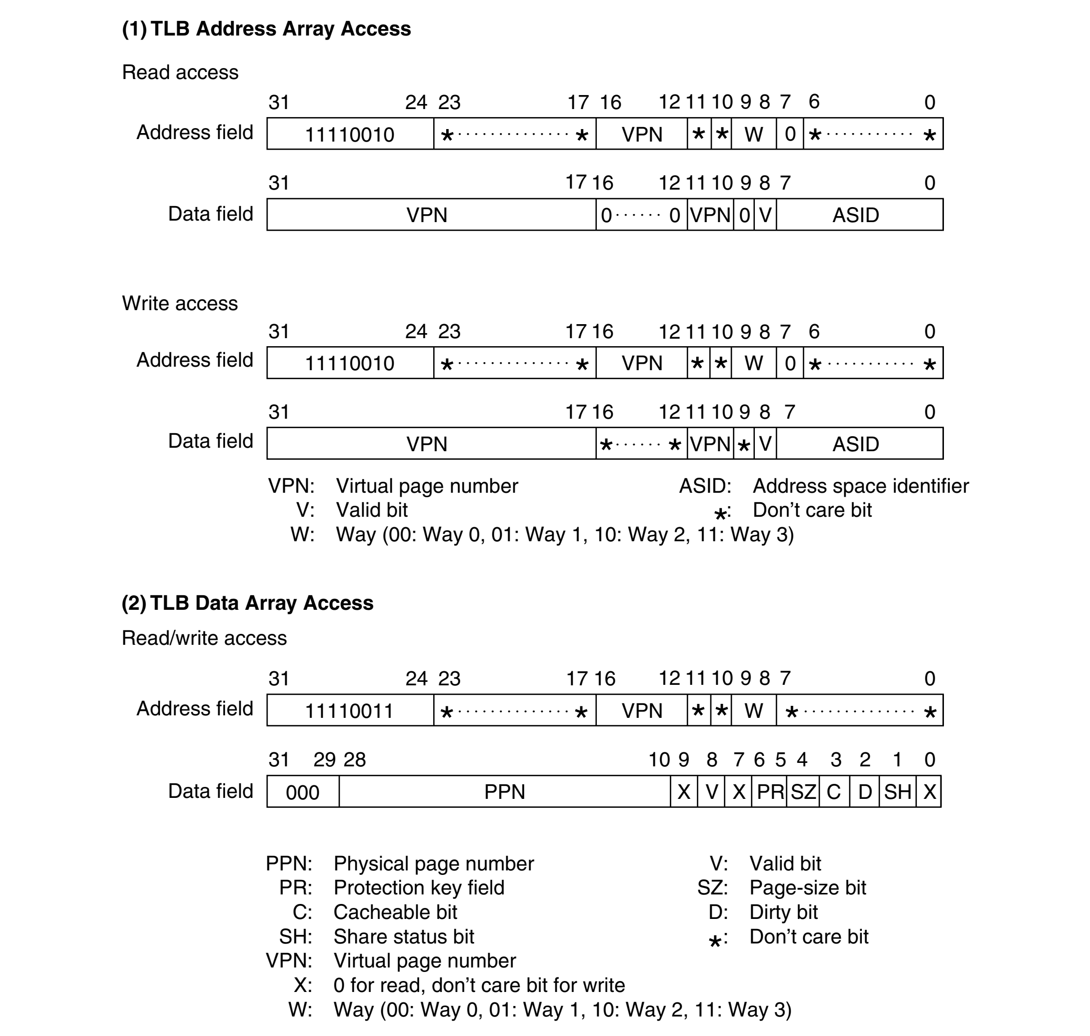


Figure 3.13 Specifying Address and Data for Memory-Mapped TLB Access

3.6.3 Usage Examples

Invalidating Specific Entries: Specific TLB entries can be invalidated by writing 0 to the entry's V bit. When the A bit is 1, the VPN and ASID specified by the write data is compared to the VPN and ASID within the TLB entry selected by the entry address and data is written to the matching way. If no match is found, there is no operation. R0 specifies the write data and R1 specifies the address.

```
 ; R0=H'1547 381C  R1=H'F201 3000
 ; MMUCR.IX=0
 ; VPN(31-17)=B'0001 0101 0100 011   VPN(11-10)=B'10   ASID=B'0001 1100
 ; corresponding entry association is made from the entry selected by
 ; the VPN(16-12)=B'1 0011 index, the V bit of the hit way is cleared to
 ; 0,achieving invalidation.
MOV.L  R0,@R1
```

Reading the Data of a Specific Entry: This example reads the data section of a specific TLB entry. The bit order indicated in the data field in figure 3.14 (2) is read. R0 specifies the address 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 Notes for usage

Instructions that manipulate the MD or BL bit in register SR (the LDC Rm, SR instruction, LDC @Rm+, SR instruction, and RTE instruction) and the following instruction, or the LDTLB instruction, should be used with the TLB disabled or in a fixed physical address space (the P1 or P2 space).

Section 4 Exception Processing

4.1 Exception Processing Function

Exception processing is separate from normal program processing, and is performed by a routine separate from the normal program. In response to an exception processing request due to abnormal termination of the executing instruction, control is passed to a user-written exception handler. However, in response to an interrupt request, normal program execution continues until the end of the executing instruction. Here, all exceptions other than resets and interrupts will be called general exceptions. There are thus three types of exceptions: resets, general exceptions, and interrupts.

4.1.1 Exception Processing Flow

In exception processing, the contents of the program counter (PC) and status register (SR) are saved in the saved program counter (SPC) and saved status register (SSR), respectively, and execution of the exception handler is invoked from a vector address. The return from exception handler (RTE) instruction is issued by the exception handler routine at the completion of the routine, restoring the contents of the PC and SR to return to the processor state at the point of interruption and the address where the exception occurred.

A basic exception processing sequence consists of the following operations:

1. The contents of the PC and SR are saved in the SPC and SSR, respectively.
2. The block (BL) bit in SR is set to 1, masking any subsequent exceptions.
3. The mode (MD) bit in SR is set to 1 to place the SH7706 in the privileged mode.
4. The register bank (RB) bit in SR is set to 1.
5. An exception code identifying the exception event is written to bits 11–0 of the exception event (EXPEVT) or interrupt event (INTEVT and INTEVT2) register.
6. Instruction execution jumps to the designated exception processing vector address to invoke the handler routine.

4.1.2 Exception Processing Vector Addresses

The reset vector address is fixed at H'A0000000. The other three events are assigned offsets from the vector base address by software. Translation look-aside buffer (TLB) miss exceptions have an offset from the vector base address of H'00000400. The vector address offset for general exception events other than TLB miss exceptions is H'00000100. The interrupt vector address offset is H'00000600. The vector base address is loaded into the vector base register (VBR) by software. The vector base address should reside in P1 or P2 fixed physical address space. Figure 4.1 shows the relationship between the vector base address, the vector offset, and the vector table.

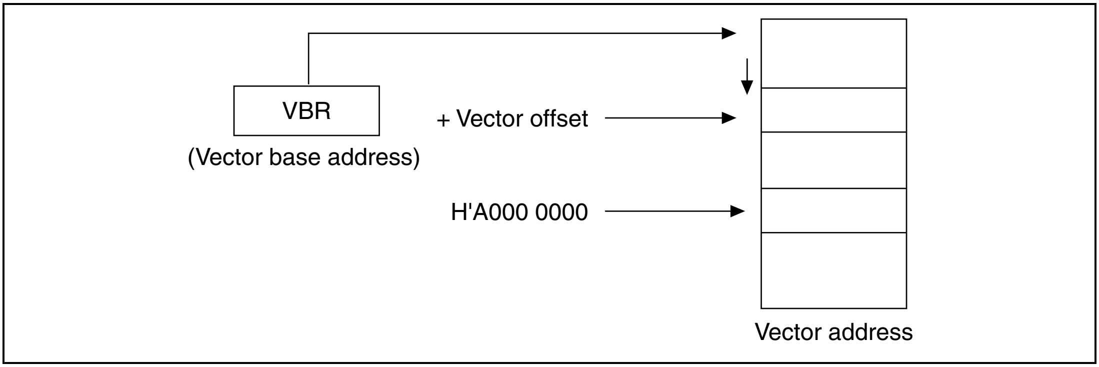


Figure 4.1 Vector Addresses

In table 4.1, exceptions and their vector addresses are listed by exception type, instruction completion state, relative acceptance priority, relative order of occurrence within an instruction execution sequence and vector address for exceptions and their vector addresses.

Table 4.1 Exception Event Vectors

Exception Type	Current Instruction	Exception Event	Priority*1	Exception Order	Vector Address	Vector Offset
Reset	Aborted	Power-on	1	—	H'A00000000	—
		Manual reset	1	—	H'A00000000	—
		H-UDI reset	1	—	H'A00000000	—
General exception events	Aborted and retried	CPU Address error (instruction access)	2	1	—	H'00000100
		TLB miss (instruction access)	2	2	—	H'00000400
		TLB invalid (instruction access)	2	3	—	H'00000100
		TLB protection violation (instruction access)	2	4	—	H'00000100
		General illegal instruction exception	2	5	—	H'00000100
		Illegal slot instruction exception	2	5	—	H'00000100
		CPU_Address error (data access)	2	6	—	H'00000100
		TLB miss (data access not in repeat loop)	2	7	—	H'00000400
		TLB invalid (data access)	2	8	—	H'00000100
		TLB protection violation (data access)	2	9	—	H'00000100
		Initial page write	2	10	—	H'00000100
	Completed	Unconditional trap (TRAPA instruction)	2	5	—	H'00000100
		User breakpoint trap	2	n*2	—	H'00000100
		DMA address error	2	12	—	H'00000100

Exception Type	Current Instruction	Exception Event	Priority*1	Exception Order	Vector Address	Vector Offset
General interrupt requests	Completed	Nonmaskable interrupt	3	—	—	H'00000600
		External hardware interrupt	4*3	—	—	H'00000600
		H-UDI interrupt	4*3	—	—	H'00000600

Notes: 1. Priorities are indicated from high to low, 1 being highest and 4 being lowest.

2. The user defines the break point traps. 1 is a break point before instruction execution and 11 is a break point after instruction execution. For an operand break point, use 11.

3. Use software to specify relative priorities of external hardware interrupts and peripheral module interrupts (see section 6, Interrupt Controller (INTC)).

4.1.3 Acceptance of Exceptions

Processor resets and interrupts are asynchronous events unrelated to the instruction stream. All exception events are prioritized to establish an acceptance order whenever two or more exception events occur simultaneously. If a power-on reset and manual reset occur simultaneously, the power-on reset takes precedence.

All general exception events occur in a relative order in the execution sequence of an instruction (i.e. execution order), but are handled at priority level 2 in instruction-stream order (i.e. program order), where an exception detected in a preceding instruction is accepted prior to an exception detected in a subsequent instruction.

Three general exception events (reserved instruction code exception, unconditional trap, and illegal slot instruction exception) are detected in the decode stage (ID stage) of different instructions and are mutually exclusive events in the instruction pipeline. They have the same execution priority. Figure 4.2 shows the order of general exception acceptance.

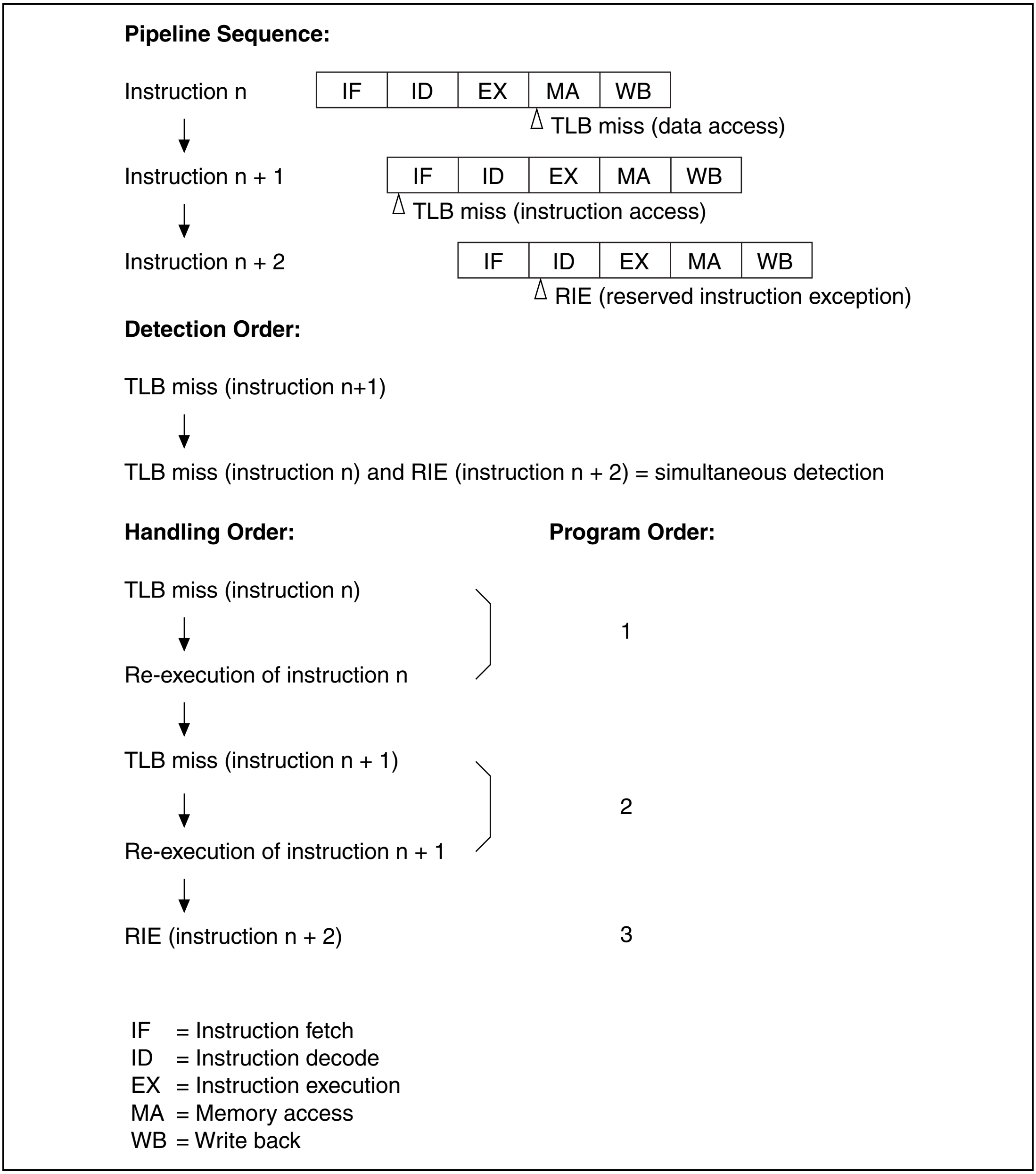

Pipeline Sequence:
Instruction n
IF
ID
EX
MA
WB
TLB miss (data access)
Instruction n + 1
TLB miss (instruction access)
Instruction n + 2
RIE (reserved instruction exception)
Detection Order:
TLB miss (instruction n+1)
TLB miss (instruction n) and RIE (instruction n + 2) = simultaneous detection
Handling Order:
Program Order:
TLB miss (instruction n)
Re-execution of instruction n
1
TLB miss (instruction n + 1)
Re-execution of instruction n + 1
2
RIE (instruction n + 2)
3
IF = Instruction fetch
ID = Instruction decode
EX = Instruction execution
MA = Memory access
WB = Write back

Figure 4.2 Example of Acceptance Order of General Exceptions

All exceptions other than a reset are detected in the pipeline ID stage, and accepted on instruction boundaries. However, an exception is not accepted between a delayed branch instruction and the delay slot. A re-execution type exception detected in a delay slot is accepted before execution of the delayed branch instruction. A completion type exception detected in a delayed branch instruction or delay slot is accepted after execution of the delayed branch instruction. The delay slot here refers to the next instruction after a delayed unconditional branch instruction, or the next instruction when a delayed conditional branch instruction is true.

4.1.4 Exception Codes

Table 4.2 lists the exception codes written to bits 11–0 of the EXPEVT register for reset or general exceptions or the INTEVT and INTEVT2 registers for general interrupt requests to identify each specific exception event. An additional exception register, the TRAPA (TRA) register, is used to hold the 8-bit immediate data in an unconditional trap (TRAPA instruction).

Table 4.2 Exception Codes

Exception Type	Exception Event	Exception Code
Reset	Power-on reset	H'000
	Manual reset	H'020
	H-UDI reset	H'000
General exception events	TLB miss/invalid exception (load)	H'040
	TLB miss/invalid exception (store)	H'060
	Initial page write exception	H'080
	TLB protection exception (load)	H'0A0
	TLB protection exception (store)	H'0C0
	CPU Address error (load)	H'0E0
	CPU Address error (store)	H'100
	Unconditional trap (TRAPA instruction)	H'160
	Reserved instruction code exception	H'180
	Illegal slot instruction exception	H'1A0
	User breakpoint trap	H'1E0
	DMA address error	H'5C0
General interrupt requests	Nonmaskable interrupt	H'1C0
	H-UDI interrupt	H'5E0
	External hardware interrupts:	
	IRL3–IRL0 = 0000	H'200
	IRL3–IRL0 = 0001	H'220
	IRL3–IRL0 = 0010	H'240
	IRL3–IRL0 = 0011	H'260
	IRL3–IRL0 = 0100	H'280
	IRL3–IRL0 = 0101	H'2A0

Exception Type	Exception Event	Exception Code
General interrupt requests (cont)	External hardware interrupts (cont):	
	IRL3–IRL0 = 0110	H'2C0
	IRL3–IRL0 = 0111	H'2E0
	IRL3–IRL0 = 1000	H'300
	IRL3–IRL0 = 1001	H'320
	IRL3–IRL0 = 1010	H'340
	IRL3–IRL0 = 1011	H'360
	IRL3–IRL0 = 1100	H'380
	IRL3–IRL0 = 1101	H'3A0
	IRL3–IRL0 = 1110	H'3C0

Note: Exception codes H'120, H'140, and H'3E0 are reserved.

4.1.5 Exception Request and BL Bit

If a general exception event occurs when the BL bit in SR is 1, the CPU’s internal registers are set to their post-reset state, other module registers retain their contents prior to the general exception, and a branch is made to the same address (H'A0000000) as for a reset.

If a general interrupt occurs when BL = 1, the request is masked (held pending) and not accepted until the BL bit is cleared to 0 by software. For reentrant exception processing, the SPC and SSR must be saved and the BL bit in SR cleared to 0.

4.1.6 Returning from Exception Processing

The RTE instruction is used to return from exception processing. When RTE is executed, the SPC value is set in the PC, and the SSR value in SR, and the return from exception processing is performed by branching to the SPC address.

If the SPC and SSR have been saved in the external memory, set the BL bit in SR to 1, then restore the SPC and SSR, and issue an RTE instruction.

4.2 Description of Registers

There are four registers related to exception processing. These are peripheral module registers, and therefore reside in area P4. They can be accessed by specifying the address in the privileged mode only.

There are following four registers related to exception processing. Registers with undefined initial values (TRAPA exception register, Interrupt event register, and Interrupt event register 2) should be initialized by software. Refer to section 23, Control Registers Table for more details of the addresses and access sizes.

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

4.2.1 Exception Event Register (EXPEVT)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	—	*	R/W	12-bit exception code

Note:* H'0000 is set in a power-on reset, and H'020 in a manual reset.

4.2.2 Interrupt Event Register (INTEVT)

The interrupt event register (INTEVT) contains a 12-bit interrupt exception code or a code indicating the interrupt priority. Which is set when an interrupt occurs depends on the interrupt source (refer to section 6, Interrupt Controller (INTC)). The exception code or interrupt priority code is set automatically by hardware when an exception occurs. INTEVT can also be modified by software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	—	—	R/W	12-bit interrupt exception code or a code indicating the interrupt priority

4.2.3 Interrupt Event Register 2 (INTEVT2)

The interrupt event register 2 (INTEVT2) contains a 12-bit exception code. The exception code set in INTEVT2 is that for an interrupt request. The exception code is set automatically by hardware when an exception occurs.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	—	—	R/W	12-bit exception code

4.2.4 TRAPA Exception Register (TRA)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 2	imm	—	R/W	8-bit immediate data
1	—	0	R	Reserved
0	—	0	R	These bits are always read as 0. The write value should always be 0.

4.3 Description of Operation

4.3.1 Reset

The reset sequence is used to power up or restart the SH7706 from the initialization state. The $\overline{\text{RESETP}}$ signal and $\overline{\text{RESETM}}$ signal are sampled every clock cycle, and in the case of a power-on reset, all processing being executed (excluding the RTC) is suspended, all unfinished events are canceled, and reset processing is executed immediately. In the case of a manual reset, however, processing to retain external memory contents is continued. The reset sequence consists of the following operations:

1. The MD bit in SR is set to 1 to place the SH7706 in privileged mode.
2. The BL bit in SR is set to 1, masking any subsequent exceptions.
3. The RB bit in SR is set to 1.
4. An encoded value of H'000 in a power-on reset or H'020 in a manual reset is written to bits 11–0 of the EXPEVT register to identify the exception event.
5. Instruction execution jumps to the user-written exception handler at address H'A0000000.

4.3.2 Interrupts

An interrupt processing request is accepted on completion of the current instruction. The interrupt acceptance sequence consists of the following operations:

1. The contents of the PC and SR are saved in SPC and SSR, respectively.
2. The BL bit in SR is set to 1, masking any subsequent exceptions.
3. The MD bit in SR is set to 1 to place the SH7706 in privileged mode.
4. The RB bit in SR is set to 1.
5. An encoded value identifying the exception event is written to bits 11–0 of the INTEVT and INTEVT2 registers.
6. Instruction execution jumps to the vector location designated by the sum of the value of the contents of the VBR and H'00000600 to invoke the exception handler.

4.3.3 General Exceptions

When the SH7706 encounters any exception condition other than a reset or interrupt request, it executes the following operations:

1. The contents of the PC and SR are saved in the SPC and SSR, respectively.
2. The BL bit in SR is set to 1, masking any subsequent exceptions.
3. The MD bit in SR is set to 1 to place the SH7706 in privileged mode.
4. The RB bit in SR is set to 1.

5. An encoded value identifying the exception event is written to bits 11–0 of the EXPEVT register.
6. Instruction execution jumps to the vector location designated by either the sum of the vector base address and offset H'00000400 in the vector table in a TLB miss trap, or by the sum of the vector base address and offset H'00000100 for exceptions other than TLB miss traps, to invoke the exception handler.

4.4 Individual Exception Operations

This section describes the conditions for specific exception processing, and the processor operations.

4.4.1 Resets

- Power-On Reset
  - Conditions: $\overline{\text{RESETP}}$ low
  - Operations: EXPEVT set to H'000, VBR and SR initialized, branch to PC = H'A0000000. Initialization sets the VBR register to H'0000000. In SR, the MD, RB and BL bits are set to 1 and the interrupt mask bits (I3 to I0) are set to B'1111. The CPU and on-chip supporting modules are initialized. See the register descriptions in the relevant sections for details. A power-on reset must always be performed when powering on.

    A high level is output from the STATUS0 and STATUS1 pins.
- Manual Reset
  - Conditions: $\overline{\text{RESETM}}$ low
  - Operations: EXPEVT set to H'020, VBR and SR initialized, branch to PC = H'A0000000. Initialization sets the VBR register to H'0000000. In SR, the MD, RB, and BL bits are set to 1 and the interrupt mask bits (I3 to I0) are set to B'1111. The CPU and on-chip supporting modules are initialized. See the register descriptions in the relevant sections for details.

    A high level is output from the STATUS0 and STATUS1 pins.
- H-UDI Reset
  - Conditions: H-UDI reset command input (see section 21, Hitachi User-Debug Interface (H-UDI))
  - Operations: EXPEVT set to H'000, VBR and SR initialized, branch to PC = H'A0000000. Initialization sets the VBR register to H'0000000. In SR, the MD, RB and BL bits are set to 1 and the interrupt mask bits (I3 to I0) are set to B'1111. The CPU and on-chip supporting modules are initialized. See the register descriptions in the relevant sections for details.

Table 4.3 Types of Reset

		Internal State	
Type	**Conditions for Transition to Reset State**	**CPU**	**On-Chip Supporting Modules**
Power-on reset	$\overline{\text{RESETP}}$ = Low	Initialized	(See register configuration in relevant sections)
Manual reset	$\overline{\text{RESETM}}$ = Low	Initialized	
H-UDI reset	H-UDI reset command input	Initialized	

4.4.2 General Exceptions

- TLB miss exception
  - Conditions: Comparison of TLB addresses shows no address match
  - Operations: The virtual address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASID of PTEH indicates the ASID at the time the exception occurred. The RC bit in MMUCR is incremented by one when all ways are valid, or way-0 is set to the RC with top priority when there is invalid way.

  The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. If the exception occurred during a read, H'040 is set in EXPEVT; if the exception occurred during a write, H'060 is set in EXPEVT. The BL, MD and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0400.

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

- TLB invalid exception
  - Conditions: Comparison of TLB addresses shows address match but V = 0.
  - Operations: The virtual address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASID of PTEH indicates the ASID at the time the exception occurred. The way that generated the exception is set in the RC bits in MMUCR.

  The PC and SR of the instruction that generated the exception are saved in the SPC and SSR, respectively. If the exception occurred during a read, H'040 is set in EXPEVT; if the exception occurred during a write, H'060 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- Initial page write exception
  - Conditions: A hit occurred to the TLB for a store access, but D = 0.
    This occurs for initial writes to the page registered by the load.

— Operations: The virtual address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASID of PTEH indicates the ASID at the time the exception occurred. The way that generated the exception is set in the RC bit in MMUCR.

The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. H'080 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs in PC = VBR + H'0100.

- TLB protection exception

— Conditions: When a hit access violates the TLB protection information (PR bits) shown below:

PR	Privileged mode	User mode
00	Only read enabled	No access
01	Read/write enabled	No access
10	Only read enabled	Only read enabled
11	Read/write enabled	Read/write enabled

— Operations: The virtual address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASID of PTEH indicates the ASID at the time the exception occurred. The way that generated the exception is set in the RC bits in MMUCR.

The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. If the exception occurred during a read, H'0A0 is set in EXPEVT; if the exception occurred during a write, H'0C0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- Address error

— Conditions: When corresponded to the following items.

a. Instruction fetch from odd address (4n + 1, 4n + 3)

b. Word data accessed from addresses other than word boundaries (4n + 1, 4n + 3)

c. Longword accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)

d. Virtual space accessed in user mode in the area H'80000000 to H'FFFFFFFF.

— Operations: The virtual address (32 bits) that caused the exception is set in TEA. The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. If the exception occurred during a read, H'0E0 is set in EXPEVT; if the exception occurred during a write, H'100 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. Refer to section 3.5.5 Processing Flow in Event of MMU Exception.

- Unconditional trap

— Conditions: TRAPA instruction executed

— Operations: The exception is a processing-completion type, so the PC of the instruction after the TRAPA instruction is saved to the SPC. SR from the time when the TRAPA instruction was executing is saved to SSR. The 8-bit immediate value in the TRAPA instruction is quadrupled and set in TRA (9–0). H'160 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- General illegal instruction exception

— Conditions: When corresponded to the following items.

a. When undefined code not in a delay slot is decoded

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

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

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

— Operations: The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. H'180 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. When an undefined instruction other than H'Fxxx is decoded, operation cannot be guaranteed.

- Illegal slot instruction exception

— Conditions: When corresponded to the following items.

a. When undefined code in a delay slot is decoded

Delay branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
Undefined instruction: H’Fxxx.

b. When an instruction that rewrites the PC in a delay slot is decoded

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

c. When a privileged instruction in a delay slot is decoded in user mode

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

— Operations: The PC of the previous delay branch instruction is saved to the SPC. SR of the instruction that generated the exception is saved to SSR. H'1A0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. When an undefined instruction other than H'Fxxx is decoded, operation cannot be guaranteed.

- User break point trap

— Conditions: When a break condition set in the user break point controller is satisfied

— Operations: When a post-execution break occurs, the PC of the instruction immediately after the instruction that set the break point is set in the SPC. If a pre-execution break occurs, the PC of the instruction that set the break point is set in the SPC. SR when the break occurs is set in SSR. H'1E0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. See section 7, User Break Controller, for more information.

- DMA Address error

  — Conditions: When corresponded to the following items.

  a. Word data accessed from addresses other than word boundaries (4n + 1, 4n + 3)
  b. Longword accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)

  — Operations: The PC of the instruction immediately after the instruction executed before the exception occurs is saved to the SPC. SR when the exception occurs is saved to SSR. H'5C0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

4.4.3 Interrupts

1. NMI

   Conditions: NMI pin edge detection

   Operations: The PC and SR after the instruction that receives the interrupt are saved to the SPC and SSR, respectively. H'01C0 is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to PC = VBR + H'0600. This interrupt is not masked by SR.IMASK and received with top priority when the SR's BL bit in SR is 0. When the BL bit is 1, the interrupt is masked. When BLMSK in ICRI is a logic zero and not masked when BLMSK in ICRI is a logic one. See section 6, Interrupt Controller (INTC), for more information.

2. IRL Interrupts

   Conditions: The value of the interrupt mask bits in SR is lower than the IRL3–IRL0 level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. SR at the time the interrupt is accepted is saved to SSR. The code corresponding to the IRL3–IRL0 level is set in INTEVT and INTEVT2. The corresponding code is given as H'200 + B' (IRL3–IRL0) × H'20. See table 6.4, Interrupt Exception Handling Source and Priority for the corresponding code. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to VBR + H'0600. The received level is not set in the interrupt mask bit of SR. See section 6, Interrupt Controller (INTC), for more information.

3. IRQ Pin Interrupts

   Conditions: IRQ pin is asserted and the interrupt mask bit of SR is lower than the IRQ priority level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. The SR at the point the interrupt is accepted is saved to the SSR. The code corresponding to the interrupt source is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to VBR + H'0600. The received level is not set to the interrupt mask bit of SR. See section 6, Interrupt Controller (INTC), for more information.

4. On-Chip Peripheral Module Interrupts

   Conditions: The interrupt mask bit of SR is lower than the on-chip peripheral module (TMU, RTC, SCI0, SCI2, A/D, LCDC, PCC, DMAC, CPG, REF) interrupt level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. The SR at the point the interrupt is accepted is saved to the SSR. The code corresponding to the interrupt source is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to VBR + H'0600. See section 6, Interrupt Controller (INTC), for more information.

5. H-UDI Interrupt

   Conditions: H-UDI interrupt command is input (see section 22.4.4, H-UDI Interrupt) and the interrupt mask bit of SR is lower than 15 and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. The SR at the point the interrupt is accepted is saved to the SSR. H'5E0 is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to VBR + H'0600. See section 6, Interrupt Controller (INTC), for more information.

4.5 Notes for Usage

- Return from exception processing
  - Check the BL bit in SR with software. When the SPC and SSR have been saved to external memory, set the BL bit in SR to 1 before restoring them.
  - Issue an RTE instruction. Set the SPC in the PC and SSR in SR with the RTE instruction, branch to the SPC address, and return from exception processing.
- Operation when exception or interrupt occurs while SR.BL = 1
  - Interrupt: Acceptance is suppressed until the BL bit in SR is set to 0 by software. If there is a request and the reception conditions are satisfied, the interrupt is accepted after the execution of the instruction that sets the BL bit in SR to 0. During the sleep or standby mode, however, the interrupt will be accepted even when the BL bit in SR is 1.

    NMI is accepted when BLMSK in ICR1 is 1.
  - Exception: No user break point trap will occur even when the break conditions are met. When one of the other exceptions occurs, a branch is made to the fixed address of the reset (H'A0000000). In this case, the values of the EXPEVT, SPC, and SSR registers are undefined.

    Differently from general reset processing, no signal is output from STATUS0 and STATUS1.
- SPC when an Exception Occurs: The PC saved to the SPC when an exception occurs is as shown below:

- Re-executing-type exceptions: The PC of the instruction that caused the exception is set in the SPC and re-executed after return from exception processing. If the exception occurred in a delay slot, however, the PC of the immediately prior delayed branch instruction is set in the SPC. If the condition of the conditional delayed branch instruction is not satisfied, the delay slot PC is set in SPC.
- Completed-type exceptions and interrupts: The PC of the instruction after the one that caused the exception is set in the SPC. If the exception was caused by a delayed conditional instruction, however, the branch destination PC is set in SPC. If the condition of the conditional delayed branch instruction is not satisfied, the delay slot PC is set in SPC.

- Initial register values after reset
  - Undefined registers

    R0_BANK0/1–R7_BANK0/1, R8–R15, GBR, SPC, SSR, MACH, MACL, PR
  - Initialized registers

    VBR = H'00000000

    SR.MD = 1, SR.BL = 1, SR.RB = 1, SR.I3–SR.I0 = H'F. Other SR bits are undefined.

    PC = H'A0000000
- Ensure that an exception is not generated at an RTE instruction delay slot, as operation is not guaranteed in this case.
- When the BL bit in the SR register is set to 1, ensure that a TLB-related exception or address error does not occur at an LDC instruction that updates the SR register and the following instruction. This occurrence will be identified as multiple exceptions, and may initiate reset processing.

Section 5 Cache

5.1 Features

- Instruction/data mixed, 16-byte cache
- 256 entries/way, 4-way set associative, 16-byte block
- Write-back/write-through selectable
- LRU replacing algorithm
- 1-stage write-back buffer
- A maximum of two ways lockable

5.1.1 Cache Structure

The cache uses a 4-way set associative system. It is composed of four ways (banks), each of which is divided into an address section and a data section. Each of the address and data sections is divided into 256 entries. The data section of the entry is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 4 kbytes (16 bytes × 256 entries), with a total of 16 kbytes in the cache as a whole (4 ways). Figure 5.1 shows the cache structure.

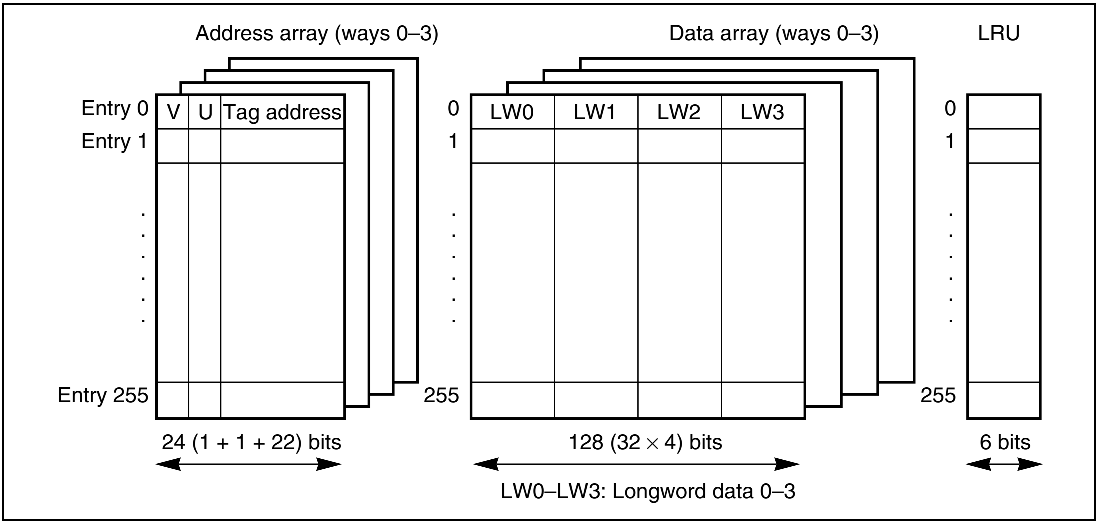


Figure 5.1 Cache Structure

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

In the SH7706, the top three of 32 physical address bits are used as shadow bits (see section 8, Bus State Controller (BSC)), and therefore in a normal replace operation the top three bits of the tag address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset, but are not initialized by a manual reset. 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 line units (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 same entry address (address bits 11 to 4) can be registered in the cache. When an entry is registered, the LRU bits show which of the four ways it is recorded in. There are six LRU bits, controlled by hardware. A least recently used (LRU) algorithm, which selects the way that has been used least recently, is used to select the way.

The LRU bits also indicate the way to be replaced when a cache miss occurs. Table 5.1 shows the relationship between the LRU bits and the way to be replaced when cache locking mechanism is disabled. (For details on the case when cache locking mechanism is enabled, see section 5.2.2, Cache Control Register 2). If a bit pattern other than those listed in table 5.1 is set in the LRU bits by software, the cache will not function correctly. When modifying the LRU bits by software, set one of the patterns listed in table 5.1.

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

Table 5.1 LRU and Way Replacement

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

5.2 Description of Registers

The cache includes the following registers. For the address and access size, see section 23, Control Registers.

- Cache control register (CCR)
- Cache control register 2 (CCR2)

5.2.1 Cache Control Register (CCR)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CF	0	R	Cache Flash When 1 is set, the V, U and LRU bits of all cache entries are cleared to 0 (flush). This bit is always read as 0. Write-back to external memory is not performed when the cache is flushed.
2	CB	0	R/W	Cache Write-back Indicates the cache's operating mode for area P1. 0: Write-through mode 1: Write-back mode
1	WT	0	R/W	Write through Indicates the cache's operating mode for area P0, U0, and P3. 0: Write-back mode 1: Write-through mode
0	CE	0	R/W	Cache enable Indicates whether to use the cache function. 0: Cache not used 1: Cache used.

5.2.2 Cache Control Register 2 (CCR2)

Cache control register 2 (CCR2) enables or disables the cache locking mechanism. This register setting is valid only in cache locking mode. Cache locking mode is enabled when the cache locking bit (bit 12) of the status register (SR) is set to 1, and disabled when it is cleared to 0.

If a cache miss occurs during prefetch instruction (PREF) execution in cache locking mode, one line size of data pointed by Rn is loaded into the cache according to the W3LOAD, W3LOCK, W2LOAD, and W2LOCK bit settings of CCR2 (bits 9, 8, 1, and 0). Table 5.2 shows the relationship between each bit setting and the way to be replaced when the prefetch instruction is executed. On the other hand, if a cache hit occurs during prefetch instruction (PREF) execution, no data is loaded into the cache and entries that have been valid in the cache are maintained. For instance, if one line size of data pointed by Rn exists at way 0, and if the prefetch instruction is executed while the cache lock, W3LOAD, and W3LOCK are set to 1s, a cache hit occurs and data is not brought to way 3.

When a cache is accessed by other than the prefetch instruction in cache locking mode, the ways to be replaced are controlled by the W3LOCK and W2LOCK bit settings. Table 5.3 shows the relationship between CCR2 bit settings and the way to be replaced.

A program to modify the CCR2 contents should be placed at an address area whose data is not cached.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	W3LOAD	0	W	W3LOAD: Way 3 load
8	W3LOCK		W	W3LOCK: Way 3 Lock When W3LOACK = 1 & W3LOAD – 1 & SR.CL is 1, the prefetched data will always be loaded into Way3. In all other conditions, the prefetched data will be loaded into the way pointed by LRU.
7 to 2	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	W2LOAD	0	W	W3LOAD: Way 2 load
0	W2LOCK		W	W3LOCK: Way 2 Lock When W3LOACK = 1 & W3LOAD – 1 & SR.CL is 1, the prefetched data will always be loaded into Way2. In all other conditions, the prefetched data will be loaded into the way pointed by LRU.

Note: Do not set 1 into W2LOAD and W3LOAD at the same time.

Whenever CCR2 bit 8 (W3LOCK) or bit 0 (W2LOCK) is high level the cache is locked. The locked data will not be overwritten unless W3LOCK bit and W2LOCK bit are reset or the PREF condition during cache locking mode watches. During cache locking mode, the LRU in table 5.1 will be replaced by tables 5.4 to 5.6.

Table 5.2 Way to be Replaced when Cache Miss Occurs during PREF Instruction Execution

CL bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replaced
0	*	*	*	*	According to LRU (table 5.1)
1	*	0	*	0	According to LRU (table 5.1)
1	*	0	0	1	According to LRU (table 5.4)
1	0	1	*	0	According to LRU (table 5.5)
1	0	1	0	1	According to LRU (table 5.6)
1	0	*	1	1	Way 2
1	1	1	0	*	Way 3

Note: *: Don't care

Do not set 1 into W2LOAD and W3LOAD at the same time.

Table 5.3 Way to be Replaced when Cache Miss Occurs during Execution of Instruction other than PREF Instruction

CL bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replaced
0	*	*	*	*	According to LRU (table 5.1)
1	*	0	*	0	According to LRU (table 5.1)
1	*	0	*	1	According to LRU (table 5.4)
1	*	1	*	0	According to LRU (table 5.5)
1	*	1	*	1	According to LRU (table 5.6)

Note: *: Don't care

Do not set 1 into W2LOAD and W3LOAD at the same time.

Table 5.4 LRU and Way Replacement (when W2LOCK=1)

LRU (5–0)	Way to be Replaced
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 5.5 LRU and Way Replacement (when W3LOCK=1)

LRU (5–0)	Way to be Replaced
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

Table 5.6 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=1)

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

5.3 Description of Operation

5.3.1 Searching the Cache

If the cache is enabled, whenever instructions or data in memory are accessed the cache will be searched to see if the desired instruction or data is in the cache. Figure 5.2 illustrates the method by which the cache is searched. The cache is a physical cache and holds physical addresses in its address section.

Entries are selected using bits 11–4 of the address (virtual) of the access to memory and the address tag of that entry is read. In parallel to reading of the address tag, the virtual address is translated to a physical address in the MMU. The physical address after translation and the physical address read from the address section are compared. The address comparison uses all four ways. When the comparison shows a match and the selected entry is valid (V = 1), a cache hit occurs. When the comparison does not show a match or the selected entry is not valid (V = 0), a cache miss occurs.

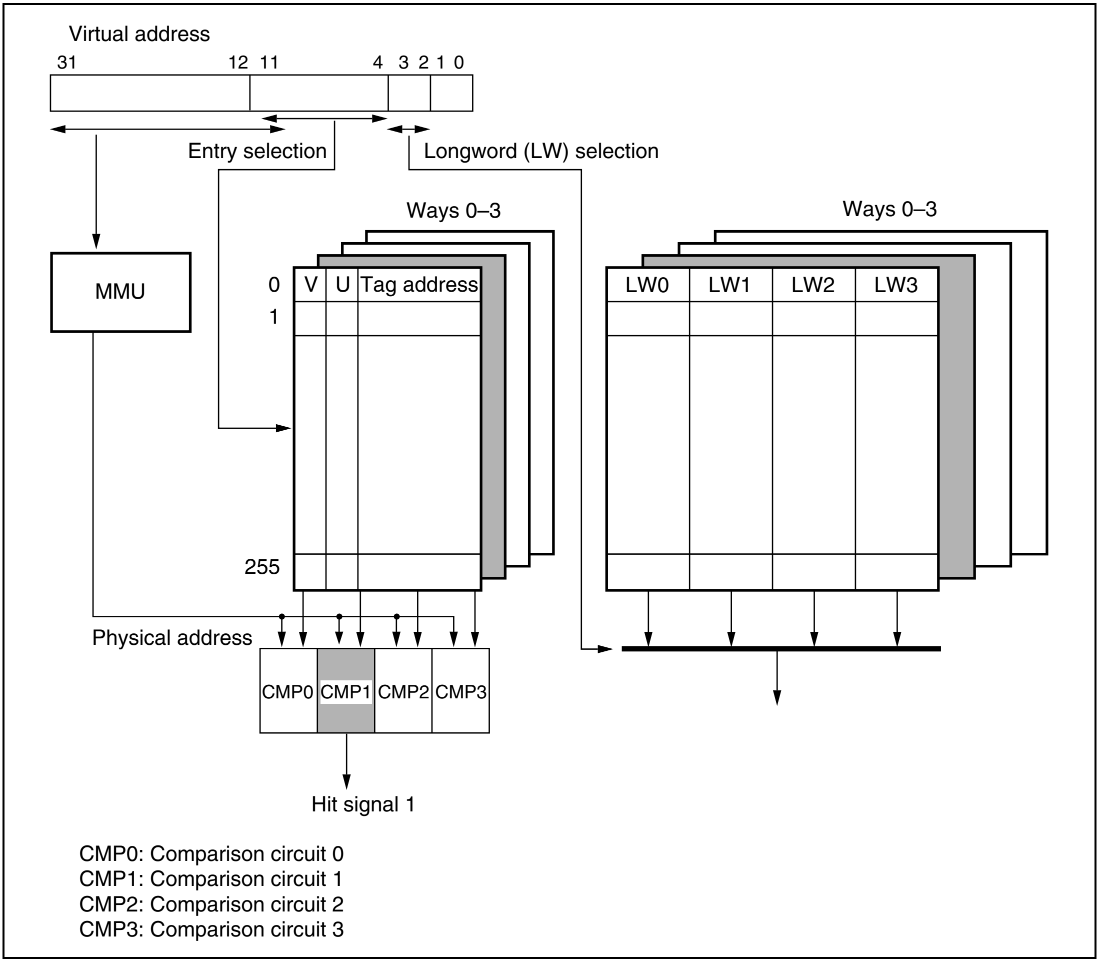


Figure 5.2 Cache Search Scheme (Normal Mode)

5.3.2 Read Access

Read Hit: In a read access, instructions and data are transferred from the cache to the CPU. The LRU is updated.

Read Miss: An external bus cycle starts and the entry is updated. The way replaced is shown in table 5.3. Entries are updated in 16-byte units. When the desired instruction or data that caused the miss is loaded from external memory to the cache, the instruction or data is transferred to the CPU in parallel with being loaded to the cache. When it is loaded in the cache, the U bit is cleared to 0 and the V bit is set to 1.

5.3.3 Prefetch Operation

Prefetch Hit: LRU is updated so that the way that has been hit to be the latest. Other contents of the cache are not updated. Instruction or data is not transferred to the CPU.

Prefetch Miss: Instruction or data is not transferred to the CPU. The way to be replaced is listed in table 5.2. Other operations are same as those in read miss.

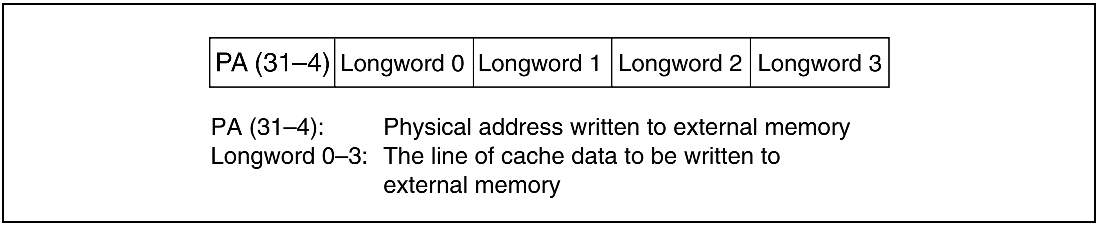
5.3.4 Write Access

Write Hit: In a write access in the write-back mode, the data is written to the cache and the U bit of the entry written is set to 1. Writing occurs only to the cache; no external memory write cycle is issued. In the write-through mode, the data is written to the cache and an external memory write cycle is issued.

Write Miss: In the write-back mode, an external write cycle starts when a write miss occurs, and the entry is updated. The way to be replaced is shown in table5.3. When the U bit of the entry to be replaced is 1, the cache fill cycle starts after the entry is transferred to the write-back buffer. The write-back unit is 16 bytes. Data is written to the cache and the U bit and V bit are set to 1. After the cache completes its fill cycle, the write-back buffer writes back the entry to the memory. In the write-through mode, no write to cache occurs in a write miss; the write is only to the external memory.

5.3.5 Write-Back Buffer

When the U bit of the entry to be replaced in the write-back mode is 1, it must be written back to the external memory. To increase performance, the entry to be replaced is first transferred to the write-back buffer and fetching of new entries to the cache takes priority over writing back to the external memory. After fetching of new entries to the cache is completed, the data in the write-back buffer is write back to the external memory. During the write back cycles, the cache can be accessed. The write-back buffer can hold one line of the cache data (16 bytes) and its physical address. Figure 5.3 shows the configuration of the write-back buffer.



PA (31–4)	Longword 0	Longword 1	Longword 2	Longword 3

PA (31–4): Physical address written to external memory
Longword 0–3: The line of cache data to be written to external memory

Figure 5.3 Write-Back Buffer Configuration

5.3.6 Coherency of Cache and External Memory

Use software to ensure coherency between the cache and the external memory. To allocate memory shared by this LSI and the external device to an address area to be cached, write back must be performed by invalidating the entries by operating the memory allocating cache. If necessary, for the memory shared by the CPU and the direct memory access controller in this LSI, write back must also be performed in the same way as described above.

5.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read and written by means of MOV instructions in the privileged mode. The cache is mapped onto the P4 area in virtual address space. The address array is mapped onto addresses H'F0000000 to H'F0FFFFFF, and the data array onto addresses H'F1000000 to H'F1FFFFFF. Only longword can be used as the access size for the address array and data array, and instruction fetches cannot be performed.

5.4.1 Address Array

The address array is mapped onto H'F0000000 to H'F0FFFFFF. To access an address array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the tag address, V bit, U bit, and LRU bits to be written to the address array (figure 5.4 (1)).

In the address field, specify the entry address for selecting the entry (bits 11 to 4), W for selecting the way (bits 13 and 12: 00 is way 0, 01 is way 1, 10 is way 2, and 11 is way 3), A for selecting the associative operation (bit 3), and H'F0 to indicate address array access (bits 31 to 24).

In data field, specify the tag address (bits 31 to 10), LRU bits (bits 9 to 4), U bit (bit 1), and V bit (bit 0). Upper three bits of the tag address (bits 31 to 29) should always be 0.

The following three operations are enabled for the address array.

Address Array Read: Read the tag address, LRU bits, U bit, and V bit of the entry specified by the entry address and the way number. When reading, no associative operation is performed regardless of the value of the associative bit (bit A) specified in the address.

Address Array Write (without associative operation): Write the tag address, LRU bits, U bit, and V bit specified in the data field to the entry specified by the entry address and the way number. The associative bit (bit A) should be 0. If data is written to the cache line in which U and V bits are set to 1, the cache line is written back, and then the tag address, LRU bits, U bit, and V bit specified in the data field are written to. However, when 0 is written to the V bit, 0 must also be written to the U bit of that entry.

Address Array Write (with associative operation): When writing while the associative bit (bit A) is 1, the addresses of four entries selected by the entry addresses are compared to the tag addresses specified in the data field. As a result of the comparison, U bit and V bit specified in the data field are written to the entry for the hit way. Note however, that the tag address and LRU bits are not changed. When no ways are hit, nothing is written to the address array and no operation occurs. This operation is used to invalidate a specific entry of the cache. When the U bit of the hit entry is 1, write back is occurs. However, when 0 is written to the V bit, 0 must also be written to the U bit of that entry.

5.4.2 Data Array

The data array is mapped onto H'F1000000 to H'F1FFFFFF. To access a data array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the longword data to be written to the data array.

In the address field, specify the entry address for selecting the entry (bits 11 to 4), L indicating the longword position within the (16-byte) line (bits 3 and 2: 00 is longword 0, 01 is longword 1, 10 is longword 2, and 11 is longword 3), W for selecting the way (bits 13 and 12: 00 is way 0, 01 is way 1, 10 is way 2, and 11 is way 3), and H'F1 to indicate data array access (bits 31 to 24).

The access size of the data array is fixed at longword, so 00 should be specified to bits 1 and 0 in the address field.

The following two operations are enabled for data array. However, information of the address array is not changed by the following operations.

Data Array Read: Reads data specified by L (bits 3 and 2) in the address field from the entry specified by the entry address and the way number.

Data Array Write: Writes a longword data specified by the data field to the position specified by L (bits 3 and 2) in the address field from the entry specified by the entry address and the way number.

1. Address array access

Address specification

Read access

31 24	23 14	13 12	11 4	3	2		0
1111 0000	*············*	W	Entry address	0	*	0	0

Write access

31 24	23 14	13 12	11 4	3	2		0
1111 0000	*············*	W	Entry address	A	*	0	0

Data specification (both read and write access)

31	30	29	10	9 4	3	2	1	0
0	0	0	Address tag (28–10)	LRU	X	X	U	V

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

Address specification

31 24	23 14	13 12	11 4	3 2	1	0
1111 0001	*············*	W	Entry address	L	0	0

Data specification

31 0
Longword

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

Figure 5.4 Specifying Address and Data for Memory-Mapped Cache Access

5.4.3 Usage Examples

1. Invalidating Specific Entries

Specific cache entries can be invalidated by writing 0 to the entry's U and V bit. When the A bit is 1, the address tag specified by the write data is compared to the address tag within the cache selected by the entry address, and data is written when a match is found. If no match is found, there is no operation. When the V bit of the entry is 1, a write back occurs.

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

2. Reading the Data of a Specific Entry

This example reads the data section of a specific cache entry. The longword indicated in the data field of the data array in figure 5.6 is read to the register.

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

Section 6 Interrupt Controller (INTC)

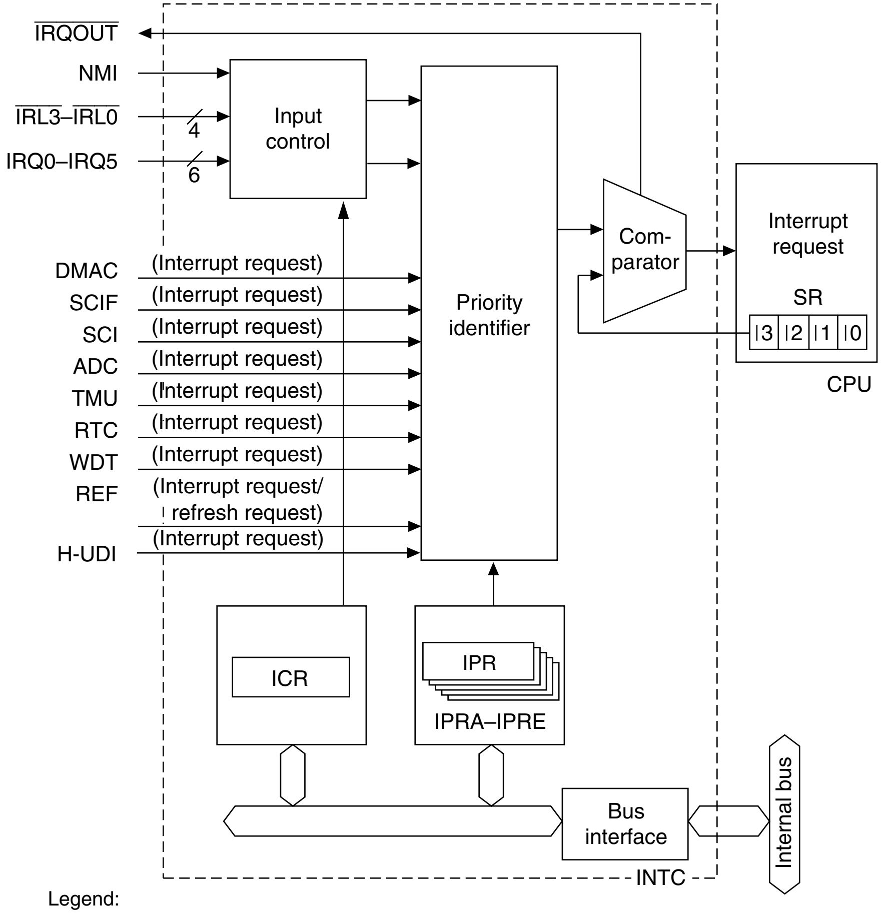
The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC has registers for setting the priority of each interrupt, and interrupt requests are handled according to the priorities set in these registers.

6.1 Features

INTC has the following features:

- 16 levels of interrupt priority can be set: By setting the five interrupt-priority registers, the priorities of on-chip peripheral module, IRQ interrupts can be selected from 16 levels for individual request sources.
- NMI noise canceler function: NMI input-level bit indicates NMI pin states. By reading this bit in the interrupt exception service routine, the pin state can be checked, enabling it to be used as a noise canceler.
- External devices can be notified that an interrupt has been received ($\overline{\text{IRQOUT}}$): When the SH7706 has released the bus right, the external bus master can be notified that an external interrupt, an on-chip peripheral module interrupt or a memory refresh request has occurred, enabling this LSI to request the bus right.

Figure 6.1 is a block diagram of the INTC.



Legend:

TMU:	Timer unit
RTC:	Realtime clock unit
SCI:	Serial communication interface
SCIF:	Serial communication interface (with FIFO)
WDT:	Watchdog timer
REF:	Refresh requests in the bus state controller
ICR:	Interrupt control register
IPRA-IPRE:	Registers A-E for setting the interrupt proprity levels
SR:	Status register
DMAC:	Direct memory access controller
ADC:	Analog-to-digital converter
H-UDI:	Hitachi user-debugging interface

Figure 6.1 INTC Block Diagram

6.2 I/O Pins

Table 6.1 lists the INTC pin configuration.

Table 6.1 Pin Configuration

Name	Abbreviation	I/O	Description
Nonmaskable interrupt input pin	NMI	I	Nonmaskable interrupt request signal input
Interrupt input pins	IRQ5–IRQ0 $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$	I	Interrupt request signal input (Maskable by interrupt mask bits in SR)
Interrupt request output pin	$\overline{\text{IRQOUT}}$	O	Output of signal that notifies external devices that an interrupt source or memory refresh has occurred

6.3 Interrupt Sources

There are 4 types of interrupt sources: NMI, IRQ, IRL, and on-chip peripheral modules. The priority of each interrupt is indicated by a priority level value (16–0), with level 16 as the highest and level 1 as the lowest. When level 0 is set, the interrupt is masked and interrupt requests are ignored.

6.3.1 NMI Interrupts

The NMI interrupt has the highest priority level of 16. When the BLMSK bit of the interrupt control register (ICR1) is 1 or the BL bit of the status register (SR) is 0, NMI interrupts are accepted when the MAI bit of the ICR1 register is 0. NMI interrupts are edge-detected. In sleep or software standby mode, the interrupt is accepted regardless of the BL. The NMI edge select bit (NMIE) in the interrupt control register 0 (ICR0) is used to select either the rising or falling edge. When the NMIE bit of the ICR0 register is changed, the NMI interrupt is not detected for 20 cycles after changing the ICR0.NMIE to avoid a false detection of the NMI interrupt. NMI interrupt exception processing does not affect the interrupt mask level bits (I3–I0) in the status register (SR).

When the BL bit is 1 and the BLMSK bit of the ICR1 register is set to 1, only NMI interrupts are accepted and the SPC register and SSR register are updated by the NMI interrupt handler, making it impossible to return to the original processing from exception processing initiated prior to the NMI. Use should therefore be restricted to cases where return is not necessary.

It is possible to wake the chip up from the software standby state with an NMI interrupt (except when the MAI bit of the ICR1 register is set to 1).

6.3.2 IRQ Interrupt

IRQ interrupts are input by priority from pins IRQ0–IRQ5 with a level or an edge. The priority level can be set by priority setting registers C–D (IPRC–IPRD) in a range from levels 0–15.

When using edge-sensing for IRQ interrupts, clear the interrupt source by having software read 1 from the corresponding bit in IRR0, then write 0 to the bit.

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

It is necessary for an edge input interrupt detection to input a pulse width more than two-cycle width by peripheral clock (Pφ) basis.

In level detection, keep the level until the CPU accepts an interrupt and starts the interrupt processing.

The interrupt mask bits (I3–I0) of the status register (SR) are not affected by IRQ interrupt processing.

Interrupts IRQ4–IRQ0 can wake the chip up from the software standby state when the relevant interrupt level is higher than I3–I0 in the SR register (but only when the RTC 32-kHz oscillator is used).

Note: When the IRQ is used in edge sensitive, pay attention to the following:

1. If an IRQ edge is input immediately before the CPU enters standby mode (the period between the SLEEP instruction executed by the CPU to high level of STATUS0), an interrupt may not be detected. In this case, when an IRQ edge is input again after STATUS0 becomes high level, an interrupt is detected.
2. If an IRQ edge is input while the frequency is changed by the FRQCR STC bit (when the WDT is counting), an interrupt may not be detected. In this case, when an IRQ edge is input again after the WDT halts counting, an interrupt is detected.

6.3.3 IRL Interrupts

IRL interrupts are input by level at pins $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$. The priority level is the higher of those indicated by pins $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$. An $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ value of 0 (0000) indicates the highest-level interrupt request (interrupt priority level 15). A value of 15 (1111) indicates no interrupt request (interrupt priority level 0). Figure 6.2 shows an examples of an IRL interrupt connection. Table 6.2 shows $\overline{\text{IRL}}$ pins and interrupt levels.

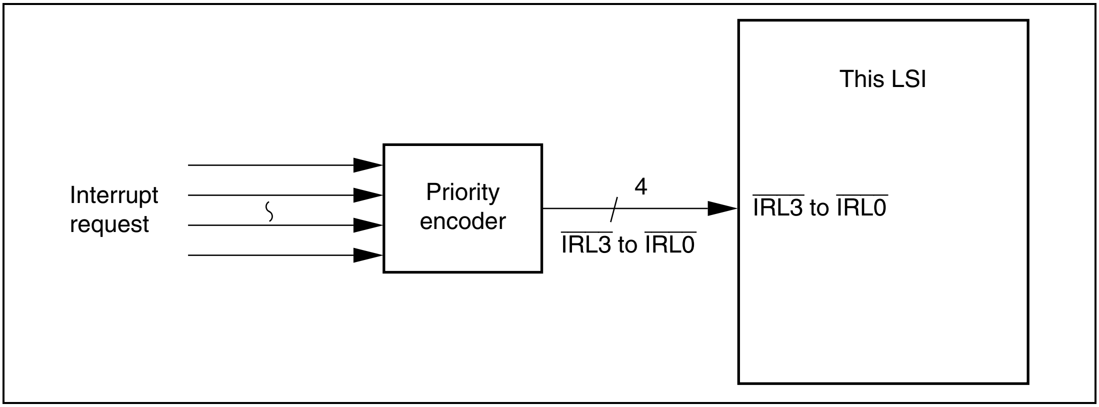

This LSI
Interrupt request
Priority encoder
4
$\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$
$\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$

Figure 6.2 Example of IRL Interrupt Connection

Table 6.2 $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ Pins and Interrupt Levels

$\overline{\text{IRL3}}$	$\overline{\text{IRL2}}$	$\overline{\text{IRL1}}$	$\overline{\text{IRL0}}$	Interrupt Priority Level	Interrupt Request
0	0	0	0	15	Level 15 interrupt request
0	0	0	1	14	Level 14 interrupt request
0	0	1	0	13	Level 13 interrupt request
0	0	1	1	12	Level 12 interrupt request
0	1	0	0	11	Level 11 interrupt request
0	1	0	1	10	Level 10 interrupt request
0	1	1	0	9	Level 9 interrupt request
0	1	1	1	8	Level 8 interrupt request
1	0	0	0	7	Level 7 interrupt request
1	0	0	1	6	Level 6 interrupt request
1	0	1	0	5	Level 5 interrupt request
1	0	1	1	4	Level 4 interrupt request
1	1	0	0	3	Level 3 interrupt request
1	1	0	1	2	Level 2 interrupt request
1	1	1	0	1	Level 1 interrupt request
1	1	1	1	0	No interrupt request

A noise-cancellation feature is built in, and the IRL interrupt is not detected unless the levels sampled at every supporting module cycle remain unchanged for two consecutive cycles, so that no transient level on the $\overline{\text{IRL}}$ pin change is detected. In the software standby mode, as the peripheral clock is stopped, noise cancellation is performed using the 32-kHz clock for the RTC instead. Therefore when the RTC is not used, interruption by means of IRL interrupts cannot be performed in software standby mode.

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

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

6.3.4 On-Chip Peripheral Module Interrupts

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

- Timer unit (TMU)
- Realtime clock (RTC)
- Serial communication interface (SCI, SCIF)
- Bus state controller (BSC)
- Watchdog timer (WDT)
- Direct memory access controller (DMAC)
- A/D converter (ADC)
- User-debugging interface (H-UDI)

Not every interrupt source is assigned a different interrupt vector, but sources are reflected in the interrupt event registers (INTEVT and INTEVT2), so it is easy to identify sources by branching with the INTEVT or INTEVT2 register value as an offset.

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

The interrupt mask bits (I3–I0) of the SR are not affected by the on-chip peripheral module interrupt processing.

TMU and RTC interrupts can restore the chip from the software standby state when the relevant interrupt level is higher than I3—I0 in the SR (but only when the RTC 32-kHz oscillator is used).

6.3.5 Interrupt Exception Processing and Priority

Tables 6.3 and 6.4 lists the codes for the interrupt event register (INTEVT and INTEVT2), and the order of interrupt priority. Each interrupt source is assigned unique code. The start address of the interrupt service routine is common to each interrupt source. This is why, for instance, the value of INTEVT or INTEVT2 is used as offset at the start of the interrupt service routine and branched to identify the interrupt source.

The order of priority of the on-chip peripheral module, IRQ, and PINT interrupts is set within the priority levels 0–15 at will by using the interrupt priority level set to registers A–E (IPRA–IPRE).

The order of priority of the on-chip peripheral module, IRQ, and PINT interrupts is set to zero by RESET.

When the order of priorities for multiple interrupt sources are set to the same level and such interrupts are generated at the same time, they are processed according to the default order listed in tables 6.3 and 6.4.

Table 6.3 Interrupt Exception Handling Sources and Priority (IRQ Mode)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		H'1C0 (H'1C0)	16	—	—	High
H-UDI		H'5E0 (H'5E0)	15	—	—	↕
IRQ	IRQ0	H'200–3C0* (H'600)	0–15 (0)	IPRC (3–0)	—	
	IRQ1	H'200–3C0* (H'620)	0–15 (0)	IPRC (7–4)	—	
	IRQ2	H'200–3C0* (H'640)	0–15 (0)	IPRC (11–8)	—	
	IRQ3	H'200–3C0* (H'660)	0–15 (0)	IPRC (15–12)	—	
	IRQ4	H'200–3C0* (H'680)	0–15 (0)	IPRD (3–0)	—	
	IRQ5	H'200–3C0* (H'6A0)	0–15 (0)	IPRD (7–4)	—	
DMAC	DEI0	H'200–3C0* (H'800)	0–15 (0)	IPRE (15–12)	High	
	DEI1	H'200–3C0* (H'820)			↕	
	DEI2	H'200–3C0* (H'840)				
	DEI3	H'200–3C0* (H'860)			Low	
SCIF (SCI2)	ERI2	H'200–3C0* (H'900)	0–15 (0)	IPRE (7–4)	High	
	RXI2	H'200–3C0* (H'920)			↕	
	BRI2	H'200–3C0* (H'940)				
	TXI2	H'200–3C0* (H'960)			Low	
ADC	ADI	H'200–3C0* (H'980)	0–15 (0)	IPRE (3–0)	—	
TMU0	TUNI0	H'400 (H'400)	0–15 (0)	IPRA (15–12)	—	
TMU1	TUNI1	H'420 (H'420)	0–15 (0)	IPRA (11–8)	—	
TMU2	TUNI2	H'440 (H'440)	0–15 (0)	IPRA (7–4)	High	
	TICPI2	H'460 (H'460)			Low	Low

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
RTC	ATI	H'480 (H'480)	0–15 (0)	IPRA (3–0)	High	High
	PRI	H'4A0 (H'4A0)			↕	↕
	CUI	H'4C0 (H'4C0)			Low	
SCI (SCI0)	ERI	H'4E0 (H'4E0)	0–15 (0)	IPRB (7–4)	High	
	RXI	H'500 (H'500)			↕	
	TXI	H'520 (H'520)				
	TEI	H'540 (H'540)			Low	
WDT	ITI	H'560 (H'560)	0–15 (0)	IPRB (15–12)	—	
BSC (REF)	RCMI	H'580 (H'580)	0–15 (0)	IPRB (11–8)	High	
	ROVI	H'5A0 (H'5A0)			Low	Low

Note: * The code corresponding to an interrupt level shown in table 6.5 is set.

Table 6.4 Interrupt Exception Handling Sources and Priority (IRL Mode)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		H'1C0 (H'1C0)	16	—	—	High
H-UDI		H'5E0 (H'5E0)	15	—	—	↕
IRL	IRL(3:0) = 0000	H'200 (H'200)	15	—	—	
	IRL(3:0) = 0001	H'220 (H'220)	14	—	—	
	IRL(3:0) = 0010	H'240 (H'240)	13	—	—	
	IRL(3:0) = 0011	H'260 (H'260)	12	—	—	
	IRL(3:0) = 0100	H'280 (H'280)	11	—	—	
	IRL(3:0) = 0101	H'2A0 (H'2A0)	10	—	—	
	IRL(3:0) = 0110	H'2C0 (H'2C0)	9	—	—	
	IRL(3:0) = 0111	H'2E0 (H'2E0)	8	—	—	
	IRL(3:0) = 1000	H'300 (H'300)	7	—	—	
	IRL(3:0) = 1001	H'320 (H'320)	6	—	—	
	IRL(3:0) = 1010	H'340 (H'340)	5	—	—	
	IRL(3:0) = 1011	H'360 (H'360)	4	—	—	
	IRL(3:0) = 1100	H'380 (H'380)	3	—	—	Low

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
IRL	IRL(3:0) = 1101	H'3A0 (H'3A0)	2	—	—	High
	IRL(3:0) = 1110	H'3C0 (H'3C0)	1	—	—	
IRQ	IRQ4	H'200–3C0* (H'680)	0–15 (0)	IPRD (3–0)	—	
	IRQ5	H'200–3C0* (H'6A0)	0–15 (0)	IPRD (7–4)	—	
DMAC	DEI0	H'200–3C0* (H'800)	0–15 (0)	IPRE (15–12)	High	
	DEI1	H'200–3C0* (H'820)				
	DEI2	H'200–3C0* (H'840)				
	DEI3	H'200–3C0* (H'860)			Low	
SCIF (SCI2)	ERI2	H'200–3C0* (H'900)	0–15 (0)	IPRE (7–4)	High	
	RXI2	H'200–3C0* (H'920)				
	BRI2	H'200–3C0* (H'940)				
	TXI2	H'200–3C0* (H'960)			Low	
ADC	ADI	H'200–3C0* (H'980)	0–15 (0)	IPRE (3–0)	—	
TMU0	TUNI0	H'400 (H'400)	0–15 (0)	IPRA (15–12)	—	
TMU1	TUNI1	H'420 (H'420)	0–15 (0)	IPRA (11–8)	—	
TMU2	TUNI2	H'440 (H'440)	0–15 (0)	IPRA (7–4)	High	
	TICPI2	H'460 (H'460)			Low	
RTC	ATI	H'480 (H'480)	0–15 (0)	IPRA (3–0)	High	
	PRI	H'4A0 (H'4A0)				
	CUI	H'4C0 (H'4C0)			Low	
SCI (SCI0)	ERI	H'4E0 (H'4E0)	0–15 (0)	IPRB (7–4)	High	
	RXI	H'500 (H'500)				
	TXI	H'520 (H'520)				
	TEI	H'540 (H'540)			Low	
WDT	ITI	H'560 (H'560)	0–15 (0)	IPRB (15–12)	—	
BSC (REF)	RCMI	H'580 (H'580)	0–15 (0)	IPRB (11–8)	High	
	ROVI	H'5A0 (H'5A0)			Low	Low

Note: * The code corresponding to an interrupt level shown in table 6.5 is set.

Table 6.5 Interrupt Level and INTEVT Code

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

6.4 Description of Registers

The INTC has the following registers. Refer to section 23, Control Registers Table, for more detail of the addresses and access size.

- Interrupt control register 0 (ICR0)
- Interrupt control register 1 (ICR1)
- 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 request register 0 (IRR0)
- Interrupt request register 1 (IRR1)
- Interrupt request register 2 (IRR2)

6.4.1 Interrupt Priority Registers A–E (IPRA–IPRE)

The interrupt priority level setting registers A–E (IPRA–IPRE) are 16-bit read/write registers that set priority levels from 0 to 15 for on-chip peripheral module interrupts. These registers are initialized to H'0000 at power-on reset, manual reset, or in hardware standby mode, but is not initialized in standby mode.

Table 6.6 lists the relationship between the interrupt sources and the IPRA and IPRE bits.

Table 6.6 Interrupt Request Sources and IPRA–IPRE

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
IPRA	TMU0	TMU1	TMU2	RTC
IPRB	WDT	REF	SCI0	Reserved*
IPRC	IRQ3	IRQ2	IRQ1	IRQ0
IPRD	Reserved*	Reserved*	IRQ5	IRQ4
IPRE	DMAC	Reserved*	SCIF	ADC

Note: * These bits are always read as 0. The write value should be 0.

As shown in table 6.6, four sets of on-chip peripheral module, IRQ interrupts are assigned to each register. 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) are set with values from H'0 (0000) to H'F (1111). Setting H'0 means priority level 0 (masking is requested); H'F is priority level 15 (the highest level). A reset initializes IPRA–IPRE to H'0000.

H'0 should be set into bits corresponding to an unused interrupt.

6.4.2 Interrupt Control Register 0 (ICR0)

The interrupt control register 0 (ICR0) is a 16-bit register that sets the input signal detection mode of the external interrupt input pin NMI and indicates the input signal level to the NMI pin. This register is initialized to H'0000 at power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	0/1*	R	NMI Input Level Sets the level of the signal input at the NMI pin. This bit can be read to determine the NMI 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 write value should always be 0.
8	NMIE	0	R/W	NMI Edge Select Selects whether the interrupt request signal is detected on the falling or rising edge of NMI input. 0: Interrupt request signal is detected on falling edge of NMI input 1: Interrupt request signal is detected on rising edge of NMI input
7 to 0	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.

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

6.4.3 Interrupt Control Register 1 (ICR1)

The interrupt control register 1 (ICR1) is a 16-bit register that specifies the detection mode to external interrupt input pins, IRQ0 to IRQ5 individually: rising edge, falling edge, or low level.

Bit	Bit Name	Initial Value	R/W	Description
15	MAI	0	R/W	Mask All Interrupts When set to 1, masks all interrupt requests when a low level is being input to the NMI pin. Masks NMI interrupts in standby mode. 0: All interrupt requests are not masked when a low level is being input to the NMI pin 1: All interrupt requests are masked when a low level is being input to the NMI pin

Bit	Bit Name	Initial Value	R/W	Description
14	IRQLVL	1	R/W	Interrupt Request Level Detect Selects whether the IRQ3–IRQ0 pins are used as four independent interrupt pins or as 15-level interrupt pins encoded as $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$. 0: Used as four independent interrupt request pins IRQ3–IRQ0 1: Used as encoded 15-level interrupt pins as $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$
13	BLMSK	0	R/W	BL Bit Mask Specifies whether NMI interrupts are masked when the BL bit of the SR register is 1. 0: NMI interrupts are masked when the BL bit is 1 1: NMI interrupts are accepted regardless of the BL bit setting
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 10	IRQ51S IRQ50S	0 0	R/W R/W	IRQ5 Sense Select Select whether the interrupt signal to the IRQ5 pin is detected at the rising edge, at the falling edge, or at low level. 00: An interrupt request is detected at IRQ5 input falling edge 01: An interrupt request is detected at IRQ5 input rising edge 10: An interrupt request is detected at IRQ5 input low level 11: Reserved (Setting prohibited)
9 8	IRQ41S IRQ40S	0 0	R/W R/W	IRQ4 Sense Select Select whether the interrupt signal to the IRQ4 pin is detected at the rising edge, at the falling edge, or at low level. 00: An interrupt request is detected at IRQ4 input falling edge 01: An interrupt request is detected at IRQ4 input rising edge 10: An interrupt request is detected at IRQ4 input low level 11: Reserved (Setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ31S	0	R/W	IRQ3 Sense Select
6	IRQ30S	0	R/W	Select whether the interrupt signal to the IRQ3 pin is detected at the rising edge, at the falling edge, or at low level. 00: An interrupt request is detected at IRQ3 input falling edge 01: An interrupt request is detected at IRQ3 input rising edge 10: interrupt request is detected at IRQ3 input low level 11: Rserved (Setting prohibited)
5	IRQ21S	0	R/W	IRQ2 Sense Select
4	IRQ20S	0	R/W	Select whether the interrupt signal to the IRQ2 pin is detected at the rising edge, at the falling edge, or at low level. 00: An interrupt request is detected at IRQ2 input falling edge 01: An interrupt request is detected at IRQ2 input rising edge 10: An interrupt request is detected at IRQ2 input low level 11: Reserved (Setting prohibited)
3	IRQ11S	0	R/W	IRQ1 Sense Select
2	IRQ10S	0	R/W	Select whether the interrupt signal to the IRQ1 pin is detected at the rising edge, at the falling edge, or at low level. 00: An interrupt request is detected at IRQ1 input falling edge 01: An interrupt request is detected at IRQ1 input rising edge 10: An interrupt request is detected at IRQ1 input low level 11: Reserved (Setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
1	IRQ01S	0	R/W	IRQ0 Sense Select
0	IRQ00S	0	R/W	Select whether the interrupt signal to the IRQ0 pin is detected at the rising edge, at the falling edge, or at low level. 00: An interrupt request is detected at IRQ0 input falling edge 01: An interrupt request is detected at IRQ0 input rising edge 10: An interrupt request is detected at IRQ0 input low level 11: Reserved (Setting prohibited)

6.4.4 Interrupt Request Register 0 (IRR0)

The interrupt request register 0 (IRR0) is an 8-bit register that indicates interrupt requests from external input pins IRQ0 to IRQ5.

When clearing IRQ5R–IRQ0R bit to 0, 0 should be written to the bit after the bit is set to 1 and the contents of 1 are read. Only 0 can be written to IRQ5R–IRQ0R.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0		These bits are always read as 0. The write value should always be 0.
5	IRQ5R	0	R	IRQ5 Interrupt Request Indicates whether an interrupt request is input to the IRQ5 pin. When edge detection mode is set for IRQ5, an interrupt request is cleared by clearing the IRQ5R bit. 0: An interrupt request is not input to IRQ5 pin 1: An interrupt request is input to IRQ5 pin

Bit	Bit Name	Initial Value	R/W	Description
4	IRQ4R	0	R/W	IRQ4 Interrupt Request Indicates whether an interrupt request is input to the IRQ4 pin. When edge detection mode is set for IRQ4, an interrupt request is cleared by clearing the IRQ4R bit. 0: An interrupt request is not input to IRQ4 pin 1: An interrupt request is input to IRQ4 pin
3	IRQ3R	0	R/W	IRQ3 Interrupt Request Indicates whether an interrupt request is input to the IRQ3 pin. When edge detection mode is set for IRQ3, an interrupt request is cleared by clearing the IRQ3R bit. 0: An interrupt request is not input to IRQ3 pin 1: An interrupt request is input to IRQ3 pin
2	IRQ2R	0	R/W	IRQ2 Interrupt Request Indicates whether an interrupt request is input to the IRQ2 pin. When edge detection mode is set for IRQ2, an interrupt request is cleared by clearing the IRQ2R bit. 0: An interrupt request is not input to IRQ2 pin 1: An interrupt request is input to IRQ2 pin
1	IRQ1R	0	R/W	IRQ1 Interrupt Request Indicates whether an interrupt request is input to the IRQ1 pin. When edge detection mode is set for IRQ1, an interrupt request is cleared by clearing the IRQ1R bit. 0: An interrupt request is not input to IRQ1 pin 1: An interrupt request is input to IRQ1 pin
0	IRQ0R	0	R/W	IRQ0 Interrupt Request (IRQ0R): Indicates whether an interrupt request is input to the IRQ0 pin. When edge detection mode is set for IRQ0, an interrupt request is cleared by clearing the IRQ0R bit. 0: An interrupt request is not input to IRQ0 pin 1: An interrupt request is input to IRQ0 pin

6.4.5 Interrupt Request Register 1 (IRR1)

The interrupt request register 1 (IRR1) is an 8-bit read-only register that indicates whether DMAC or IrDA interrupt requests are generated.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	DEI3R	0	R	DEI3 Interrupt Request Indicates whether a DEI3 (DMAC) interrupt request is generated. 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 generated. 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 generated. 0: A DEI1 interrupt request is not generated 1: A DEI1 interrupt request is generated
0	DEI0R	0	R	DEI0 Interrupt Request Indicates whether a DEI0 (DMAC) interrupt request is generated. 0: A DEI0 interrupt request is not generated 1: A DEI0 interrupt request is generated

6.4.6 Interrupt Request Register 2 (IRR2)

The interrupt request register 2 (IRR2) is an 8-bit read-only register that indicates whether A/D converter, or SCIF interrupt requests are generated.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	ADIR	0	R	ADI Interrupt Request Indicates whether an ADI (ADC) interrupt request is generated. 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 (SCIF) interrupt request is generated. 0: TXI2 interrupt request is not generated 1: A TXI2 interrupt request is generated
2	BRI2R	0	R	BRI2 Interrupt Request Indicates whether a BRI2 (SCIF) interrupt request is generated. 0: A BRI2 interrupt request is not generated 1: A BRI2 interrupt request is generated
1	RXI2R	0	R	RXI2 Interrupt Request Indicates whether an RXI2 (SCIF) interrupt request is 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 (SCIF) interrupt request is generated. 0: An ERI2 interrupt request is not generated 1: An ERI2 interrupt request is generated

6.5 Description of Operation

6.5.1 Interrupt Sequence

The sequence of interrupt operations is explained below. Figure 6.3 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest priority interrupt from the interrupt requests sent, following the priority levels set in interrupt priority registers A to E (IPRA to IPRE). Lower priority interrupts are held pending. If two of these interrupts have the same priority level or if multiple interrupts occur within a single module, the interrupt with the highest default priority or the highest priority within its IPR setting unit (as indicated in table 6.3 and table 6.4) is selected.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask bits (I3–I0) in the status register (SR) of the CPU. If the request priority level is higher than the level in bits I3–I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU. When the interrupt controller receives an interrupt, a low level is output from the $\overline{\text{IRQOUT}}$ pin.
4. Detection timing: The INTC operates in synchronization with the peripheral clock (Pϕ), and reports the interrupt request to the CPU. The CPU receives an interrupt at a break in instruction.
5. The interrupt source code is set in the interrupt event registers (INTEVT and INTEVT2).
6. The SR and PC are saved to SSR and SPC, respectively.
7. The BL, MD, and RB in SR are set to 1.
8. The CPU jumps to the start address of the interrupt handler (the sum of the value set in the vector base register (VBR) and H'00000600). This jump is not a delayed branch. The interrupt handler may branch with the INTEVT register value as its offset in order to identify the interrupt source. This enables it to branch to the processing routine for the individual interrupt source.

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

2. $\overline{\text{IRQOUT}}$ outputs a low level until the interrupt request is cleared. However, if the interrupt source is masked by an interrupt mask bit, the $\overline{\text{IRQOUT}}$ pin returns to the high level. The level is output without regard to the BL bit.

3. The interrupt source flag should be cleared in the interrupt handler. The interrupt source flag should be cleared in the interrupt handler. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read the source flag after it has been cleared, then wait for the interval shown in "Time for priority decision and SR mask bit comparison" in table 6.7 before clearing the BL bit or executing an RTE instruction.

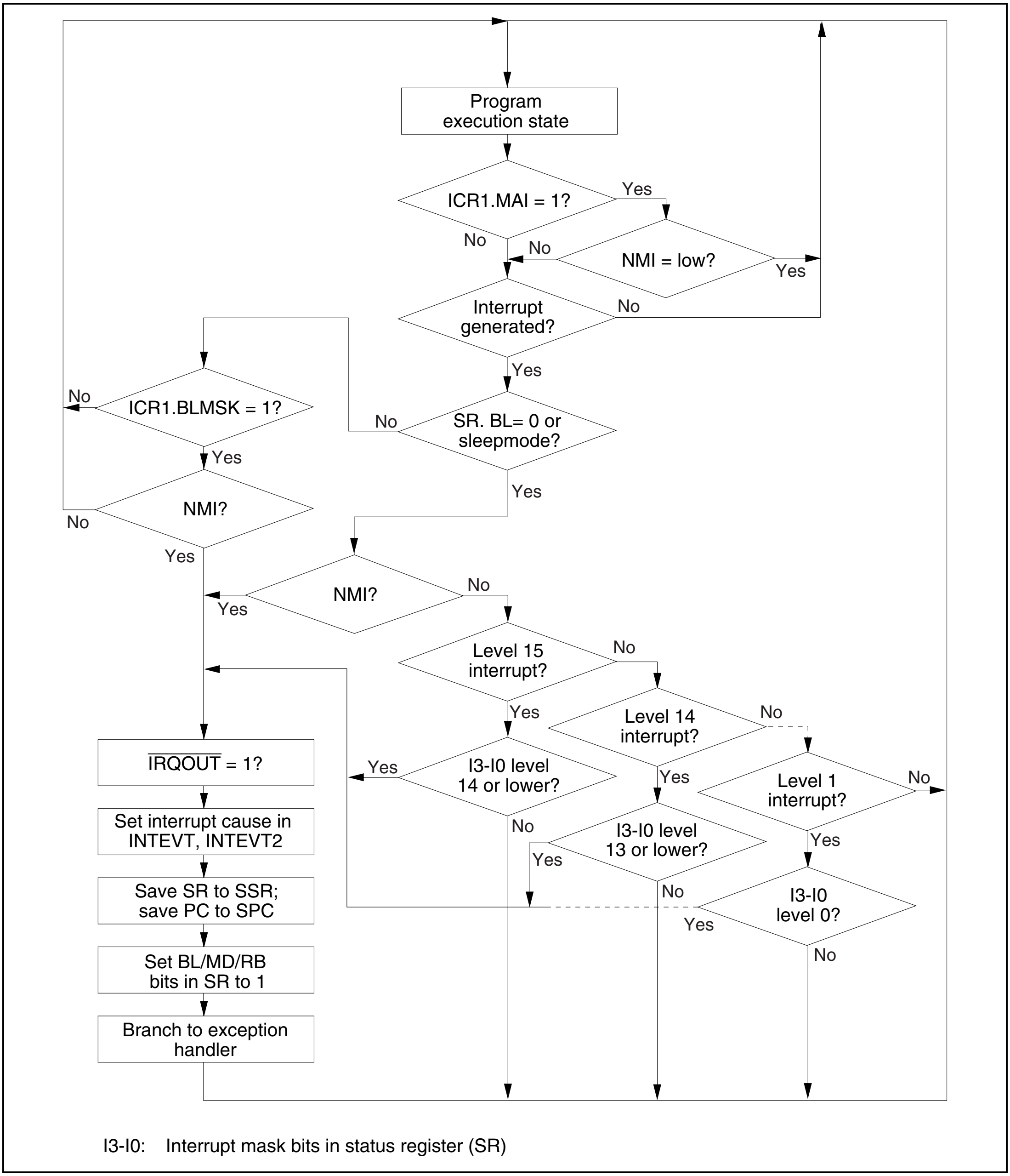


Figure 6.3 Interrupt Operation Flowchart

6.5.2 Multiple Interrupts

When multiple interrupts are used, the structure of the interrupt service routine should be as follows.

1. Branch to a specific interrupt handler corresponding to a code set in INTEVT and INTEVT2. The code in INTEVT and INTEVT2 can be used as a branch-offset for branching to the specific handler.
2. Clear the cause of the interrupt in each specific handler.
3. Save SSR and SPC to the memory.
4. Clear the BL bit in SR, and set the accepted interrupt level in the interrupt mask bits in SR.
5. Handle the interrupt.
6. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted after clearing BL in step 4.

6.6 Interrupt Response Time

The time from generation of an interrupt request until interrupt exception processing is performed and fetching of the first instruction of the exception handler is started (the interrupt response time) is shown in table 6.7. Figure 6.4 shows an example of pipeline operation when an IRL interrupt is accepted. When SR.BL is 1, interrupt exception processing is masked, and is kept waiting until completion of an instruction that clears BL to 0.

The response time is represented by the clock number of Iϕ. Depending on the Pϕ phase when an interrupt is occurred, one clock period of Pϕ may vary from the contents of this table.

Table 6.7 Interrupt Response Time

	Number of States				
Item	NMI	IRQ	IRL	Peripheral Modules	Notes
Time for priority decision and SR mask bit comparison	0.5 × Icyc + 1.5 × Bcyc	1.5 × Icyc + 0.5 × Bcyc + 2 × Pcyc*2	0.5 × Icyc + 0.5 × Bcyc + 3.5 × Pcyc	0.5 × Icyc + 1.5 × Pcyc*3 0.5 × Icyc + 3 × Pcyc*4	
Wait time until end of sequence being executed by CPU	X (≥ 0) × Icyc	X (≥ 0) × Icyc	X (≥ 0) × Icyc	X (≥ 0) × Icyc	Interrupt exception processing is kept waiting until the executing instruction ends. If the number of instruction execution states is S*1, the maximum wait time is: X = S – 1. However, if BL is set to 1 by instruction execution or by an exception, interrupt exception processing is deferred until completion of an instruction that clears BL to 0. If the following instruction masks interrupt exception processing, the processing may be further deferred.
Time from interrupt exception processing (save of SR and PC) until fetch of first instruction of exception service routine is started	5 × Icyc	5 × Icyc	5 × Icyc	5 × Icyc	

		Number of States				
Item		NMI	IRQ	IRL	Peripheral Modules	Notes
Response time	Total	(5.5 + X) × Icyc + 1.5 × Bcyc	(6.5 + X) × Icyc + 0.5 × Bcyc + 2 × Pcyc*4	(5.5 + X) × Icyc + 0.5 × Bcyc + 3.5 × Pcyc	(5.5 + X) × Icyc + 1.5 × Pcyc*3 (5.5 + X) × Icyc + 3 × Pcyc*4	
	Minimum case	7	9	9.5	7*3/8.5*4	Iφ: Bφ: Pφ = 1:1:1
	Maximum case	10.5 + S	15.5 + S	20.5 + S	10.5 + S*3 16.5 + S*4	Iφ: Bφ: Pφ = 4:1:1

Icyc: Duration of one cycle of Iφ.

Bcyc: Duration of one cycle of Bφ.

Pcyc: Duration of one cycle of Pφ.

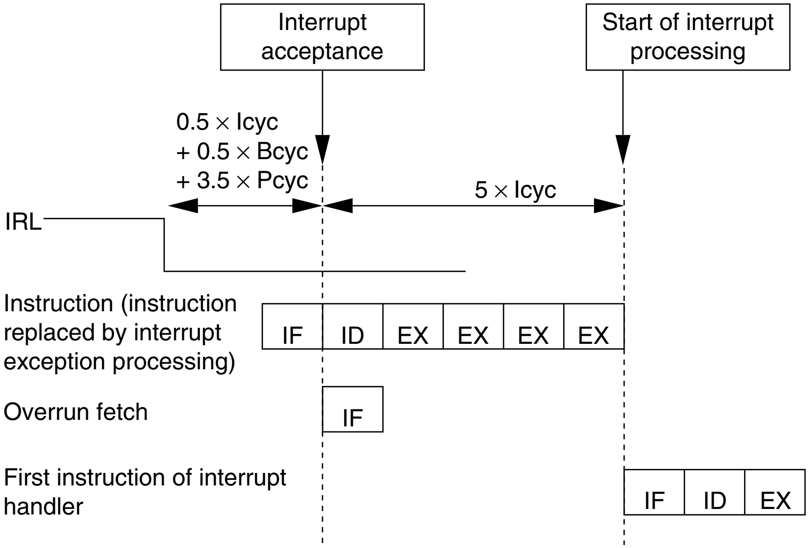
Notes: 1. S also includes the memory access wait time.

The processing requiring the maximum execution time is LDC.L @Rm+, SR. When the memory access is a cache-hit, this requires seven instruction execution cycles. When the external access is performed, the corresponding number of cycles must be added. There are also instructions that perform two external memory accesses; if the external memory access is slow, the number of instruction execution cycles will increase accordingly.

2. Edge detection.

3. Extended modules: TMU, RTC, SCI, WDT, REFC

4. Extended modules: DMAC, ADC, SCIF


Interrupt acceptance
Start of interrupt processing
0.5 × Icyc
+ 0.5 × Bcyc
+ 3.5 × Pcyc
5 × Icyc
IRL
Instruction (instruction replaced by interrupt exception processing)
IF
ID
EX
EX
EX
EX
Overrun fetch
IF
First instruction of interrupt handler
IF
ID
EX

IF: Instruction fetch: Instruction is fetched from memory in which program is stored.
ID: Instruction decode: Fetched instruction is decoded.
EX: Instruction execution: Data operation and address calculation are performed in accordance with result of decoding.

Figure 6.4 Example of Pipeline Operations when IRL Interrupt is Accepted

Section 7 User Break Controller

The UBC provides functions that simplify program debugging. Using this function, a self-monitor debugger can be easily prepared, and a program can be debugged using this LSI alone, without using an in-circuit emulator. Instruction fetches, data read/write, data size, data contents, address values, and the timing to stop execution at instruction fetch can be set to the UBC. The UBC block diagram is shown in figure 7.1.

7.1 Features

The UBC has the following features:

- The following break comparison conditions can be set.

  Number of break channels: (channels A and B)

  Address: comparison bits are masked in units of 32 bits.

  One of the two address buses (the virtual address bus (LAB) and the internal address bus (IAB)) can be selected

  Data: only on channel B, 32-bit maskable

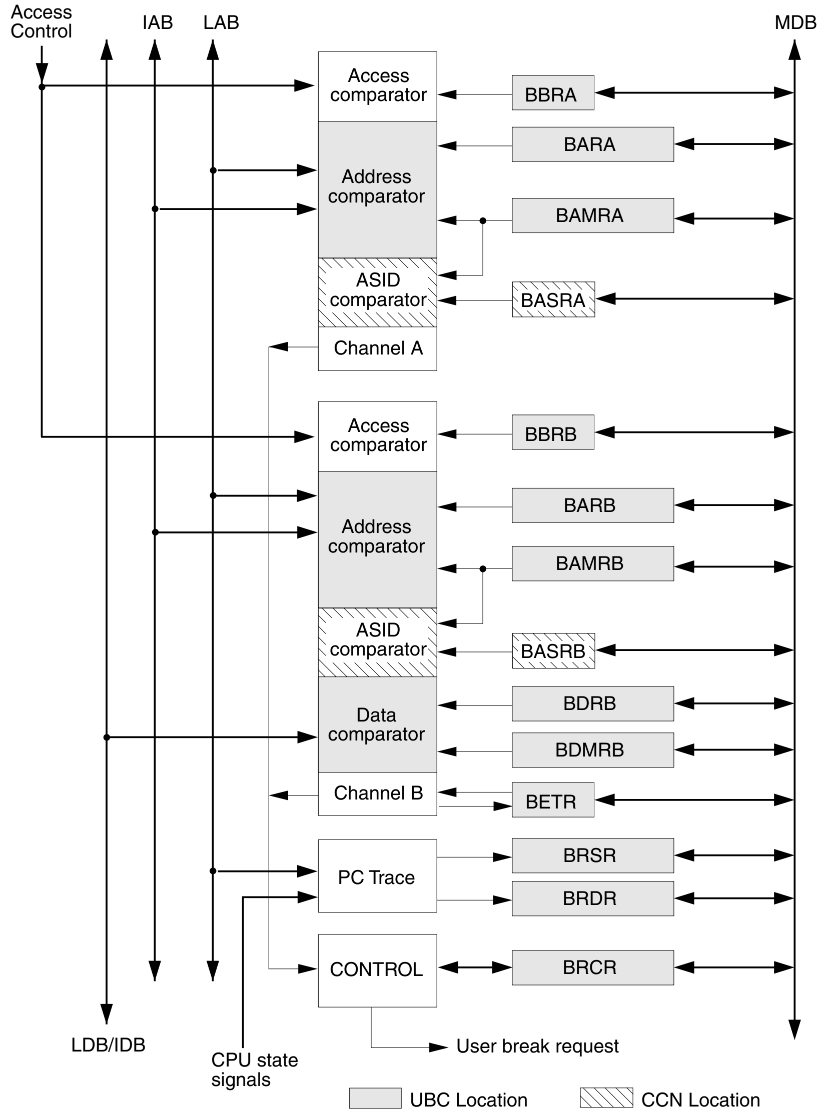
  One of two data buses (the virtual data bus (LDB) or the internal data bus (IDB)) can be selected.

  Bus master: CPU cycle or DMAC cycle

  Bus cycle: instruction fetch or data access

  Read/write

  Operand size: byte, word, or longword
- A user-designed user-break 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 instruction is executed.
- The number of repeat times can be specified as a break condition. (It is only for channel B)
- Maximum repeat times for the break condition: $2^{12} - 1$ times.
- Eight pairs of branch source/destination buffers.



Legend

BBRA:	Break bus cycle register A	BASRB:	Break ASID register B
BARA:	Break address register A	BDRB:	Break data register B
BAMRA:	Break address mask register A	BDMRB:	Break data mask register B
BASRA:	Break ASID register A	BETR:	Break execution times register
BBRB:	Break bus cycle register B	BRSR:	Branch source register
BARB:	Break address register B	BRDR:	Branch destination register
BAMRB:	Break address mask register B	BRCR:	Break control register

Figure 7.1 Block Diagram of User Break Controller

7.2 Description of Registers

The UBC has the following registers. Refer to section 23, Control Registers Table, for more detail of the addresses and access size.

- Break address register A (BARA)
- Break address mask register A (BAMRA)
- Break bus cycle register A (BBRA)
- Break address register B (BARB)
- 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 count break register (BETR)
- Branch source register (BRSR)
- Branch destination register (BRDR)
- Break ASID register A (BASRA)
- Break ASID register B (BASRB)

7.2.1 Break Address Register A (BARA)

BARA is a 32-bit read/write register. BARA specifies the address used as a break condition in channel A.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAA31 to BAA0	0	R/W	Break Address Stores the address on the LAB or IAB that specifies break conditions of channel A.

7.2.2 Break Address Mask Register A (BAMRA)

BAMRA is a 32-bit read/write register. BAMRA specifies bits masked in the break address specified by BARA.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAMA31 to BAMA0	0	R/W	Break Address Mask Bit Specifies bits masked in the channel A break address bits specified by BARA (BAA31–BAA0). 0: Break address bit BAAn of channel A is included in the break condition 1: Break address bit BAAn of channel A is masked and is not included in the break condition

Note: N: 31 to 0.

7.2.3 Break Bus Cycle Register A (BBRA)

Break bus cycle register A (BBRA) is a 16-bit read/write register, which specifies (1) CPU cycle or DMAC cycle, (2) instruction fetch or data access, (3) read or write, and (4) operand size in the break conditions of 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 should always be 0.
7	CDA1	0	R/W	CPU Cycle/DMAC Cycle Select A
6	CDA0	0	R/W	Selects the CPU cycle or DMAC cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed X1: The break condition is the CPU cycle 10: The break condition is the DMAC cycle
5	IDA1	0	R/W	Instruction Fetch/Data Access Select A
4	IDA0	0	R/W	Selects the instruction fetch cycle or data access cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycle 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycle or data access cycle

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

Note: X :Don’t care

7.2.4 Break Address Register B (BARB)

BARB is a 32-bit read/write register. BARB specifies the address used as a break condition in channel B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAB31 to BAB0	0	R/W	Break Address Stores the address of LAB or IAB that specifies the break conditions of channel B.

7.2.5 Break Address Mask Register B (BAMRB)

BAMRB is a 32-bit read/write register. BAMRB specifies bits masked in the break address specified by BARB.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAMB31 to BAMB0	0	R/W	Break Address Mask Specifies bits masked in the channel B break address bits specified by BARB (BAB31—BAB0). 0: Break address BABn of channel B is included in the break condition 1: Break address BABn of channel B is masked and is not included in the break condition

Note: n:31 to 0

7.2.6 Break Data Register B (BDRB)

BDRB is a 32-bit read/write register.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDB31 to BDB0	0	R/W	Break Data Bit

7.2.7 Break Data Mask Register B (BDMRB)

BDMRB is a 32-bit read/write register. BDMRB specifies bits masked in the break data specified by BDRB.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDMB31 to BDMB0	0	R/W	Break Data Mask 0: Break data BDBn of channel B is included in the break condition 1: Break data BDBn of channel B is masked and is not included in the break condition

Notes: n: 31 to 0

1. Specify an operand size when including the value of the data bus in the break condition.
2. When a byte size is selected as a break condition, the break data must be set in bits 15-8 in BDRB for an even break address and bits 7-0 for an odd break address.

7.2.8 Break Bus Cycle Register B (BBRB)

Break bus cycle register B (BBRB) is a 16-bit read/write register, which specifies, (1) CPU cycle or DMAC cycle, (2) instruction fetch or data access, (3) read/write, and (4) operand size in the break conditions of channel B.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	0	R	Reserved These bits are always read as 0. These bits are always read as 0.
7	CDB1	0	R/W	CPU Cycle/DMAC Cycle Select B
6	CDB0	0	R/W	Select the CPU cycle or DMAC cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed X1: The break condition is the CPU cycle 10: The break condition is the DMAC cycle
5	IDB1	0	R/W	Instruction Fetch/Data Access Select B
4	IDB0	0	R/W	Select the instruction fetch cycle or data access cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycle 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycle or data 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 cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or write cycle
1	SZB1	0	R/W	Operand Size Select B
0	SZB0	0	R/W	Select the operand size of the bus cycle for the channel B break condition. 00: The break condition does not include operand size 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

Note: X:Don't care

7.2.9 Break Control Register (BRCR)

BRCR sets the following conditions:

1. Channels A and B are used in two independent channels condition or under the sequential condition.
2. A break is set before or after instruction execution.
3. A break is set by the number of execution times.
4. Determine whether to include data bus on channel B in comparison conditions.
5. Enable PC trace.
6. Enable the ASID check.

The break control register (BRCR) is a 32-bit read/write register that has break conditions match flags and bits for 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 value should always be 0.
21	BASMA	0	R/W	Break ASID Mask A Specifies whether the bits of the channel A break ASID7-ASID0 (BASA7 to BASA0) set in BASRA are masked or not. 0: All BASRA bits are included in break condition, ASID is checked 1: No BASRA bits are included in break condition, ASID is not checked
20	BASMB	0	R/W	Break ASID Mask B Specifies whether the bits of channel B break ASID7-ASID0 (BASB7 to BASB0) set in BASRB are masked or not. 0: All BASRB bits are included in break condition, ASID is checked 1: No BASRB bits are included in break condition, ASID is not checked
19 to 16	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
15	SCMFCA	0	R/W	CPU Condition Match Flag A When the CPU bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The CPU cycle condition for channel A does not match 1: The CPU cycle condition for channel A matches
14	SCMFCB	0	R/W	CPU Condition Match Flag B When the CPU bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The CPU cycle condition for channel B does not match 1: The CPU cycle condition for channel B matches
13	SCMFDA	0	R/W	DMAC Condition Match Flag A When the on-chip DMAC bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The DMAC cycle condition for channel A does not match 1: The DMAC cycle condition for channel A matches
12	SCMFDB	0	R/W	DMAC Condition Match Flag B When the on-chip DMAC bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The DMAC cycle condition for channel B does not match 1: The DMAC cycle condition for channel B matches

Bit	Bit Name	Initial Value	R/W	Description
11	PCTE	0	R/W	PC Trace Enable Enables PC trace. 0: Disables PC trace 1: Enables PC trace
10	PCBA	0	R/W	PC Break Select A (PCBA): Selects the break timing of the instruction fetch cycle for channel A as before or after instruction execution. 0: PC break of channel A is set before instruction execution 1: PC break of channel A is set after instruction execution
9	—	0	R	Reserved
8	—	0	R	These bits are always read as 0. The write value should always be 0.
7	DBEB	0	R/W	Data Break Enable B Selects whether or not the data bus condition is included in the break condition of channel B. 0: No data bus condition is included in the condition of channel B 1: The data bus condition is included in the condition of channel B
6	PCBB	0	R/W	PC Break Select B Selects the break timing of the instruction fetch cycle for channel B as before or after instruction execution. 0: PC break of channel B is set before instruction execution 1: PC break of channel B is set after instruction execution
5	—	0	R	Reserved
4	—	0	R	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	SEQ	0	R/W	Sequence Condition Select Selects two conditions of channels A and B as independent or sequential. 0: Channels A and B are compared under the independent condition 1: Channels A and B are compared under the sequential condition
2	—	0	R	Reserved
1	—	0	R	These bits are always read as 0. The write value should always be 0.
0	ETBE	0	R/W	The Number of Execution Times Break Enable Enable the execution-times break condition only on channel B. If this bit is 1 (break enable), a user break is issued when the number of break conditions matches with the number of execution times that is specified by the BETR register. 0: The execution-times break condition is masked on channel B 1: The execution-times break condition is enabled on channel B

7.2.10 Execution Times Break Register (BETR)

When the execution-times break condition of channel B is enabled, this register specifies the number of execution times to make the break. The maximum number is $2^{12} - 1$ times. Everytime the break condition is satisfied, BETR is decremented by 1. A break is issued when the break condition is satisfied after the BETR becomes H'0001.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	—	0	R/W	Number of execution times

7.2.11 Branch Source Register (BRSR)

BRSR is a 32-bit read register. BRSR stores the last fetched address before branch and the pointer (3 bits) which indicates the number of cycles from fetch to execution for the last executed instruction. BRSR has the flag bit that is set to 1 when branch occurs. This flag bit is cleared to 0, when BRSR is read and also initialized by power-on resets or manual resets. Other bits are not initialized by reset. Eight BRSR registers have queue structure and a stored register is shifted every branch.

Bit	Bit Name	Initial Value	R/W	Description
31	SVF	0	R	BRSR Valid Flag Indicates whether the address and the pointer by which the branch source address can be calculated. When a branch source address is fetched, this flag is set to 1. This flag is cleared to 0 in reading BRSR. 0: The value of BRSR register is invalid 1: The value of BRSR register is valid
30 to 28	PID2 to PID0	*	R	Instruction Decode Pointer PID is a 3-bit binary pointer (0–7). These bits indicate the instruction buffer number which stores the last executed instruction before branch. Even: PID indicates the instruction buffer number. Odd: PiD+2 indicates the instruction buffer number
27 to 0	BSA27 to BSA0	*	R	Branch Source Address These bits store the last fetched address before branch.

Note: * Undefined value

7.2.12 Branch Destination Register (BRDR)

BRDR is a 32-bit read register. BRDR stores the branch destination fetch address. BRDR has the flag bit that is set to 1 when branch occurs. This flag bit is cleared to 0, when BRDR is read and also initialized by power-on resets or manual resets. Other bits are not initialized by resets. Eight BRDR registers have queue structure and a stored register is shifted every branch.

Bit	Bit Name	Initial Value	R/W	Description
31	DVF	0	R	BRDR Valid Flag Indicates whether a branch destination address is stored. When a branch destination address is fetched, this flag is set to 1. This flag is set to 0 in reading BRDR. 0: The value of BRDR register is invalid 1: The value of BRDR register is valid
30 to 28	—	*	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	BDA27 to BDA0	*	R	Branch Destination Address These bits store the first fetched address after branch.

Note: * Undefined value

7.2.13 Break ASID Register A (BASRA)

Break ASID register A (BASRA) is an 8-bit read/write register that specifies the ASID that serves as the break condition for channel A. It is not initialized by resets. It is located in CCN.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BASA7 to BASA0	*	R/W	Break ASID These bits store the ASID (bits 7 to 0) that is the channel A break condition.

Note: * Undefined value

7.2.14 Break ASID Register B (BASRB)

Break ASID register B (BASRB) is an 8-bit read/write register that specifies the ASID that serves as the break condition for channel B. It is not initialized by resets. It is located in CCN.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BASB7 to BASB0	*	R/W	Break ASID These bits store the ASID (bits 7 to 0) that is the channel B break condition.

Note: * Undefined value

7.3 Description of Operation

7.3.1 Flow of the User Break Operation

The flow from setting of break conditions to user break exception processing is described below:

1. The break addresses and the corresponding ASIDs are loaded in the BARA, BARB, BASRA and BASRB. The masked addresses are set in the BAMRA and BAMRB. The break data is set in the BDRB. The masked data is set in the BDMRB. The breaking bus conditions are set in the BBRA and BBRB. Three groups of the BBRA and BBRB (CPU cycle/DMAC cycle select, instruction fetch/data access select, and read/write select) are each set. No user break will be generated if even one of these groups is set with 00. The respective conditions are set in the bits of the BRCR.
2. When the break conditions are satisfied, the UBC sends a user break request to the interrupt controller. The break type will be sent to CPU indicating the instruction fetch, pre/post instruction break, or data access break. When conditions match up, the CPU condition match flags (SCMFCA and SCMFCB) and DMAC condition match flags (SCMFDA and SCMFDB) for the respective channels are set.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCMFDB) can be used to check if the set conditions match or not. The matching of the conditions sets flags, 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 break occur around the same time, there will be only one break request to the CPU, but these two break channel match flags could be both set.

7.3.2 Break on Instruction Fetch Cycle

1. When CPU/instruction fetch/read/word or longword is set in the break bus cycle registers (BBRA/BBRB), the break condition becomes the CPU instruction fetch cycle. Whether it then breaks before or after the execution of the instruction can then be selected with the PCBA/PCBB bits of the break control register (BRCR) for the appropriate channel.
2. An instruction set for a break before execution breaks when it is confirmed that the instruction has been fetched and will be executed. This means this feature cannot be used on instructions fetched by overrun (instructions fetched at a branch or during an interrupt transition, but not to be executed). When this kind of break is set for the delay slot of a delay branch instruction, the break is generated prior to execution of the instruction that then first accepts the break. Meanwhile, the break set for pre-instruction-break on delay slot instruction and post-instruction-break on SLEEP instruction are also prohibited.
3. When the condition is specified to be occurred after execution, the instruction set with the break condition is executed and then the break is generated prior to the execution of the next instruction. As with pre-execution breaks, this cannot be used with overrun fetch instructions.

When this kind of break is set for a delay branch instruction, the break is generated at the instruction that then first accepts the break.

4. When an instruction fetch cycle is set for channel B, break data register B (BDRB) is ignored. There is thus no need to set break data for the break of the instruction fetch cycle.

7.3.3 Break by Data Access Cycle

1. The memory cycles in which CPU data access breaks occur are from instructions.
2. The relationship between the data access cycle address and the comparison condition for operand size are listed in table 7.1:

Table 7.1 Data Access Cycle Addresses and Operand Size Comparison Conditions

Access Size	Address Compared
Longword	Compares break address register bits 31–2 to address bus bits 31–2
Word	Compares break address register bits 31–1 to address bus bits 31–1
Byte	Compares break address register bits 31–0 to address bus bits 31–0

This means that when address H'00001003 is set without specifying the size condition, for example, the bus cycle in which the break condition is satisfied is as follows (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 B channel:

   When the data value is included in the break conditions, either longword, word, or byte is specified as the operand size of the break bus cycle registers (BBRA and BBRB). When data values are included in break conditions, a break is generated when the address conditions and data conditions both match. To specify byte data for this case, set the same data in two bytes at bits 15–8 and bits 7–0 of the break data register B (BDRB) and break data mask register B (BDMRB). When word or byte is set, bits 31–16 of BDRB and BDMRB are ignored.
4. When the DMAC data access is included in the break condition:

   When the address is included in the break condition on DMAC data access, the operand size of the break bus cycle registers (BBRA and BBRB) should be byte, word or no specified operand size. When the data value is included, select either byte or word.

7.3.4 Sequential Break

1. By specifying SEQ in BRCR is set to 1, the sequential break is issued when channel B break condition matches after channel A break condition matches. A user break is ignored even if

channel B break condition matches before channel A break condition matches. When channels A and B condition match at the same time, the sequential break is not issued.

2. In sequential break specification, logical or internal bus can be selected and the execution times break condition can be also specified. For example, when the execution times break condition is specified, the break condition is satisfied at channel B condition match with BETR = H'0001 after channel A condition match.

7.3.5 Value of Saved Program Counter

The PC when a break occurs is saved to the SPC in user breaks. The PC value saved is as follows depending on the type of break.

1. When instruction fetch (before instruction execution) is specified as a break condition:

   The value of the program counter (PC) saved is the address of the instruction that matches the break condition. The fetched instruction is not executed, and a break occurs before it.
2. When instruction fetch (after instruction execution) is specified as a break condition:

   The PC value saved is the address of the instruction to be executed following the instruction in which the break condition matches. The fetched instruction is executed, and a break occurs before the execution of the next instruction.
3. When data access (address only) is specified as a break condition:

   The PC value is the address of the instruction to be executed following the instruction that matched the break condition. The instruction that matched the condition is executed and the break occurs before the next instruction is executed.
4. When data access (address + data) is specified as a break condition:

   The PC value is the start address of the instruction that follows the instruction already executed when break processing started up. When a data value is added to the break conditions, the place where the break will occur cannot be specified exactly. The break will occur before the execution of an instruction fetched around the data access where the break occurred.

7.3.6 PC Trace

1. Setting PCTE in BRCR to 1 enables PC traces. When branch (branch instruction, repeat, and interrupt) is generated, the address from which the branch source address can be calculated and the branch destination address are stored in BRSR and BRDR, respectively. The branch address and the pointer, which corresponds to the branch, are included in BRSR.
2. The branch address before branch occurs can be calculated from the address and the pointer stored in BRSR. The expression from BSA (the address in BRSR), PID (the pointer in BRSR), and IA (the instruction address before branch occurs) is as follows: IA = BSA – 2 * PID.

Notes are needed when an interrupt (a branch) is issued before the branch destination instruction is executed. In case of the next figure, the instruction “Exec” executed immediately before branch is calculated by IA = BSA – 2 * PID. However, when branch “branch” has delay slot and the destination address is 4n + 2 address, the address “Dest” which is specified by branch instruction is stored in BRSR (Dest = BSA). Therefore, as IA = BSA – 2 * PID is not applied to this case, this PID is invalid. The case where BSA is 4n + 2 boundary is applied only to this case and then some cases are classified as follows:

```
Exec:branch  Dest
Dest:instr         (not executed)
     interrupt
Int: interrupt routine
```

If the PID value is odd, instruction buffer indicates PID+2 buffer. However, these expressions in this table are accounted for it. Therefore, the true branch source address is calculated with BSA and PID values stored in BRSR.

3. The branch address before branch occurrence, IA, has different values due to some kinds of branch.

   a. Branch instruction

      The branch instruction address

   b. Interrupt

      The last instruction executed before interrupt

      The top address of interrupt routine is stored in BRDR.

4. BRSR and BRDR have eight pairs of queue structures. The top of queues is read first when the address stored in the PC trace register is read. BRSR and BRDR share the read pointer. Read BRSR and BRDR in order, the queue only shifts after BRDR is read. When reading BRDR, longword access should be used. Also, the PC trace has a trace pointer, which initially points to the bottom of the queues. The first pair of branch addresses will be stored at the bottom of the queues, then push up when next pairs come into the queues. The trace pointer will points to the next branch address to be executed, unless it got push out of the queues. When the branch address has been executed, the trace pointer will shift down to next pair of addresses, until it reaches the bottom of the queues. After switching the PCTE bit (in BRCR) off and on, the values in the queues are invalid. The read pointer stay at the position before PCTE is switched, but the trace pointer restart at the bottom of the queues.

7.3.7 Usage Examples

Break Condition Specified to a CPU Instruction Fetch Cycle

1. Register specifications

   BARA = H'00000404, BAMRA = H'00000000, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300400

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00000404, Address mask: H'00000000

     Bus cycle: CPU/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

     No ASID check is included

   - Channel B

     Address: H'00008010, Address mask: H'00000006

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

     No ASID check is included

   A user break occurs after an instruction of address H'00000404 is executed or before instructions of adresses H'00008010 to H'00008016 are executed.

2. Register specifications

   BARA = H'00037226, BAMRA = H'00000000, BBRA = H'0056, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B sequence mode

   - Channel A

     Address: H'00037226, Address mask: H'00000000, ASID = H'80

     Bus cycle: CPU/instruction fetch (before instruction execution)/read/word

   - Channel B

     Address: H'0003722E, Address mask: H'00000000, ASID = H'70

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/read/word

   An instruction with ASID = H'80 and address H'00037226 is executed, and a user break occurs before an instruction with ASID = H'70 and address H'0003722E is executed.

3. Register specifications

   BARA = H'00027128, BAMRA = H'00000000, BBRA = H'005A, BARB = H'00031415, BAMRB = H'00000000, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300000

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00027128, Address mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/write/word

     No ASID check is included

   - Channel B

     Address: H'00031415, Address mask: H'00000000

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

     No ASID check is included

   On channel A, no user break occurs since instruction fetch is not a write cycle. On channel B, no user break occurs since instruction fetch is performed for an even address.

4. Register specifications

   BARA = H'00037226, BAMRA = H'00000000, BBRA = H'005A, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B sequence mode

   - Channel A

     Address: H'00037226, Address mask: H'00000000, ASID: H'80

     Bus cycle: CPU/instruction fetch (before instruction execution)/write/word

   - Channel B

     Address: H'0003722E, Address mask: H'00000000, ASID: H'70

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/read/word

   Since instruction fetch is not a write cycle on channel A, a sequence condition does not match. Therefore, no user break occurs.

5. Register specifications

   BARA = H'00000500, BAMRA = H'00000000, BBRA = H'0057, BARB = H'00001000, BAMRB = H'00000000, BBRB = H'0057, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300001, BETR = H'0005

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00000500, Address mask: H'00000000

Bus cycle: CPU/instruction fetch (before instruction execution)/read/longword

- Channel B

  Address: H'00001000, Address mask: H'00000000

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: CPU/instruction fetch (before instruction execution)/read/longword

  The number of execution-times break enable (5 times)

On channel A, a user break occurs before an instruction of address H'00000500 is executed. On channel B, a user break occurs before the fifth instruction execution after instructions of address H'00001000 are executed four times.

6. Register specifications

   BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, 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: CPU/instruction fetch (after instruction execution)/read (operand size 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: CPU/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

   A user break occurs after an instruction with ASID = H'80 and address H'00008000 to H'00008FFE is executed or before instructions with ASID = H'70 and addresses H'00008010 to H'00008016 are executed.

Break Condition Specified to a CPU Data Access Cycle

1. Register specifications

   BARA = H'00123456, BAMRA = H'00000000, BBRA = H'0064, BARB = H'000ABCDE, BAMRB = H'000000FF, BBRB = H'006A, BDRB = H'0000A512, BDMRB = H'00000000, BRCR = H'00000080, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00123456, Address mask: H'00000000

     Bus cycle: CPU/data access/read (operand size is not included in the condition)

   - Channel B

     Address: H'000ABCDE, Address mask: H'000000FF, ASID: H'70

Data: H'0000A512, Data mask: H'00000000

Bus cycle: CPU/data access/write/word

On channel A, a user break occurs with ASID = H'80 during longword read to address H'00123454, word read to address H'00123456, or byte read to address H'00123456. On channel B, a user break occurs with ASID = H'70 when word H'A512 is written in addresses H'000ABC00 to H'000ABCFE.

Break Condition Specified to a DMAC Data Access Cycle

1. Register specifications:

   BARA = H'00314156, BAMRA = H'00000000, BBRA = H'0094, BARB = H'00055555, BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00000078, BDMRB = H'0000000F, 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: DMAC/instruction fetch/read (operand size is not included in the condition)

   - Channel B

     Address: H'00055555, Address mask: H'00000000, ASID: H'70

     Data: H'00000078, Data mask: H'0000000F

     Bus cycle: DMAC/data access/write/byte

   On channel A, no user break occurs since instruction fetch is not performed in DMAC cycles. On channel B, a user break occurs with ASID = H'70 when the DMAC writes byte H'7* in address H'00055555.

7.4 Notes for Usage

1. Only CPU can read/write UBC registers.
2. UBC cannot monitor CPU and DMAC access in the same channel.
3. Notes in specification of sequential break are described below:
   a. A condition match occurs when B-channel match occurs in a bus cycle after an A-channel match occurs in another bus cycle in sequential break setting. Therefore, no condition match occurs even if a bus cycle, in which an A-channel match and a channel B match occur simultaneously, is set.
   b. Since the CPU has a pipeline configuration, the pipeline determines the order of an instruction fetch cycle and a memory cycle. Therefore, when a channel condition matches in the order of bus cycles, a sequential condition is satisfied.
   c. When the bus cycle condition for channel A is specified as a break before execution (PCBA = 0 in BRCR) and an instruction fetch cycle (in BBRA), the attention is as follows. A break is issued and condition match flags in BRCR are set to 1, when the bus cycle conditions both for channels A and B match simultaneously.

4. The change of a UBC register value is executed in MA (memory access) stage. Therefore, even if the break condition matches in the instruction fetch address following the instruction in which the pre-execution break is specified as the break condition, no break occurs. In order to know the timing UBC register is changed, read the last written register. Instructions after then are valid for the newly written register value.
5. The branch instruction should not be executed as soon as PC trace register BRSR and BRDR are read.
6. When PC breaks and TLB exceptions or errors occur in the same instruction. The priority is as follows:
    a. Break and instruction fetch exceptions: Instruction fetch exception occurs first.
    b. Break before execution and operand exception: Break before execution occurs first.
    c. Break after execution and operand exception: Operand exception occurs first.

Section 8 Bus State Controller (BSC)

The bus state controller (BSC) divides physical address space and output control signals for various types of memory and bus interface specifications. BSC functions enable this LSI to link directly with DRAM, synchronous DRAM, SRAM, ROM, and other memory storage devices without an external circuit. The BSC also allows direct connection to PCMCIA interfaces, simplifying system design and allowing high-speed data transfers in a compact system.

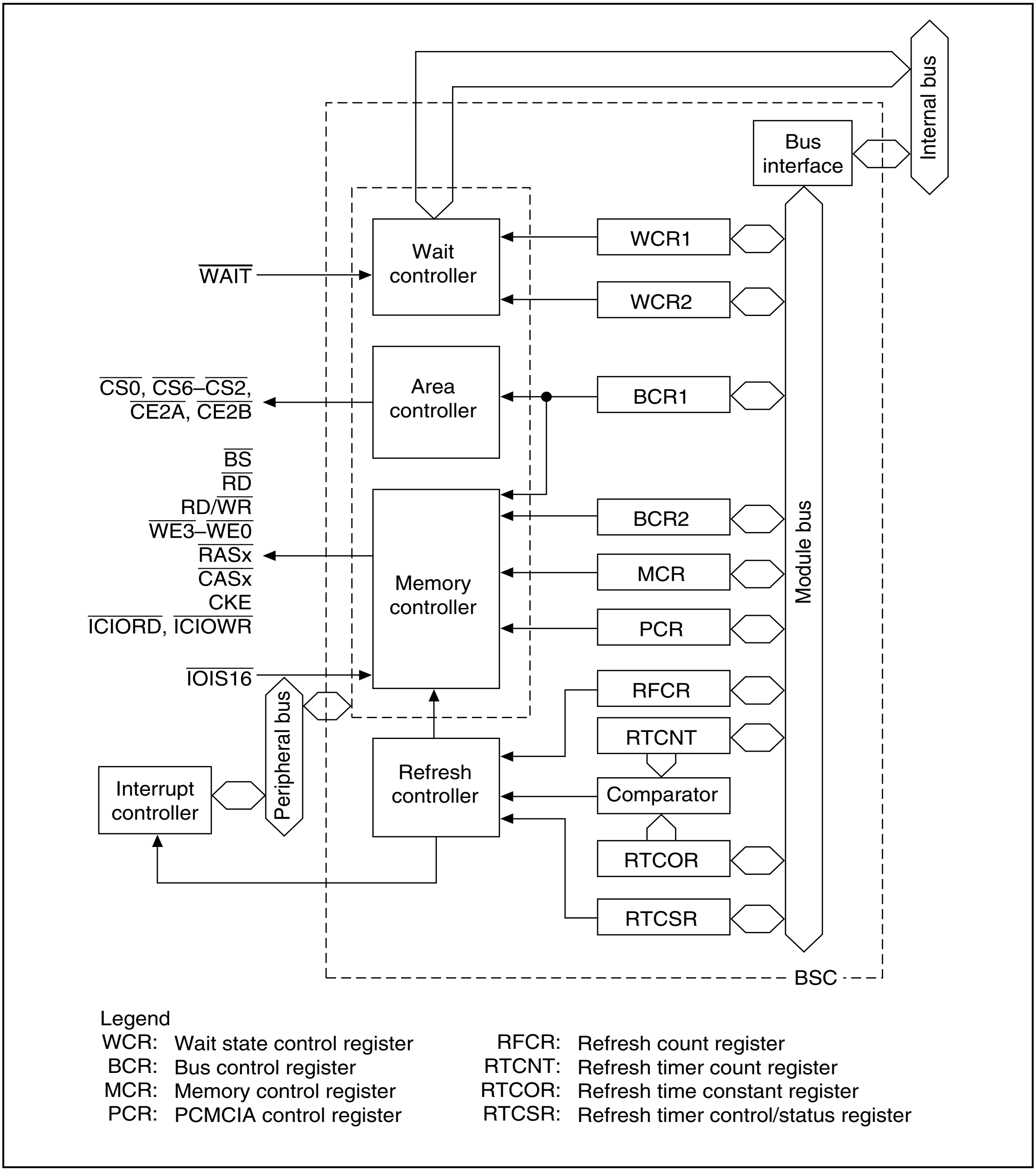
Figure 8.1 shows the block diagram of the BSC.

8.1 Features

The BSC has the following features:

- Physical address space is divided into six areas
  - A maximum 64 Mbytes for each of the six areas, 0, 2–6
  - Area bus width can be selected by register (area 0 is set by external pin)
  - Wait states can be inserted using the $\overline{\text{WAIT}}$ pin
  - Wait state insertion can be controlled through software. Register settings can be used to specify the insertion of 1–10 cycles independently for each area (1–38 cycles for areas 5 and 6 and the PCMCIAT interface only)
  - The type of memory connected can be specified for each area, and control signals are output for direct memory connection
  - Wait cycles are automatically inserted to avoid data bus conflict for continuous memory accesses to different areas or writes directly following reads of the same area
- Direct interface to synchronous DRAM
  - Multiplexes row/column addresses according to synchronous DRAM capacity
  - Supports burst operation
  - Supports bank active mode
  - Has both auto-refresh and self-refresh functions
  - Controls timing of synchronous DRAM direct-connection control signals according to register setting
- Burst ROM interface
  - Insertion of wait states controllable through software
  - Register setting control of burst transfers
- PCMCIA direct-connection interface
  - Insertion of wait states controllable through software
  - Bus sizing function for I/O bus width (only in the little endian mode)

- Refresh function
  - Refresh cycles will be automatically maintained in the sleep mode even after the external bus frequency is reduced to 1/4 of its normal operating frequency
- The refresh counter can be used as an interval timer
  - Outputs an interrupt request signal using the compare-matching function
  - Outputs an interrupt request signal when the refresh counter overflows



WCR:	Wait state control register	RFCR:	Refresh count register
BCR:	Bus control register	RTCNT:	Refresh timer count register
MCR:	Memory control register	RTCOR:	Refresh time constant register
PCR:	PCMCIA control register	RTCSR:	Refresh timer control/status register

Figure 8.1 BSC Functional Block Diagram

8.2 I/O Pins

Table 8.1 lists the BSC pin configuration.

Table 8.1 Pin Configuration

Pin Name	Signal	I/O	Description
Address bus	A25–A0	O	Address output
Data bus	D15–D0	I/O	Data I/O
	D31–D16	I/O	When 32-bit bus width, data I/O
Bus cycle start	$\overline{\text{BS}}$	O	Shows start of bus cycle. During burst transfers, asserts every data cycle.
Chip select 0, 2–4	$\overline{\text{CS0}}$, $\overline{\text{CS2}}$–$\overline{\text{CS4}}$	O	Chip select signal to indicate area being accessed.
Chip select 5, 6	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$, $\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	O	Chip select signal to indicate area being accessed. $\overline{\text{CS5}}$/$\overline{\text{CE1A}}$ and $\overline{\text{CS6}}$/$\overline{\text{CE1B}}$ can also be used as $\overline{\text{CE1A}}$ and $\overline{\text{CE1B}}$ of PCMCIA.
PCMCIA card select	$\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$	O	When PCMCIA is used, CE2A and CE2B
Read/write	RD/$\overline{\text{WR}}$	O	Data bus direction indicator signal. Synchronous DRAM write indicator signal.
Row address strobe L	$\overline{\text{RASL}}$	O	When synchronous DRAM is used, RASL for lower 32-Mbyte address.
Row address strobe U	$\overline{\text{RASU}}$	O	When synchronous DRAM is used, RASU for upper 32-Mbyte address.
Column address strobe	$\overline{\text{CASL}}$	O	When synchronous DRAM is used, CASL signal for lower 32-Mbyte address.
Column address strobe	$\overline{\text{CASU}}$	O	When synchronous DRAM is used, CASU signal for upper 32-Mbyte address.
Data enable 0	$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	O	When memory other than synchronous DRAM is used, selects D7–D0 write strobe signal. When synchronous DRAM is used, selects D7–D0.
Data enable 1	$\overline{\text{WE1}}$/$\overline{\text{DQMLU}}$/ $\overline{\text{WE}}$	O	When memory other than synchronous DRAM is used, selects D15–D8 write strobe signal. When synchronous DRAM is used, selects D15–D8. When PCMCIA is used, strobe signal that indicates the write cycle.
Data enable 2	$\overline{\text{WE2}}$/$\overline{\text{DQMUL}}$/ $\overline{\text{ICIORD}}$	O	When memory other than synchronous DRAM is used, selects D23–D16 write strobe signal. When synchronous DRAM is used, selects D23–D16. When PCMCIA is used, strobe signal indicating I/O read.

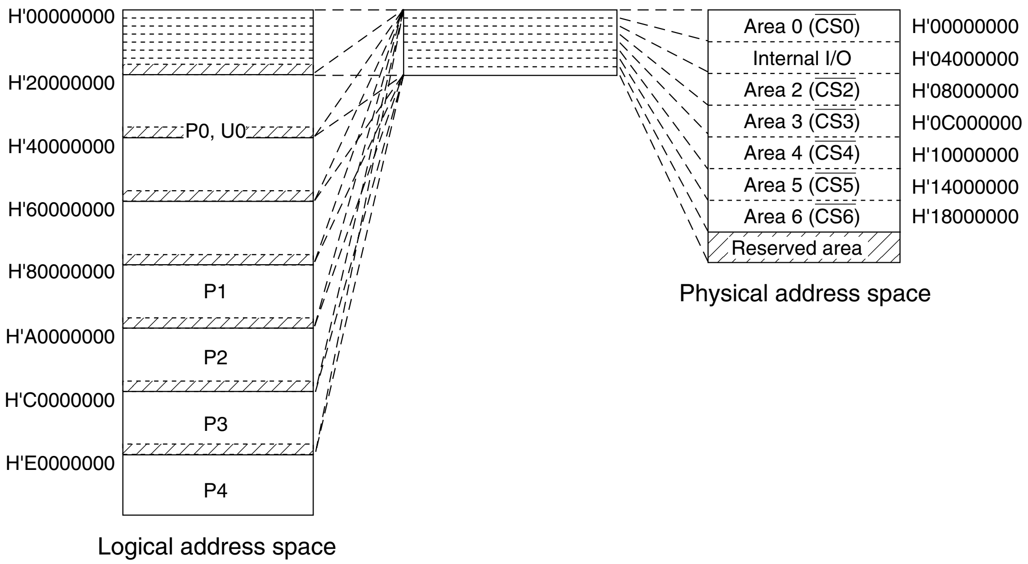
Pin Name	Signal	I/O	Description
Data enable 3	$\overline{\mathrm{WE3}}$/$\overline{\mathrm{DQMUU}}$/ $\overline{\mathrm{ICIOWR}}$	O	When memory other than synchronous DRAM is used, selects D31–D24 write strobe signal. When synchronous DRAM is used, selects D31–D24. When PCMCIA is used, strobe signal indicating I/O write.
Read	$\overline{\mathrm{RD}}$	O	Strobe signal indicating read cycle
Wait	$\overline{\mathrm{WAIT}}$	I	Wait state request signal
Clock enable	CKE	O	Clock enable control signal of synchronous DRAM
IOIS16	$\overline{\mathrm{IOIS16}}$	I	Signal indicating PCMCIA 16-bit I/O. Valid only in little-endian mode.
Bus release request	$\overline{\mathrm{BREQ}}$	I	Bus release request signal
Bus release acknowledgment	$\overline{\mathrm{BACK}}$	O	Bus release acknowledge signal

8.3 Area Overview

Space Allocation: In the architecture of this LSI, both logical spaces and physical spaces have 32-bit address spaces. The logical space is divided into five areas by the value of the upper bits of the address. The physical space is divided into eight areas.

Logical space can be allocated at physical spaces using a memory management unit (MMU). For details, refer to section 3, Memory Management Unit, which describes area allocation for physical spaces.

As listed in table 8.2, this LSI can be connected directly to six areas of memory/PCMCIA interface, and it outputs chip select signals ($\overline{\mathrm{CS0}}$, $\overline{\mathrm{CS2}}$–$\overline{\mathrm{CS6}}$, $\overline{\mathrm{CE2A}}$, $\overline{\mathrm{CE2B}}$) for each of them. $\overline{\mathrm{CS0}}$ is asserted during area 0 access; $\overline{\mathrm{CS6}}$ is asserted during area 6 access. When PCMCIA interface is selected in area 5 or 6, in addition to $\overline{\mathrm{CS5}}$/$\overline{\mathrm{CS6}}$, $\overline{\mathrm{CE2A}}$/$\overline{\mathrm{CE2B}}$ are asserted for the corresponding bytes accessed.



Note: For logical address spaces P0 and P3, when the memory management unit (MMU) is on, it can optionally generate a physical address for the logical address. It can be applied when the MMU is off and when the MMU is on and each physical address for the logical address is equal except for upper three bits.

Figure 8.2 Corresponding to Logical Address Space and Physical Address Space

Table 8.2 Physical Address Space Map

Area	Connectable Memory	Physical Address	Capacity	Access Size
0	Ordinary memory*1, burst ROM	H'00000000 to H'03FFFFFF	64 Mbytes	8, 16, 32*2
		H'00000000 + H'20000000 × n to H'03FFFFFF + H'20000000 × n	Shadow	n: 1–6
1	Internal I/O registers*8	H'04000000 to H'07FFFFFF	64 Mbytes	8, 16, 32*3
		H'04000000 + H'20000000 × n to H'07FFFFFF + H'20000000 × n	Shadow	n: 1–6
2	Ordinary memory*1, synchronous DRAM	H'08000000 to H'0BFFFFFF	64 Mbytes	8, 16, 32*3, *4
		H'08000000 + H'20000000 × n to H'0BFFFFFF + H'20000000 × n	Shadow	n: 1–6
3	Ordinary memory*1, synchronous DRAM	H'0C000000 to H'0FFFFFFF	64 Mbytes	8, 16, 32*3, *5
		H'0C000000 + H'20000000 × n to H'0FFFFFFF + H'20000000 × n	Shadow	n: 1–6
4	Ordinary memory*1	H'10000000 to H'13FFFFFF	64 Mbytes	8, 16, 32*3
		H'10000000 + H'20000000 × n to H'13FFFFFF + H'20000000 × n	Shadow	n: 1–6
5	Ordinary memory*1, PCMCIA, burst ROM	H'14000000 to H'15FFFFFF	32 Mbytes	8, 16, 32 *3, *6
		H'16000000 to H'17FFFFFF	32 Mbytes	
		H'14000000 + H'20000000 × n to H'17FFFFFF + H'20000000 × n	Shadow	n: 1–6
6	Ordinary memory*1, PCMCIA, burst ROM	H'18000000 to H'19FFFFFF	32 Mbytes	8, 16, 32 *3, *6
		H'1A000000 to H'1BFFFFFF		
		H'18000000 + H'20000000 × n to H'1BFFFFFF + H'20000000 × n	Shadow	n: 1–6
7*7	Reserved area	H'1C000000 + H'20000000 × n to H'1FFFFFFF + H'20000000 × n		n: 0–7

Notes: 1. Memory with interface such as SRAM or ROM.

2. Use external pin to specify memory bus width.

3. Use register to specify memory bus width.

4. With synchronous DRAM interfaces, bus width must be 16 or 32 bits.

5. With synchronous DRAM interfaces, bus width must be 16 or 32 bits.

6. With PCMCIA interface, bus width must be 8 or 16 bits.

7. Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.

8. When the control register in area 1 is not used for address translation by the MMU, set the top three bits of the logical address to 101 to allocate in the P2 space.

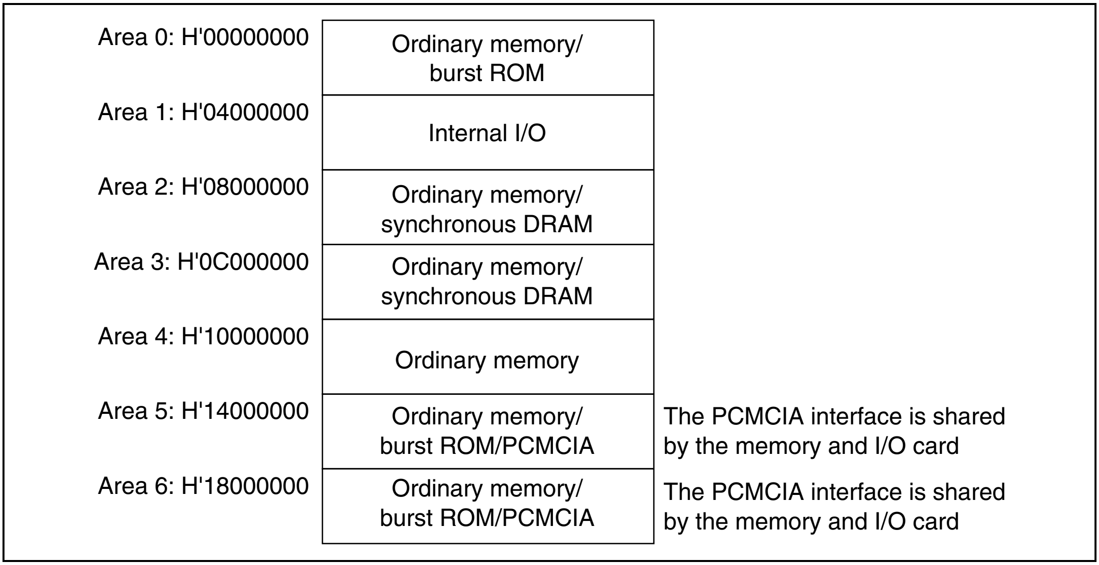

Area 0: H'00000000
Ordinary memory/ burst ROM
Area 1: H'04000000
Internal I/O
Area 2: H'08000000
Ordinary memory/ synchronous DRAM
Area 3: H'0C000000
Ordinary memory/ synchronous DRAM
Area 4: H'10000000
Ordinary memory
Area 5: H'14000000
Ordinary memory/ burst ROM/PCMCIA
The PCMCIA interface is shared by the memory and I/O card
Area 6: H'18000000
Ordinary memory/ burst ROM/PCMCIA
The PCMCIA interface is shared by the memory and I/O card

Figure 8.3 Physical Space Allocation

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

Table 8.3 Correspondence between External Pins (MD4 and MD3) and Memory Size

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

For areas 2–6, byte, word, and longword may be chosen for the bus width using bus control register 2 (BCR2) whenever ordinary memory, ROM, or burst ROM are used. When the synchronous DRAM interface is used, word or longword can be chosen as the bus width.

When the PCMCIA interface is used, set the bus width to byte or word. When synchronous DRAM is connected to both area 2 and area 3, set the same bus width for areas 2 and 3. When using port A or B, set a bus width of 8 or 16 bits for all areas. For more information, see section 8.4.2, Bus Control Register 2 (BCR2).

Shadow Space: Areas 0, 2–6 are decoded by physical addresses A28–A26, which correspond to areas 000 to 110. Address bits 31–29 are ignored. This means that the range of area 0 addresses, for example, is H'00000000 to H'03FFFFFF, and its corresponding shadow space is the address space obtained by adding to it H'20000000 × n (n = 1–6). The address range for area 7, which is on-chip I/O space, is H'1C000000 to H'1FFFFFFF. The address space H'1C000000 + H'20000000 × n–H'1FFFFFFF + H'20000000 × n (n = 0–7) corresponding to the area 7 shadow space is reserved, so do not use it.

8.3.1 PCMCIA Support

This LSI supports PCMCIA standard interface specifications in physical space areas 5 and 6 (except for WP).

The interfaces supported are basically the "IC memory card interface" and "I/O card interface" stipulated in JEIDA Specifications Ver. 4.2 (PCMCIA2.1).

Table 8.4 PCMCIA Interface Characteristics

Item	Feature
Access	Random access
Data bus	8/16 bits
Memory type	Mask ROM, OTPROM, EPROM, EEPROM, flash memory, SRAM
Memory capacity	Maximum 32 Mbytes
I/O space capacity	Maximum 32 Mbytes
Others	Dynamic bus sizing of I/O bus width* The PCMCIA interface can be accessed from the address translation area or non-address translation area.

Note: * Dynamic bus sizing of I/O bus width is supported only in the little endian mode.

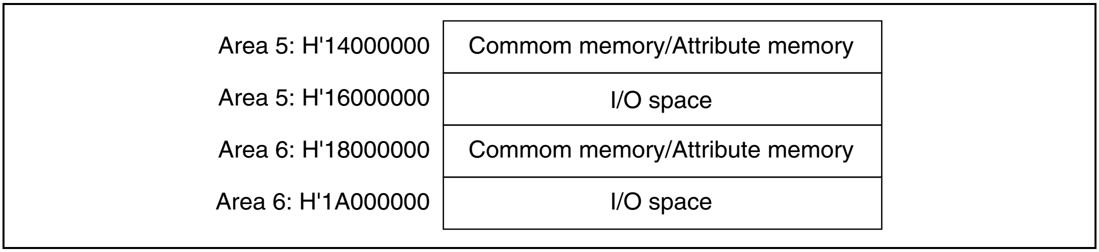


Figure 8.4 PCMCIA Space Allocation

Table 8.5 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			
Pin	**Signal**	**I/O**	**Function**	**Signal**	**I/O**	**Function**	**SH7706 Pin**
1	GND	—	Ground	GND	—	Ground	—
2	D3	I/O	Data	D3	I/O	Data	D3
3	D4	I/O	Data	D4	I/O	Data	D4
4	D5	I/O	Data	D5	I/O	Data	D5
5	D6	I/O	Data	D6	I/O	Data	D6
6	D7	I/O	Data	D7	I/O	Data	D7
7	$\overline{\text{CE1}}$	I	Card enable	$\overline{\text{CE1}}$	I	Card enable	$\overline{\text{CE1A}}$ or $\overline{\text{CE1B}}$
8	A10	I	Address	A10	I	Address	A10
9	OE	I	Output enable	$\overline{\text{OE}}$	I	Output enable	$\overline{\text{RD}}$
10	A11	I	Address	A11	I	Address	A11
11	A9	I	Address	A9	I	Address	A9
12	A8	I	Address	A8	I	Address	A8
13	A13	I	Address	A13	I	Address	A13
14	A14	I	Address	A14	I	Address	A14
15	$\overline{\text{WE}}$/$\overline{\text{PGM}}$	I	Write enable	$\overline{\text{WE}}$/$\overline{\text{PGM}}$	I	Write enable	$\overline{\text{WE}}$
16	RDY/$\overline{\text{BSY}}$	O	Ready/Busy	$\overline{\text{IREQ}}$	O	Ready/Busy	—
17	V_{CC}		Operation power	V_{CC}		Operation power	—
18	V_{PP1}		Program power	V_{PP1}		Program/ peripheral power	—
19	A16	I	Address	A16	I	Address	A16
20	A15	I	Address	A15	I	Address	A15
21	A12	I	Address	A12	I	Address	A12
22	A7	I	Address	A7	I	Address	A7
23	A6	I	Address	A6	I	Address	A6
24	A5	I	Address	A5	I	Address	A5
25	A4	I	Address	A4	I	Address	A4
26	A3	I	Address	A3	I	Address	A3
27	A2	I	Address	A2	I	Address	A2
28	A1	I	Address	A1	I	Address	A1
29	A0	I	Address	A0	I	Address	A0
30	D0	I/O	Data	D0	I/O	Data	D0

	IC Memory Card Interface			I/O Card Interface			
Pin	Signal	I/O	Function	Signal	I/O	Function	SH7706 Pin
31	D1	I/O	Data	D1	I/O	Data	D1
32	D2	I/O	Data	D2	I/O	Data	D2
33	WP*	O	Write protect	$\overline{\text{IOIS16}}$	O	16-bit I/O port	$\overline{\text{IOIS16}}$
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	$\overline{\text{CD1}}$	O	Card detection	$\overline{\text{CD1}}$	O	Card detection	—
37	D11	I/O	Data	D11	I/O	Data	D11
38	D12	I/O	Data	D12	I/O	Data	D12
39	D13	I/O	Data	D13	I/O	Data	D13
40	D14	I/O	Data	D14	I/O	Data	D14
41	D15	I/O	Data	D15	I/O	Data	D15
42	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2A}}$ or $\overline{\text{CE2B}}$
43	$\overline{\text{VS1}}$	I	Voltage sense 1	$\overline{\text{VS1}}$	I	Voltage sense 1	—
44	RFU		Reserved	$\overline{\text{IORD}}$	I	I/O read	$\overline{\text{ICIORD}}$
45	RFU		Reserved	$\overline{\text{IOWR}}$	I	I/O write	$\overline{\text{ICIOWR}}$
46	A17	I	Address	A17	I	Address	A17
47	A18	I	Address	A18	I	Address	A18
48	A19	I	Address	A19	I	Address	A19
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	V_{CC}		Power supply	V_{CC}		Power supply	—
52	V_{PP2}		Program power	V_{PP2}		Program/ peripheral power	—
53	A22	I	Address	A22	I	Address	A22
54	A23	I	Address	A23	I	Address	A23
55	A24	I	Address	A24	I	Address	A24
56	A25	I	Address	A25	I	Address	A25
57	$\overline{\text{VS2}}$	I	Voltage sense 2	$\overline{\text{VS2}}$	I	Voltage sense 2	—
58	RESET	I	Reset	RESET	I	Reset	—
59	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{WAIT}}$	O	Wait request	—
60	RFU		Reserved	$\overline{\text{INPACK}}$	O	Input acknowledge	—

Note: * This LSI does not support WP.

	IC Memory Card Interface			I/O Card Interface			
Pin	Signal	I/O	Function	Signal	I/O	Function	SH7706 Pin
61	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{REG}}$	I	Attribute memory space select	—
62	BVD2	O	Battery voltage detection	$\overline{\text{SPKR}}$	O	Digital voice signal	—
63	BVD1	O	Battery voltage detection	$\overline{\text{STSCHG}}$	O	Card state change	—
64	D8	I/O	Data	D8	I/O	Data	D8
65	D9	I/O	Data	D9	I/O	Data	D9
66	D10	I/O	Data	D10	I/O	Data	D10
67	$\overline{\text{CD2}}$	O	Card detection	$\overline{\text{CD2}}$	O	Card detection	—
68	GND		Ground	GND		Ground	—

8.4 Description of Registers

The BSC has 11 registers. The synchronous DRAM also has a built-in synchronous DRAM mode register. These registers control direct connection interfaces to memory, wait states and refreshes.

Refer to section 23, Control Registers Table, for more detail of the address and access size.

- Bus control register 1 (BCR1)
- Bus control register 2 (BCR2)
- Wait state control register 1 (WCR1)
- Wait state control register 2 (WCR2)
- Individual memory control register (MCR)
- PCMCIA control register (PCR)
- Synchronous DRAM mode register (SDMR)
- Refresh timer control/status register (RTCSR)
- Refresh timer counter (RTCNT)
- Refresh time constant register (RTCOR)
- Refresh count register (RFCR)

8.4.1 Bus Control Register 1 (BCR1)

Bus control register 1 (BCR1) is a 16-bit read/write register that sets the functions and bus cycle state for each area. It is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or by standby mode. Do not access external memory outside area 0 until BCR1 register initialization is complete.

Bit	Bit Name	Initial Value	R/W	Description
15	PULA	0	R/W	Pin A25 to A0 Pull-Up Specifies whether or not pins A25 to A0 are pulled up for 4 cycles immediately after $\overline{\text{BACK}}$ is asserted. 0: Not pulled up 1: Pulled up
14	PULD	0	R/W	Pin D31 to D0 Pull-Up Specifies whether or not pins D31 to D0 are pulled up when not in use. 0: Not pulled up 1: Pulled up
13	HIZMEM	0	R/W	Hi-Z memory control Specifies the state of A25-0, $\overline{\text{BS}}$, $\overline{\text{CS}}$, RD/$\overline{\text{WR}}$, $\overline{\text{WE}}$/DQM, $\overline{\text{RD}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$ and DRAK0/1 in standby mode. 0: High-impedance state in standby mode. 1: Driven in standby mode.
12	HIZCNT	0	R/W	High-Z Control Specifies the state of the $\overline{\text{RAS}}$ and the $\overline{\text{CAS}}$ signals at standby and bus right release. 0: High-impedance state at standby and bus right release. 1: Driven at standby and bus right release.

Bit	Bit Name	Initial Value	R/W	Description
11	ENDIAN	0/1*	R	Endian Flag Samples the value of the external pin designating endian upon a power-on reset. Endian for all physical spaces is decided by this bit, which is read-only. 0: (On reset) Endian setting external pin (MD5) is low. Indicates the SH7706 is set as big endian. 1: (On reset) Endian setting external pin (MD5) is high. Indicates the SH7706 is set as little endian.
10	A0BST1	0	R/W	Area 0 Burst ROM Control
9	A0BST0	0	R/W	Specify whether to use burst ROM in physical space area 0. When burst ROM is used, set the number of burst transfers. 00: Access area 0 as ordinary memory 01: Access area 0 as burst ROM (4 consecutive accesses). Can be used when bus width is 8, 16, or 32. 10: Access area 0 as burst ROM (8 consecutive accesses). Can be used when bus width is 8 or 16. 01: Access area 0 as burst ROM (16 consecutive accesses). Can be used only when bus width is 8.
8	A5BST1	0	R/W	Area 5 Burst Enable
7	A5BST0	0	R/W	Specify whether to use burst ROM and PCMCIA burst mode in physical space area 5. When burst ROM and PCMCIA burst mode are used, set the number of burst transfers. 00: Access area 5 as ordinary memory 01: Burst access of area 5 (4 consecutive accesses). Can be used when bus width is 8, 16, or 32. 10: Burst access of area 5 (8 consecutive accesses). Can be used when bus width is 8 or 16. 11: Burst access of area 5 (16 consecutive accesses). Can be used only when bus width is 8.

Bit	Bit Name	Initial Value	R/W	Description
6	A6BST1	0	R/W	Area 6 Burst Enable
5	A6BST0	0	R/W	Specify whether to use burst ROM and PCMCIA burst mode in physical space area 6. When burst ROM and PCMCIA burst mode are used, set the number of burst transfers. 00: Access area 6 as ordinary memory 01: Burst access of area 6 (4 consecutive accesses). Can be used when bus width is 8, 16, or 32. 10: Burst access of area 6 (8 consecutive accesses). Can be used when bus width is 8 or 16. 11: Burst access of area 6 (16 consecutive accesses). Can be used only when bus width is 8.
4	DRAMTP2	0	R/W	Area 2, Area 3 Memory Type
3	DRAMTP1	0	R/W	Designate the types of memory connected to physical space areas 2 and 3. Ordinary memory, such as ROM, SRAM, or flash ROM, can be directly connected. Synchronous DRAM can also be directly connected. 000: Areas 2 and 3 are ordinary memory 001: Reserved (Setting prohibited) 010: Area 2: ordinary memory; area 3: synchronous DRAM*2 011: Areas 2 and 3 are synchronous DRAM*1,*2 100: Reserved (Setting prohibited) 101: Reserved (Setting prohibited) 110: Reserved (Setting prohibited) 111: Reserved (Setting prohibited) Notes: 1. When selecting this mode, set the same bus width for area 2 and area 3. 2. Do not access to the SRAM when the clock ratio is Iø:Bø = 1:1.
2	DRAMTP0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
1	A5PCM	0	R/W	Area 5 Bus Type Designates whether to access physical space area 5 as PCMCIA space. 0: Access physical space area 5 as ordinary memory 1: Access physical space area 5 as PCMCIA space
0	A6PCM	0	R/W	Area 6 Bus Type Designates whether to access physical space area 6 as PCMCIA space. 0: Access physical space area 6 as ordinary memory 1: Access physical space area 6 as PCMCIA space

Notes: 1. Samples the value of the external pin (MD5) designating endian at power-on reset.

2. When selecting this mode, set the same bus width for area 2 and 3.

3. Do not access to the SRAM when the clock ratio is I ϕ : B ϕ = 1:1.

8.4.2 Bus Control Register 2 (BCR2)

The bus control register 2 (BCR2) is a 16-bit read/write register that selects the bus-size width and 8-bit port of each area. It is initialized to H'3FF0 by a power-on reset, but is not initialized by a manual reset or by standby mode. Do not access external memory outside area 0 until BCR2 register initialization is complete.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved
14	—	0	R	These bits are always read as 0. The write value should always be 0.
13	A6SZ1	1	R/W	Area 6 Bus Size Specification
12	A6SZ0	1	R/W	Specify the bus sizes of physical space area 6. • When port A/B is unused. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Longword (32-bit) size • When port A/B is used. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Reserved (Setting prohibited)
11	A5SZ1	1	R/W	Area 5 Bus Size Specification
10	A5SZ0	1	R/W	Specify the bus sizes of physical space area 5. • When port A/B is unused. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Longword (32-bit) size • When port A/B is used. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Reserved (Setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
9	A4SZ1	1	R/W	Area 4 Bus Size Specification
8	A4SZ0	1	R/W	Specify the bus sizes of physical space area 4. • When port A/B is unused. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Longword (32-bit) size • When port A/B is used. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Reserved (Setting prohibited)
7	A3SZ1	1	R/W	Area 3 Bus Size Specification
6	A3SZ0	1	R/W	Specify the bus sizes of physical space area 3. • When port A/B is unused. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Longword (32-bit) size • When port A/B is used. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Reserved (Setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
5	A2SZ1	1	R/W	Area 2 Bus Size Specification
4	A2SZ0	1	R/W	Specify the bus sizes of physical space area 2. • When port A/B is unused. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Longword (32-bit) size • When port A/B is used. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Reserved (Setting prohibited)
3 to 0	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.

8.4.3 Wait State Control Register 1 (WCR1)

Wait state control register 1 (WCR1) is a 16-bit read/write register that specifies the number of idle (wait) state cycles inserted for each area. For some memories, the drive of the data bus may not be turned off quickly even when the read signal from the external device is turned off. This can result in conflicts between data buses when consecutive memory accesses are to different memories or when a write immediately follows a memory read. This LSI automatically inserts idle states equal to the number set in WCR1 in those cases.

WCR1 is initialized to H'3FF3 by a power-on reset. It is not initialized by a manual reset or by standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	WAITSEL	0	R/W	WAIT Sampling Timing Select Specifies the WAIT signal sampling timing. 0: Set 1 to use the WAIT signal. 1: The WAIT signal is sampled at the falling edge of CKIO.
14	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	A6IW1	1	R/W	Area 6 Intercycle Idle Specification
12	A6IW0	1	R/W	Specify the number of idles inserted between bus cycles when switching between physical space area 6 to another space or between a read access to a write access in the same physical space. 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted
11	A5IW1	1	R/W	Area 5 Intercycle Idle Specification
10	A5IW0	1	R/W	Specify the number of idles inserted between bus cycles when switching between physical space area 5 to another space or between a read access to a write access in the same physical space. 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted
9	A4IW1	1	R/W	Area 4 Intercycle Idle Specification
8	A4IW0	1	R/W	Specify the number of idles inserted between bus cycles when switching between physical space area 4 to another space or between a read access to a write access in the same physical space. 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
7	A3IW1	1	R/W	Area 3 Intercycle Idle Specification
6	A3IW0	1	R/W	Specify the number of idles inserted between bus cycles when switching between physical space area 3 to another space or between a read access to a write access in the same physical space. 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted
5	A2IW1	1	R/W	Area 2 Intercycle Idle Specification
4	A2IW0	1	R/W	Specify the number of idles inserted between bus cycles when switching between physical space area 2 to another space or between a read access to a write access in the same physical space. 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted
3	—	0	R	Reserved
2	—	0	R	These bits are always read as 0. The write value should always be 0.
1	A0IW1	1	R/W	Area 0 Intercycle Idle Specification
0	A0IW0	1	R/W	Specify the number of idles inserted between bus cycles when switching between physical space area 0 to another space or between a read access to a write access in the same physical space. 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted

8.4.4 Wait State Control Register 2 (WCR2)

Wait state control register 2 (WCR2) is a 16-bit read/write register that specifies the number of wait state cycles inserted for each area. It also specifies the pitch of data access for burst memory accesses. This allows direct connection of even low-speed memories without an external circuit.

Bit	Bit Name	Initial Value	R/W	Description
15	A6W2	1	R/W	Area 6 Wait Control Specify the number of wait states inserted into physical space area 6. Also specify the burst pitch for burst transfer. Refer to table 8.6 for details.
14	A6W1	1	R/W	
13	A6W0	1	R/W	
12	A5W2	1	R/W	Area 5 Wait Control Specify the number of wait states inserted into physical space area 5. Also specify the burst pitch for burst transfer. Refer to table 8.7 for details.
11	A5W1	1	R/W	
10	A5W0	1	R/W	
9	A4W2	1	R/W	Area 4 Wait Control Specify the number of wait states inserted into physical space area 4. Refer to table 8.8 for details.
8	A4W1	1	R/W	
7	A4W0	1	R/W	
6	A3W1	1	R/W	Area 3 Wait Control Specify the number of wait states inserted into physical space area 3. • For Ordinary memory Inserted Wait States $\overline{\text{WAIT}}$ Pin 00: 0 Ignored 01: 1 Enable 10: 2 Enable 11: 3 Enable • For Synchronus DRAM Synchronus DRAM :CAS Latency 00: 1 01: 1 10: 2 11: 3
5	A3W0	1	R/W	

Bit	Bit Name	Initial Value	R/W	Description
4	A2W1	1	R/W	Area 2 Wait Control
3	A2W0	1	R/W	Specify the number of wait states inserted into physical space area 2. • For Ordinary memory Inserted Wait States — $\overline{\text{WAIT}}$ Pin 00: 0 Ignored 01: 1 Enabled 10: 2 Enabled 11: 3 Enabled • For Synchronus DRAM Synchronus DRAM :CAS Latency 00: 1 01: 1 10: 2 11: 3
2	A0W2	1	R/W	Area 0 Wait Control
1	A0W1	1	R/W	Specify the number of wait states inserted into physical space area 0. Also specify the burst pitch for burst transfer. Refer to table 8.9 for details.
0	A0W0	1	R/W	

Table 8.6 Area 6 Wait Control

			Description			
WCR2's bits			**First Cycle**		**Burst Cycle (Excluding First Cycle)**	
Bit 15: A6W2	**Bit 14: A6W1**	**Bit 13: A6W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**	**Number of States Per Data Transfer**	**$\overline{\text{WAIT}}$ Pin**
			0	Ignored	2	Enable
		1	1	Enable	2	Enable
	1	0	2	Enable	3	Enable
		1	3	Enable	4	Enable
1	0	0	4	Enable	4	Enable
		1	6	Enable	6	Enable
	1	0	8	Enable	8	Enable
		1	10	Enable	10	Enable

Table 8.7 Area 5 Wait Control

			Description			
WCR2's bits			First Cycle		Burst Cycle (Excluding First Cycle)	
Bit 12: A5W2	Bit 11: A5W1	Bit 10: A5W0	Inserted Wait States	$\overline{\text{WAIT}}$ Pin	Number of States Per Data Transfer	$\overline{\text{WAIT}}$ Pin
			0	Ignored	2	Enable
		1	1	Enable	2	Enable
	1	0	2	Enable	3	Enable
		1	3	Enable	4	Enable
1	0	0	4	Enable	4	Enable
		1	6	Enable	6	Enable
	1	0	8	Enable	8	Enable
		1	10	Enable	10	Enable

Table 8.8 Area 4 Wait Control

WCR2's bits			Description	
Bit 9: A4W2	Bit 8: A4W1	Bit 7: A4W0	Inserted Wait State	$\overline{\text{WAIT}}$ Pin
0	0	0	0	Ignored
		1	1	Enable
	1	0	2	Enable
		1	3	Enable
1	0	0	4	Enable
		1	6	Enable
	1	0	8	Enable
		1	10	Enable

Table 8.9 Area 0 Wait Control

			Description			
WCR2's bits			First Cycle		Burst Cycle (Excluding First Cycle)	
Bit 2: A0W2	Bit 1: A0W1	Bit 0: A0W0	Inserted Wait States	$\overline{\text{WAIT}}$ Pin	Number of States Per Data Transfer	$\overline{\text{WAIT}}$ Pin
			0	Ignored	2	Enable
		1	1	Enable	2	Enable
	1	0	2	Enable	3	Enable
		1	3	Enable	4	Enable
1	0	0	4	Enable	4	Enable
		1	6	Enable	6	Enable
	1	0	8	Enable	8	Enable
		1	10	Enable	10	Enable

8.4.5 Individual Memory Control Register (MCR)

The individual memory control register (MCR) is a 16-bit read/write register that specifies $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ timing and burst control for synchronous DRAM (areas 2 and 3), specifies address multiplexing, and controls refresh. This enables direct connection of synchronous DRAM without external circuits.

The MCR is initialized to H'0000 by power-on resets, but is not initialized by manual resets or standby mode. The bits TPC1–TPC0, RCD1–RCD0, TRWL1–TRWL0, TRAS1–TRAS0, RASD and AMX3–AMX0 are written to at the initialization after a power-on reset and are not then modified again. When RFSH and RMODE are written to, write the same values to the other bits. When using synchronous DRAM, do not access areas 2 and 3 until this register is initialized.

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>15</td><td>TPC1</td><td>0</td><td>R/W</td><td>RAS Precharge Time</td></tr>
<tr><td>14</td><td>TPC0</td><td>0</td><td>R/W</td><td>When synchronous DRAM interface is selected as connected memory, they set the minimum number of cycles until output of the next bank-active command after precharge.
The number of cycles to be inserted immediately after issuing a precharge all banks (PALL) command in auto-refresh or a precharge (PRE) command in bank-active mode is one cycle less than the normal value. In bank-active mode, neither TPC1 nor TPC0 should be cleared to 0.
<table>
<tr><th>Normal Operation</th><th>Immediately after* Precharge Command</th><th>Immediately after Self-Refresh</th></tr>
<tr><td>00: 1 cycle</td><td>0 cycle</td><td>2 cycles</td></tr>
<tr><td>01: 2 cycles</td><td>1 cycle</td><td>5 cycles</td></tr>
<tr><td>10: 3 cycles</td><td>2 cycles</td><td>8 cycles</td></tr>
<tr><td>11: 4 cycles</td><td>3 cycles</td><td>11 cycles</td></tr>
</table>
Note: * Immediately after a precharge all banks (PALL) command in auto-refresh and a precharge (PRE) command in bank-active mode.</td></tr>
<tr><td>13</td><td>RCD1</td><td>0</td><td>R/W</td><td>RAS–CAS Delay</td></tr>
<tr><td>12</td><td>RCD0</td><td>0</td><td>R/W</td><td>When synchronous DRAM interface is selected as connected memory, sets the bank active read/write command delay time.
00: 1 cycle
01: 2 cycles
10: 3 cycles
11: 4 cycles</td></tr>
</table>

Bit	Bit Name	Initial Value	R/W	Description
11	TRWL1	0	R/W	Write-Precharge Delay
10	TRWL0	0	R/W	The TRWL bits set the synchronous DRAM write-precharge delay time. This designates the time between the end of a write cycle and the next bank-active command. This is valid only when synchronous DRAM is connected. After the write cycle, the next bank-active command is not issued for the period TPC + TRWL. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: Reserved (Setting prohibited)
9	TRAS1	0	R/W	$\overline{\text{CAS}}$-Before-$\overline{\text{RAS}}$ Refresh $\overline{\text{RAS}}$ Assert Time
8	TRAS0	0	R/W	When synchronous DRAM interface is selected as connected memory, no bank-active command is issues during the period TPC + TRAS after an auto-refresh command. 00: 2 cycles 01: 3 cycles 10: 4 cycles 11: 5 cycles
7	RASD	0	R/W	Synchronous DRAM Bank Active Specifies whether synchronous DRAM is used in bank active mode or auto-precharge mode. 0: Auto-precharge mode 1: Bank active mode

Bit	Bit Name	Initial Value	R/W	Description
6	AMX3	0	R/W	Address Multiplex The AMX bits specify address multiplexing for synchronous DRAM. The actual address shift value differs between DRAM interface and synchronous DRAM interface. For Synchronous DRAM interface: 0000: Reserved (Setting prohibited) 0001: Reserved (Setting prohibited) 0010: Reserved (Setting prohibited) 0011: Reserved (Setting prohibited) 0100: The row address begins with A9. (The A9 value is output at A1 when the row address is output. 64 M (1 M × 16 bits × 4 banks)) 0101: The row address begins with A10. (The A10 value is output at A1 when the row address is output. 128 M (2 M × 16 bits × 4 banks), 64 M (1 M × 16 bits × 4 banks)) 0110: Cannot be set. 0111: The row address begins with A9. (The A9 value is output at A1 when the row address is output. 64 M (512 k × 32 bits × 4 banks)*2) 1000: Reserved (Setting prohibited) 1001: Reserved (Setting prohibited) 1010: Reserved (Setting prohibited) 1011: Reserved (Setting prohibited) 1100: Reserved (Setting prohibited) 1101: The row address begins with A10. (The A10 value is output at A1 when the row address is output. 256 M (4 M × 16 bits × 4 banks)) 1110: The row address begins with A11. (The A11 value is output at A1 when the row address is output. 512 M (8 M × 16 bits × 4 banks)*1) 1111: Reserved (Setting prohibited) Notes: 1. Cannot be set when using a 32-bit bus width. 2. Cannot be set when using a 16-bit bus width.
5	AMX2	0	R/W	
4	AMX1	0	R/W	
3	AMX0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
2	RFSH	0	R/W	Refresh Control The RFSH bit determines whether or not the refresh operation of the DRAM and synchronous DRAM is performed. The timer for generation of the refresh request frequency can also be used as an interval timer. 0: No refresh 1: Refresh
1	RMODE	0	R/W	Refresh Mode The RMODE bit selects whether to perform an ordinary refresh or a self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 0, a CAS-before-RAS refresh or an auto-refresh is performed on synchronous DRAM at the period set by the refresh-related registers RTCNT, RTCOR and RTCSR. When a refresh request occurs during an external bus cycle, the bus cycle will be ended and the refresh cycle performed. When the RFSH bit is 1 and this bit is also 1, the synchronous DRAM will wait for the end of any executing external bus cycle before going into a self-refresh. All refresh requests to memory that is in the self-refresh state are ignored. 0: CAS-before-RAS refresh (RFSH must be 1) 1: Self-refresh (RFSH must be 1)
0	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.

8.4.6 PCMCIA Control Register (PCR)

The PCMCIA control register (PCR) is a 16-bit read/write register that specifies the timing for the assertion or negation of the OE and WE signals for the PCMCIA interface connected to areas 5 and 6. The width for assertion of the OE and WE signals is set by the wait control bit in the WCR2 register.

Bit	Bit Name	Initial Value	R/W	Description
15	A6W3	0	R/W	Area 6 Wait Control (): The A6W3 bit specifies the number of inserted wait states for area 6 combined with bits A6W2–A6W0 in WCR2. It also specifies the number of transfer states in burst transfer. Set this bit to 0 when area 6 is not set to PCMCIA. Refer to table 8.10 for details.
14	A5W3	0	R/W	Area 5 Wait Control The A5W3 bit specifies the number of inserted wait states for area 5 combined with bits A5W2–A5W0 in WCR2. It also specifies the number of transfer states in burst transfer. Set this bit to 0 when area 5 is not set to PCMCIA. The relationship between the setting value and the number of waits is the same as A6W3.
13	—	0	R/W	Reserved
12	—	0	R/W	These bits are always read as 0. The write value should always be 0.
11, 7, 6	A5TED2 to A5TED0	0	R/W	Area 5 Address $\overline{\text{OE}}$/$\overline{\text{WE}}$ Assert Delay The A5TED bits specify the address to $\overline{\text{OE}}$/$\overline{\text{WE}}$ assert delay time for the PCMCIA interface connected to area 5. 000: 0.5-cycle delay 001: 1.5-cycle delay 010: 2.5-cycle delay 011: 3.5-cycle delay 100: 4.5-cycle delay 101: 5.5-cycle delay 110: 6.5-cycle delay 111: 7.5-cycle delay

Bit	Bit Name	Initial Value	R/W	Description
10	A6TED2	0	R/W	Area 6 Address $\overline{\text{OE}}$/$\overline{\text{WE}}$ Assert Delay
5	A6TED1	0	R/W	The A6TED bits specify the address to $\overline{\text{OE}}$/$\overline{\text{WE}}$ assert delay time for the PCMCIA interface connected to area 6.
4	A6TED0	0	R/W	000: 0.5-cycle delay 001: 1.5-cycle delay 010: 2.5-cycle delay 011: 3.5-cycle delay 100: 4.5-cycle delay 101: 5.5-cycle delay 110: 6.5-cycle delay 111: 7.5-cycle delay
9	A5TEH2	0	R/W	Area 5 $\overline{\text{OE}}$/$\overline{\text{WE}}$ Negate Address Delay
3	A5TEH1	0	R/W	The A5TEH bits specify the $\overline{\text{OE}}$/$\overline{\text{WE}}$ negate address delay time for the PCMCIA interface connected to area 5.
2	A5TEH0	0	R/W	000: 0.5-cycle delay 001: 1.5-cycle delay 010: 2.5-cycle delay 011: 3.5-cycle delay 100: 4.5-cycle delay 101: 5.5-cycle delay 110: 6.5-cycle delay 111: 7.5-cycle delay
8	A6TEH2	0	R/W	Area 6 $\overline{\text{OE}}$/$\overline{\text{WE}}$ Negate Address Delay
1	A6TEH1	0	R/W	The A6TEH bits specify the $\overline{\text{OE}}$/$\overline{\text{WE}}$ negate address delay time for the PCMCIA interface connected to area 6.
0	A6TEH0	0	R/W	000: 0.5-cycle delay 001: 1.5-cycle delay 010: 2.5-cycle delay 011: 3.5-cycle delay 100: 4.5-cycle delay 101: 5.5-cycle delay 110: 6.5-cycle delay 111: 7.5-cycle delay

Note: The bit numbers are out of sequence.

Table 8.10 Area 6 Wait Control

				Description			
				Top Cycle		Burst Cycle	
PCR		WCR2					
A6W3	A6W2	A6W1	A6W0	Inserted Wait State	$\overline{\text{WAIT}}$ Pin	Number of States per One-data Transfer	$\overline{\text{WAIT}}$ Pin
0	0	0	0	0	Ignored	2	Enabled
0	0	0	1	1	Enabled	2	Enabled
0	0	1	0	2	Enabled	3	Enabled
0	0	1	1	3	Enabled	4	Enabled
0	1	0	0	4	Enabled	5	Enabled
0	1	0	1	6	Enabled	7	Enabled
0	1	1	0	8	Enabled	9	Enabled
0	1	1	1	10	Enabled	11	Enabled
1	0	0	0	12	Enabled	13	Enabled
1	0	0	1	14	Enabled	15	Enabled
1	0	1	0	18	Enabled	19	Enabled
1	0	1	1	22	Enabled	23	Enabled
1	1	0	0	26	Enabled	27	Enabled
1	1	0	1	30	Enabled	31	Enabled
1	1	1	0	34	Enabled	35	Enabled
1	1	1	1	38	Enabled	39	Enabled

8.4.7 Synchronous DRAM Mode Register (SDMR)

The synchronous DRAM mode register (SDMR) is written to via the synchronous DRAM address bus and is an 8-bit write-only register. It sets synchronous DRAM mode for areas 2 and 3. SDMR must be set before synchronous DRAM is accessed.

Writes to the synchronous DRAM mode register use the address bus rather than the data bus. If the value to be set is X and the SDMR address is Y, the value X is written in the synchronous DRAM mode register by writing in address X + Y. Since, with a 32-bit bus width, A0 of the synchronous DRAM is connected to A2 of the chip and A1 of the synchronous DRAM is connected to A3 of the chip, the value actually written to the synchronous DRAM is the X value shifted two bits right. With a 16-bit bus width, the value written is the X value shifted one bit right. For example, with a 32-bit bus width, when H'0230 is written to the SDMR register of area 2, random data is written to the address H'FFFFD000 (address Y) + H'08C0 (value X), or H'FFFFD8C0. As a result, H'0230 is written to the SDMR register. The range for value X is H’0000 to H’0FFC. When H'0230 is written to the SDMR register of area 3, random data is written to the address H'FFFFE000 (address Y) + H'08C0 (value X), or H'FFFFE8C0. As a result, H'0230 is written to the SDMR register. The range for value X is H'0000 to H'0FFC.

8.4.8 Refresh Timer Control/Status Register (RTCSR)

The refresh timer control/status register (RTCSR) is a 16-bit read/write register that specifies the refresh cycle, whether to generate an interrupt, and that interrupt’s cycle. It is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or standby mode and holds its values unchanged. Make the RTCOR setting before setting bits CKS2 to CKS0 in RTCSR.

Note: Writing to the RTCSR differs from that to general registers to ensure the RTCSR is not rewritten incorrectly. Use the word-transfer instruction to set the upper byte as B'10100101 and the lower byte as the write data. For the byte-transfer instruction, writing is disabled. Read data in 16 bits. 0 is read from undefined bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CMF	0	R/W	Compare Match Flag (): The CMF status flag indicates that the values of RTCNT and RTCOR match. 0: The values of RTCNT and RTCOR do not match. Clear condition: When a refresh is performed After 0 has been written in CMF and RFSH = 1 and RMODE = 0 (to perform a CBR refresh). 1: The values of RTCNT and RTCOR match. Set condition: RTCNT = RTCOR * Note: * Contents do not change when 1 is written to CMF.
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables an interrupt request caused when the CMF of RTCSR is set to 1. Do not set this bit to 1 when using auto-refresh. 0: Disables an interrupt request caused by CMF 1: Enables an interrupt request caused by CMF
5	CKS2	0	R/W	Clock Select Bits Select the clock input to RTCNT. The source clock is the external bus clock (CKIO). The RTCNT count clock is CKIO divided by the specified ratio. RTCOR should be set before setting CKS2 to CKS0. 000: Disables clock input 001: Bus clock (CKIO)/4 010: CKIO/16 011: CKIO/64 100: CKIO/256 101: CKIO/1024 110: CKIO/2048 111: CKIO/4096
4	CKS1	0	R/W	
3	CKS0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
2	OVF	0	R/W	Refresh Count Overflow Flag The OVF status flag indicates when the number of refresh requests indicated in the refresh count register (RFCR) exceeds the limit set in the LMTS bit of RTCSR. 0: RFCR has not exceeded the count limit value set in LMTS Clear Conditions: When 0 is written to OVF 1: RFCR has exceeded the count limit value set in LMTS Set Conditions: When the RFCR value has exceeded the count limit value set in LMTS* Note: * Contents don't change when 1 is written to OVF.
1	OVIE	0	R/W	Refresh Count Overflow Interrupt Enable OVIE selects whether to suppress generation of interrupt requests by OVF when the OVF bit of RTCSR is set to 1. 0: Disables interrupt requests from the OVF 1: Enables interrupt requests from the OVF
0	LMTS	0	R/W	Refresh Count Overflow Limit Select Indicates the count limit value to be compared to the number of refreshes indicated in the refresh count register (RFCR). When the value RFCR overflows the value specified by LMTS, the OVF flag is set. 0: Count limit value is 1024 1: Count limit value is 512

8.4.9 Refresh Timer Counter (RTCNT)

RTCNT is a 16-bit read/write register. RTCNT is an 8-bit counter that counts up with input clocks. The clock select bits (CKS2–CKS0) of RTCSR select the input clock. When RTCNT matches RTCOR, the CMF bit of RTCSR is set and RTCNT is cleared. RTCNT is initialized to H'00 by a power-on reset; it continues incrementing after a manual reset; it is not initialized by standby mode and holds its values unchanged.

Note: Writing to the RTCNT differs from that to general registers to ensure the RTCNT is not rewritten incorrectly. Use the word-transfer instruction to set the upper byte as B'10100101 and the lower byte as the write data. For the byte-transfer instruction, writing is disabled. Read data in 16 bits. 0 is read from undefined bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	0	R	Reserved These bits are always read as 0.
7 to 0	—	0	R/W	8-bit. counter

8.4.10 Refresh Time Constant Register (RTCOR)

The refresh time constant register (RTCOR) is a 16-bit read/write register. The values of RTCOR and RTCNT (bottom 8 bits) are constantly compared. When the values match, the CMF of RTCSR is set and RTCNT is cleared to 0. When the refresh bit (RFSH) of the individual memory control register (MCR) is set to 1 and the refresh mode is set to auto refresh, a memory refresh cycle occurs when the CMF bit is set. RTCOR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset or standby mode, but holds its contents. Make the RTCOR setting before setting bits CKS2 to CKS0 in RTCSR.

Note: Writing to the RTCOR differs from that to general registers to ensure the RTCOR is not rewritten incorrectly. Use the word-transfer instruction to set the upper byte as B'10100101 and the lower byte as the write data. For the byte-transfer instruction, writing is disabled. Read data in 16 bits. 0 is read from undefined bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	0	R	Reserved These bits are always read as 0.
7 to 0	—	0	R/W	Upper limit of the counter (8 bits)

8.4.11 Refresh Count Register (RFCR)

The refresh count register (RFCR) is a 16-bit read/write register. It is a 10-bit counter that increments every time RTCOR and RTCNT match. When RFCR exceeds the count limit value set in the LMTS of RTCSR, RTCSR's OVF bit is set and RFCR clears. RFCR is initialized to H'0000 when a power-on reset is performed. It is not initialized by a manual reset or standby mode, but holds its contents.

Note: Writing to the RFCR differs from that to general registers to ensure the RFCR is not rewritten incorrectly. Use the word-transfer instruction to set the MSB and followed six bits of upper bytes as B'101001 and remaining bits as the write data. For the byte-transfer instruction, writing is disabled. Read data in 16 bits. 0 is read from undefined bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	0	R	Reserved These bits are always read as 0.
9 to 0	—	0	R/W	10-bit counter

8.5 Description of Operation

8.5.1 Endian/Access Size and Data Alignment

This LSI supports both big endian, in which the 0 address is the most significant byte in the byte data, and little endian, in which the 0 address is the least significant byte. This switchover is designated by an external pin (MD5 pin) at the time of a power-on reset. After a power-on reset, big endian is engaged when MD5 is low; little endian is engaged when MD5 is high.

Three data bus widths are available for ordinary memory (byte, word, longword) and two data bus widths (word and longword) for synchronous DRAM. For the PCMCIA interface, choose from byte and word. This means data alignment is done by matching the device's data width and endian. The access unit must also be matched to the device's bus width. This also means that when longword data is read from a byte-width device, the read operation must happen 4 times. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.

Tables 8.11 through 8.16 show the relationship between endian, device data width, and access unit.

Table 8.11 32-Bit External Device/Big Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	D31–D24	D23–D16	D15–D8	D7–D0	$\overline{\text{WE3}}$, $\overline{\text{DQMUU}}$	$\overline{\text{WE2}}$, $\overline{\text{DQMUL}}$	$\overline{\text{WE1}}$, $\overline{\text{DQMLU}}$	$\overline{\text{WE0}}$, $\overline{\text{DQMLL}}$
Byte access at 0	Data 7–0	—	—	—	Assert			
Byte access at 1	—	Data 7–0	—	—		Assert		
Byte access at 2	—	—	Data 7–0	—			Assert	
Byte access at 3	—	—	—	Data 7–0				Assert
Word access at 0	Data 15–8	Data 7–0	—	—	Assert	Assert		
Word access at 2	—	—	Data 15–8	Data 7–0			Assert	Assert
Longword access at 0	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Assert	Assert	Assert	Assert

Table 8.12 16-Bit External Device/Big Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		D31–D24	D23–D16	D15–D8	D7–D0	$\overline{\text{WE3}}$, $\overline{\text{DQMUU}}$	$\overline{\text{WE2}}$, $\overline{\text{DQMUL}}$	$\overline{\text{WE1}}$, $\overline{\text{DQMLU}}$	$\overline{\text{WE0}}$, $\overline{\text{DQMLL}}$
Byte access at 0		—	—	Data 7–0	—			Assert	—
Byte access at 1		—	—	—	Data 7–0				Assert
Byte access at 2		—	—	Data 7–0	—			Assert	—
Byte access at 3		—	—	—	Data 7–0				Assert
Word access at 0		—	—	Data 15–8	Data 7–0			Assert	Assert
Word access at 2		—	—	Data 15–8	Data 7–0			Assert	Assert
Longword access at 0	1st time at 0	—	—	Data 31–24	Data 23–16			Assert	Assert
	2nd time at 2	—	—	Data 15–8	Data 7–0			Assert	Assert

Table 8.13 8-Bit External Device/Big Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	$\overline{\text{WE3}}$, $\overline{\text{DQMUU}}$	$\overline{\text{WE2}}$, $\overline{\text{DQMUL}}$	$\overline{\text{WE1}}$, $\overline{\text{DQMLU}}$	$\overline{\text{WE0}}$, $\overline{\text{DQMLL}}$
Byte access at 0		—	—	—	Data 7–0				Assert
Byte access at 1		—	—	—	Data 7–0				Assert
Byte access at 2		—	—	—	Data 7–0				Assert
Byte access at 3		—	—	—	Data 7–0				Assert
Word access at 0	1st time at 0	—	—	—	Data 15–8				Assert
	2nd time at 1	—	—	—	Data 7–0				Assert
Word access at 2	1st time at 2	—	—	—	Data 15–8				Assert
	2nd time at 3	—	—	—	Data 7–0				Assert
Longword access at 0	1st time at 0	—	—	—	Data 31–24				Assert
	2nd time at 1	—	—	—	Data 23–16				Assert
	3rd time at 2	—	—	—	Data 15–8				Assert
	4th time at 3	—	—	—	Data 7–0				Assert

Table 8.14 32-Bit External Device/Little Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	D31–D24	D23–D16	D15–D8	D7–D0	$\overline{\text{WE3}}$, $\overline{\text{DQMUU}}$	$\overline{\text{WE2}}$, $\overline{\text{DQMUL}}$	$\overline{\text{WE1}}$, $\overline{\text{DQMLU}}$	$\overline{\text{WE0}}$, $\overline{\text{DQMLL}}$
Byte access at 0	—	—	—	Data 7–0				Assert
Byte access at 1	—	—	Data 7–0	—			Assert	
Byte access at 2	—	Data 7–0	—	—		Assert		
Byte access at 3	Data 7–0	—	—	—	Assert			
Word access at 0	—	—	Data 15–8	Data 7–0			Assert	Assert
Word access at 2	Data 15–8	Data 7–0	—	—	Assert	Assert		
Longword access at 0	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Assert	Assert	Assert	Assert

Table 8.15 16-Bit External Device/Little Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		D31–D24	D23–D16	D15–D8	D7–D0	$\overline{\text{WE3}}$, $\overline{\text{DQMUU}}$	$\overline{\text{WE2}}$, $\overline{\text{DQMUL}}$	$\overline{\text{WE1}}$, $\overline{\text{DQMLU}}$	$\overline{\text{WE0}}$, $\overline{\text{DQMLL}}$
Byte access at 0		—	—	—	Data 7–0				Assert
Byte access at 1		—	—	Data 7–0	—			Assert	
Byte access at 2		—	—	—	Data 7–0				Assert
Byte access at 3		—	—	Data 7–0	—			Assert	
Word access at 0		—	—	Data 15–8	Data 7–0			Assert	Assert
Word access at 2		—	—	Data 15–8	Data 7–0			Assert	Assert
Longword access at 0	1st time at 0	—	—	Data 15–8	Data 7–0			Assert	Assert
	2nd time at 2	—	—	Data 31–24	Data 23–16			Assert	Assert

Table 8.16 8-Bit External Device/Little Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	$\overline{\text{WE3}}$, $\overline{\text{DQMUU}}$	$\overline{\text{WE2}}$, $\overline{\text{DQMUL}}$	$\overline{\text{WE1}}$, $\overline{\text{DQMLU}}$	$\overline{\text{WE0}}$, $\overline{\text{DQMLL}}$
Byte access at 0		—	—	—	Data 7–0				Assert
Byte access at 1		—	—	—	Data 7–0				Assert
Byte access at 2		—	—	—	Data 7–0				Assert
Byte access at 3		—	—	—	Data 7–0				Assert
Word access at 0	1st time at 0	—	—	—	Data 7–0				Assert
	2nd time at 1	—	—	—	Data 15–8				Assert
Word access at 2	1st time at 2	—	—	—	Data 7–0				Assert
	2nd time at 3	—	—	—	Data 15–8				Assert
Longword access at 0	1st time at 0	—	—	—	Data 7–0				Assert
	2nd time at 1	—	—	—	Data 15–8				Assert
	3rd time at 2	—	—	—	Data 23–16				Assert
	4th time at 3	—	—	—	Data 31–24				Assert

8.5.2 Description of Areas

Area 0: Area 0 physical addresses A28–A26 are 000. Addresses A31–A29 are ignored and the address range is H'00000000 + H'20000000 × n – H'03FFFFFF + H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

Ordinary memories such as SRAM, ROM, and burst ROM can be connected to this space. Byte, word, or longword can be selected as the bus width using external pins MD3 and MD4. When the area 0 space is accessed, a $\overline{\text{CS0}}$ signal is asserted. An $\overline{\text{RD}}$ signal that can be used as $\overline{\text{OE}}$ and the $\overline{\text{WE0}}$–$\overline{\text{WE3}}$ signals for write control are also asserted. The number of bus cycles is selected between 0 and 10 wait cycles using the A0W2–A0W0 bits of WCR2. When the burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2–10 according to the number of waits.

Area 1: Area 1 physical addresses A28–A26 are 001. Addresses A31–A29 are ignored and the address range is H'04000000 + H'20000000 × n – H'07FFFFFF + H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

Area 1 is the area specifically for the internal peripheral modules. The external memories cannot be connected.

Control registers of peripheral modules shown below are mapped to this area 1. Their addresses are physical address, to which logical addresses can be mapped with the MMU enabled:

DMAC, PORT, SCIF, ADC, DAC, INTC (except INTEVT, IPRA, IPRB)

Those registers must be set not to be cached.

Area 2: Area 2 physical addresses A28–A26 are 010. Addresses A31–A29 are ignored and the address range is H'08000000 + H'20000000 × n – H'0BFFFFFF + H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

Ordinary memories like SRAM and ROM, as well as synchronous DRAM, can be connected to this space. Byte, word, or longword can be selected as the bus width using the A2SZ1–A2SZ0 bits of BCR2 for ordinary memory.

When the area 2 space is accessed, a $\overline{\text{CS2}}$ signal is asserted. When ordinary memories are connected, an $\overline{\text{RD}}$ signal that can be used as $\overline{\text{OE}}$ and the $\overline{\text{WE0}}$–$\overline{\text{WE3}}$ signals for write control are also asserted and the number of bus cycles is selected between 0 and 3 wait cycles using the A2W1 to A2W0 bits of WCR2.

When synchronous DRAM is connected, the $\overline{\text{RASU}}$, $\overline{\text{RASL}}$ signal, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$ signal, RD/$\overline{\text{WR}}$ signal, and byte controls $\overline{\text{DQMHH}}$, $\overline{\text{DQMHL}}$, $\overline{\text{DQMLH}}$, and $\overline{\text{DQMLL}}$ are all asserted and addresses multiplexed. Control of $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, data timing, and address multiplexing is set with MCR.

Area 3: Area 3 physical addresses A28–A26 are 011. Addresses A31–A29 are ignored and the address range is H'0C000000 + H'20000000 × n – H'0FFFFFFF + H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

Ordinary memories like SRAM and ROM, as well as synchronous DRAM, can be connected to this space. Byte, word or longword can be selected as the bus width using the A3SZ1–A3SZ0 bits of BCR2 for ordinary memory.

When area 3 space is accessed, $\overline{\text{CS3}}$ is asserted.

When ordinary memories are connected, an RD signal that can be used as $\overline{\text{OE}}$ and the $\overline{\text{WE0}}$–$\overline{\text{WE3}}$ signals for write control are asserted and the number of bus cycles is selected between 0 and 3 wait cycles using the A3W1 to A3W0 bits of WCR2.

When synchronous DRAM is connected, the $\overline{\text{RASU}}$, $\overline{\text{RASL}}$ signal, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$ signal, RD/$\overline{\text{WR}}$ signal, and byte controls $\overline{\text{DQMHH}}$, $\overline{\text{DQMHL}}$, $\overline{\text{DQMLH}}$, and $\overline{\text{DQMLL}}$ are all asserted and addresses multiplexed. Control of $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and data timing and of address multiplexing is set with MCR.

Area 4: Area 4 physical addresses A28–A26 are 100. Addresses A31–A29 are ignored and the address range is H'10000000 + H'20000000 × n – H'13FFFFFF + H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

Only ordinary memories like SRAM and ROM can be connected to this space. Byte, word, or longword can be selected as the bus width using the A4SZ1–A4SZ0 bits of BCR2. When the area 4 space is accessed, a $\overline{\text{CS4}}$ signal is asserted. An $\overline{\text{RD}}$ signal that can be used as $\overline{\text{OE}}$ and the $\overline{\text{WE0}}$–$\overline{\text{WE3}}$ signals for write control are also asserted. The number of bus cycles is selected between 0 and 10 wait cycles using the A4W2–A4W0 bits of WCR2.

Area 5: Area 5 physical addresses A28–A26 are 101. Addresses A31–A29 are ignored and the address range is the 64 Mbytes at H'14000000 + H'20000000 × n – H'17FFFFFF + H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

Ordinary memories like SRAM and ROM as well as burst ROM and PCMCIA interfaces can be connected to this space. When the PCMCIA interface is used, the IC memory card interface address range comprises the 32 Mbytes at H'14000000 + H'20000000 x n to H'15FFFFFF + H'20000000 x n (where n = 0—6, and n = 1—6 represents shadow space), and the I/O card interface address range comprises the 32 Mbytes at H'16000000 + H'20000000 x n to H'17FFFFFF + H'20000000 x n (where n = 0—6, and n = 1—6 represents shadow space).

For ordinary memory and burst ROM, byte, word, or longword can be selected as the bus width using the A5SZ1–A5SZ0 bits of BCR2. For the PCMCIA interface, byte, and word can be selected as the bus width using the A5SZ1–A5SZ0 bits of BCR2.

When the area 5 space is accessed and ordinary memory is connected, a $\overline{\text{CS5}}$ signal is asserted. An RD signal that can be used as $\overline{\text{OE}}$ and the $\overline{\text{WE0}}$–$\overline{\text{WE3}}$ signals for write control are also asserted. When the PCMCIA interface is used, the $\overline{\text{CE1A}}$ signal, $\overline{\text{CE2A}}$ signal, $\overline{\text{RD}}$ signal as $\overline{\text{OE}}$ signal, and $\overline{\text{WE1}}$, $\overline{\text{ICIORD}}$, and $\overline{\text{ICIOWR}}$ signals are asserted.

The number of bus cycles is selected between 0 and 10 wait cycles using the A5W2–A5W0 bits of WCR2. With the PCMCIA interface, from 0 to 38 wait cycles can be selected using the A5W2–A5W0 bits of WCR2 and the A5W3 bit of PCR. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$). When a burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2–11 (2–39 for the PCMCIA interface) according to the number of waits. The setup and hold times of address/$\overline{\text{CS5}}$ for the read/write strobe signals can be set in the range 0.5–7.5 using A5TED2–A5TED0 and A5TEH2 to A5TEH0 bits of the PCR register.

Area 6: Area 6 physical addresses A28–A26 are 110. Addresses A31–A29 are ignored and the address range is the 64 Mbytes at H'18000000 + H'20000000 × n – H'1BFFFFFF + H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

Ordinary memories like SRAM and ROM as well as burst ROM and PCMCIA interfaces can be connected to this space. When the PCMCIA interface is used, the IC memory card interface address range is 32 Mbytes at H'18000000 + H'20000000 × n – H'19FFFFFF + H'20000000 × n and the I/O card interface address range is 32 Mbytes at H'1A000000 + H'20000000 × n – H'1BFFFFFF + H'20000000 × n (n = 0–6 and n = 1–6 are the shadow spaces).

For ordinary memory and burst ROM, byte, word, or longword can be selected as the bus width using the A6SZ1–A6SZ0 bits of BCR2. For the PCMCIA interface, byte, and word can be selected as the bus width using the A6SZ1–A6SZ0 bits of BCR2.

When the area 6 space is accessed and ordinary memory is connected, a $\overline{\text{CS6}}$ signal is asserted. An $\overline{\text{RD}}$ signal that can be used as $\overline{\text{OE}}$ and the $\overline{\text{WE0}}$–$\overline{\text{WE3}}$ signals for write control are also asserted. When the PCMCIA interface is used, the $\overline{\text{CE1B}}$ signal, $\overline{\text{CE2B}}$ signal, $\overline{\text{RD}}$ signal as $\overline{\text{OE}}$ signal, and $\overline{\text{WE}}$, $\overline{\text{ICIORD}}$, and $\overline{\text{ICIOWR}}$ signals are asserted.

The number of bus cycles is selected between 0 and 10 wait cycles using the A6W2–A6W0 bits of WCR2. With the PCMCIA interface, from 0 to 38 wait cycles can be selected using the A6W2–A6W0 bits of WCR2 and the A6W3 bit of PCR. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$). The bus cycle pitch of the burst cycle is determined within a range of 2–11 (2–39 for the PCMCIA interface) according to the number of waits. The setup and hold times of address/$\overline{\text{CS6}}$ for the read/write strobe signals can be set in the range 0.5–7.5 using A6TED2–A6TED0 and A6TEH2–A6TEH0 bits of the PCR register.

8.5.3 Basic Interface

Basic Timing: The basic interface of this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. Figure 8.5 shows the basic timing of normal space accesses. A no-wait normal access is completed in two cycles. The $\overline{\text{BS}}$ signal is asserted for one cycle to indicate the start of a bus cycle. The $\overline{\text{CSn}}$ signal is negated on the T2 clock falling edge to secure the negation period. Therefore, in case of access at minimum pitch, there is a half-cycle negation period.

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always read in case of a 32-bit device, and 16 bits in case of a 16-bit device. When writing, only the $\overline{\text{WE}}$ signal for the byte to be written is asserted. For details, see section 8.5.1, Endian/Access Size and Data Alignment.

Read/write for cache fill or write-back follows the set bus width and transfers a total of 16 bytes continuously. The bus is not released during this transfer. For cache misses that occur during byte or word operand accesses or branching to odd word boundaries, the fill is always performed by longword accesses on the chip-external interface. Write-through-area write access and non-cacheable read/write access are based on the actual address size.

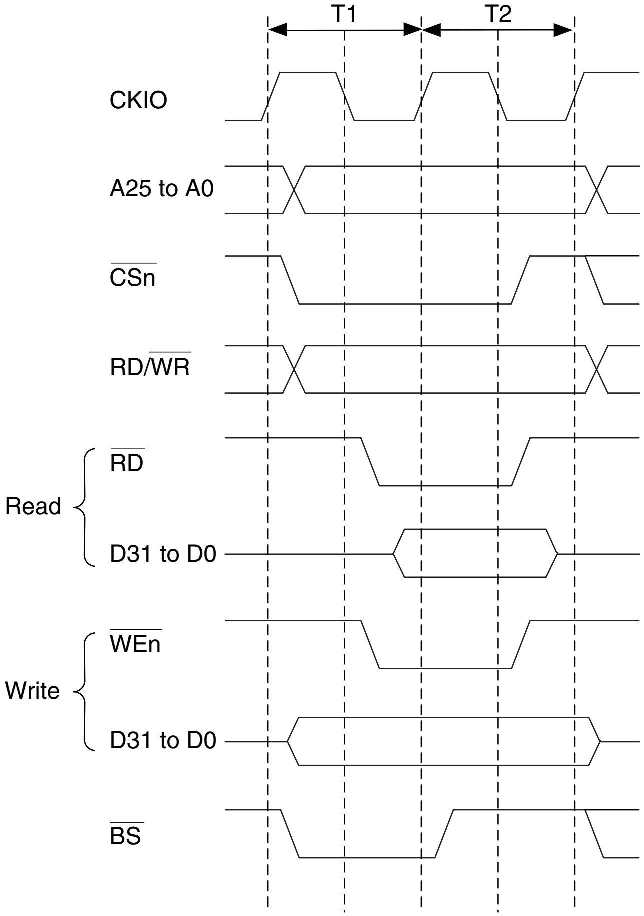


Figure 8.5 Basic Timing of Basic Interface

Figures 8.6, 8.7, and 8.8 show examples of connection to 32, 16, and 8-bit data-width static RAM, respectively.

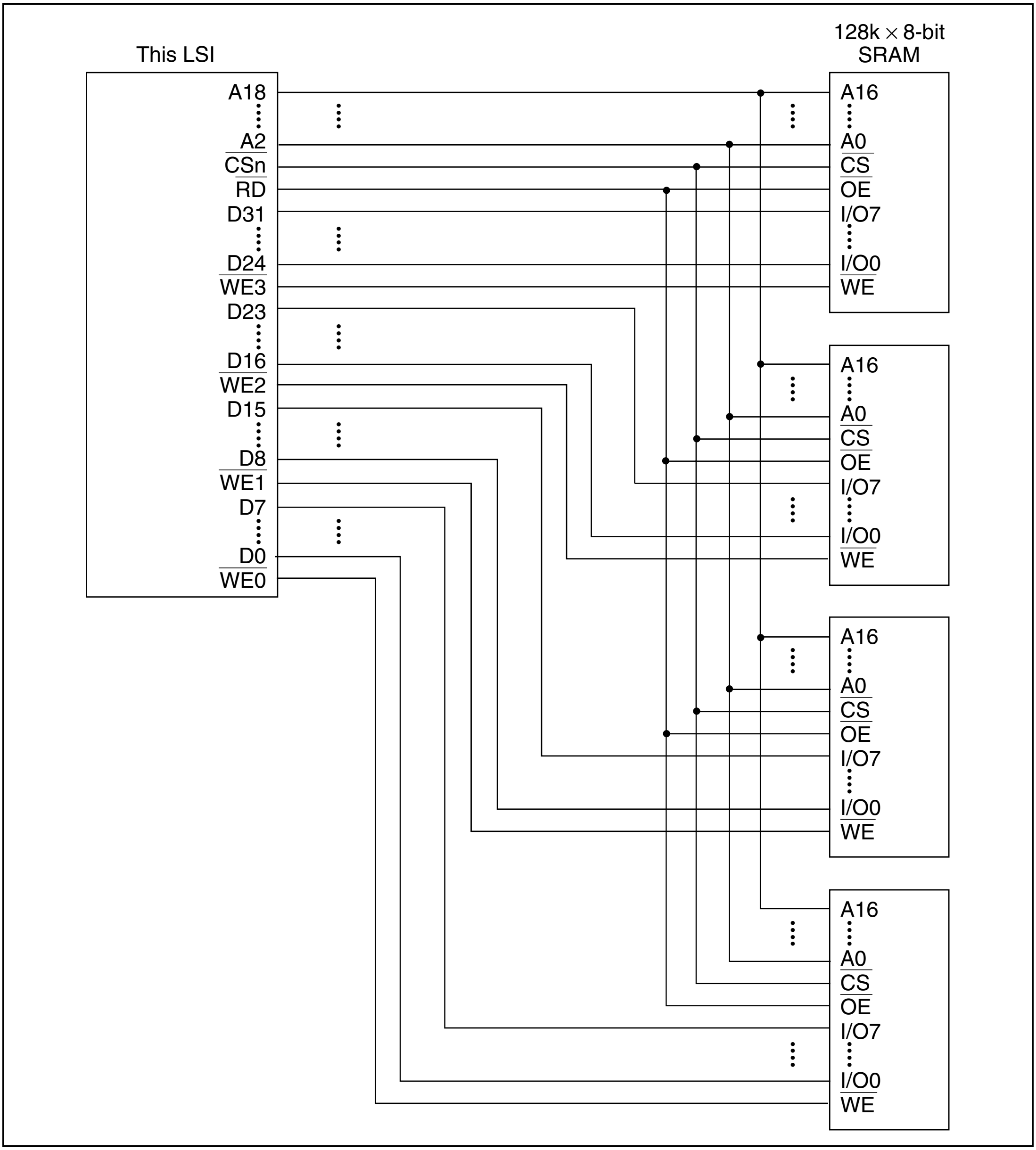


Figure 8.6 Example of 32-Bit Data-Width Static RAM Connection

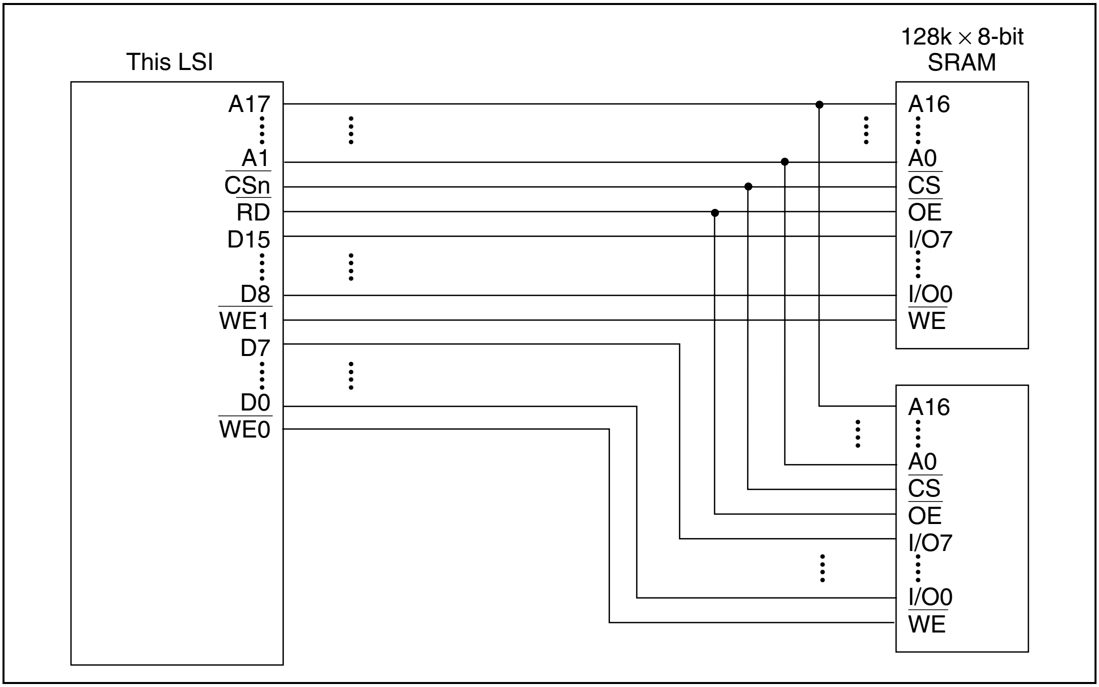


Figure 8.7 Example of 16-Bit Data-Width Static RAM Connection

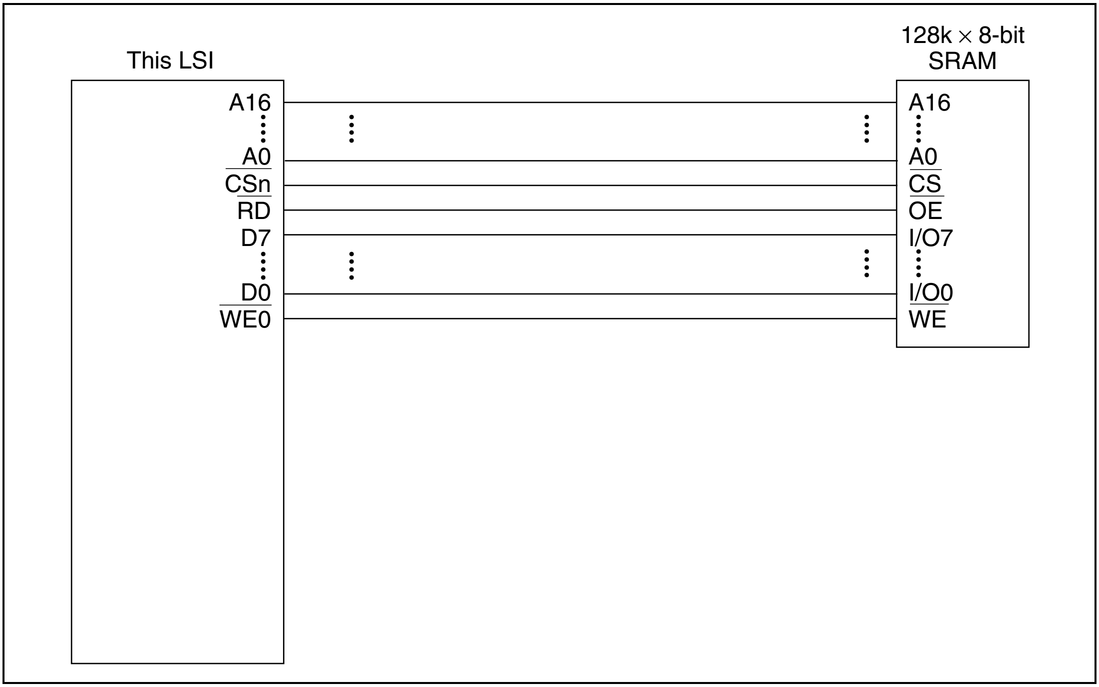


Figure 8.8 Example of 8-Bit Data-Width Static RAM Connection

Wait State Control: Wait state insertion on the basic interface can be controlled by the WCR2 settings. If the WCR2 wait specification bits corresponding to a particular area are not zero, a software wait is inserted in accordance with that specification. For details, see section 8.4.4, Wait Control Register 2 (WCR2).

The specified number of Tw cycles are inserted as wait cycles using the basic interface wait timing shown in figure 8.9.

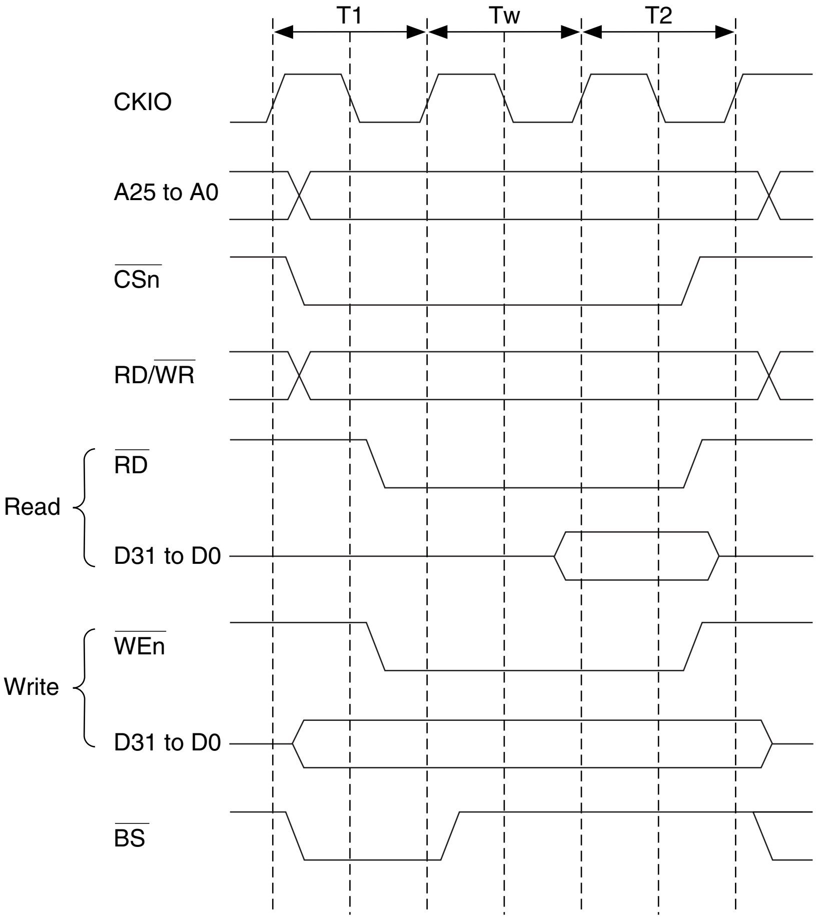


Figure 8.9 Basic Interface Wait Timing (Software Wait Only)

When software wait insertion is specified by WCR2, the external wait input $\overline{\text{WAIT}}$ signal is also sampled. $\overline{\text{WAIT}}$ pin sampling is shown in figure 8.10. A 2-cycle wait is specified as a software wait. Sampling is performed at the transition from the Tw state to the T2 state; therefore, if the $\overline{\text{WAIT}}$ signal has no effect if asserted in the T1 cycle or the first Tw cycle.

When the WAITSEL bit in the WCR1 register is set to 1, the $\overline{\text{WAIT}}$ signal is sampled at the falling edge of the clock. If the setup time and hold times with respect to the falling edge of the clock are not satisfied, the value sampled at the next falling edge is used.

However, the $\overline{\text{WAIT}}$ signal is ignored in the following cases:

- In 16-byte DMA transfer or dual addressing mode, or when writing data to the external address area
- In 16-byte DMA transfer or single addressing mode, or when transferring data from an external device with DACK to the external address area
- When accessing cache for write back

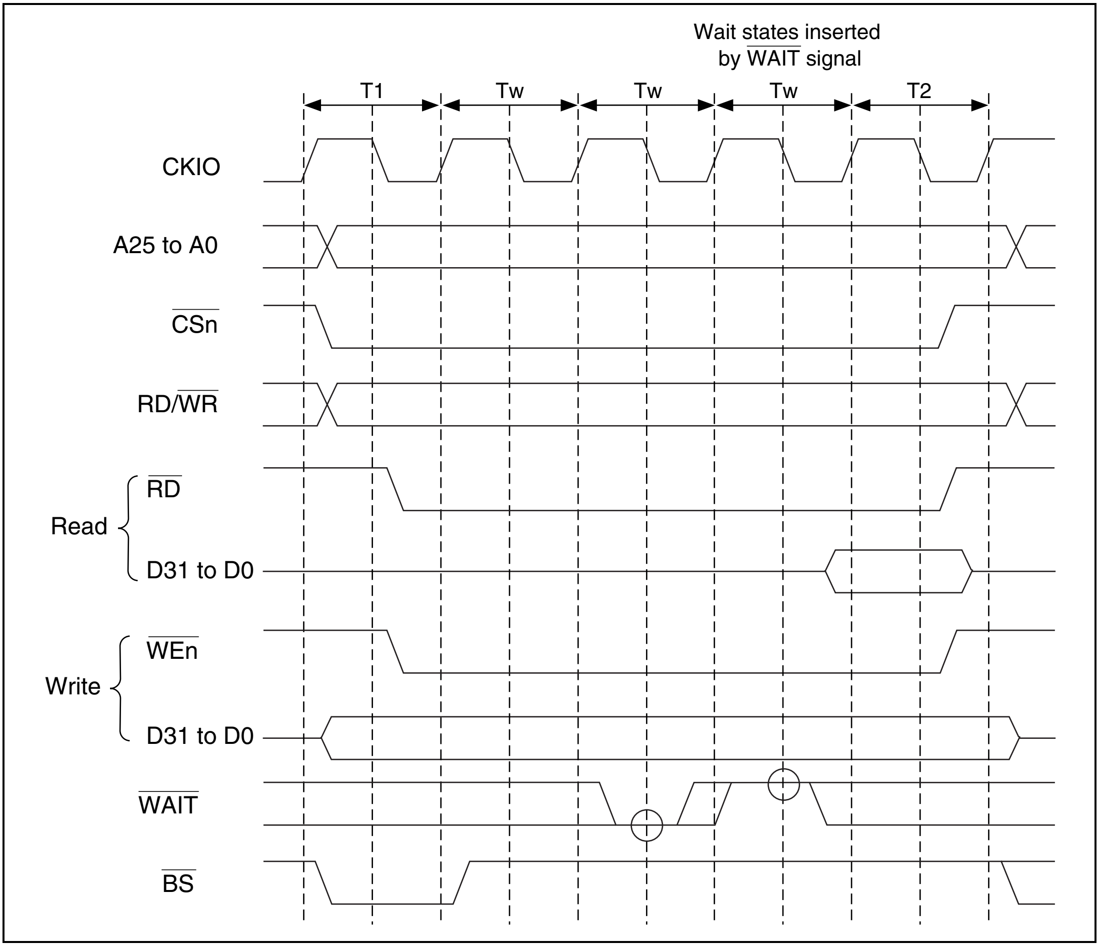


Figure 8.10 Basic Interface Wait State Timing (Wait State Insertion by $\overline{\text{WAIT}}$ Signal WAITSEL = 1)

8.5.4 Synchronous DRAM Interface

- Synchronous DRAM Direct Connection

Since synchronous DRAM can be selected by the $\overline{\text{CS}}$ signal, physical space areas 2 and 3 can be connected using $\overline{\text{RAS}}$ and other control signals in common. If the memory type bits (DRAMTP2–0) in BCR1 are set to 010, area 2 is ordinary memory space and area 3 is synchronous DRAM space; if set to 011, areas 2 and 3 are both synchronous DRAM space. However, do not access to the synchronous DRAM when clock ratio is Iø:Bø = 1:1.

With this LSI, burst length 1 burst read/single write mode is supported as the synchronous DRAM operating mode. A data bus width of 16 or 32 bits can be selected. A 16-byte burst transfer is performed in a cache fill/write-back cycle, and only one access is performed in a write-through area write or a non-cacheable area read/write.

The control signals for direct connection of synchronous DRAM are $\overline{\text{RASL}}$, $\overline{\text{RASU}}$, $\overline{\text{CASL}}$, $\overline{\text{CASU}}$, RD/$\overline{\text{WR}}$, $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$, $\overline{\text{DQMUU}}$, $\overline{\text{DQMUL}}$, $\overline{\text{DQMLU}}$, $\overline{\text{DQMLL}}$, and CKE. All the signals other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than CKE are valid and fetched to the synchronous DRAM only when $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$ is asserted. Synchronous DRAM can therefore be connected in parallel to a number of areas. CKE is negated (low) only when self-refreshing is performed, and is always asserted (high) at other times.

In the refresh cycle and mode-register write cycle, $\overline{\text{RASU}}$ and $\overline{\text{RASL}}$ or $\overline{\text{CASU}}$ and $\overline{\text{CASL}}$ are output.

Commands for synchronous DRAM are specified by $\overline{\text{RASL}}$, $\overline{\text{RASU}}$, $\overline{\text{CASL}}$, $\overline{\text{CASU}}$, RD/$\overline{\text{WR}}$, and special address signals. The commands are NOP, auto-refresh (REF), self-refresh (SELF), precharge all banks (PALL), row address strobe bank active (ACTV), read (READ), read with precharge (READA), write (WRIT), write with precharge (WRITA), and mode register write (MRS).

Byte specification is performed by $\overline{\text{DQMUU}}$, $\overline{\text{DQMUL}}$, $\overline{\text{DQMLU}}$, and $\overline{\text{DQMLL}}$. A read/write is performed for the byte for which the corresponding DQM is low. In big-endian mode, $\overline{\text{DQMUU}}$ specifies an access to address 4n, and $\overline{\text{DQMLL}}$ specifies an access to address 4n + 3. In little-endian mode, $\overline{\text{DQMUU}}$ specifies an access to address 4n + 3, and $\overline{\text{DQMLL}}$ specifies an access to address 4n.

Figures 8.11 shows examples of the connection of two 1M × 16-bit × 4-bank synchronous DRAMs and figure 8.12 shows one 1M × 16-bit × 4-bank synchronous DRAM, respectively.

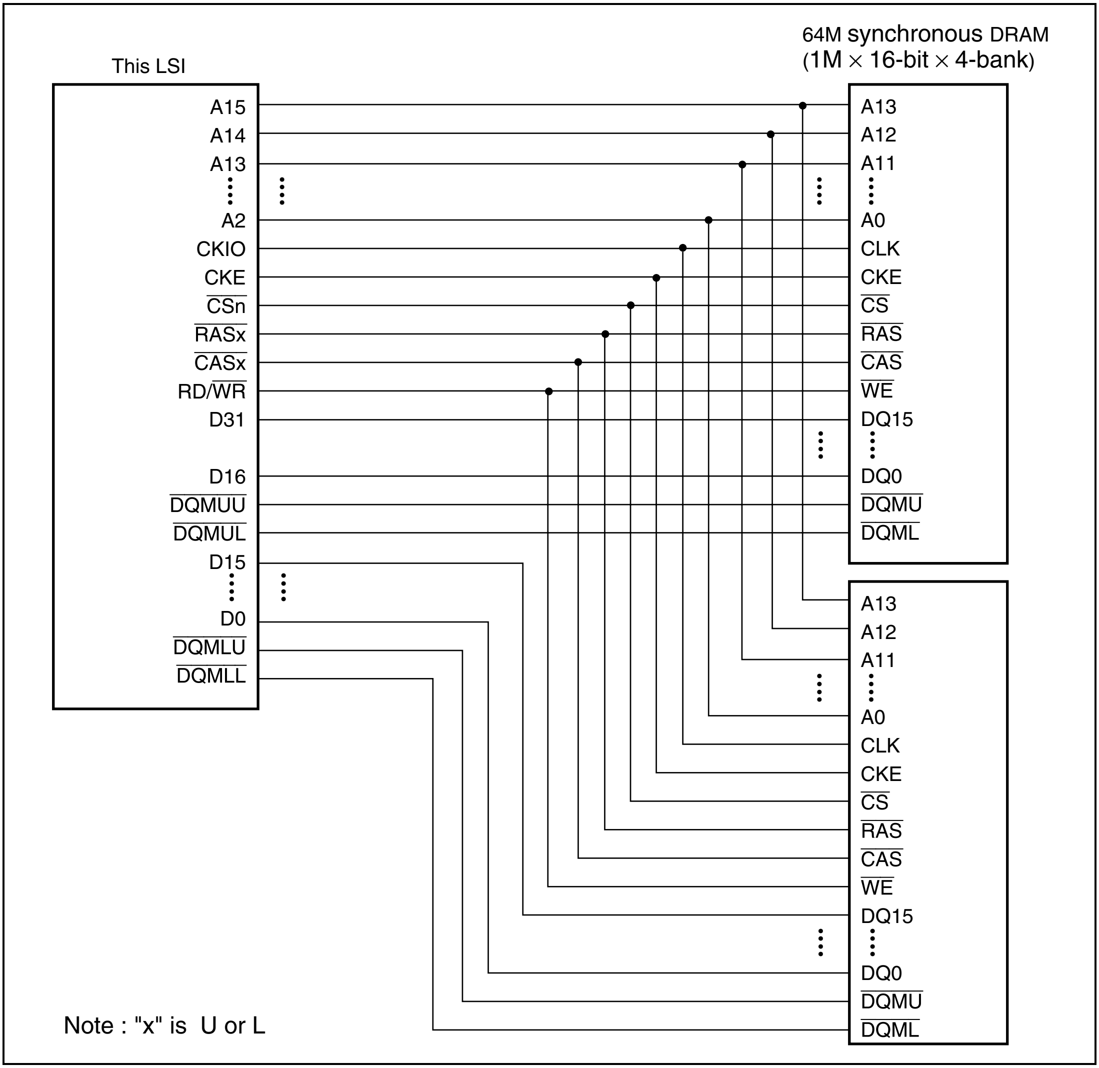


Figure 8.11 Example of 64-Mbit Synchronous DRAM Connection (32-Bit Bus Width)

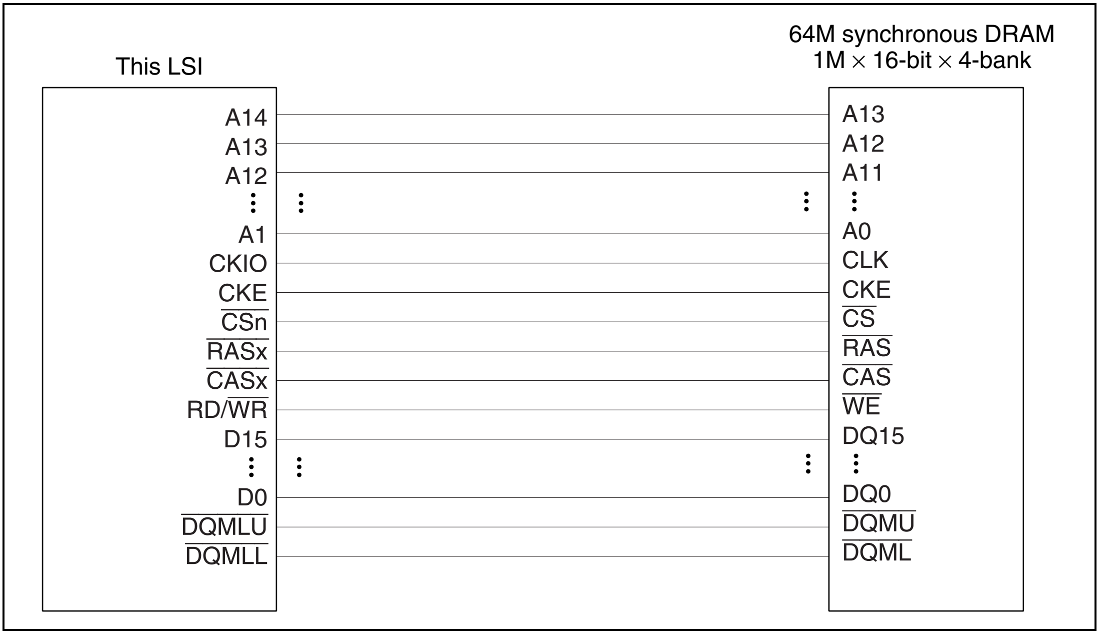


Figure 8.12 Example of 64-Mbit Synchronous DRAM (16-Bit Bus Width)

- Address Multiplexing

Synchronous DRAM can be connected without external multiplexing circuitry in accordance with the address multiplex specification bits AMX3-AMX0 in MCR. Table 8.17 shows the relationship between the address multiplex specification bits and the bits output at the address pins.

A25–A17 and A0 are not multiplexed; the original values are always output at these pins.

When A0, the LSB of the synchronous DRAM address, is connected to this LSI, it performs longword address specification. Connection should therefore be made in the following order: connect pin A0 of the synchronous DRAM to pin A2 of this LSI, then connect pin A1 to pin A3.

Table 8.18 shows an example of the connection of address pins when AMX[3:0] = 0100 with 32-bit bus width.

Table 8.17 Relationship between Bus Width, AMX, and Address Multiplex Output

Bus	Memory	Setting				Output	External Address Pins								
Width	Type	AMX3	AMX2	AMX1	AMX0	Timing	A1–A8	A9	A10	A11	A12	A13	A14	A15	A16
32 bits	4M × 16 bits × 4 banks*1	1	1	0	1	Column address	A1–A8	A9	A10	A11	L/H*3	A13	A23	A24*4	A25*4
						Row address	A10–A17	A18	A19	A20	A21	A22	A23	A24	A25*4
	2M × 16 bits × 4 banks*2	0	1	0	1	Column address	A1–A8	A9	A10	A11	L/H*3	A13	A23*4	A24*4	
						Row address	A10–A17	A18	A19	A20	A21	A22	A23*4	A24*4	
	1M × 16 bits × 4 banks*2	0	1	0	0	Column address	A1–A8	A9	A10	A11	L/H*3	A13	A22*4	A23*4	
						Row address	A9–A16	A17	A18	A19	A20	A21	A22*4	A23*4	
	2M × 8 bits × 4 banks*2	0	1	0	1	Column address	A1–A8	A9	A10	A11	L/H*3	A13	A23*4	A24*4	
						Row address	A10–A17	A18	A19	A20	A21	A22	A23*4	A24*4	
	512k × 32 bits × 4 banks*2	0	1	1	1	Column address	A1–A8	A9	A10	A11	L/H*3	A21*4	A22*4	A15	
						Row address	A9–A16	A17	A18	A19	A20	A21*4	A22*4	A23	
16 bits	8M × 16 bits × 4 banks*1	1	1	1	0	Column address	A1–A8	A9	A10	L/H*3	A12	A23	A24*4	A25*4	
						Row address	A11–A18	A19	A20	A21	A22	A23	A24*4	A25*4	
	4M × 16 bits × 4 banks*2	1	1	0	1	Column address	A1–A8	A9	A10	L/H*3	A12	A22	A23*4	A24*4	
						Row address	A10–A17	A18	A19	A20	A21	A22	A23*4	A24*4	
	2M × 16 bits × 4 banks*2	0	1	0	1	Column address	A1–A8	A9	A10	L/H*3	A12	A22*4	A23*4	A24	
						Row address	A10–A17	A18	A19	A20	A21	A22*4	A23*4	A24	
	1M × 16 bits × 4 banks*2	0	1	0	0	Column address	A1–A8	A9	A10	L/H*3	A12	A21*4	A22*4	A15	
						Row address	A9–A16	A17	A18	A19	A20	A21*4	A22*4	A23	
	2M × 8 bits × 4 banks*2	0	1	0	1	Column address	A1–A8	A9	A10	L/H*3	A12	A22*4	A23*4	A24	
						Row address	A10–A17	A18	A19	A20	A21	A22*4	A23*4	A24	

Notes: 1. Only $\overline{\text{RASL}}$/$\overline{\text{CASL}}$ are output.

2. $\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are output for upper 32-Mbyte addresses, and $\overline{\text{RASL}}$ and $\overline{\text{CASL}}$ for lower 32-Mbyte addresses.

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

4. Bank address specification

Table 8.18 Example of Correspondence between this LSI and Synchronous DRAM Address Pins (AMX (3–0) = 0100 (32-Bit Bus Width))

Address Pin of this LSI			Synchronous DRAM Address Pin	
	RAS Cycle	CAS Cycle		Function
A15	A23	A23	A13(BA1)	BANK select address
A14	A22	A22	A12(BA0)	
A13	A21	A13	A11	Address
A12	A20	L/H	A10	Address/precharge setting
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1	Not used	
A0	A8	A0	Not used	

- Burst Read

Figure 8.13 shows the timing chart for a burst read. In the example below, it is assumed that four 2M × 8-bit synchronous DRAMs are connected and a 32-bit data width is used, and the burst length is 1. Following the Tr cycle in which ACTV command output is performed, a READ command is issued in the Tc1, Tc2, and Tc3 cycles, and a READA command in the Tc4 cycle, and the read data is accepted on the rising edge of the external command clock (CKIO) from cycle Td1 to cycle Td4. The Tpc cycle is used to wait for completion of auto-precharge based on the READA command inside the synchronous DRAM; no new access command can be issued to the same bank during this cycle, but access to synchronous DRAM for another area is possible. In the this LSI, the number of Tpc cycles is determined by the TPC bit specification in MCR, and commands cannot be issued for the same synchronous DRAM during this interval.

The example in figure 8.13 shows the basic timing. To connect low-speed synchronous DRAM, the cycle can be extended by setting WCR2 and MCR bits. The number of cycles from the ACTV command output cycle, Tr, to the READ command output cycle, Tc1, can be specified by the RCD bit in MCR, with a values of 0 to 3 specifying 1 to 4 cycles, respectively. In case of 2 or more cycles, a Trw cycle, in which an NOP command is issued for the synchronous DRAM, is inserted between the Tr cycle and the Tc cycle. The number of cycles from READ and READA command output cycles Tc1-Tc4 to the first read data latch cycle, Td1, can be specified as 1 to 3 cycles independently for areas 2 and 3 by means of A2W1 and A2W0 or A3W1 and A3W0 in WCR2. This number of cycles corresponds to the number of synchronous DRAM CAS latency cycles.

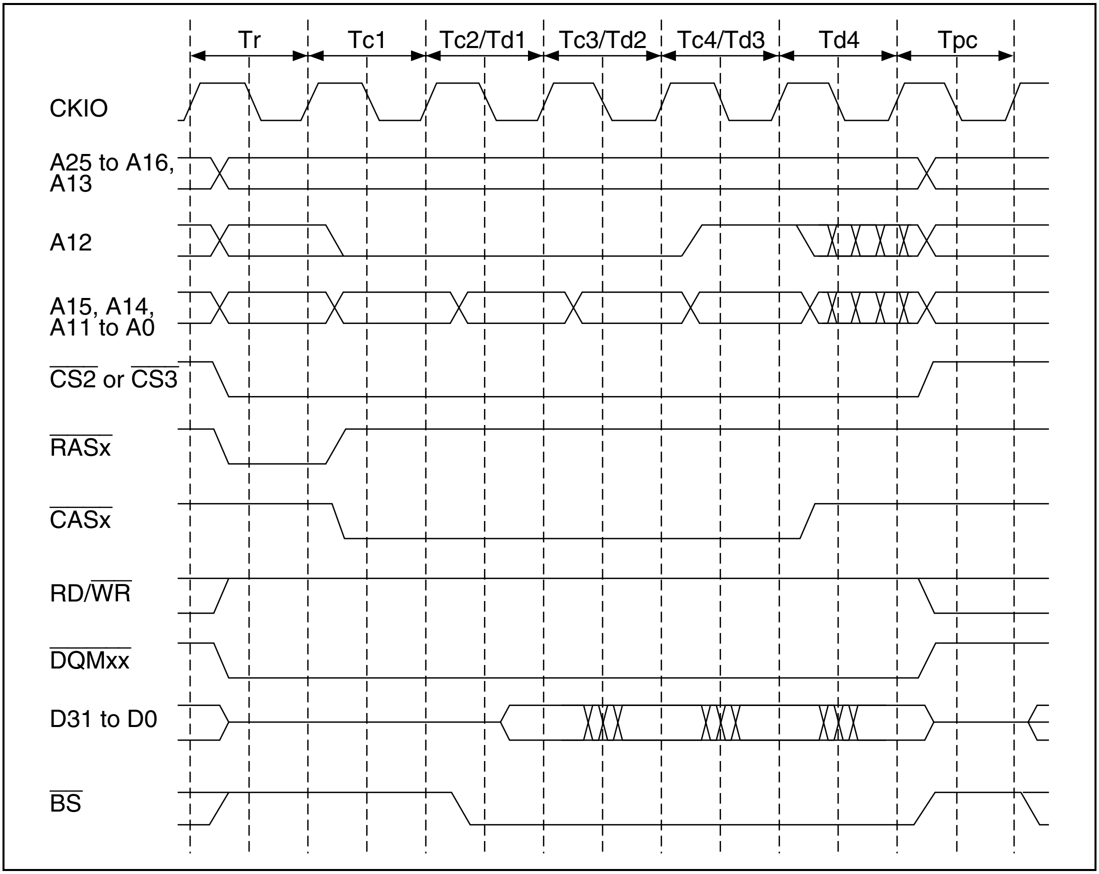


Figure 8.13 Basic Timing for Synchronous DRAM Burst Read

Figure 8.14 shows the burst read timing when RCD is set to 1, A3W1 and A3W0 are set to 10, and TPC is set to 1.

The $\overline{\text{BS}}$ cycle, which is asserted for one cycle at the start of a bus cycle for normal space access, is asserted in each of cycles Td1–Td4 in a synchronous DRAM cycle. When a burst read is performed, the address is updated each time $\overline{\text{CAS}}$ is asserted. As the unit of burst transfer is 16 bytes, address updating is performed for A3 and A2 only (A3, A2, and A1 for a 16-bit bus width). The order of access is as follows: in a fill operation in the event of a cache miss, the missed data is read first, then 16-byte boundary data including the missed data is read in wraparound mode.

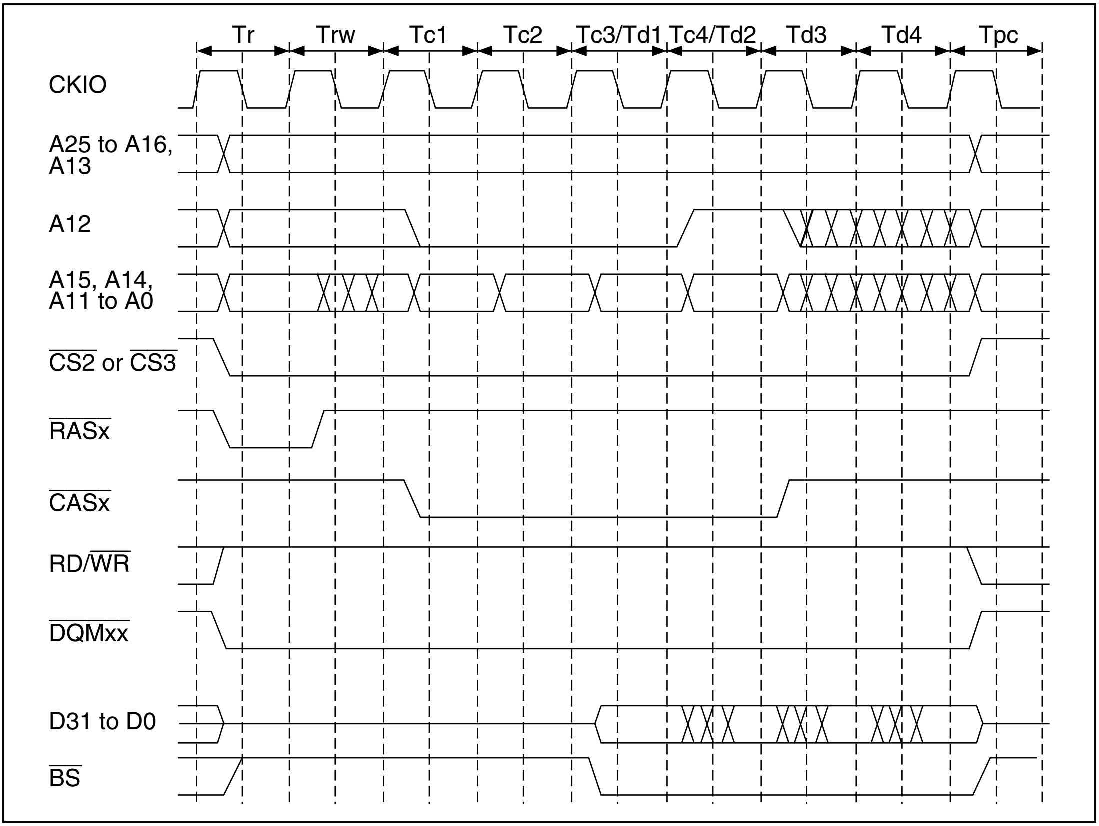


Figure 8.14 Synchronous DRAM Burst Read Wait Specification Timing

- Single Read

Figure 8.15 shows the timing when a single address read is performed. As the burst length is set to 1 in synchronous DRAM burst read/single write mode, only the required data is output. Consequently, no unnecessary bus cycles are generated even when a cache-through area is accessed.

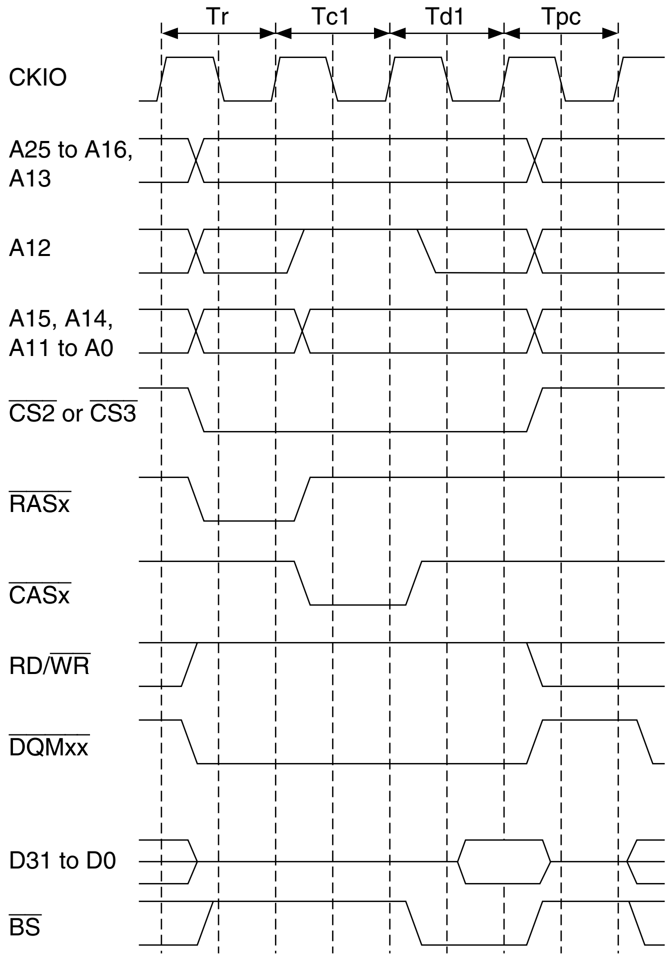


Figure 8.15 Basic Timing for Synchronous DRAM Single Read

- Burst Write

The timing chart for a burst write is shown in figure 8.16. In this LSI, a burst write occurs only in the event of cache write-back or 16-byte transfer by DMAC. In a burst write operation, following the Tr cycle in which ACTV command output is performed, a WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and a WRITA command that performs auto-precharge is issued in the Tc4 cycle. In the write cycle, the write data is output at the same time as the write command. In case of the write with auto-precharge command, precharging of the relevant bank is performed in the synchronous DRAM after completion of the write command, and therefore no command can be issued for the same bank until precharging is completed. Consequently, in addition to the precharge wait cycle, Tpc, used in a read access, cycle Trwl is also added as a wait interval until precharging is started following the write command. Issuance of a new command for the same bank is postponed during this interval. The number of Trwl cycles can be specified by the TRWL bit in MCR.

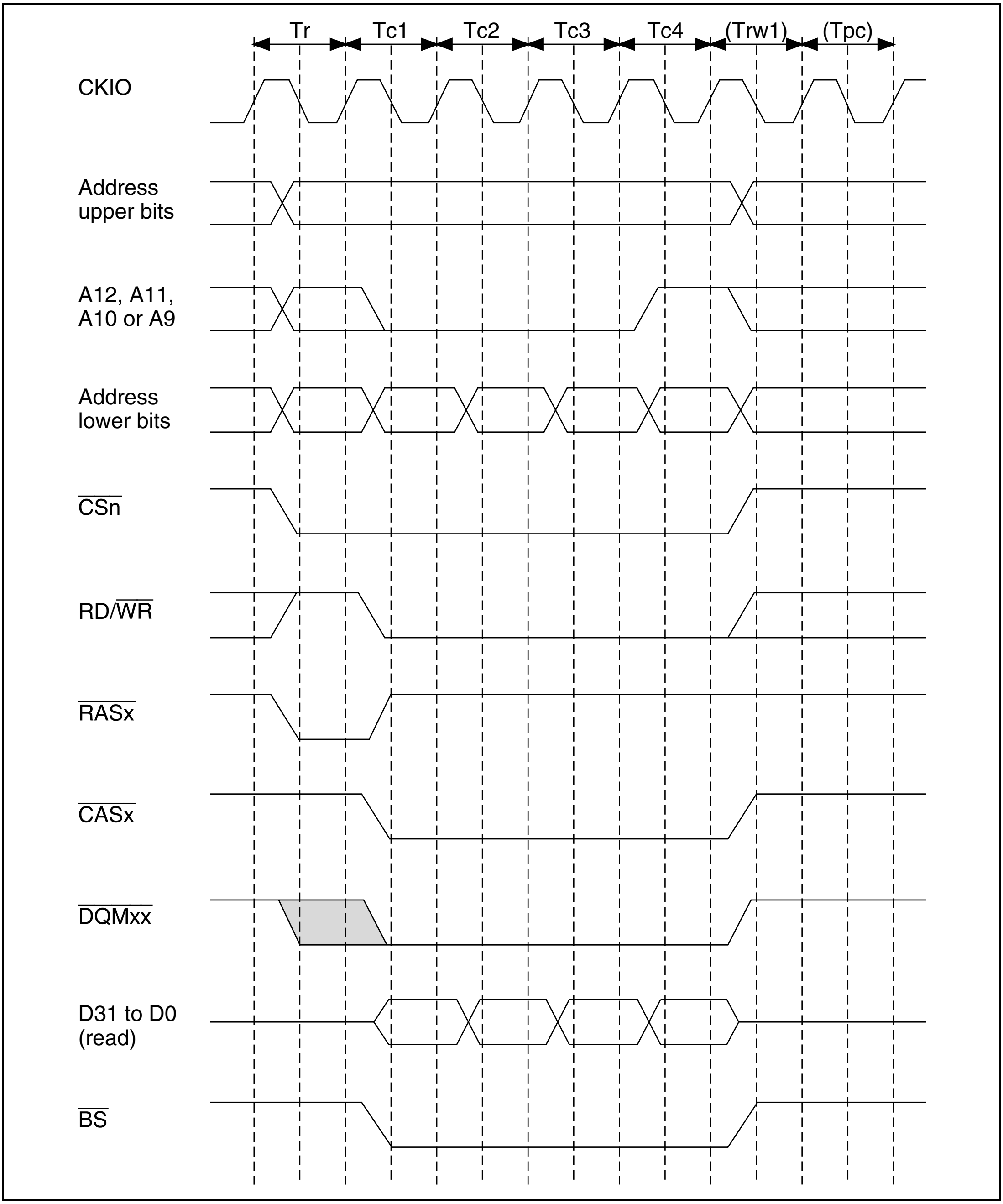


Figure 8.16 Basic Timing for Synchronous DRAM Burst Write

- Single Write

The basic timing chart for write access is shown in figure 8.17. In a single write operation, following the Tr cycle in which ACTV command output is performed, a WRITA command that performs auto-precharge is issued in the Tc1 cycle. In the write cycle, the write data is output at the same time as the write command. In case of the write with auto-precharge command, precharging of the relevant bank is performed in the synchronous DRAM after completion of the write command, and therefore no command can be issued for the same bank until precharging is completed. Consequently, in addition to the precharge wait cycle, Tpc, used in a read access, cycle Trwl is also added as a wait interval until precharging is started following the write command. Issuance of a new command for the same bank is postponed during this interval. The number of Trwl cycles can be specified by the TRWL bit in MCR.

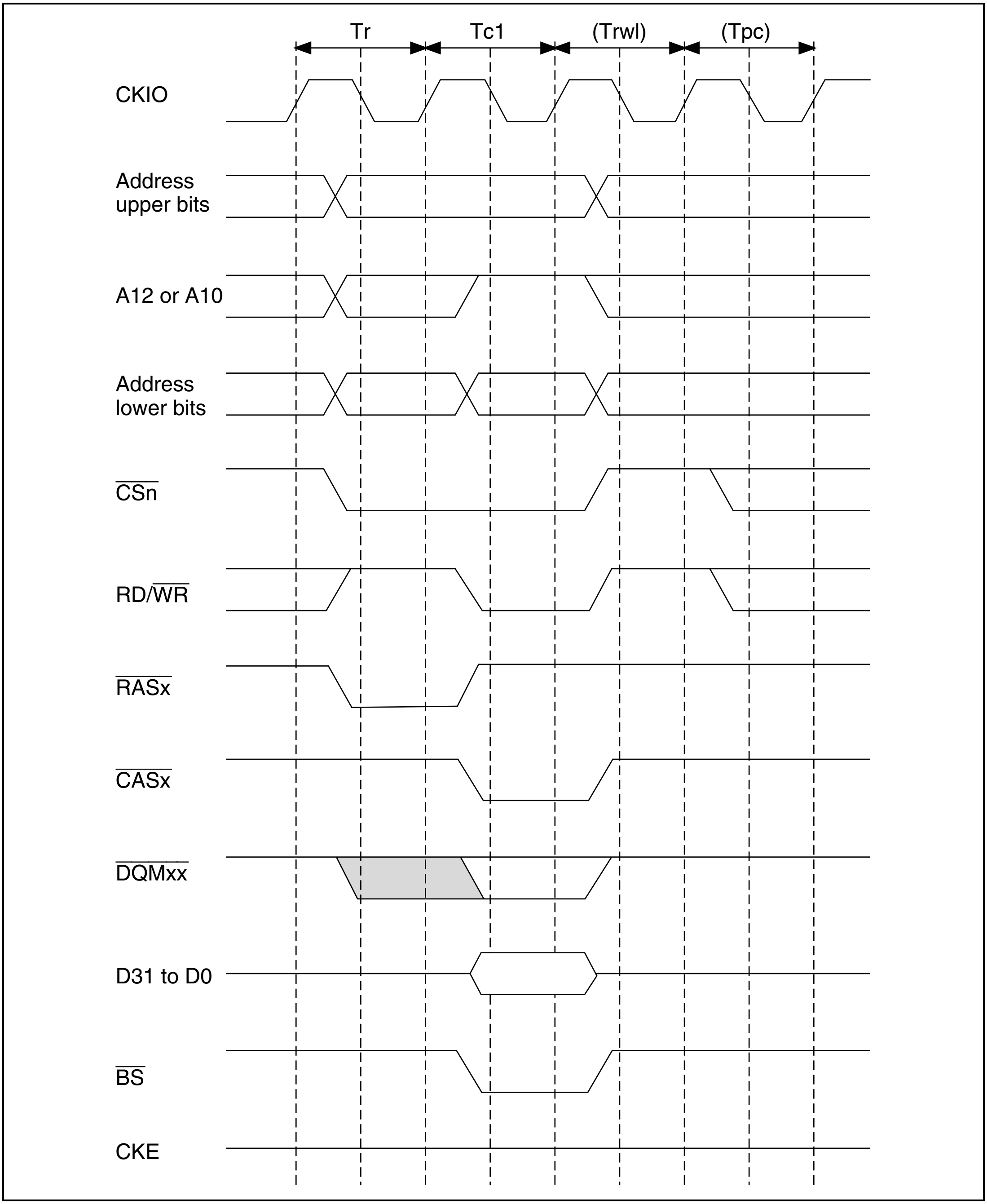


Figure 8.17 Basic Timing for Synchronous DRAM Single Write

- Bank Active

The synchronous DRAM bank function is used to support high-speed accesses to the same row address. When the RASD bit in MCR is 1, read/write command accesses are performed using commands without auto-precharge (READ, WRIT). In this case, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately, without issuing an ACTV command. As synchronous DRAM is internally divided into two or four banks, it is possible to activate one row address in each bank. If the next access is to a different row address, a PRE command is first issued to precharge the relevant bank, then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is issued.

In a write, when auto-precharge is performed, a command cannot be issued for a period of Trwl + Tpc cycles after issuance of the WRITA command. When bank active mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by Trwl + Tpc cycles for each write. The number of cycles between issuance of the precharge command and the row address strobe command is determined by the TPC bit in MCR.

Whether faster execution speed is achieved by use of bank active mode or by use of basic access is determined by the probability of accessing the same row address (P1), and the average number of cycles from completion of one access to the next access (Ta). If Ta is greater than Tpc, the delay due to the precharge wait when reading is imperceptible. If Ta is greater than Trw1 + Tpc, the delay due to the precharge wait when writing is imperceptible. In this case, the access speed for bank active mode and basic access is determined by the number of cycles from the start of access to issuance of the read/write command: (Tpc + Trcd) × (1 – P1) and Trcd, respectively.

There is a limit on Tras, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of Tras. In this way, it is possible to observe the restrictions on the maximum active state time for each bank. If auto-refresh is not used, measures must be taken in the program to ensure that the banks do not remain active for longer than the prescribed time.

A burst read cycle without auto-precharge is shown in figure 8.18, a burst read cycle for the same row address in figure 8.19, and a burst read cycle for different row addresses in figure 8.20. Similarly, a burst write cycle without auto-precharge is shown in figure 8.21, a burst write cycle for the same row address in figure 8.22, and a burst write cycle for different row addresses in figure 8.23.

A Tnop cycle, in which no operation is performed, is inserted before the Tc1 cycle in which the READ command is issued in figure 8.19, but when synchronous DRAM is read, there is a two-cycle latency for the $\overline{\text{DQMxx}}$ signal that performs the byte specification. If the Tc1 cycle were performed immediately, without inserting a Tnop cycle, it would not be possible to perform the $\overline{\text{DQMxx}}$ signal specification for Td1 cycle data output. This is the reason for inserting the Tnop cycle. If the CAS latency is two cycles or longer, Tnop cycle insertion is not performed, since the timing requirements will be met even if the $\overline{\text{DQMxx}}$ signal is set after the Tc1 cycle.

When bank active mode is set, if only accesses to the respective banks in the area 3 space are considered, as long as accesses to the same row address continue, the operation starts with the cycle in figure 8.18 or 8.21, followed by repetition of the cycle in figure 8.19 or 8.22. An access to a different area 3 space during this time has no effect. If there is an access to a different row address in the bank active state, after this is detected the bus cycle in figure 8.19 or 8.22 is executed instead of that in figure 8.19 or 8.22. In bank active mode, too, all banks become inactive after a refresh cycle or after the bus is released as the result of bus arbitration.

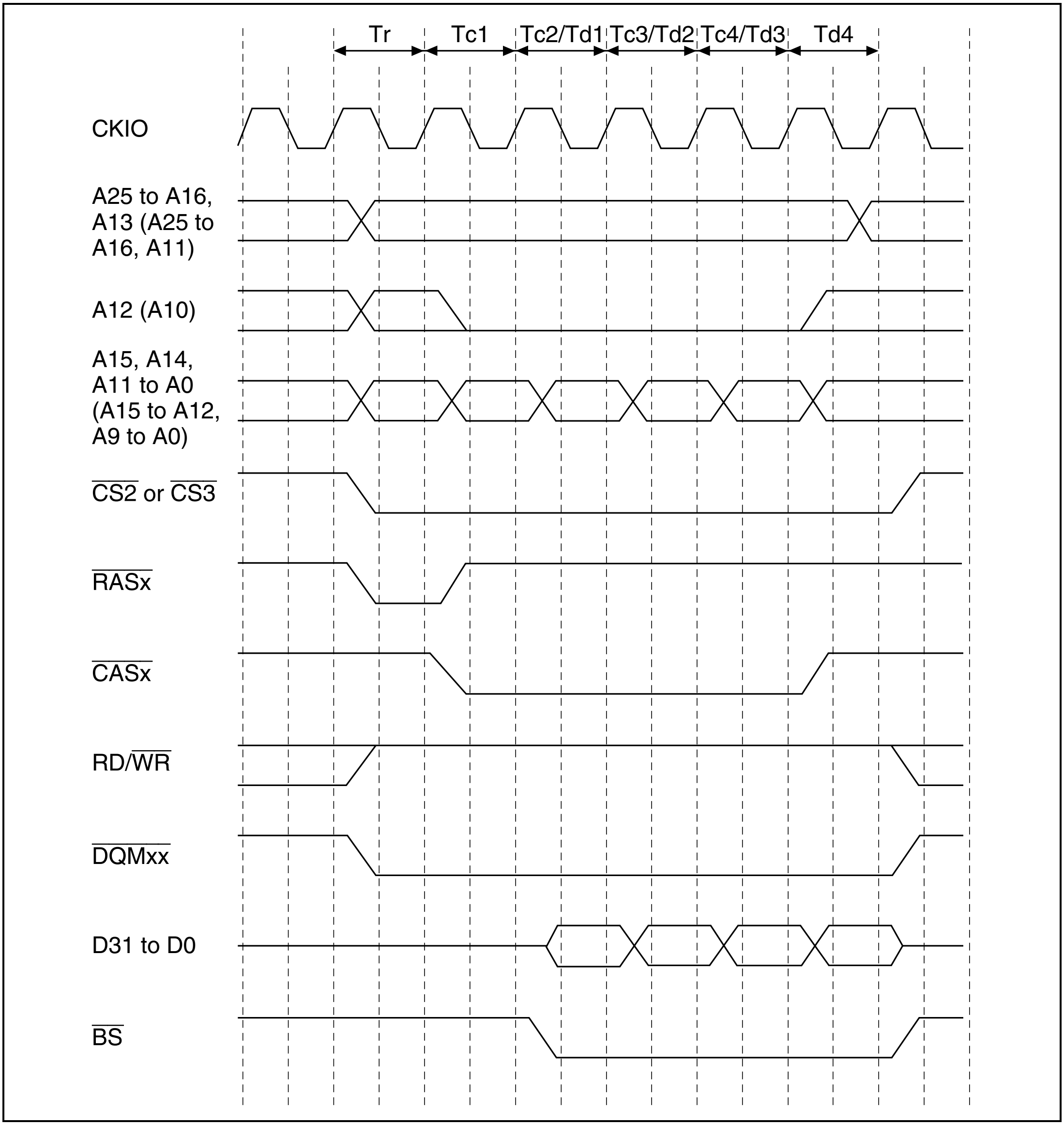


Figure 8.18 Burst Read Timing (No Precharge)

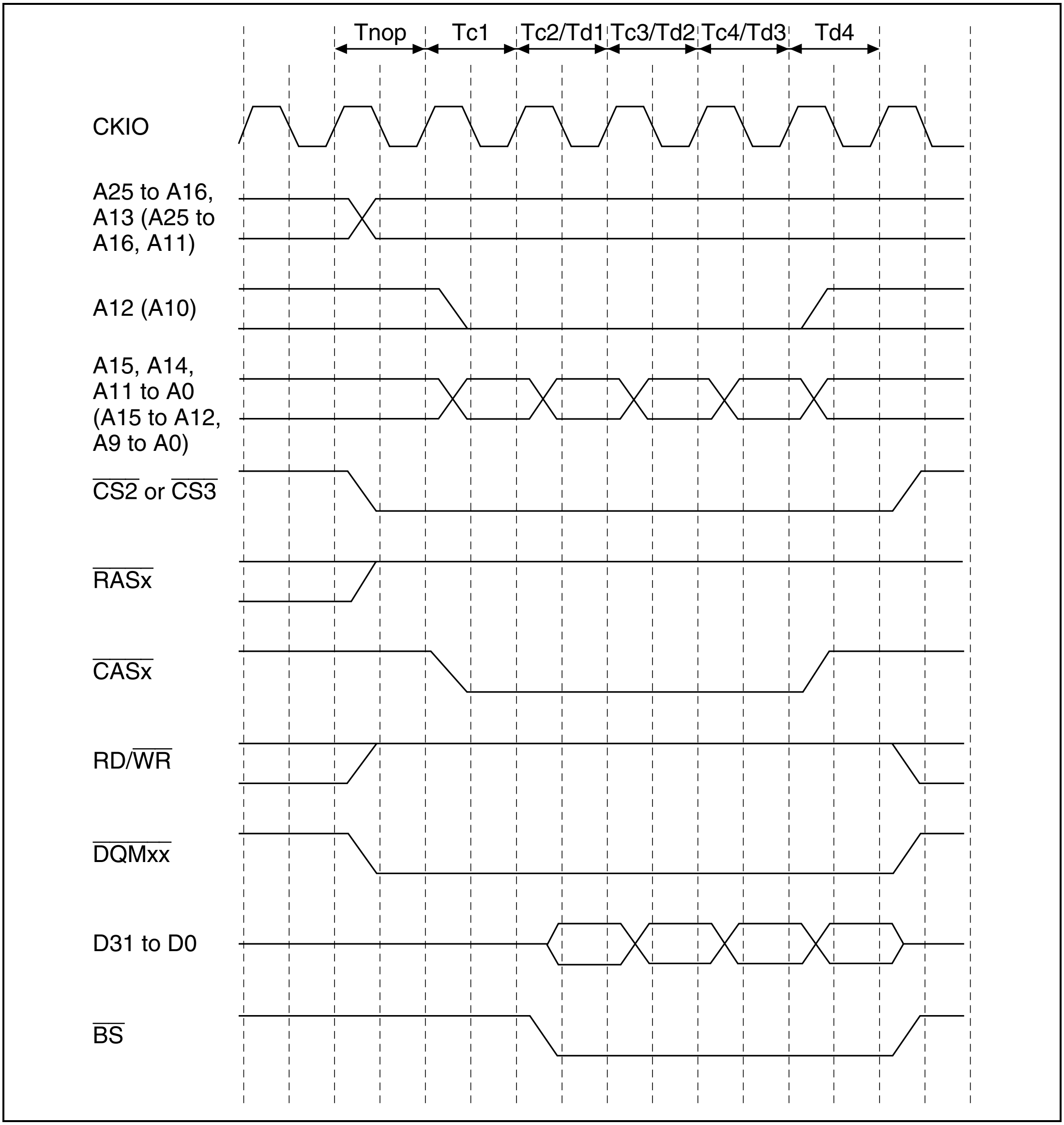

Tnop
Tc1
Tc2/Td1
Tc3/Td2
Tc4/Td3
Td4
CKIO
A25 to A16, A13 (A25 to A16, A11)
A12 (A10)
A15, A14, A11 to A0 (A15 to A12, A9 to A0)
$\overline{\text{CS2}}$ or $\overline{\text{CS3}}$
$\overline{\text{RASx}}$
$\overline{\text{CASx}}$
RD/$\overline{\text{WR}}$
$\overline{\text{DQMxx}}$
D31 to D0
$\overline{\text{BS}}$

Figure 8.19 Burst Read Timing (Same Row Address)

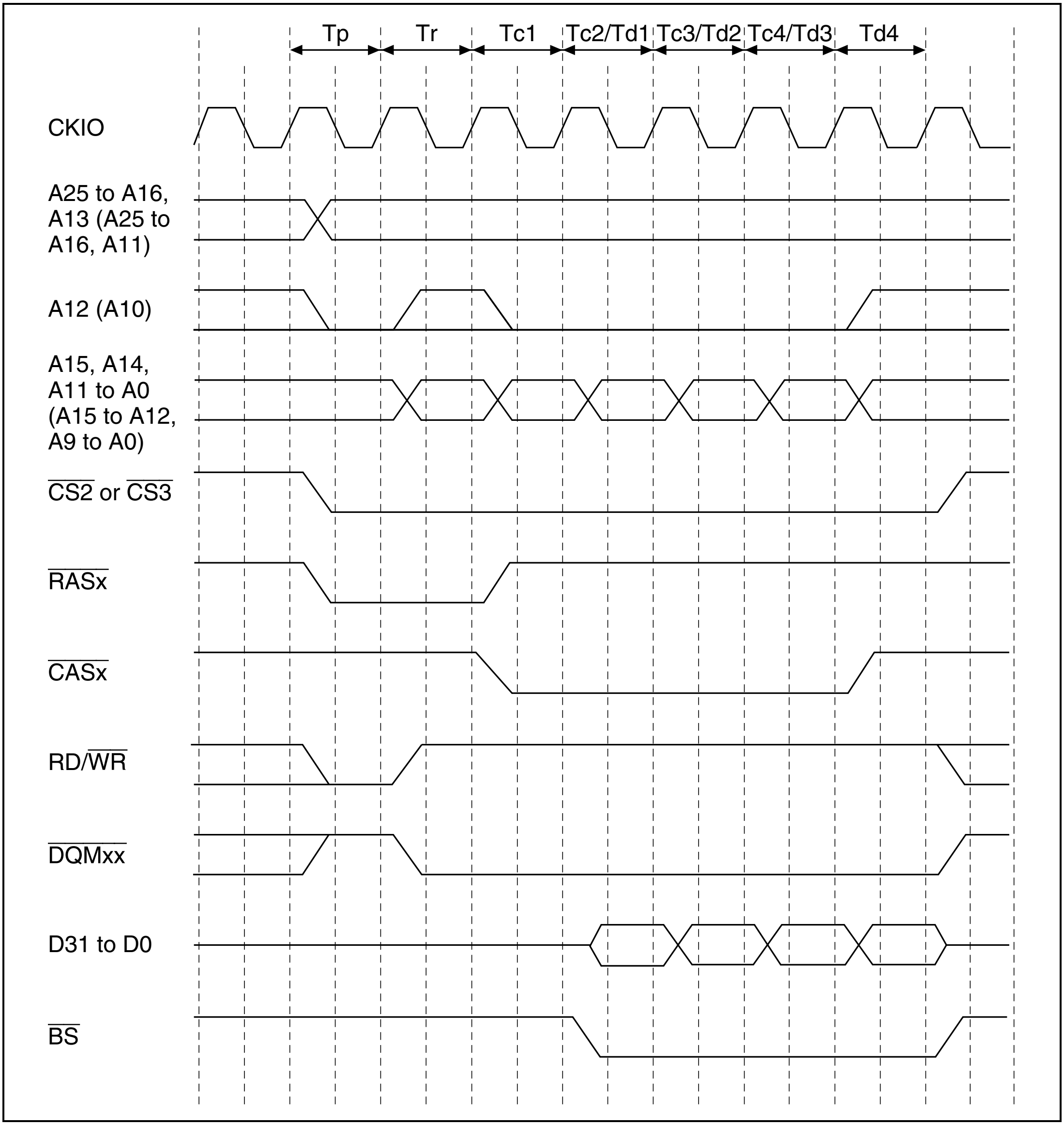

Tp
Tr
Tc1
Tc2/Td1
Tc3/Td2
Tc4/Td3
Td4
CKIO
A25 to A16, A13 (A25 to A16, A11)
A12 (A10)
A15, A14, A11 to A0 (A15 to A12, A9 to A0)
$\overline{\text{CS2}}$ or $\overline{\text{CS3}}$
$\overline{\text{RASx}}$
$\overline{\text{CASx}}$
RD/$\overline{\text{WR}}$
$\overline{\text{DQMxx}}$
D31 to D0
$\overline{\text{BS}}$

Figure 8.20 Burst Read Timing (Different Row Addresses)

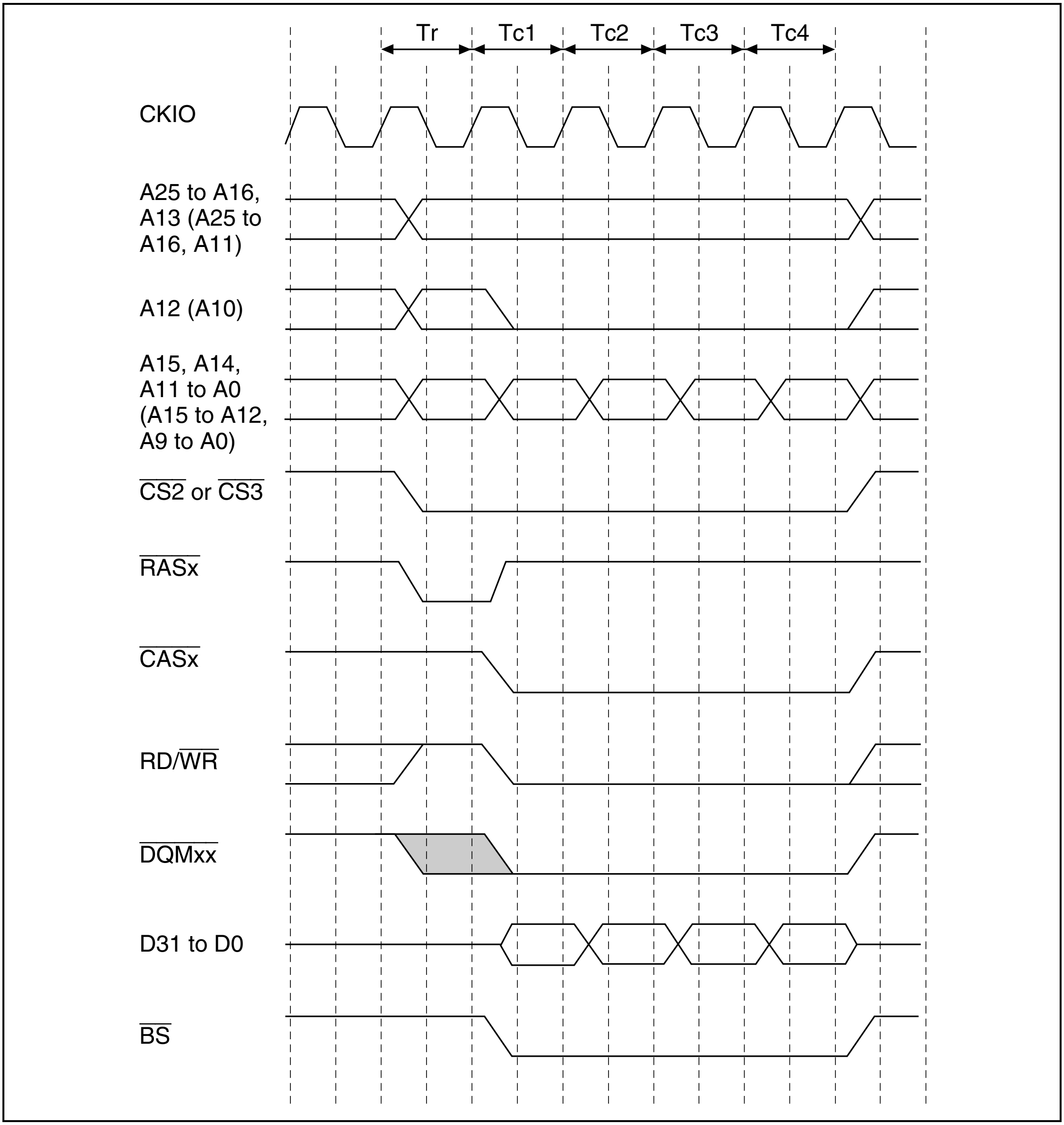

Tr
Tc1
Tc2
Tc3
Tc4
CKIO
A25 to A16, A13 (A25 to A16, A11)
A12 (A10)
A15, A14, A11 to A0 (A15 to A12, A9 to A0)
CS2 or CS3
RASx
CASx
RD/WR
DQMxx
D31 to D0
BS

Figure 8.21 Burst Write Timing (No Precharge)

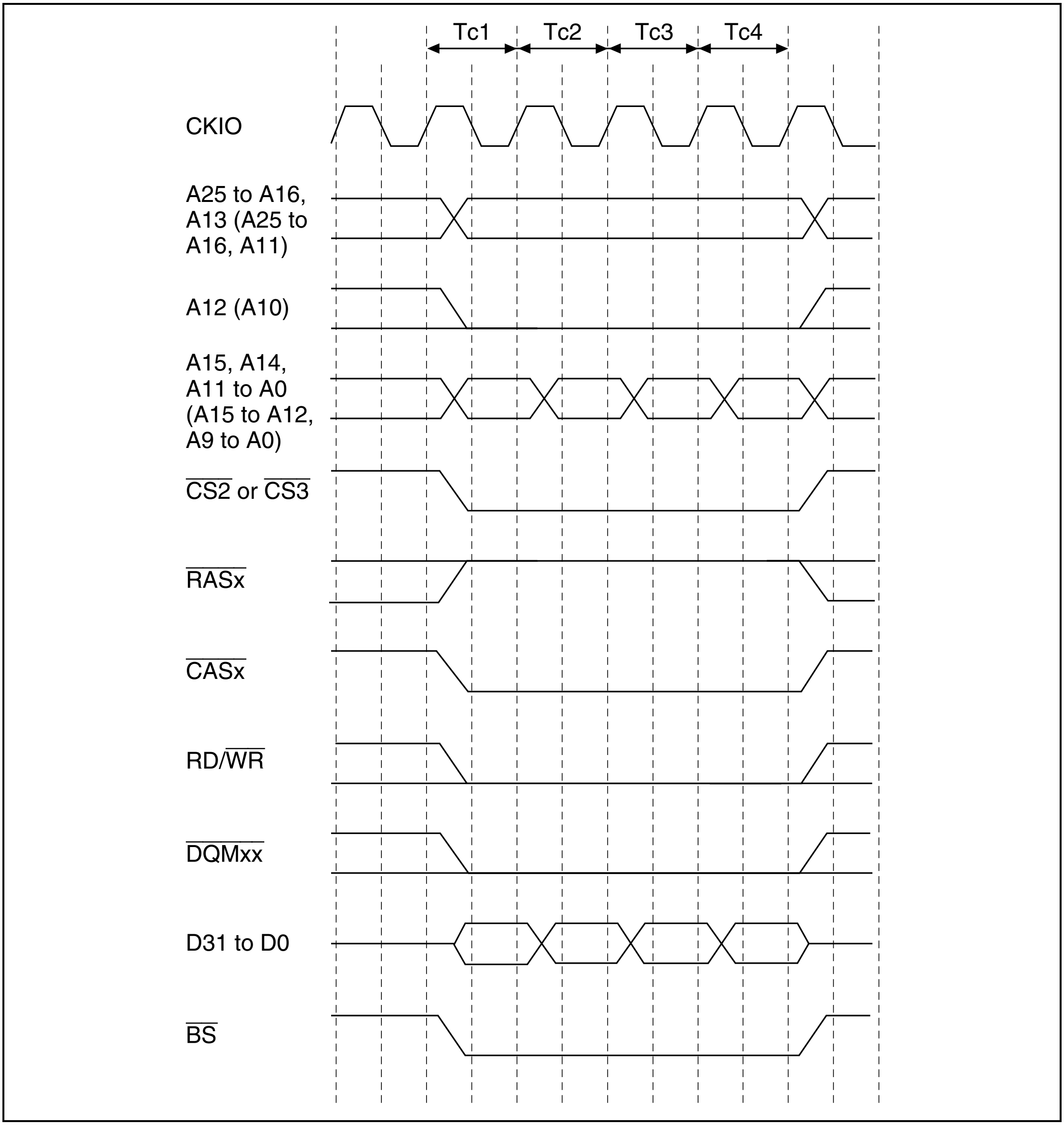


Figure 8.22 Burst Write Timing (Same Row Address)

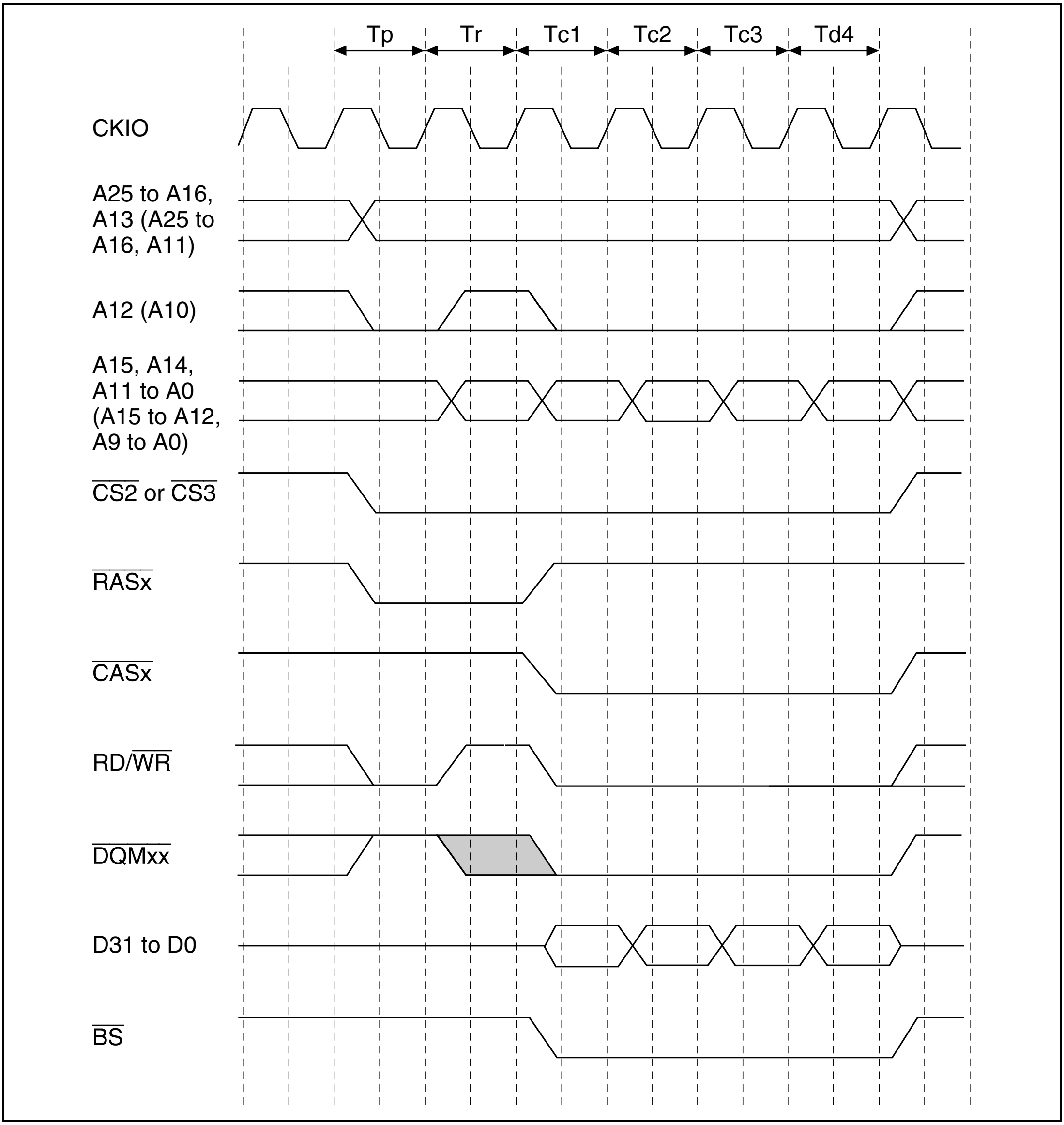

Tp
Tr
Tc1
Tc2
Tc3
Td4
CKIO
A25 to A16, A13 (A25 to A16, A11)
A12 (A10)
A15, A14, A11 to A0 (A15 to A12, A9 to A0)
$\overline{\text{CS2}}$ or $\overline{\text{CS3}}$
$\overline{\text{RASx}}$
$\overline{\text{CASx}}$
RD/$\overline{\text{WR}}$
$\overline{\text{DQMxx}}$
D31 to D0
$\overline{\text{BS}}$

Figure 8.23 Burst Write Timing (Different Row Addresses)

- Refreshing

The bus state controller is provided with a function for controlling synchronous DRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in MCR. If synchronous DRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

1. Auto-Refreshing

   Figure 8.24 shows the auto-refreshing operation.

   Refreshing is performed at intervals determined by the input clock selected by bits CKS2-0 in RTCSR, and the value set in RTCOR. The value of bits CKS2-0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the synchronous DRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in MCR, then make the CKS2-CKS0 setting. When the clock is selected by CKS2-CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed. At the same time, RTCNT is cleared to zero and the count-up is restarted. Figure 8.25 shows the auto-refresh cycle timing.

   All-bank precharging is performed in the Tp cycle, then an REF command is issued in the TRr cycle following the interval specified by the TPC bits in MCR. After the TRr cycle, new command output cannot be performed for the duration of the number of cycles specified by the TRAS bits in MCR plus the number of cycles specified by the TPC bits in MCR. The TRAS and TPC bits must be set so as to satisfy the synchronous DRAM refresh cycle time stipulation (active/active command delay time).

   Auto-refreshing is performed in normal operation, in sleep mode, and in case of a manual reset.

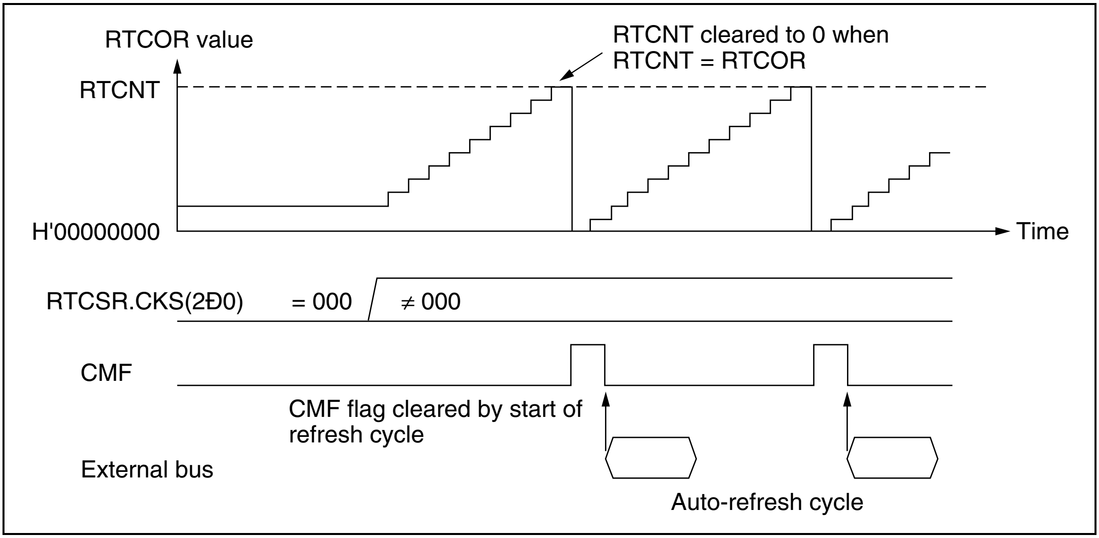


Figure 8.24 Auto-Refresh Operation

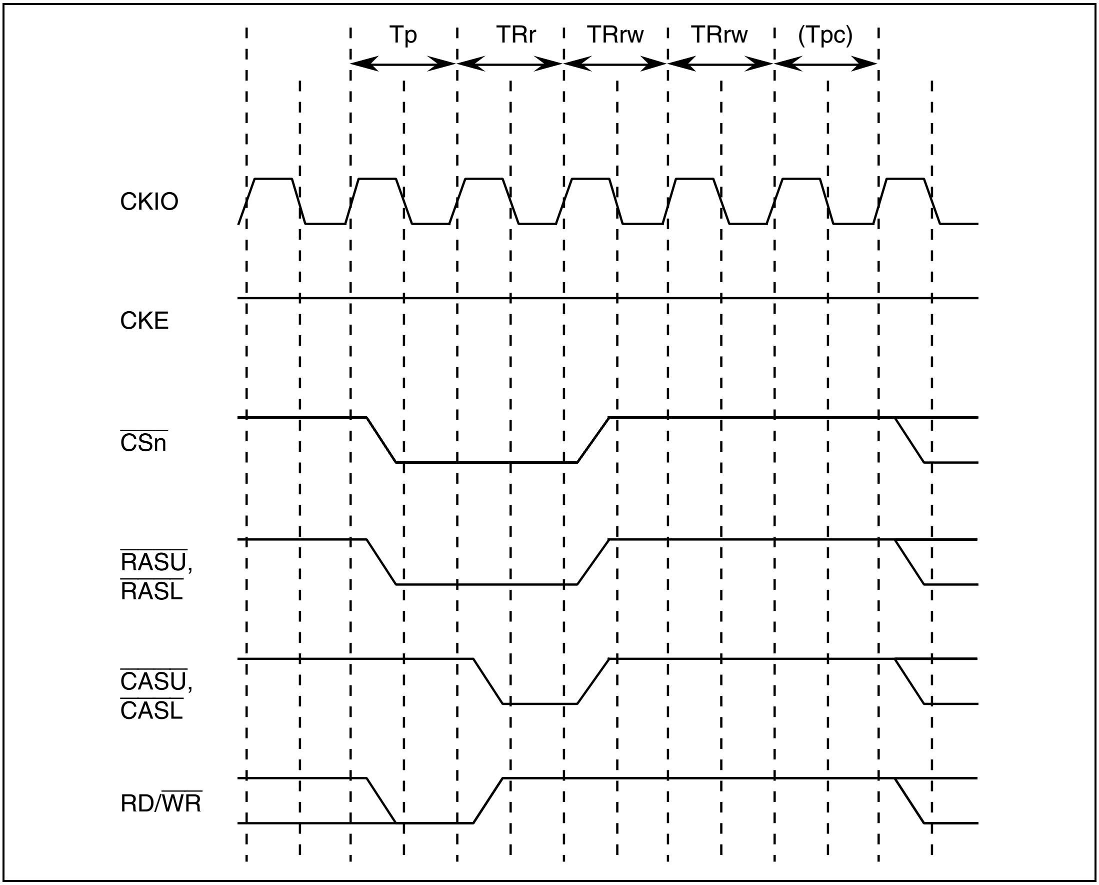


Figure 8.25 Synchronous DRAM Auto-Refresh Timing

2. Self-Refreshing

Self-refresh mode is a kind of standby mode in which the refresh timing and refresh addresses are generated within the synchronous DRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit to 1. The self-refresh state is maintained while the CKE signal is low. Synchronous DRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the TPC bits in MCR. Self-refresh timing is shown in figure 8.26. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if RFSH is set to 1 and RMODE is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the this LSI standby function, and is maintained even after recovery from standby mode other than through a power-on reset. In case of a power-on reset, the bus state controller’s registers are initialized, and therefore the self-refresh state is cleared.

Self-refreshing is performed in normal operation, in sleep mode, in standby mode, and in case of a manual reset.

When using synchronous DRAM, use the following procedure to initiate self-refreshing.

1. Clear the refresh control bit to 0.
2. Write H'00 to the RTCNT register.
3. Set the refresh control bit and refresh mode bit to 1.

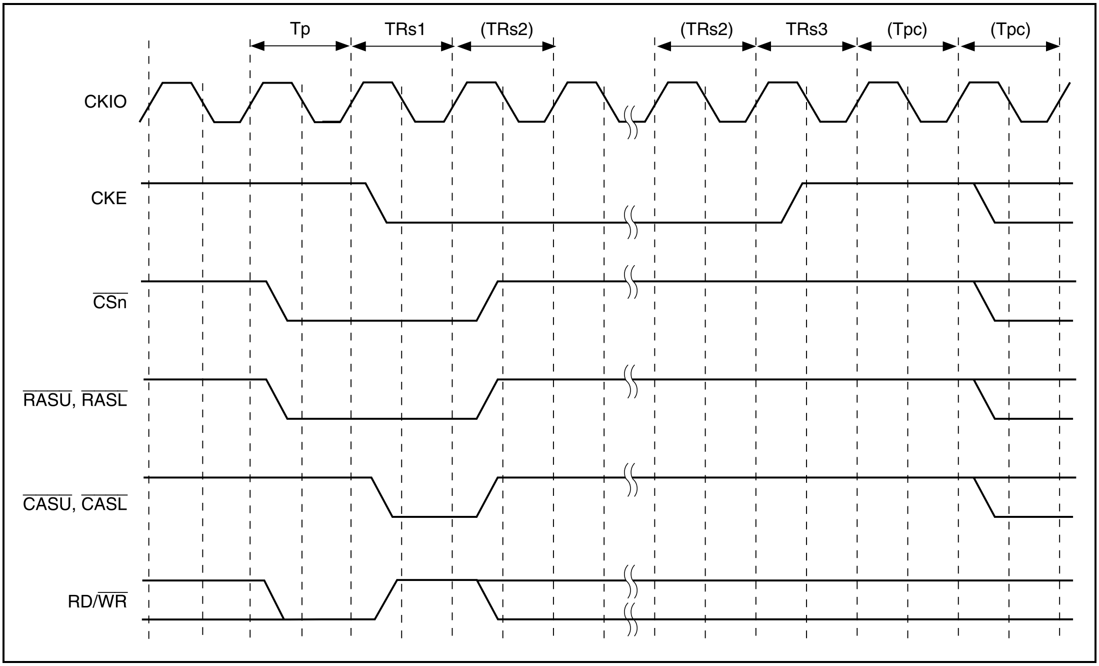

Tp
TRs1
(TRs2)
(TRs2)
TRs3
(Tpc)
(Tpc)
CKIO
CKE
$\overline{\text{CSn}}$
$\overline{\text{RASU}}$, $\overline{\text{RASL}}$
$\overline{\text{CASU}}$, $\overline{\text{CASL}}$
RD/$\overline{\text{WR}}$

Figure 8.26 Synchronous DRAM Self-Refresh Timing

3. Relationship between Refresh Requests and Bus Cycle Requests

   If a refresh request is generated during execution of a bus cycle, execution of the refresh is deferred until the bus cycle is completed. If a refresh request occurs when the bus has been released by the bus arbiter, refresh execution is deferred until the bus is acquired. If a match between RTCNT and RTCOR occurs while a refresh is waiting to be executed, so that a new refresh request is generated, the previous refresh request is eliminated. In order for refreshing to be performed normally, care must be taken to ensure that no bus cycle or bus mastership occurs that is longer than the refresh interval. When a refresh request is generated, the $\overline{\text{IRQOUT}}$ pin is asserted (driven low). Therefore, normal refreshing can be performed by having the $\overline{\text{IRQOUT}}$ pin monitored by a bus master other than this LSI requesting the bus, or the bus arbiter, and returning the bus to this LSI. When refreshing is started, and if no other interrupt request has been generated, the $\overline{\text{IRQOUT}}$ pin is negated (driven high).

- Power-On Sequence

In order to use synchronous DRAM, mode setting must first be performed after powering on. To perform synchronous DRAM initialization correctly, the bus state controller registers must first be set, followed by a write to the synchronous DRAM mode register. In synchronous DRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and RD/$\overline{\text{WR}}$ signals. If the value to be set is X, the bus state controller provides for value X to be written to the synchronous DRAM mode register by performing a write to address H'FFFFD000 + X for area 2 synchronous DRAM, and to address H'FFFFE000 + X for area 3 synchronous DRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/single write, CAS latency 1 to 3, wrap type = sequential, and burst length 1 supported by this LSI, arbitrary data is written in a byte-size access to the following addresses.

		Area 2	Area 3
32-bit	CAS latency 1	FFFFD840	FFFFE840
Bus width	CAS latency 2	FFFFD880	FFFFE880
	CAS latency 3	FFFFD8C0	FFFFE8C0

		Area 2	Area 3
16-bit	CAS latency 1	FFFFD420	FFFFE420
Bus width	CAS latency 2	FFFFD440	FFFFE440
	CAS latency 3	FFFFD460	FFFFE460

Mode register setting timing is shown in figure 8.27.

As a result of the write to address H'FFFFD000 + X or H'FFFFE000 + X, a precharge all banks (PALL) command is first issued in the TRp1 cycle, then a mode register write command is issued in the TMw1 cycle.

Address signals, when the mode-register write command is issued, are as follows:

32-bit	A15-A9	0000100 (burst read and single write)
Bus width	A8-A6	CAS latency
	A5	0 (burst type = sequential)
	A4-A2	000 (burst length 1)

16-bit	A14-A8	0000100 (burst read and single write)
Bus width	A7-A5	CAS latency
	A4	0 (burst type = sequential)
	A3-A1	000 (burst length 1)

Before mode register setting, a 100 µs idle time (depending on the memory manufacturer) must be guaranteed after powering on requested by the synchronous DRAM. If the reset signal pulse width is greater than this idle time, there is no problem in performing mode register setting immediately. The number of dummy auto-refresh cycles specified by the manufacturer (usually 8) or more must be executed. This is usually achieved automatically while various kinds of initialization are being performed after auto-refresh setting, but a way of carrying this out more dependably is to set a short refresh request generation interval just while these dummy cycles are being executed. With simple read or write access, the address counter in the synchronous DRAM used for auto-refreshing is not initialized, and so the cycle must always be an auto-refresh cycle.

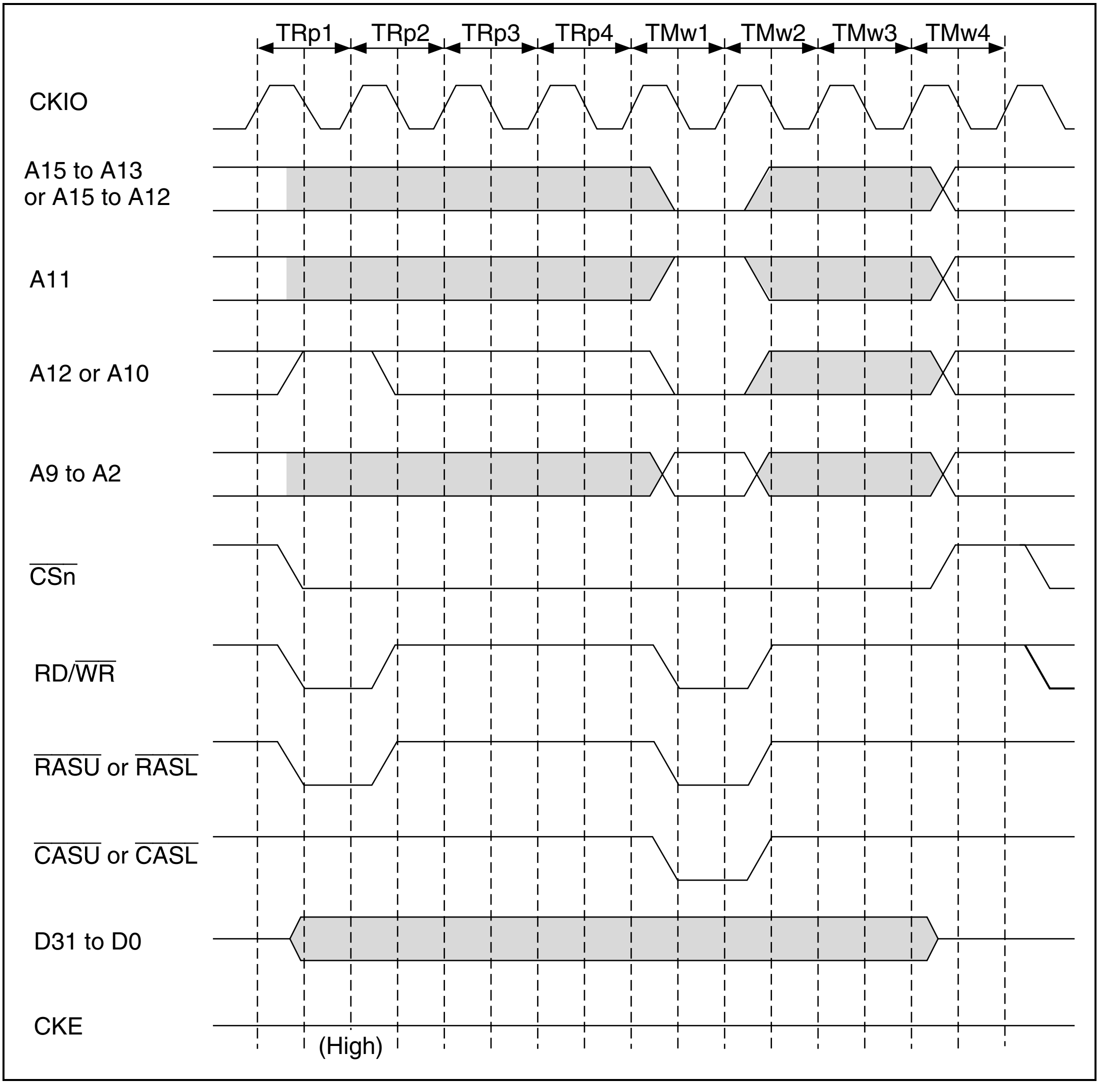

TRp1
TRp2
TRp3
TRp4
TMw1
TMw2
TMw3
TMw4
CKIO
A15 to A13
or A15 to A12
A11
A12 or A10
A9 to A2
$\overline{\text{CSn}}$
RD/$\overline{\text{WR}}$
$\overline{\text{RASU}}$ or $\overline{\text{RASL}}$
$\overline{\text{CASU}}$ or $\overline{\text{CASL}}$
D31 to D0
CKE
(High)

Figure 8.27 Synchronous DRAM Mode Write Timing

8.5.5 Burst ROM Interface

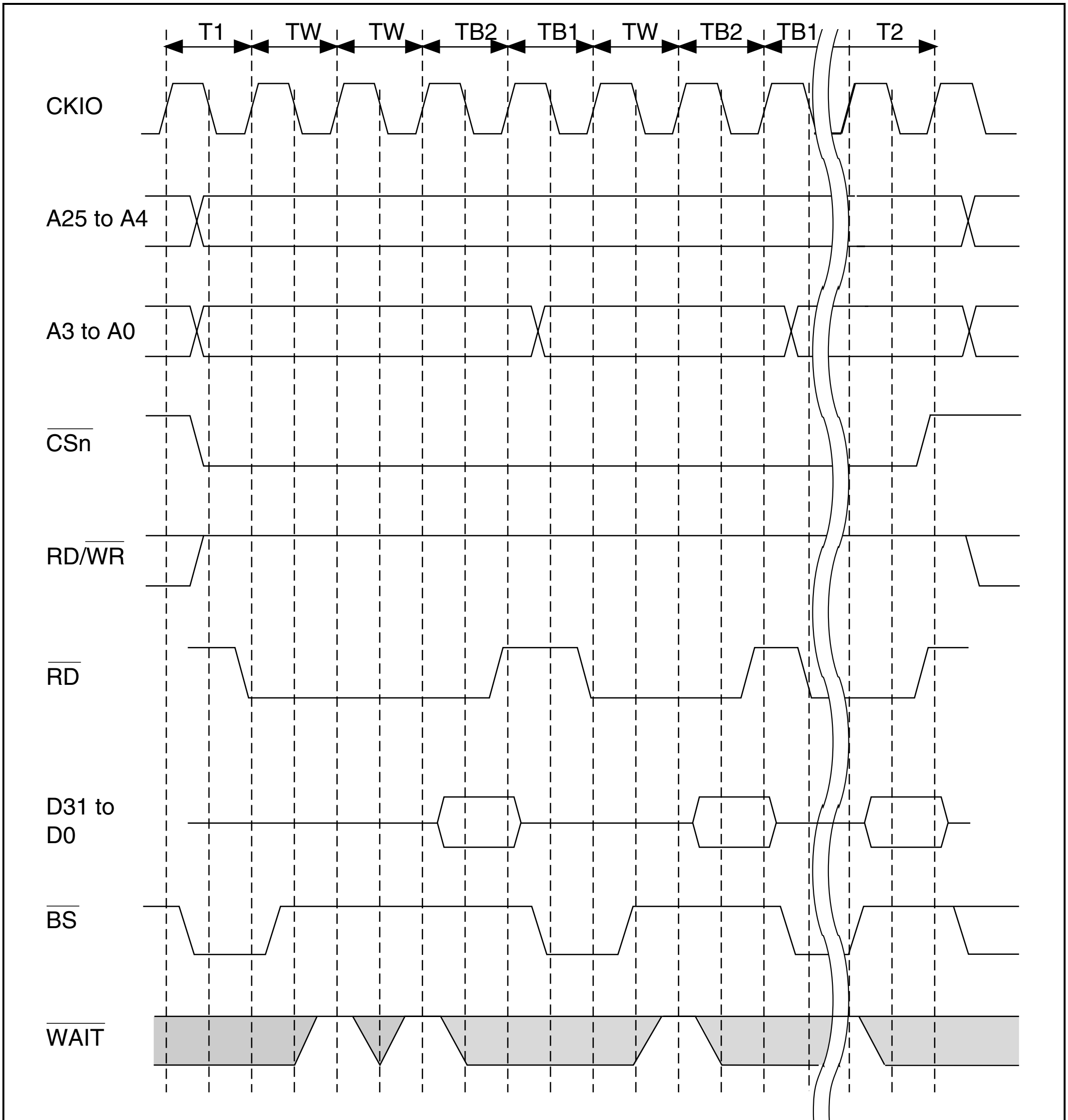
Setting bits A0BST (1–0), A5BST (1–0), and A6BST (1–0) in BCR1 to a non-zero value allows burst ROM to be connected to areas 0, 5, and 6. The burst ROM interface provides high-speed access to ROM that has a nibble access function. The timing for nibble access to burst ROM is shown in figure 8.28. Two wait cycles are set. Basically, access is performed in the same way as for normal space, but when the first cycle ends the $\overline{\text{CS0}}$ signal is not negated, and only the address is changed before the next access is executed. When 8-bit ROM is connected, the number of consecutive accesses can be set as 4, 8, or 16 by bits A0BST (1–0), A5BST (1–0), or A6BST (1–0). When 16-bit ROM is connected, 4 or 8 can be set in the same way. When 32-bit ROM is connected, only 4 can be set.

$\overline{\text{WAIT}}$ pin sampling is performed in the first access if one or more wait states are set, and is always performed in the second and subsequent accesses.

The second and subsequent access cycles also comprise two cycles when a burst ROM setting is made and the wait specification is 0. The timing in this case is shown in figure 8.29.

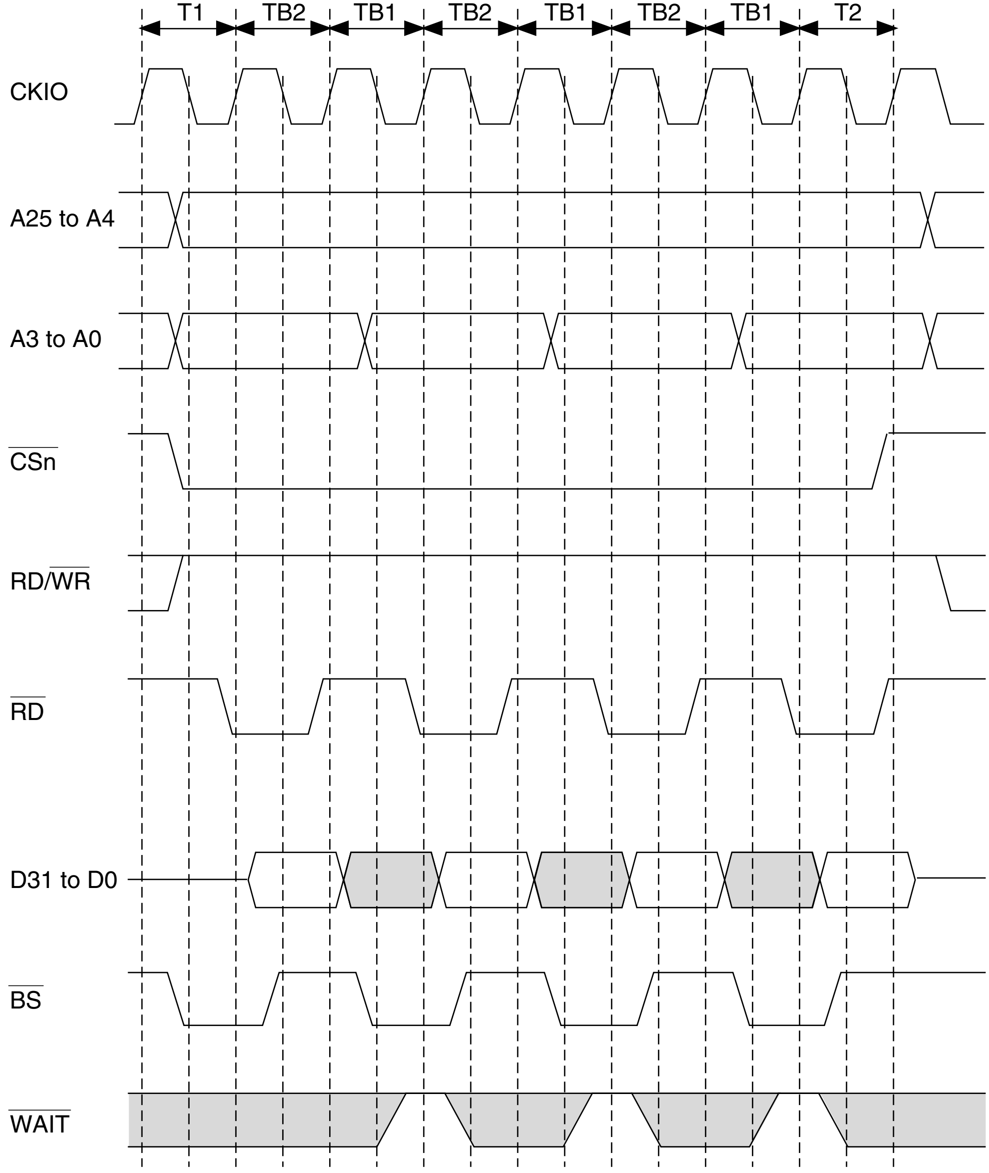
However, the $\overline{\text{WAIT}}$ signal is ignored in the following cases:

- In 16-byte DMA transfer or dual addressing mode, or when writing data to the external address area
- In 16-byte DMA transfer or single addressing mode, or when transferring data from an external device with DACK to the external bus area
- When accessing cache for write back



Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 8.28 Burst ROM Wait Access Timing



Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 8.29 Burst ROM Basic Access Timing

8.5.6 PCMCIA Interface

In this LSI, setting the A5PCM bit in BCR1 to 1 makes the bus interface for physical space area 5 an IC memory card and I/O card interface as stipulated in JEIDA version 4.2 (PCMCIA2.1). Setting the A6PCM bit to 1 makes the bus interface for physical space area 6 an IC memory card and I/O card interface as stipulated in JEIDA version 4.2.

Figure 8.30 shows the PCMCIA space allocation.

When the PCMCIA interface is used, a bus size of 8 or 16 bits can be set by bits A5SZ1 and A5SZ0, or A6SZ1 and A6SZ0, in BCR2.

Figure 8.31 shows an example of PCMCIA card connection to this LSI. To enable active insertion of the PCMCIA cards (i.e. insertion or removal while system power is being supplied), a 3-state buffer must be connected between this LSI bus interface and the PCMCIA cards.

As operation in big-endian mode is not explicitly stipulated in the JEIDA/PCMCIA specifications, the PCMCIA interface for this LSI in big-endian mode is stipulated independently.

However, the $\overline{\text{WAIT}}$ signal is ignored in the following cases:

- In 16-byte DMA transfer or dual addressing mode, or when writing data to the external address area
- In 16-byte DMA transfer or single addressing mode, or when transferring data from an external device with DACK to the external bus area
- When accessing cache for write back

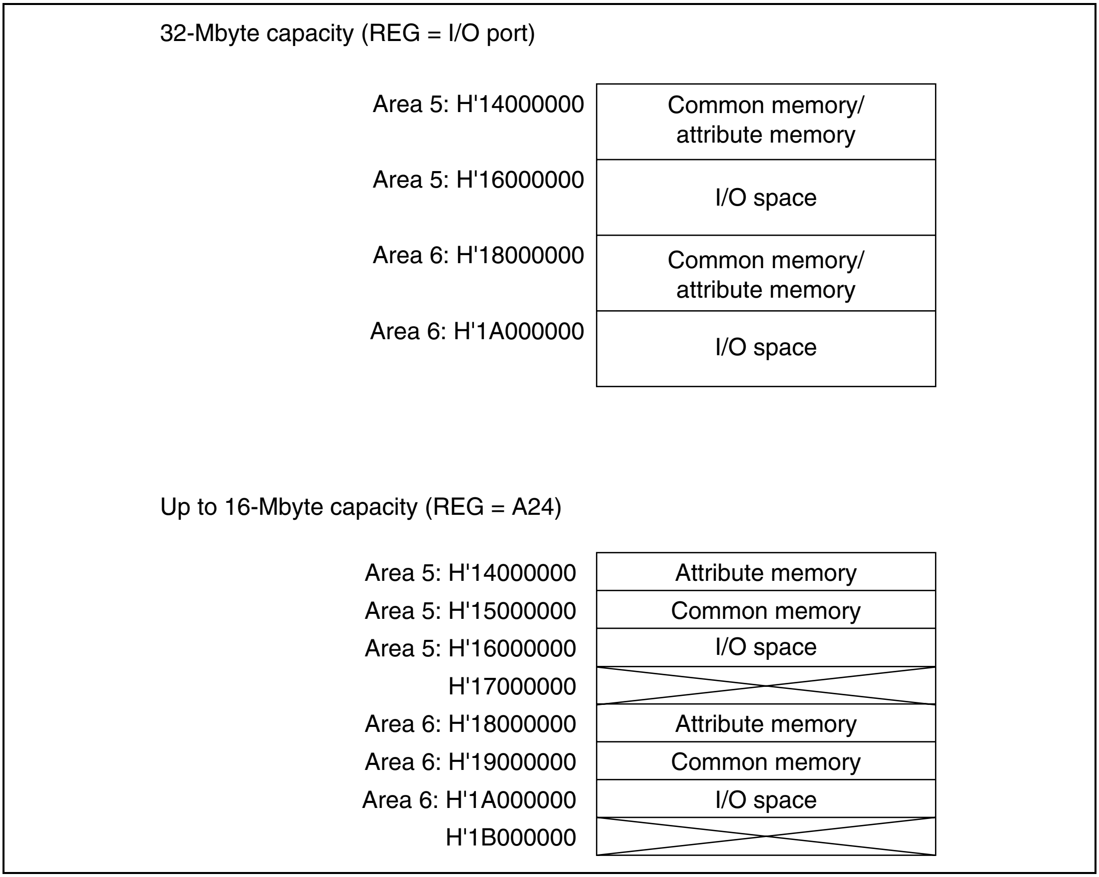


Figure 8.30 PCMCIA Space Allocation

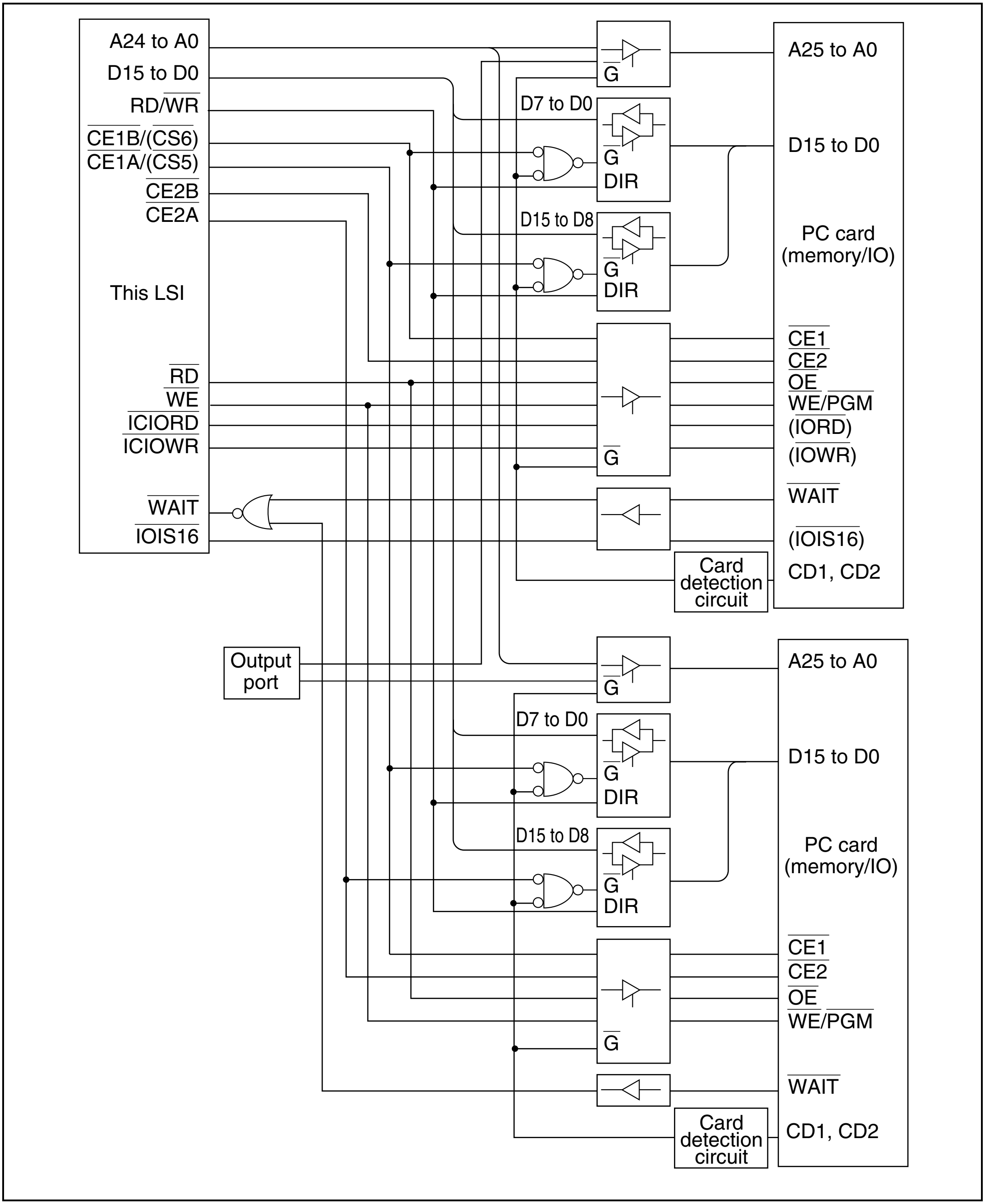


Figure 8.31 Example of PCMCIA Interface

Memory Card Interface Basic Timing: Figure 8.32 shows the basic timing for the PCMCIA IC memory card interface. When physical space areas 5 and 6 are designated as PCMCIA interface areas, bus accesses are automatically performed as IC memory card interface accesses.

With a high external bus frequency (CKIO), the setup and hold times for the address (A24–A0), card enable ($\overline{\text{CS5}}$, $\overline{\text{CE2A}}$, $\overline{\text{CS6}}$, $\overline{\text{CE2B}}$), and write data (D15–D0) in a write cycle, become insufficient with respect to $\overline{\text{RD}}$ and $\overline{\text{WR}}$ (the $\overline{\text{WE}}$ pin in this LSI). This LSI provides for this by enabling setup and hold times to be set for physical space areas 5 and 6 in the PCR register. Also, software waits by means of a WCR2 register setting and hardware waits by means of the $\overline{\text{WAIT}}$ pin can be inserted in the same way as for the basic interface. Figure 8.33 shows the PCMCIA memory bus wait timing.

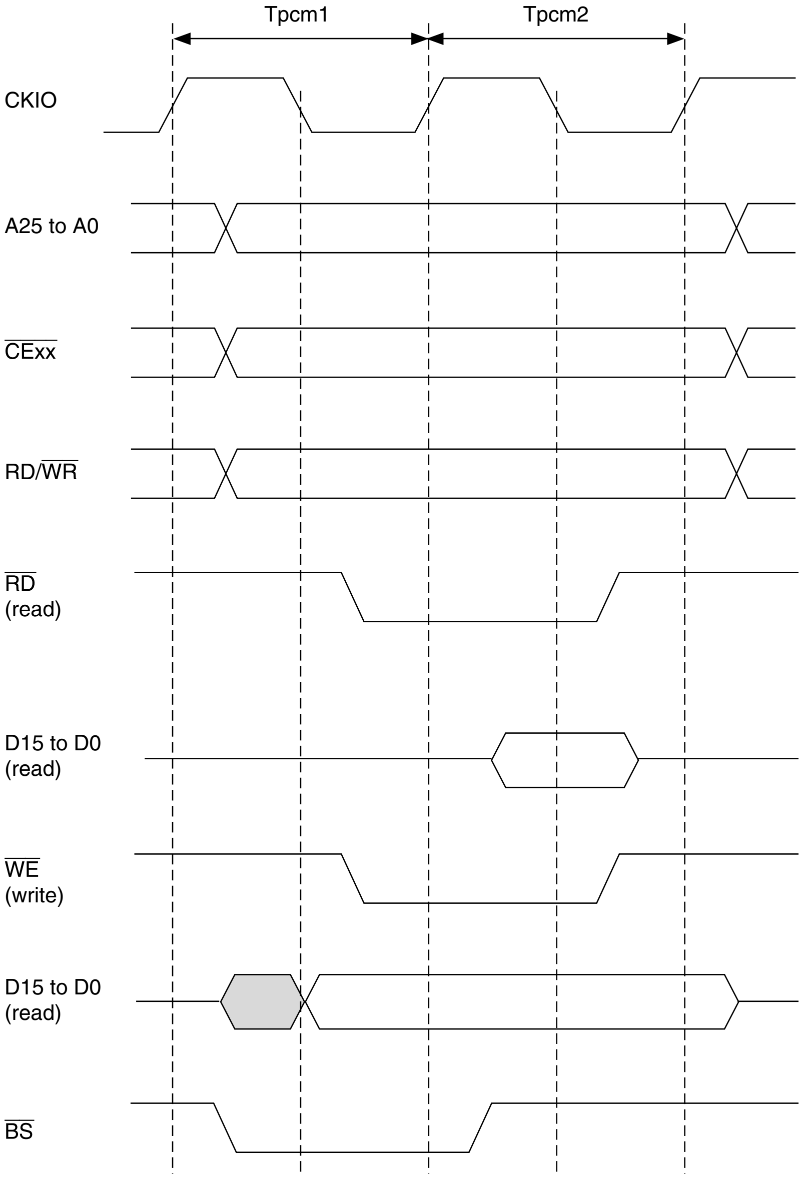


Figure 8.32 Basic Timing for PCMCIA Memory Card Interface

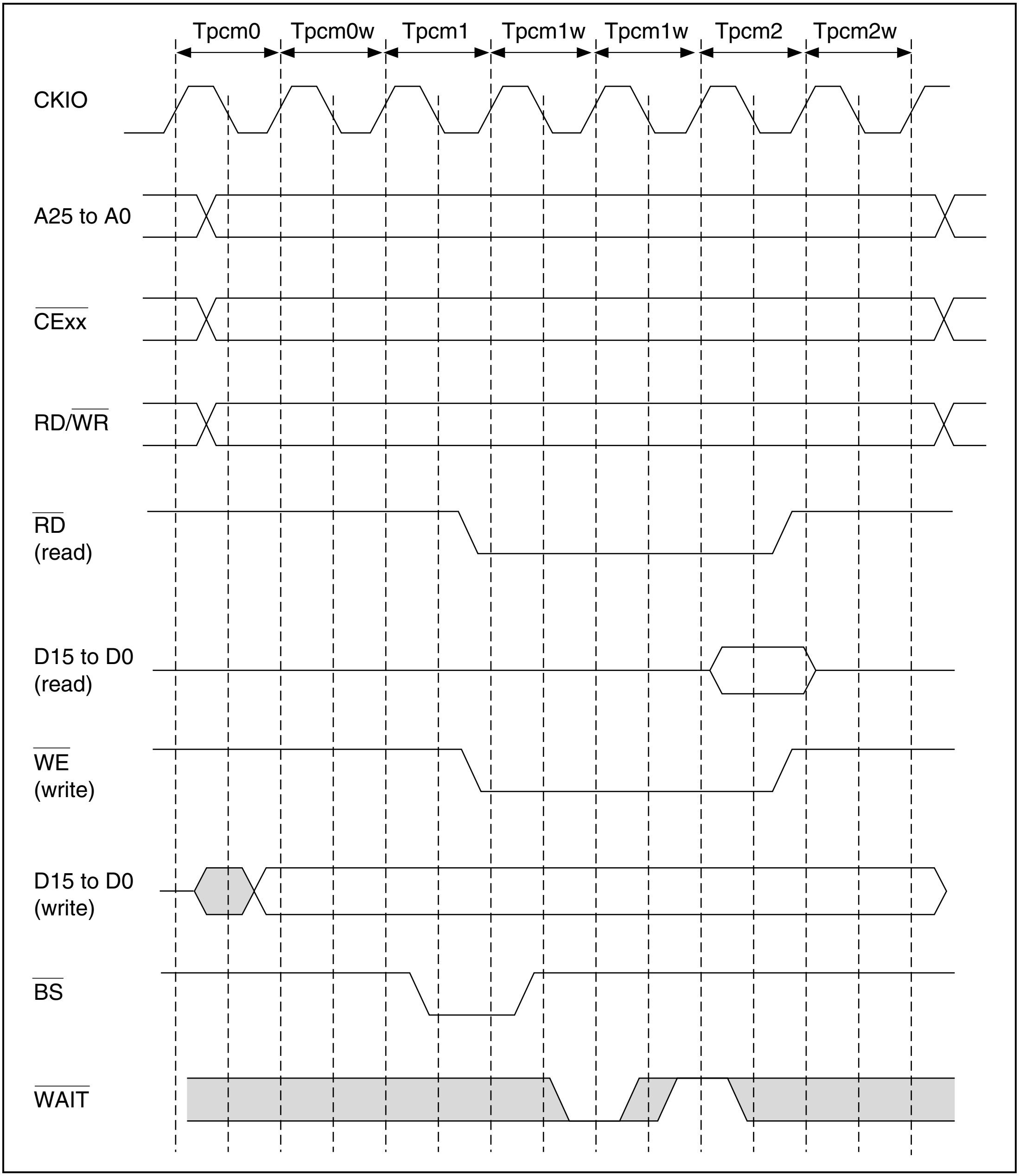


Figure 8.33 Wait Timing for PCMCIA Memory Card Interface

Memory Card Interface Burst Timing: In this LSI, when the IC memory card interface is selected, page mode burst access mode can be used, for read access only, by setting bits A5BST1 and A5BST0 in BCR1 for physical space area 5, or bits A6BST1 and A6BST0 in BCR1 for area 6. This burst access mode is not stipulated in JEIDA version 4.2 (PCMCIA2.1), but allows high-speed data access using ROM provided with a burst mode, etc.

Burst access mode timing is shown in figures 8.34 and 8.35.

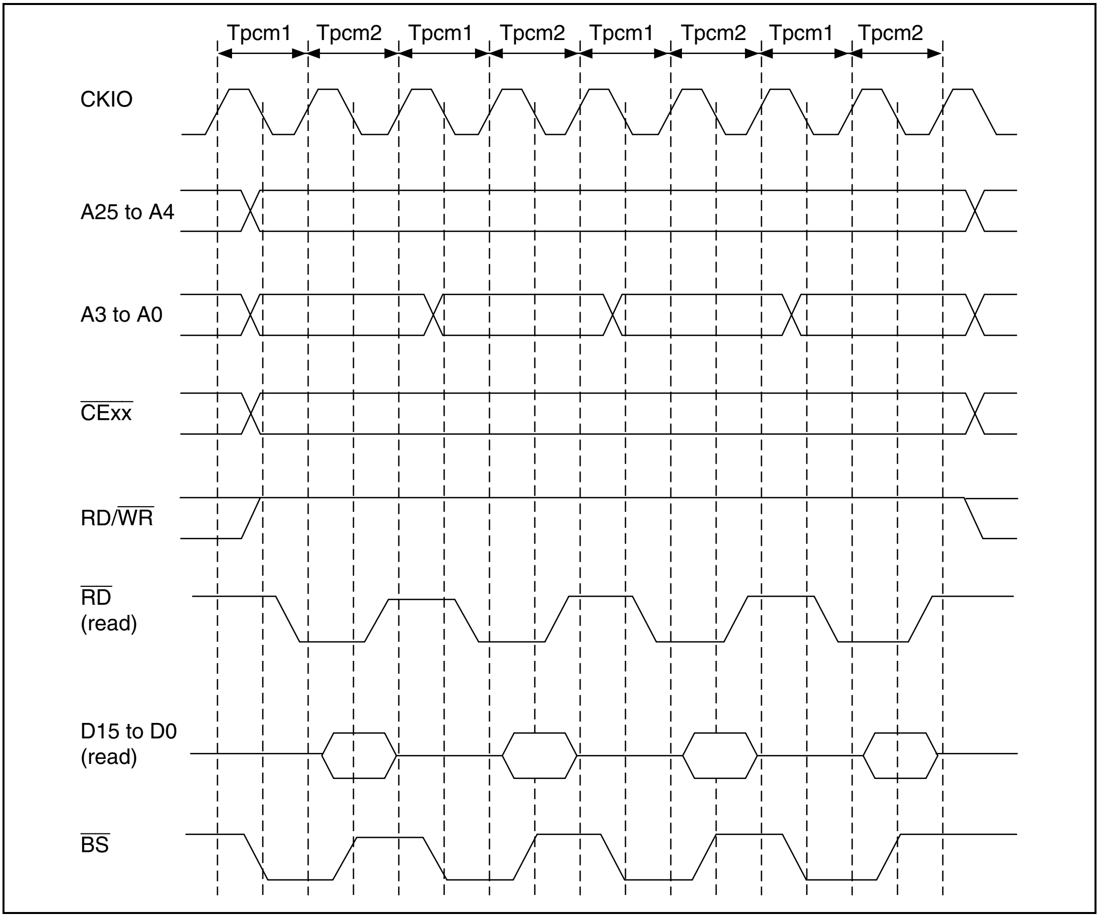


Figure 8.34 Basic Timing for PCMCIA Memory Card Interface Burst Access

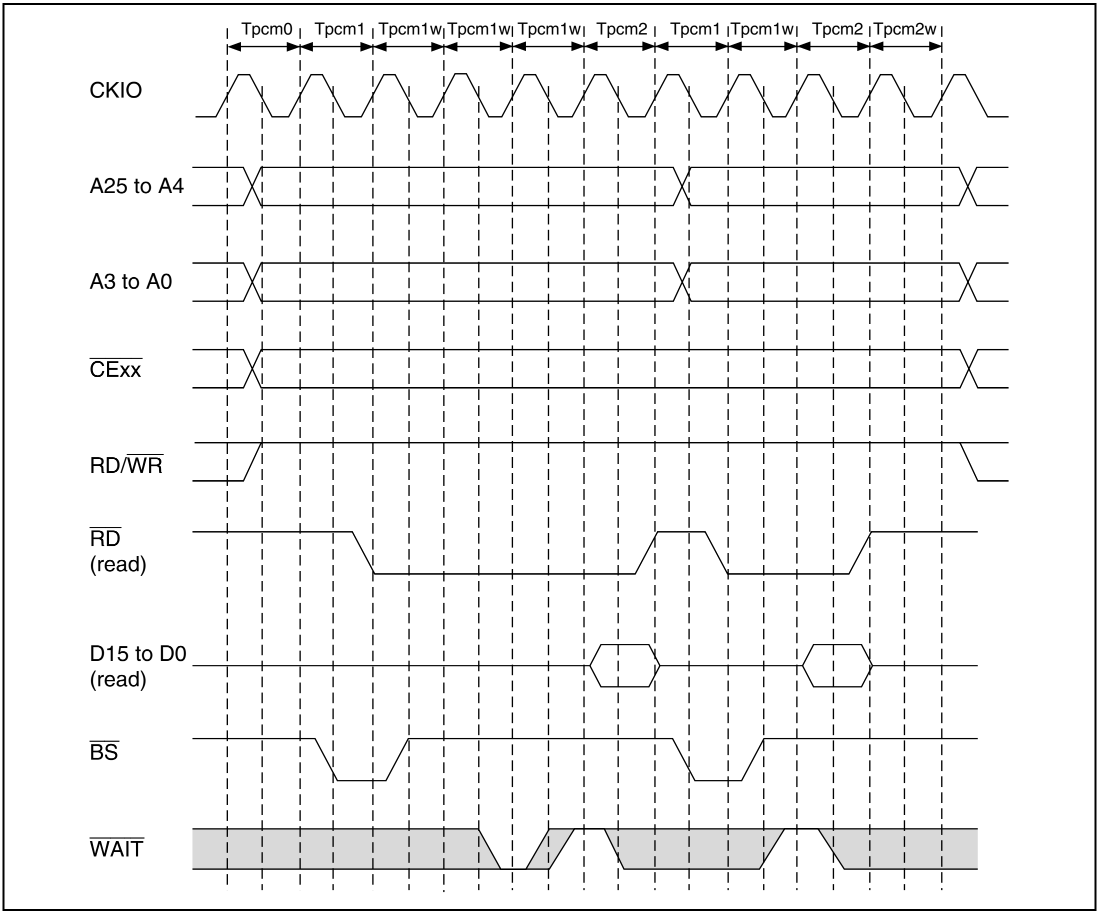


Figure 8.35 Wait Timing for PCMCIA Memory Card Interface Burst Access

When the entire 32-Mbyte memory space is used as IC memory card interface space, the common memory/attribute memory switching signal $\overline{\text{REG}}$ is generated using a port, etc. If 16-Mbytes or less of memory space is sufficient, using 16 Mbytes of memory space as common memory space and 16 Mbytes as attribute memory space enables the A24 pin to be used for the $\overline{\text{REG}}$ signal.

I/O Card Interface Timing: Figures 8.36 and 8.37 show the timing for the PCMCIA I/O card interface.

Switching between the I/O card interface and the IC memory card interface is performed according to the accessed address. When PCMCIA is designed for physical space area 5, the bus access is automatically performed as an I/O card interface access when a physical address from H'16000000 to H'17FFFFFF is accessed. When PCMCIA is designated for physical space area 6, the bus access is automatically performed as an I/O card interface access when a physical address from H'1A000000 to H'1BFFFFFF is accessed.

When accessing a PCMCIA I/O card, the access should be performed using a non-cacheable area in virtual space (P2 or P3 space) or an area specified as non-cacheable by the MMU.

When an I/O card interface access is made to a PCMCIA card in little-endian mode, dynamic sizing of the I/O bus width is possible using the $\overline{\text{IOIS16}}$ pin. When a 16-bit bus width is set for area 6, if the $\overline{\text{IOIS16}}$ signal is high during a word-size I/O bus cycle, the I/O port is recognized as being 8 bits in width. In this case, a data access for only 8 bits is performed in the I/O bus cycle being executed, followed automatically by a data access for the remaining 8 bits.

Figure 8.38 shows the basic timing for dynamic bus sizing.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal is not supported.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal should be fixed low.

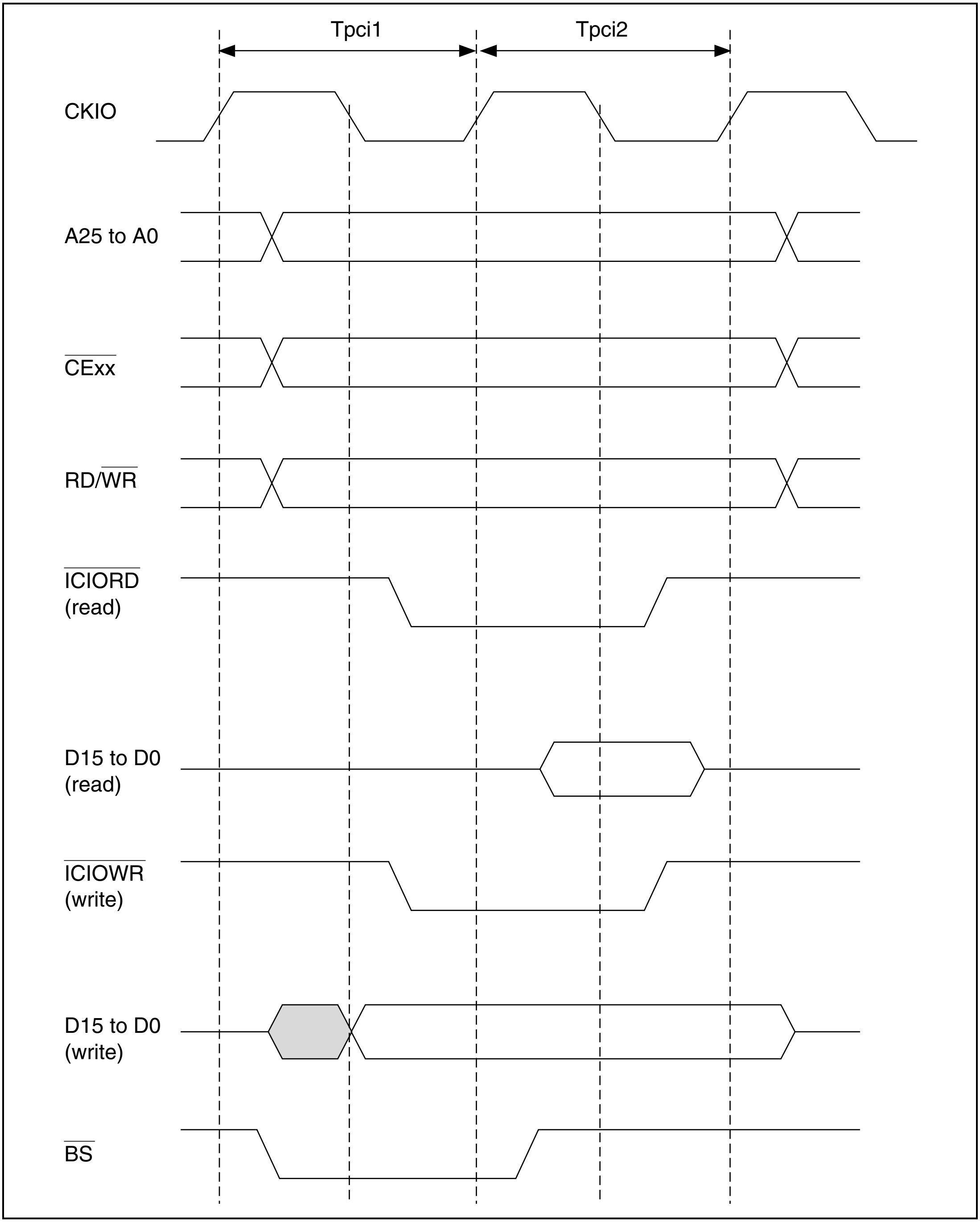


Figure 8.36 Basic Timing for PCMCIA I/O Card Interface

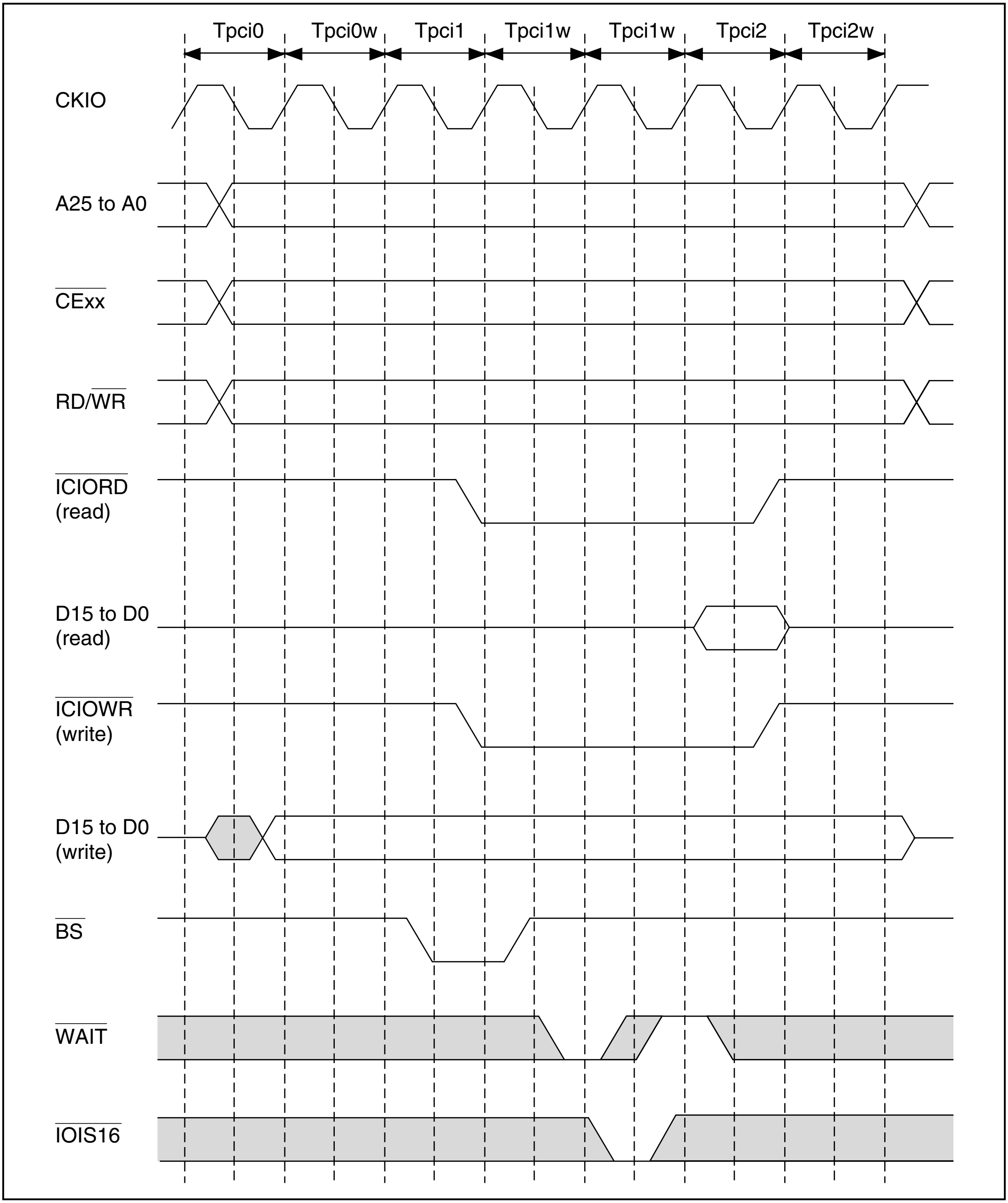


Figure 8.37 Wait Timing for PCMCIA I/O Card Interface

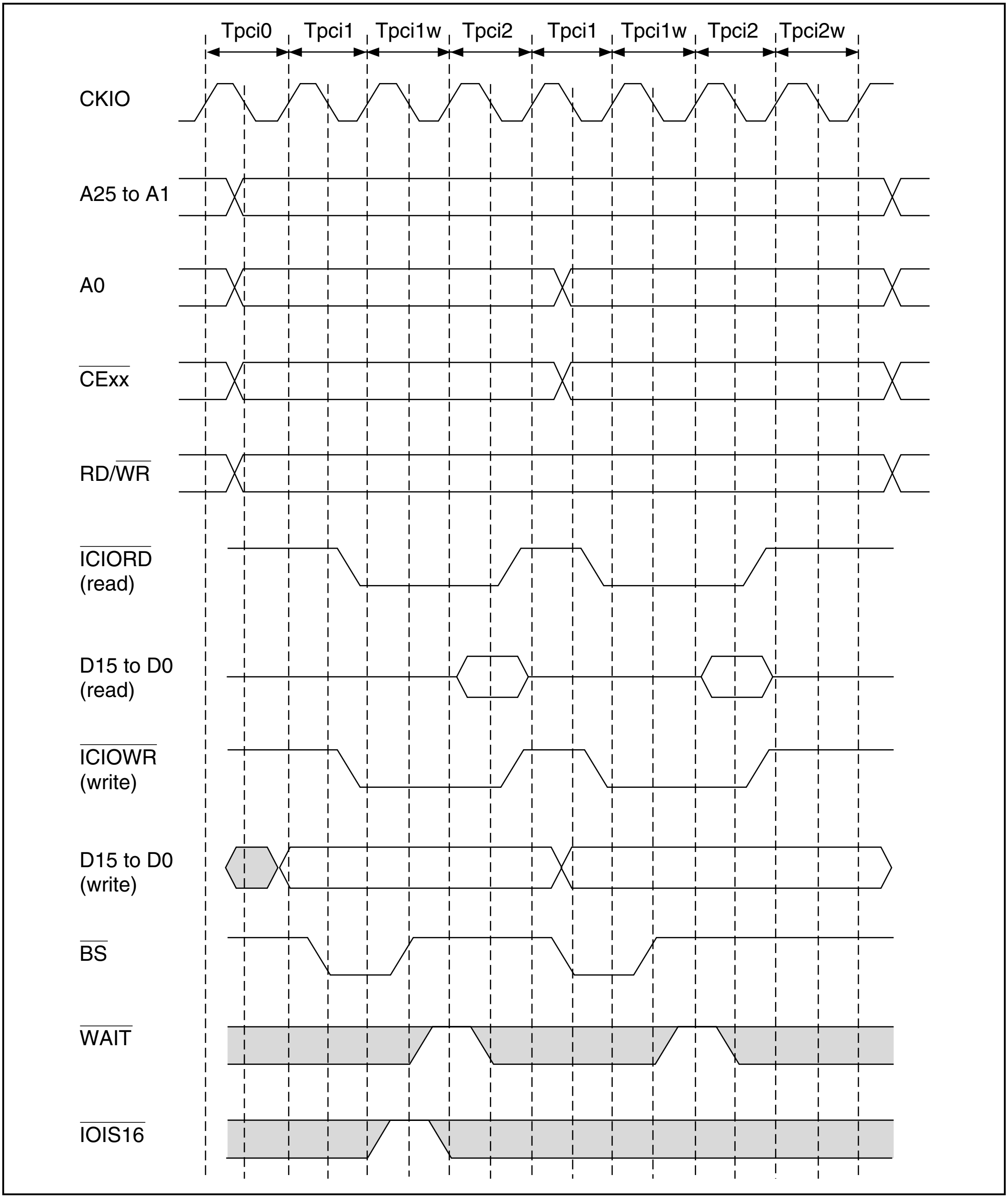


Figure 8.38 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

8.5.7 Waits between Access Cycles

A problem associated with higher external memory bus operating frequencies is that data buffer turn-off on completion of a read from a low-speed device may be too slow, causing a collision with data in the next access. This results in lower reliability or incorrect operation. To avoid this problem, a data collision prevention feature has been provided. This memorizes the preceding access area and the kind of read/write. If there is a possibility of a bus collision when the next access is started, a wait cycle is inserted before the access cycle thus preventing a data collision. There are two cases in which a wait cycle is inserted: when an access is followed by an access to a different area, and when a read access is followed by a write access from this LSI. When this LSI performs consecutive write cycles, the data transfer direction is fixed (from this LSI to other memory) and there is no problem. With read accesses to the same area, in principle, data is output from the same data buffer, and wait cycle insertion is not performed. Bits AnIW1 and AnIW0 (n = 0, 2–6) in WCR1 specify the number of idle cycles to be inserted between access cycles when a physical space area access is followed by an access to another area, or when this LSI performs a write access after a read access to physical space area n. If there is originally space between accesses, the number of idle cycles inserted is the specified number of idle cycles minus the number of empty cycles.

Waits are not inserted between accesses when bus arbitration is performed, since empty cycles are inserted for arbitration purposes.

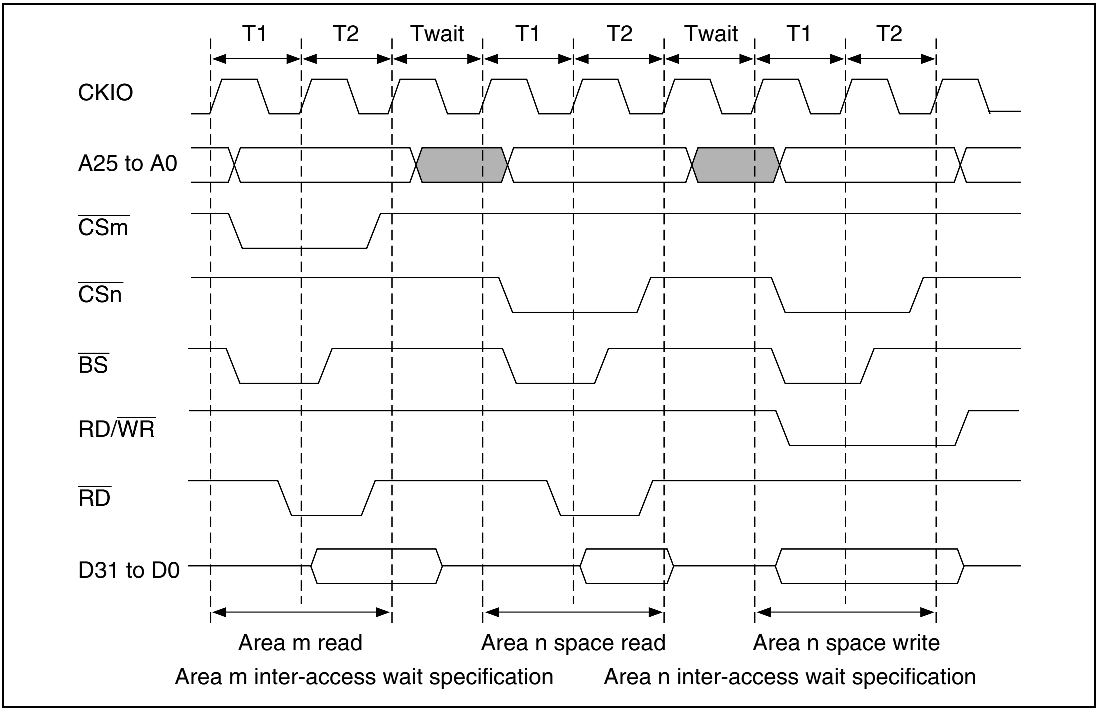


Figure 8.39 Waits between Access Cycles

8.5.8 Bus Arbitration

When a bus release request ($\overline{\text{BREQ}}$) is received from an external device, buses are released after the bus cycle being executed is completed and a bus grant signal ($\overline{\text{BACK}}$) is output. The bus is not released during burst transfers for cache fills or a write back and TAS instruction execution between the read cycle and write cycle. Bus arbitration is not executed in multiple bus cycles that are generated when the data bus width is shorter than the access size; i.e. in the bus cycles when longword access is executed for the 8-bit memory. At the negation of $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$ is negated and bus use is restarted. See Appendix A, Pin States, for the pin state when the bus is released.

This LSI sometimes needs to retrieve a bus it has released. For example, when memory generates a refresh request or an interrupt request internally, this LSI must perform the appropriate processing. This LSI has a bus request signal ($\overline{\text{IRQOUT}}$) for this purpose. When it must retrieve the bus, it asserts the $\overline{\text{IRQOUT}}$ signal. Devices asserting an external bus release request receive the assertion of the $\overline{\text{IRQOUT}}$ signal and negate the $\overline{\text{BREQ}}$ signal to release the bus. This LSI retrieves the bus and carries out the processing.

$\overline{\text{IRQOUT}}$ Pin Assertion Conditions:

- When a memory refresh request has been generated but the refresh cycle has not yet begun
- When an interrupt is generated with an interrupt request level higher than the setting of the interrupt mask bits (I3–I0) in the status register (SR). (This does not depend on the SR.BL bit.)

8.5.9 Bus Pull-Up

With this LSI, address pin pull-up can be performed when the bus is released by setting the PULA bit in BCR1 to 1. The address pins are pulled up for a 4-clock period after $\overline{\text{BACK}}$ is asserted. Figure 8.40 shows the address pin pull-up timing. Similarly, data pin pull-up can be performed by setting the PULD bit in BCR1 to 1. The data pins should be pulled up when the data bus is not in use. The data pin pull-up timing for a read cycle is shown in figure 8.41, and the timing for a write cycle in figure 8.42.

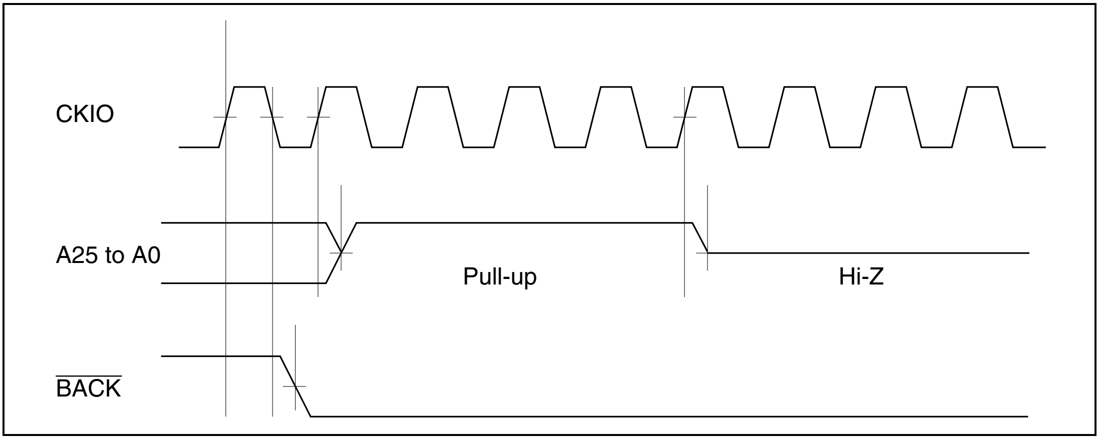


Figure 8.40 Pins A25 to A0 Pull-Up Timing

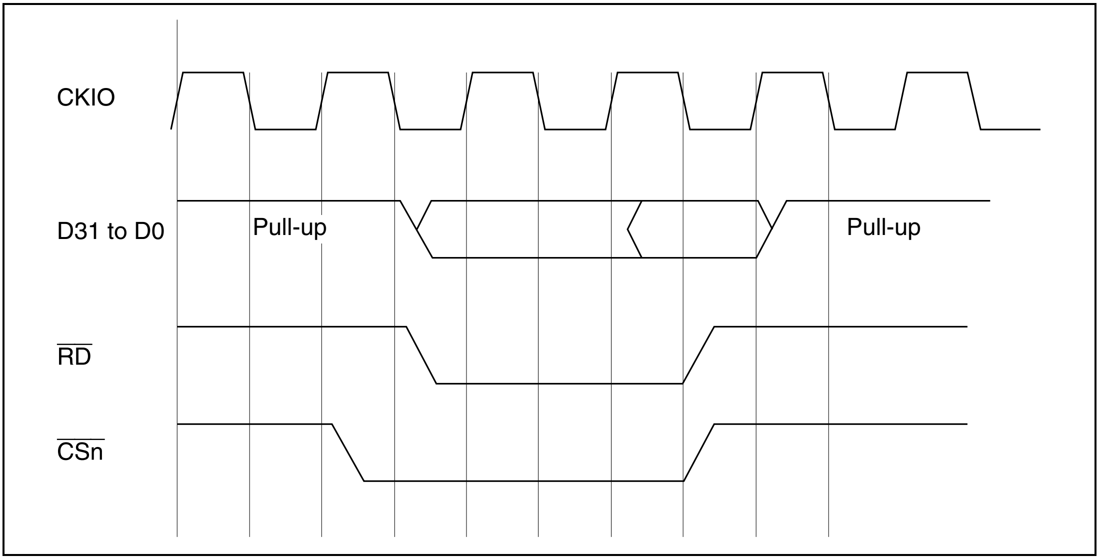


Figure 8.41 Pins D31 to D0 Pull-Up Timing (Read Cycle)

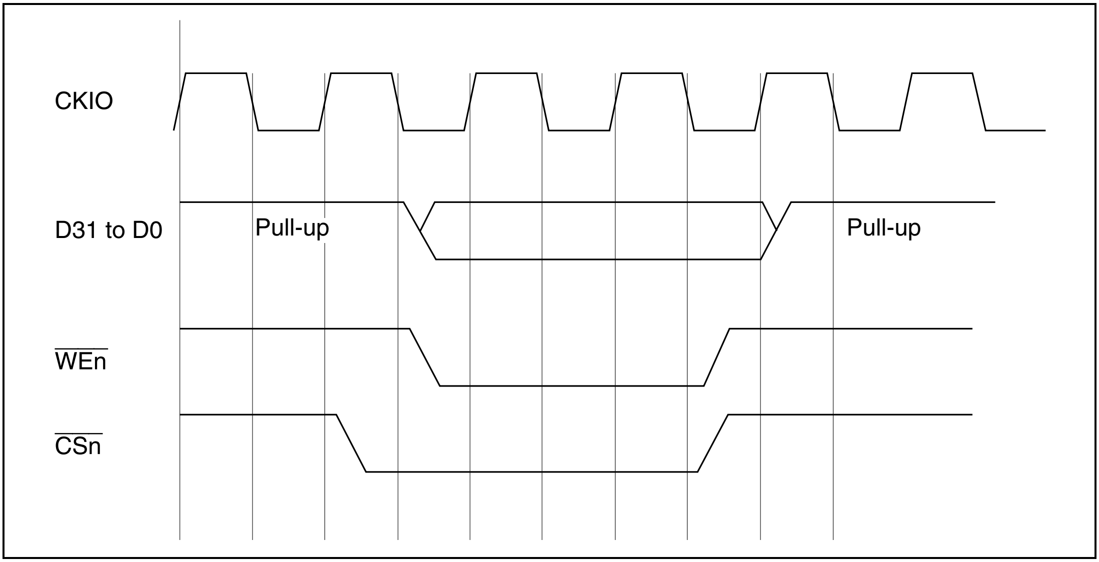


Figure 8.42 Pins D31 to D0 Pull-Up Timing (Write Cycle)

Section 9 Direct Memory Access Controller (DMAC)

This chip includes a four-channel direct memory access controller (DMAC). The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, memory-mapped external devices, and on-chip peripheral modules (SCIF, A/D converter, and D/A converter). Using the DMAC reduces the burden on the CPU and increases overall operating efficiency.

Figure 9.1 shows a block diagram of the DMAC.

9.1 Features

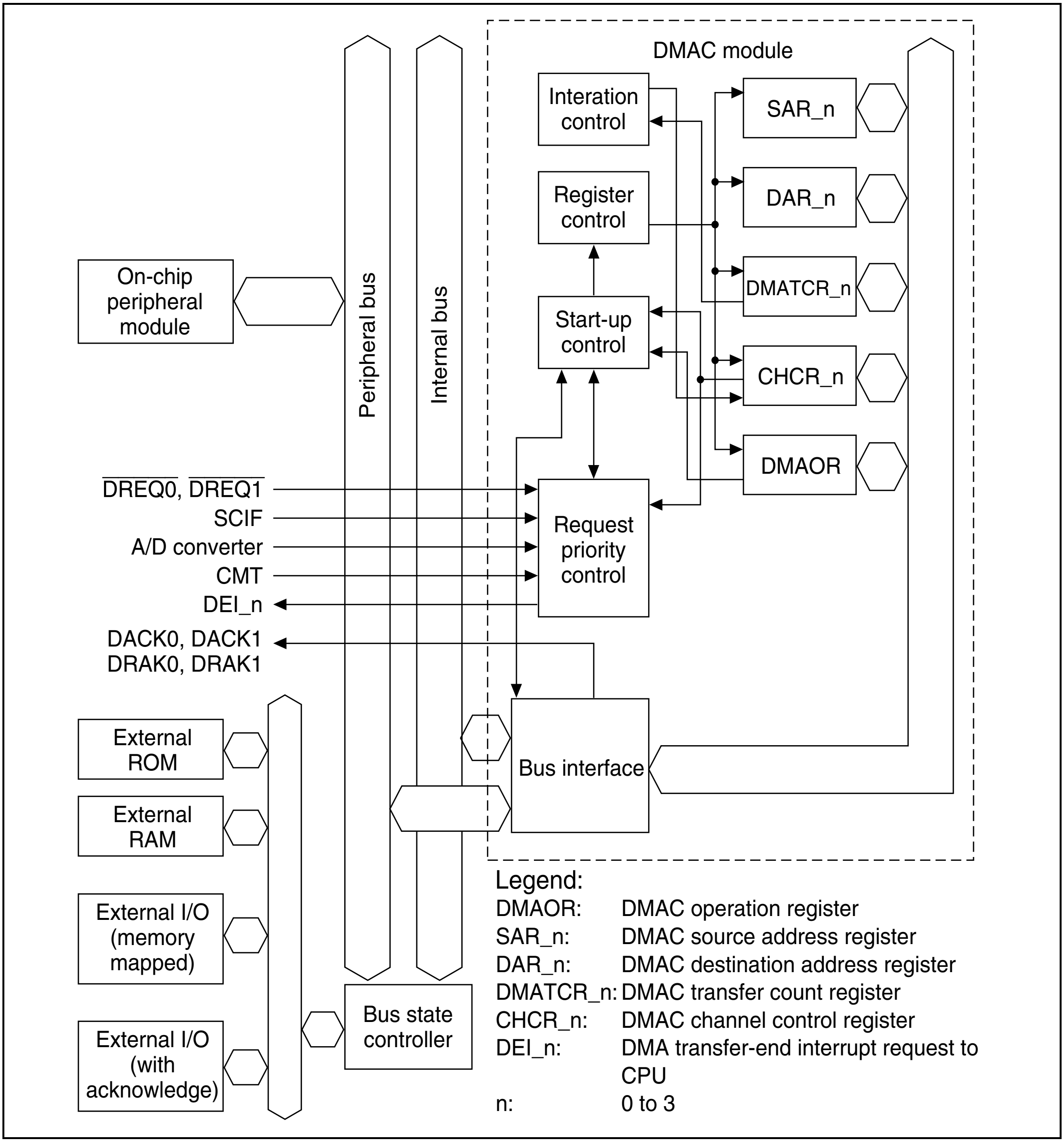
The DMAC has the following features.

- Four channels
- Address space: Architecturally 4-Gbytes
- 8-bit, 16-bit, 32-bit, or 16-byte transfer (In 16-byte transfer, four 32-bit reads are executed, followed by four 32-bit writes.)
- Maximum transfer counter: 16 Mbytes (16777216 transfers)
- Supports dual address mode
  - Direct address transfer mode: The values specified in the DMAC registers indicates the transfer source and transfer destination. Two bus cycles are required for one data transfer.
  - Indirect address transfer mode: Data is transferred with the address stored prior to the address specified in the transfer source address in the DMAC. Other operations are the same as those of direct address transfer mode. This function is only valid in channel 3. Four bus cycles are required for one data transfer.
- Supports single address mode:
  - Either the transfer source or transfer destination peripheral device is accessed (selected) by means of the DACK signal, and the other device is accessed by address. One bus cycle is required for one data transfer.
- Channel functions: Transfer mode that can be specified is different in each channel.
  - Channel 0: External request can be accepted.
  - Channel 1: External request can be accepted.
  - Channel 2: This channel has a source address reload function, which reloads a source address for each 4 transfers.
  - Channel 3: In this channel, direct address transfer mode or indirect address transfer mode can be specified.
- Reload function: The value that was specified in the source address register can be automatically reloaded every 4 DMA transfers. This function is only valid in channel 2.
- Three typo of Transfer requests

  - External request: From two $\overline{\text{DREQ}}$ pins (channels 0 and 1 only). $\overline{\text{DREQ}}$ can be detected either by the folling edge or by the low level.
  - On-chip module request: Requests from on-chip peripheral modules such as serial communications interface (SCIF), A/D converter (A/D), and a timer (CMT). This request can be accepted in all the channels.
  - Auto request: The transfer request is generated automatically within the DMAC.
- Selectable bus modes: Cycle-steal mode or burst mode
- Selectable channel priority levels:

  Fixed mode: The channel priority is fixed.

  Round-robin mode: The priority of the channel in which the execution request was accepted is made the lowest.
- Interrupt request: An interrupt request can be generated to the CPU after transfers end by the specified counts.



Legend:

DMAOR:	DMAC operation register
SAR_n:	DMAC source address register
DAR_n:	DMAC destination address register
DMATCR_n:	DMAC transfer count register
CHCR_n:	DMAC channel control register
DEI_n:	DMA transfer-end interrupt request to CPU
n:	0 to 3

Figure 9.1 DMAC Block Diagram

9.2 I/O Pins

Table 9.1 shows the DMAC pins.

Table 9.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	DMA transfer request	$\overline{\text{DREQ0}}$	I	DMA transfer request input from external device to channel 0
	DREQ acknowledge	DACK0	O	Strobe output to an external I/O at DMA transfer request from external device to channel 0
	DMA request acknowledge	DRAK0	O	Output showing that $\overline{\text{DREQ0}}$ has been accepted
1	DMA transfer request	$\overline{\text{DREQ1}}$	I	DMA transfer request input from external device to channel 1
	DREQ acknowledge	DACK1	O	Strobe output to an external I/O at DMA transfer request from external device to channel 1
	DMA request acknowledge	DRAK1	O	Output showing that $\overline{\text{DREQ1}}$ has been accepted

9.3 Description of Registers

DMAC has a total of 17 registers. Each channel has four control registers. One other control register is shared by all channels.

Refer to section 23, Control Registers Table, for more detail of the addresses and access size.

Channel 0

- DMA source address register 0 (SAR0)
- DMA destination address register 0 (DAR0)
- DMA transfer count register 0 (DMATCR0)
- DMA channel control register 0 (CHCR0)

Channel 1

- DMA source address register 1 (SAR1)
- DMA destination address register 1 (DAR1)
- DMA transfer count register 1 (DMATCR1)
- DMA channel control register 1 (CHCR1)

Channel 2

- DMA source address register 2 (SAR2)

- DMA destination address register 2 (DAR2)
- DMA transfer count register 2 (DMATCR2)
- DMA channel control register 2 (CHCR2)

Channel 3

- DMA source address register 3 (SAR3)
- DMA destination address register 3 (DAR3)
- DMA transfer count register 3 (DMATCR3)
- DMA channel control register 3 (CHCR3)
- DMA operation register (DMAOR)

9.3.1 DMA Source Address Registers 0–3 (SAR_0–SAR_3)

DMA source address registers 0–3 (SAR_0–SAR_3) are 32-bit read/write registers that specify the source address of a DMA transfer. These registers include count functions, and during a DMA transfer, these registers indicate the next source address.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address boundary. When transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set for the source address value. Specifying other addresses does not guarantee operation.

The initial value is undefined by resets. The previous value is held in standby mode.

When accessed in 16 bits, the other 16-bit data which has not been accessed is held.

9.3.2 DMA Destination Address Registers 0–3 (DAR_0–DAR_3)

DMA destination address registers 0–3 (DAR_0–DAR_3) are 32-bit read/write registers that specify the destination address of a DMA transfer. These registers include count functions, and during a DMA transfer, these registers indicate the next destination address.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address boundary. Specifying other addresses does not guarantee operation.

The initial value is undefined by resets. The previous value is held in standby mode.

When accessed in 16 bits, the other 16-bit data which has not been accessed is held.

9.3.3 DMA Transfer Count Registers 0–3 (DMATCR_0–DMATCR_3)

DMA transfer count registers 0–3 (DMATCR_0–DMATCR_3) are 24-bit read/write registers that specify the DMA transfer count (bytes, words, or longwords) in each channel. The number of transfers is 1 when the setting is H'000001, and 16777216 (the maximum) when H'000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one.

Upper eight bits in DMATCR are reserved. These bits are always read as 0. The write value should always be 0.

When using 16-byte transfer, an integral multiple of 4 (4n) must be set for the number of transfers to ensure normal operation.

The initial value is undefined by resets. The previous value is held in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	—	—	R/W	24- bit register

9.3.4 DMA Channel Control Registers 0–3 (CHCR_0–CHCR_3)

DMA channel control registers 0–3 (CHCR_0–CHCR_3) are 32-bit read/write registers that specifies operation mode, transfer method, or others in each channel.

These register values are initialized to 0s by resets. The previous value is held in standby mode.

When accessed in 16 bits, the other 16-bit data which has not been accessed is held.

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	DI	0	(R/W)*[2]	Direct/Indirect Selection DI selects direct address mode or indirect address mode in channel 3. This bit is only valid in CHCR_3 and is not used in CHCR_0 to CHCR_2. Writing to this bit is invalid in CHCR_0 to CHCR_2; 0 is read if this bit is read. When using 16-byte transfer, direct address mode must be specified. Operation is not guaranteed if indirect address mode is specified. 0: Direct address mode 1: Indirect address mode

Bit	Bit Name	Initial Value	R/W	Description
19	RO	0	(R/W)*[2]	Source Address Reload RO selects whether the source address initial value is reloaded in channel 2. This bit is only valid in CHCR_2 and is not used in CHCR_0 to CHCR_1, or CHCR_3. Writing to this bit is invalid in CHCR_0, CHCR_1, and CHCR_3; 0 is read if this bit is read. When using 16-byte transfer, this bit must be cleared to 0, specifying non-reloading. Operation is not guaranteed if reloading is specified. 0: A source address is not reloaded 1: A source address is reloaded
18	RL	0	(R/W)*[2]	Request Check Level RL specifies the DRAK (acknowledge of $\overline{\text{DREQ}}$) signal output is high active or low active. This bit is only valid in CHCR_0 and CHCR_1. Writing to this bit is invalid in CHCR_2 and CHCR_3; 0 is read if this bit is read. 0: Low-active output of DRAK 1: High-active output of DRAK
17	AM	0	(R/W)*[2]	Acknowledge Mode AM specifies whether DACK is output in data read cycle or in data write cycle in dual address mode. This bit is only valid in CHCR_0 and CHCR_1. Writing to this bit is invalid in CHCR_2 and CHCR_3; 0 is read if this bit is read. 0: DACK output in read cycle 1: DACK output in write cycle
16	AL	0	(R/W)*[2]	Acknowledge Level AL specifies the DACK (acknowledge) signal output is high active or low active. This bit is only valid in CHCR_0 and CHCR_1. Writing to this bit is invalid in CHCR_2 and CHCR_3; 0 is read if this bit is read. 0: Low-active output of DACK 1: High-active output of DACK

Bit	Bit Name	Initial Value	R/W	Description
15	DM1	0	R/W	Destination Address Mode
14	DM0	0	R/W	DM1 and DM0 select whether the DMA destination address is incremented, decremented, or left fixed. 00: Fixed destination address (Initial value) 01: Destination address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer; illegal setting in 16-byte transfer) 11: Reserved (Setting prohibited)
13	SM1	0	R/W	Source Address Mode
12	SM0	0	R/W	SM1 and SM0 select whether the DMA source address is incremented, decremented, or left fixed. 00: Fixed source address 01: Source address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer) 10: Source address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer; illegal setting in 16-byte transfer) 11: Reserved (Setting prohibited) Notes: If the transfer source is specified in indirect address, specify the address, in which the data to be transferred is stored and which is stored as data (indirect address), SAR_3. Specification of SAR_3 increment or decrement in indirect address mode depends on SM1 and SM0 settings. In this case, however, the SAR_3 increment or decrement value is +4, –4, or fixed to 0 regardless of the transfer data size specified in TS1 and TS0.

Bit	Bit Name	Initial Value	R/W	Description
11	RS3	0	R/W	Resource Select
10	RS2	0	R/W	RS3–RS0 specify which transfer requests will be sent to the DMAC.
9	RS1	0	R/W	0000: External request*, dual address mode 0001: Reserved (Setting prohibited) 0010: External request / Single address mode External address space → external device with DACK 0011: External request / Single address mode External device with DACK → external address space 0100: Auto request 0101: Reserved (Setting prohibited) 0110: Reserved (Setting prohibited) 0111: Reserved (Setting prohibited) 1000: Reserved (Setting prohibited) 1001: Reserved (Setting prohibited) 1010: Reserved (Setting prohibited) 1011: Reserved (Setting prohibited) 1100: SCIF transmission 1101: SCIF reception 1110: Internal A/D 1111: CMT
8	RS0	0	R/W	Note: 1. External request specification is valid only in channels 0 and 1. None of the request sources can be selected in channels 2 and 3. 2. When using 16-byte transfer, the following settings must not be made: 1100 SCIF transmission 1101 SCIF reception 1110 A/D converter 1111 CMT Operation is not guaranteed if these settings are made.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read 0. The write value should always be 0.
6	DS	0	(R/W)*[2]	$\overline{\text{DREQ}}$ Select Bit DS selects the sampling method of the $\overline{\text{DREQ}}$ pin that is used in external request mode is detection in low level or at the falling edge. This bit is only valid in CHCR_0 and CHCR_1. Writing to this bit is invalid in CHCR_2 and CHCR_3; 0 is read if this bit is read. In channel 0 and 1, if an on-chip peripheral module is specified as a transfer request source or an auto request is specified, specification of this bit is ignored and detection at the falling edge is fixed except in an auto-request. 0: $\overline{\text{DREQ}}$ detected in low level 1: $\overline{\text{DREQ}}$ detected at falling edge
5	TM	0	R/W	Transmit Mode TM specifies the bus mode when transferring data. 0: Cycle steal mode 1: Burst mode
4	TS1	0	R/W	Transmit Size Bits 1 and 0
3	TS0	0	R/W	TS1 and TS0 specify the size of data to be transferred. 00: Byte size (8 bits) 01: Word size (16 bits) 10: Longword size (32 bits) 11: 16-byte unit (4 longword transfers)
2	IE	0	R/W	Interrupt Enable Bit Setting this bit to 1 generates an interrupt request when data transfer end (TE = 1) by the count specified in DMATCR. 0: Interrupt request is not generated even if data transfer ends by the specified count 1: Interrupt request is generated if data transfer ends by the specified count

Bit	Bit Name	Initial Value	R/W	Description
1	TE	0	R/(W)*1	Transfer End TE is set to 1 when data transfer ends by the count specified in DMATCR. At this time, if the IE bit is set to 1, an interrupt request is generated. Before this bit is set to 1, if data transfer ends due to an NMI interrupt, a DMAC address error, or clearing the DE bit or the DME bit in DMAOR, this bit is not set to 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: Data transfer does not end by the count specified in DMATCR Clear condition: Writing 0 after TE = 1 read at power-on reset or manual reset 1: Data transfer ends by the specified count
0	DE	0	R/W	DMAC Enable DE enables channel operation. 0: Disables channel operation 1: Enables channel operation Notes: If an auto request is specifies (specified in RS3 to RS0), transfer starts when this bit is set to 1. In an external request or an internal module request, transfer starts if transfer request is generated after this bit is set to 1. Clearing this bit during transfer can terminate transfer. Even if the DE bit is set, transfer is not enabled if the TE bit is 1, the DME bit in DMAOR is 0, or the NMIF bit in DMAOR is 1.

Notes: 1 Only 0 can be written to the TE bit after 1 is read.
2 DI, RO, RL, AM, AL, and DS bits are not included in some channels.

9.3.5 DMA Operation Register (DMAOR)

The DMA operation register (DMAOR) is a 16-bit read/write register that controls the DMAC transfer mode.

This register's values are initialized to 0s by resets. The previous value is held in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PR1	0	R/W	Priority Mode
8	PR0	0	R/W	PR1 and PR0 select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: CH0 > CH1 > CH2 > CH3 01: CH0 > CH2 > CH3 > CH1 10: CH2 > CH0 > CH1 > CH3 11: Round-robin
7 to 3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	AE	0	R/(W)*	Address Error Flag AE indicates that an address error occurred during DMA transfer. If this bit is set during data transfer, transfers on all channels are suspended. The CPU cannot write 1 to this bit. 0: No DMAC address error. DMA transfer is enabled. Clearing conditions: Writing AE = 0 after AE = 1 read, power-on reset, manual reset 1: DMAC address error. DMA transfer is disabled. Setting condition: This bit is set by occurrence of a DMAC address error.

Bit	Bit Name	Initial Value	R/W	Description
1	NMIF	0	R/(W)*	NMI Flag NMIF indicates that an NMI interrupt occurred. This bit is set regardless of whether DMAC is in operating or halt state. If this bit is set during data transfer, the transfer on all channels are suspended. The CPU cannot write 1 to this bit. Only 0 can be written to clear this bit after 1 is read. 0: No NMI input. DMA transfer is enabled. (Initial value) Clearing condition: Writing NMIF = 0 after NMIF = 1 read, power-on reset, manual reset 1: NMI input. DMA transfer is disabled. Setting condition: This bit is set by occurrence of an NMI interrupt.
0	DME	0	R/W	DMA Master Enable DME enables or disables DMA transfers on all channels. If the DME bit and the DE bit corresponding to each channel in CHCR are set to 1s, transfer is enabled in the corresponding channel. If this bit is cleared during transfer, transfers on all the channels can be suspended. Even if the DME bit is set, transfer is not enabled if the TE bit is 1 or the DE bit is 0 in CHCR, or the AE bit is 1 or the NMIF bit is 1 in DMAOR. 0: Disable DMA transfers on all channels 1: Enable DMA transfers on all channels

Note: * Only 0 can be written to the AE and NMIF bits after 1 is read.

9.4 Description of Operation

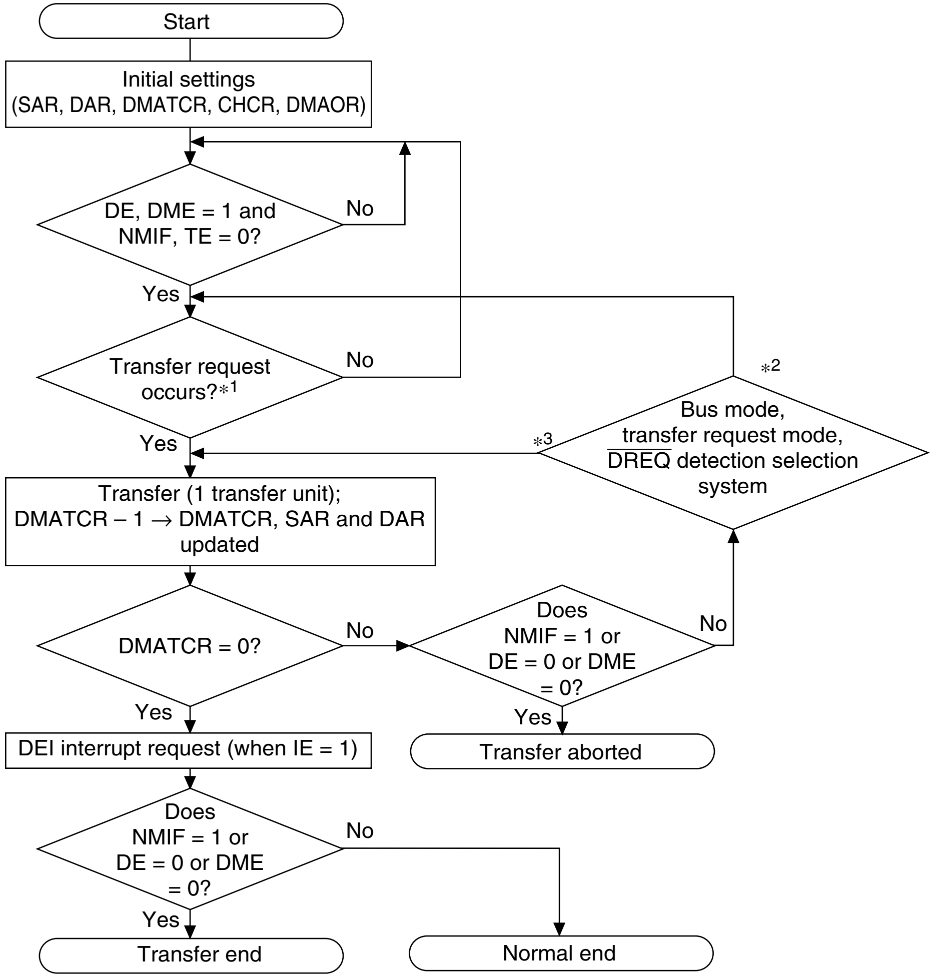
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. The dual address mode has direct address transfer mode and indirect address transfer mode. In the bus mode, the burst mode or the cycle steal mode can be selected.

9.4.1 DMA Transfer Flow

After the DMA source address registers (SAR_0 – SAR_3), DMA destination address registers (DAR_0 – DAR_3), DMA transfer count registers (DMATCR_0 – DMATCR_3), DMA channel control registers (CHCR_0 – CHCR_3), and DMA operation register (DMAOR) are set, the DMAC transfers data according to the following procedure:

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

Figure 9.2 is a flowchart of this procedure.



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

Figure 9.2 DMAC Transfer Flowchart

9.4.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated by external devices and on-chip peripheral modules that are neither the source nor the destination. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. The request mode is selected in the RS3–RS0 bits of CHCR_0–CHCR_3.

Auto-Request Mode: When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits of CHCR_0–CHCR_3 and the DME bit of the DMAOR are set to 1, the transfer begins so long as the TE bits of CHCR_0–CHCR_3 and the AE but and NMIF bit of DMAOR are all 0.

External Request Mode: In this mode a transfer is performed at the request signal ($\overline{\text{DREQ}}$) of an external device. Choose one of the modes shown in table 9.2 according to the application system. When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon a request at the $\overline{\text{DREQ}}$ input. Choose to detect $\overline{\text{DREQ}}$ by either the falling edge or low level of the signal input with the DS bit of CHCR_0–CHCR_1 (DS = 0 is level detection, DS = 1 is edge detection). The source of the transfer request does not have to be the data transfer source or destination.

Table 9.2 Selecting External Request Modes with the RS Bits

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

Note: * External memory, memory-mapped external device, on-chip memory, on-chip peripheral module (excluding DMAC, UBC, and BSC)

On-Chip Peripheral Module Request: In this mode a transfer is performed at the transfer request signal (interrupt request signal) of an on-chip peripheral module. This mode cannot be set in case of 16-byte transfer. The transfer request signals include 4 signals: the receive data full interrupts (RXI) and the transmit data empty interrupts (TXI) from serial communication interfaces (SCIF), the A/D conversion end interrupt (ADI) of the A/D converter, and the compare match timer interrupt (CMI) of the CMT. When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon the input of a transfer request signal. The source of the transfer request does not have to be the data transfer source or destination. When RXI is set as the transfer request, however, the transfer source must be the SCI's receive data register (RDR). Likewise, when TXI is set as the transfer request, the transfer source must be the SCI's transmit data register (TDR). And if the transfer requester is the A/D converter, the data transfer source must be the A/D data register (ADDR).

Table 9.3 Selecting On-Chip Peripheral Module Request Modes with the RS Bit

RS3	RS2	RS1	RS0	DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
1	0	1	0					
1	0	1	1					
1	1	0	0	SCIF transmitter	TXI2 (SCIF transmit data empty interrupt transfer request)	Any*	TDR2	Burst/ cycle steal
1	1	0	1	SCIF receiver	RXI2 (SCIF receive data full interrupt transfer request)	RDR1	Any*	Burst/ cycle steal
1	1	1	0	A/D converter	ADI (A/D conversion end interrupt)	ADDR	Any*	Burst/ cycle steal
1	1	1	1	CMT	CMI (Compare match timer interrupt)	Any*	Any*	Burst/ cycle steal

ADDR: A/D data register of A/D converter

Note: * External memory, memory-mapped external device, on-chip peripheral module (excluding DMAC, UBC , and BSC)

When outputting transfer requests from on-chip peripheral modules, the appropriate interrupt enable bits must be set to output the interrupt signals.

If the interrupt request signal of the on-chip peripheral module is used as a DMA transfer request signal, an interrupt is not generated to the CPU.

The DMA transfer request signals of table 9.3 are automatically withdrawn when the corresponding DMA transfer is performed. If the cycle-steal mode is being used, they are withdrawn at the first transfer; if the burst mode is being used, they are withdrawn at the last transfer.

9.4.3 Channel Priority

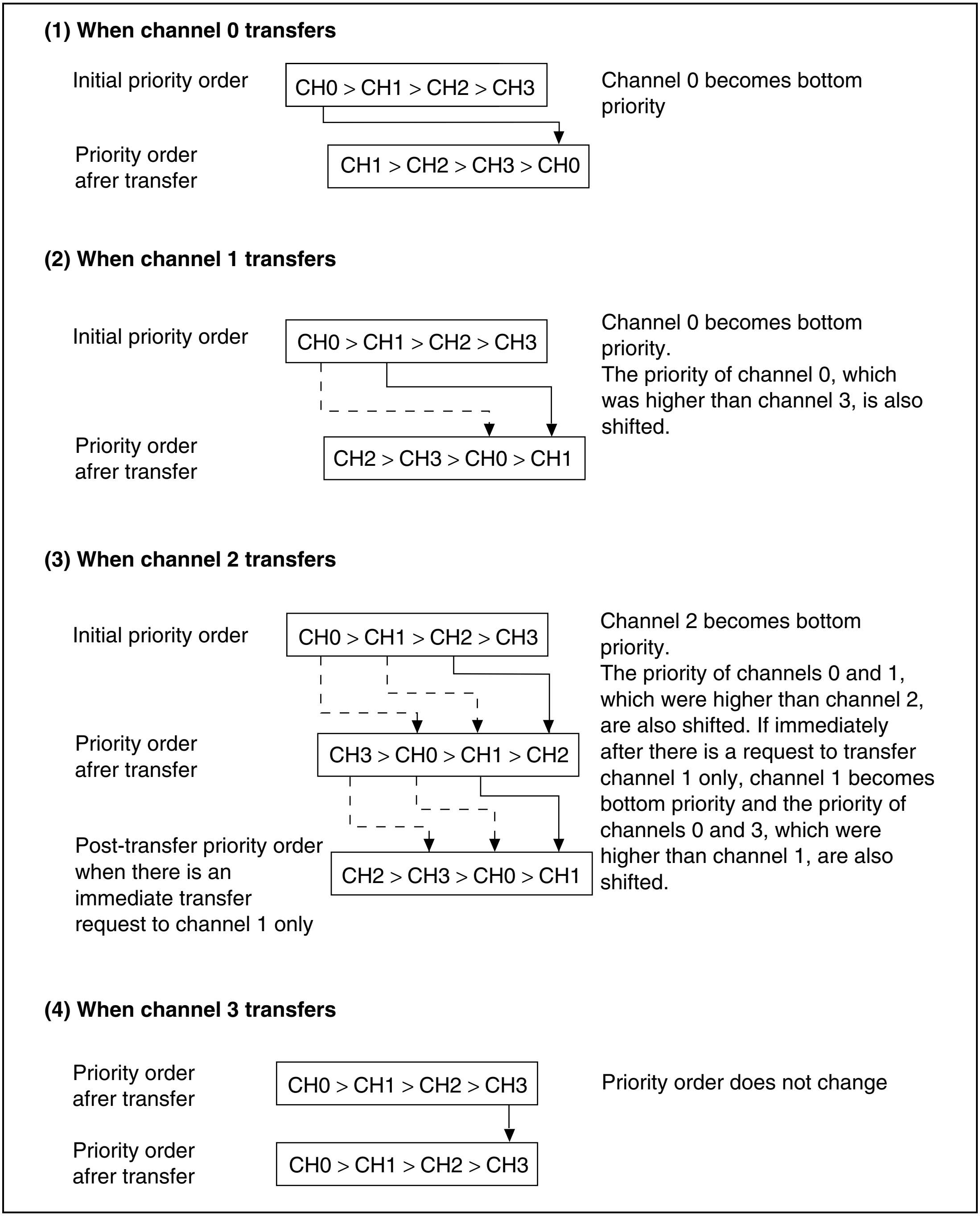
When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order. Two modes (fixed mode and round-robin mode) are selected by the priority bits PR1 and PR0 in the DMA operation register (DMAOR).

Fixed Mode: In this mode, the priority levels among the channels remain fixed. There are three kinds of fixed modes as follows:

- CH0 > CH1 > CH2 > CH3
- CH0 > CH2 > CH3 > CH1
- CH2 > CH0 > CH1 > CH3

These are selected by the PR1 and the PR0 bits in the DMA operation register (DMAOR).

Round-Robin Mode: Each time one word, byte, or longword, or 16-byte data is transferred on one channel, the priority order is rotated. The channel on which the transfer was just finished rotates to the bottom of the priority order. The round-robin mode operation is shown in figure 9.3. The priority of the round-robin mode is CH0 > CH1 > CH2 > CH3 immediately after a reset.



Channel 0 becomes bottom priority

Channel 0 becomes bottom priority.
The priority of channel 0, which was higher than channel 3, is also shifted.

Channel 2 becomes bottom priority.
The priority of channels 0 and 1, which were higher than channel 2, are also shifted. If immediately after there is a request to transfer channel 1 only, channel 1 becomes bottom priority and the priority of channels 0 and 3, which were higher than channel 1, are also shifted.

Priority order does not change

Figure 9.3 Round-Robin Mode

Figure 9.4 shows how the priority order changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 becomes lowest priority.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 becomes lowest priority.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 shift downward in priority so that channel 3 becomes the lowest priority.

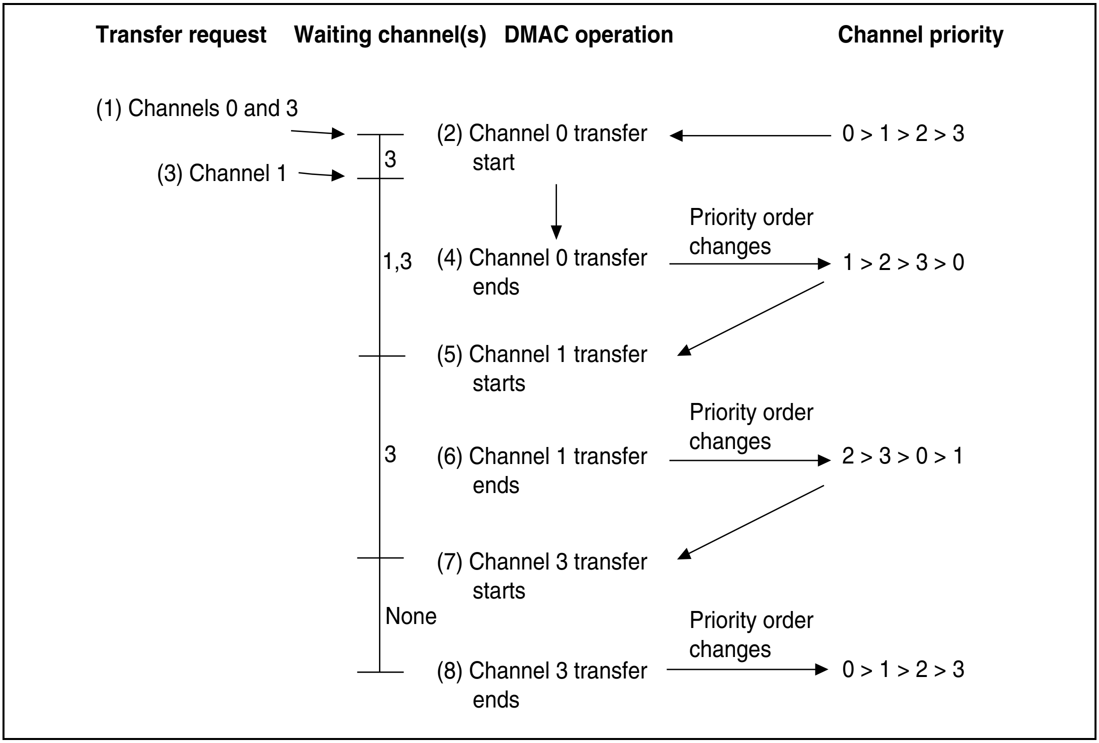


Figure 9.4 Changes in Channel Priority in Round-Robin Mode

9.4.4 DMA Transfer Types

The DMAC supports the transfers shown in table 9.4. The dual address mode has the direct address mode and the indirect address mode. In the direct address mode, an output address value is the data transfer target address; in the indirect address mode, the value stored in the output address, not the output address value itself, is the data transfer target address. A data transfer timing depends on the bus mode, which has cycle steal mode and burst mode.

Table 9.4 Supported DMA Transfers

	Destination			
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**
External device with DACK	Not available	Dual, single	Dual, single	Not available
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. The dual address mode includes the direct address mode and the indirect address mode.
4. 16-byte transfer is not available for on-chip peripheral modules.

Address Modes:

- Dual Address Mode

  In the dual address mode, both the transfer source and destination are accessed (selectable) by an address. The source and destination can be located externally or internally. The dual address mode has (1) direct address transfer mode and (2) indirect address transfer mode.

(1) In the direct address transfer mode, DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC. In the transfer between external memories as shown in figure 9.5, data is read to the DMAC from one external memory in a data read cycle, and then that data is written to the other external memory in a write cycle. Figures 9.6 to 9.8 show examples of the timing at this time.

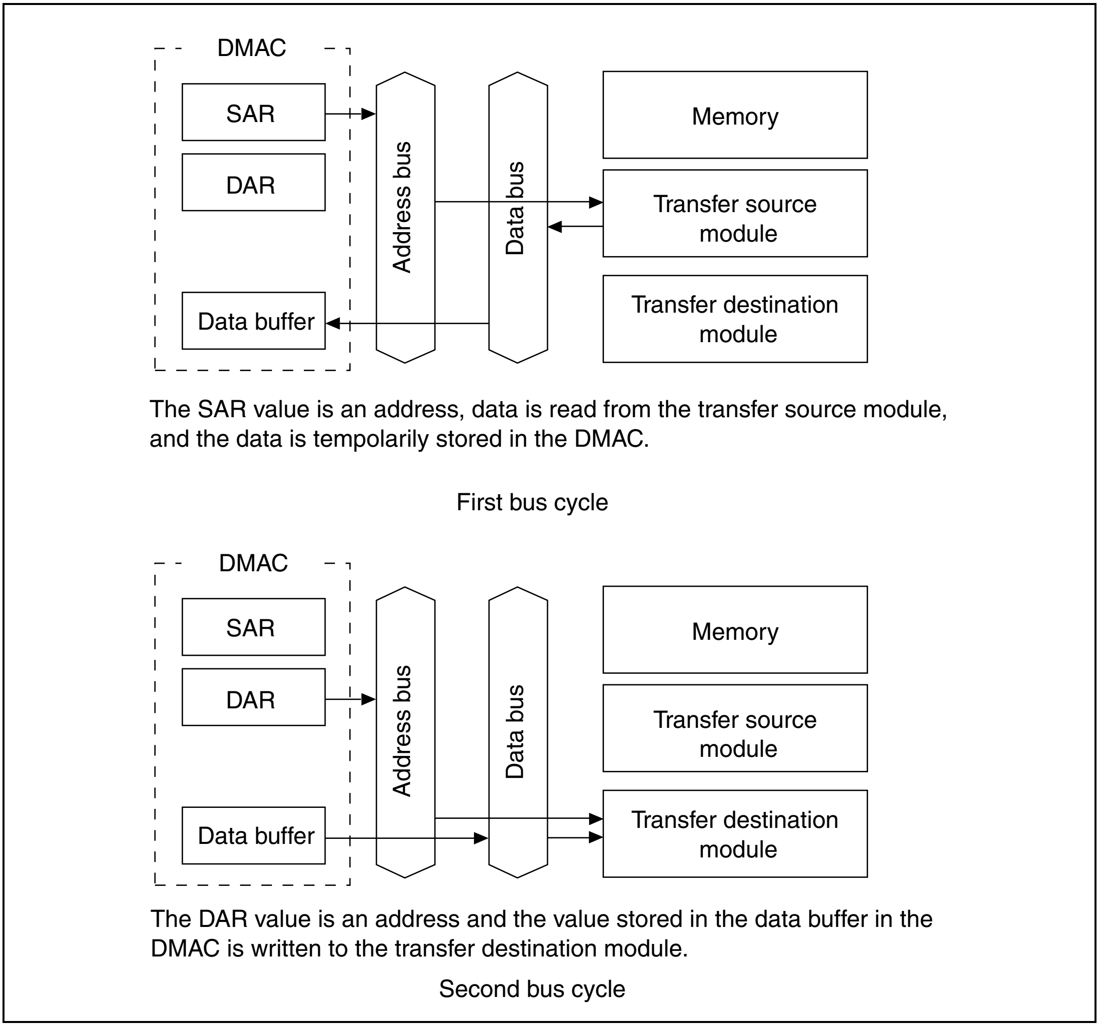
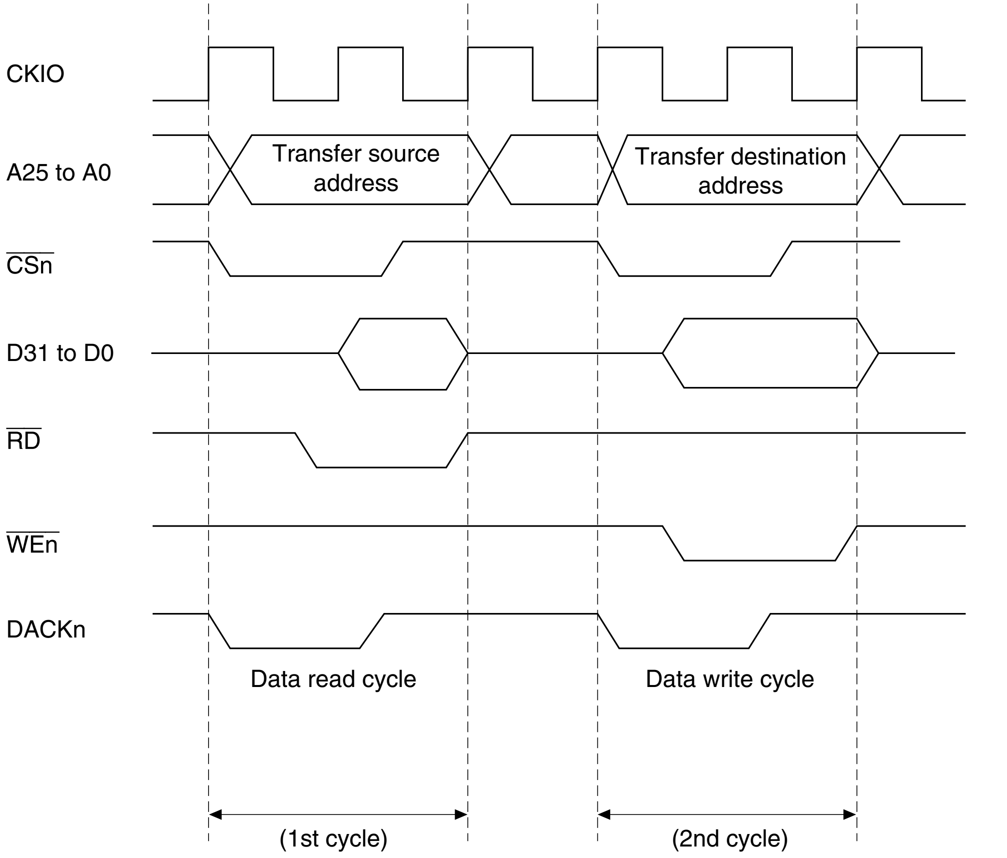
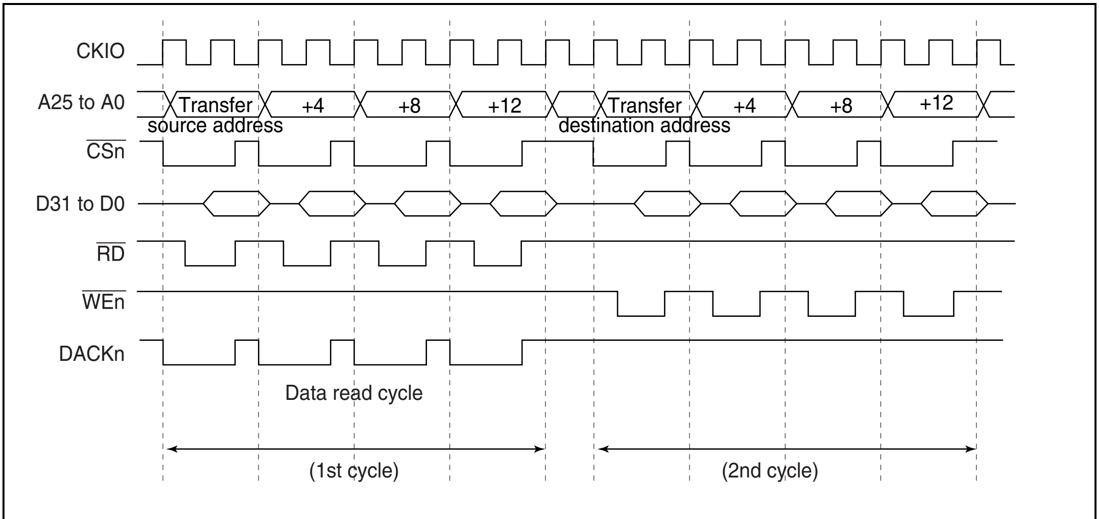

DMAC
SAR
DAR
Data buffer
Address bus
Data bus
Memory
Transfer source module
Transfer destination module
The SAR value is an address, data is read from the transfer source module, and the data is tempolarily stored in the DMAC.
First bus cycle
DMAC
SAR
DAR
Data buffer
Address bus
Data bus
Memory
Transfer source module
Transfer destination module
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 9.5 Operation in the Direct Address Mode in the Dual Address Mode



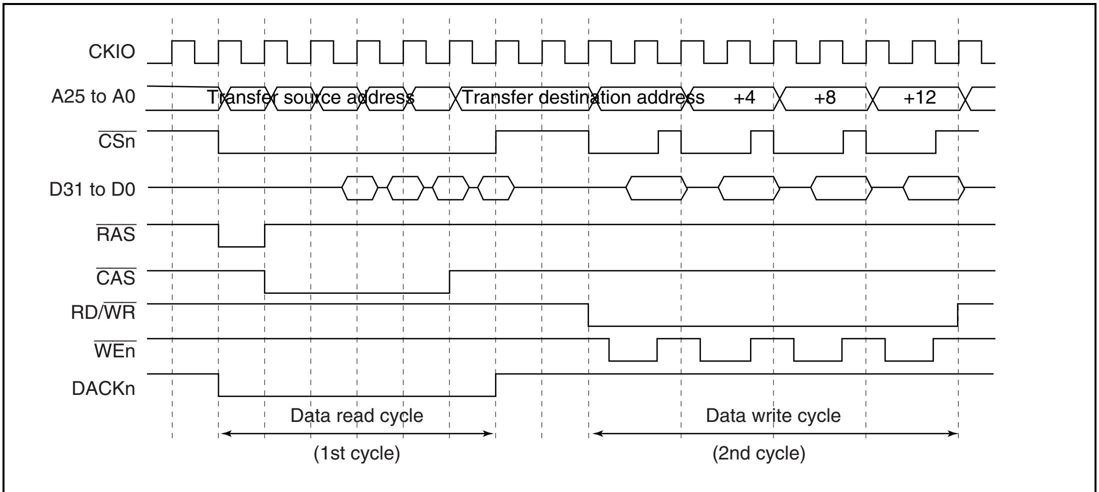
Note: Transfer between external memories, DACK output in a read cycle DACK output timing is the same as that of $\overline{\text{CSn}}$.

Figure 9.6 Example of DMA Transfer Timing in the Direct Address Mode in the Dual Address Mode (Transfer Source: Ordinary Memory, Transfer Destination: Ordinary Memory)



Note: Transfer between external memories, DACK output in a read cycle DACK output timing is the same as that of $\overline{\text{CSn}}$.

Figure 9.7 Example of DMA Transfer Timing in the Direct Address Mode in the Dual Address Mode (16-byte Transfer, Transfer Source: Ordinary Memory, Transfer Destination: Ordinary Memory)

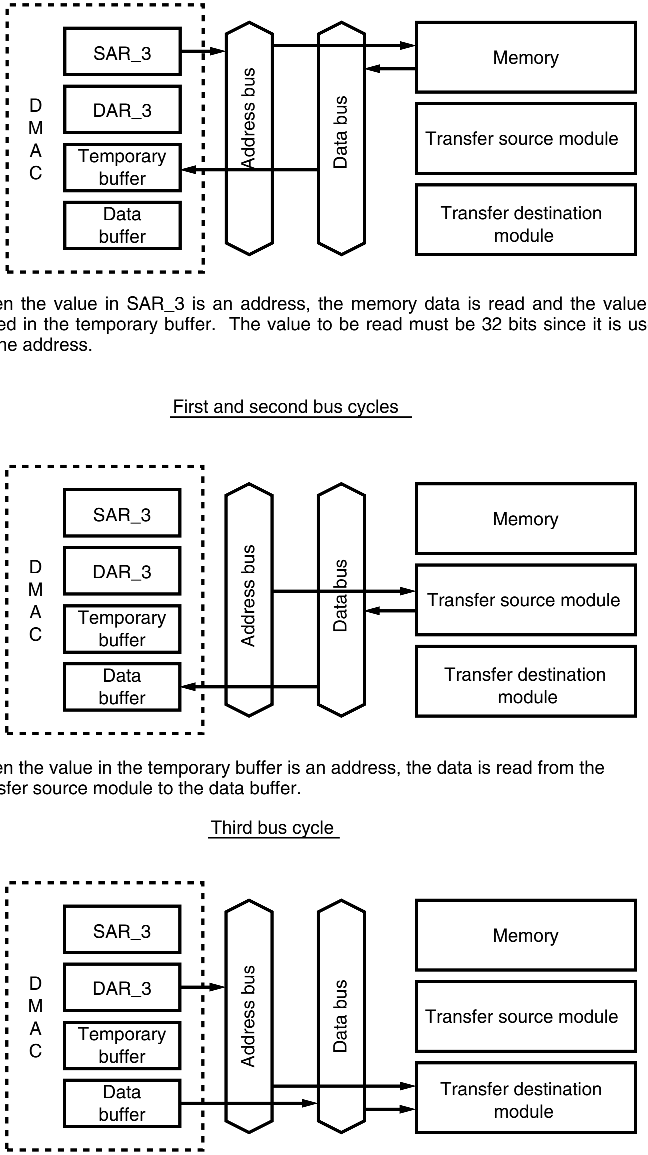


Note: Transfer between external memories, DACK output in a read cycle DACK output timing is the same as that of $\overline{\text{CSn}}$.

Figure 9.8 Example of DMA Transfer Timing in the Direct Address Mode in the Dual Address Mode (16-byte Transfer, Transfer Source: Synchronous DRAM, Transfer Destination: Ordinary Memory)

(2) In the indirect address transfer mode, the address of memory in which data to be transferred is stored is specified in the transfer source address register (SAR_3) in the DMAC. In this mode, the address value specified in the transfer source address register in the DMAC is read first. This value is temporarily stored in the DMAC. Next, the read value is output as an address, and the value stored in that address is stored in the DMAC again. Then, the value read afterwards is written to the address specified in the transfer destination address; this completes one DMA transfer. 16-byte transfer is not possible.

Figure 9.9 shows one example. In this example, the transfer destination, the transfer source, and the storage destination of the indirect address are external memories with a 16-bit width in the indirect address mode, and transfer data is 16 or 8 bits. Figure 9.10 shows an example of the transfer timing.



When the value in SAR_3 is an address, the memory data is read and the value is stored in the temporary buffer. The value to be read must be 32 bits since it is used for the address.

First and second bus cycles

When the value in the temporary buffer is an address, the data is read from the transfer source module to the data buffer.

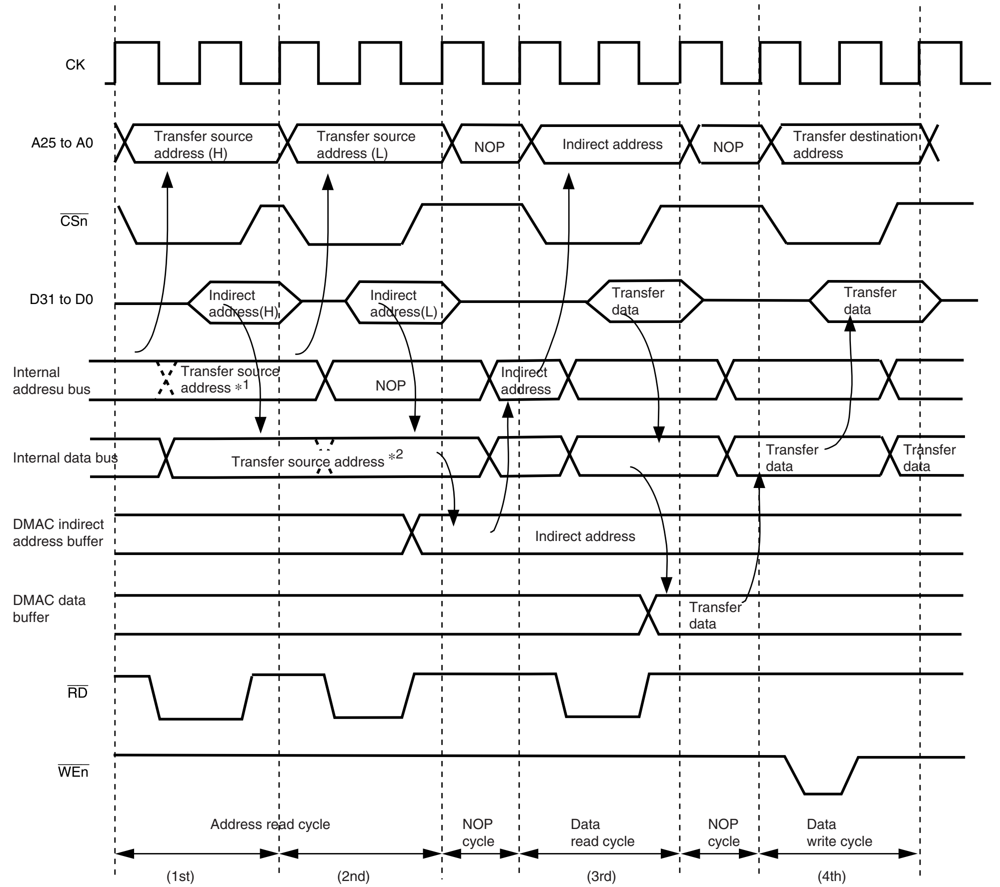
Third bus cycle

When the value in SAR_3 is an address, the value in the data buffer is written to the transfer source module.

Fourth bus cycle

Note: The above description uses the memory, transfer source module, or transfer destination module; in practice, any module can be connected in the addressing space.

Figure 9.9 Operation in the Indirect Address mode in the Dual Address Mode (When the External Memory Space has a 16-bit Width)



Notes: 1. The internal address bus value does not change, and controlled by the port.
2. The DMAC does not fetch the value until 32-bit data is output to the internal data bus.

Figure 9.10 Example of Transfer Timing in the Indirect Address Mode in the Dual Address Mode

- Single Address Mode

  In the single address mode, either the transfer source or transfer destination external device is accessed (selected) by means of the DACK signal, and the other device is accessed by address. In this mode, the DMAC performs one DMA transfer in one bus cycle, accessing one of the external devices by outputting the DACK transfer request acknowledge signal to it, and at the same time outputting an address to the other device involved in the transfer. For example, in the case of transfer between external memory and an external device with DACK, as shown in figure 9.11, when the external device outputs data to the data bus, that data is written to the external memory in the same bus cycle.

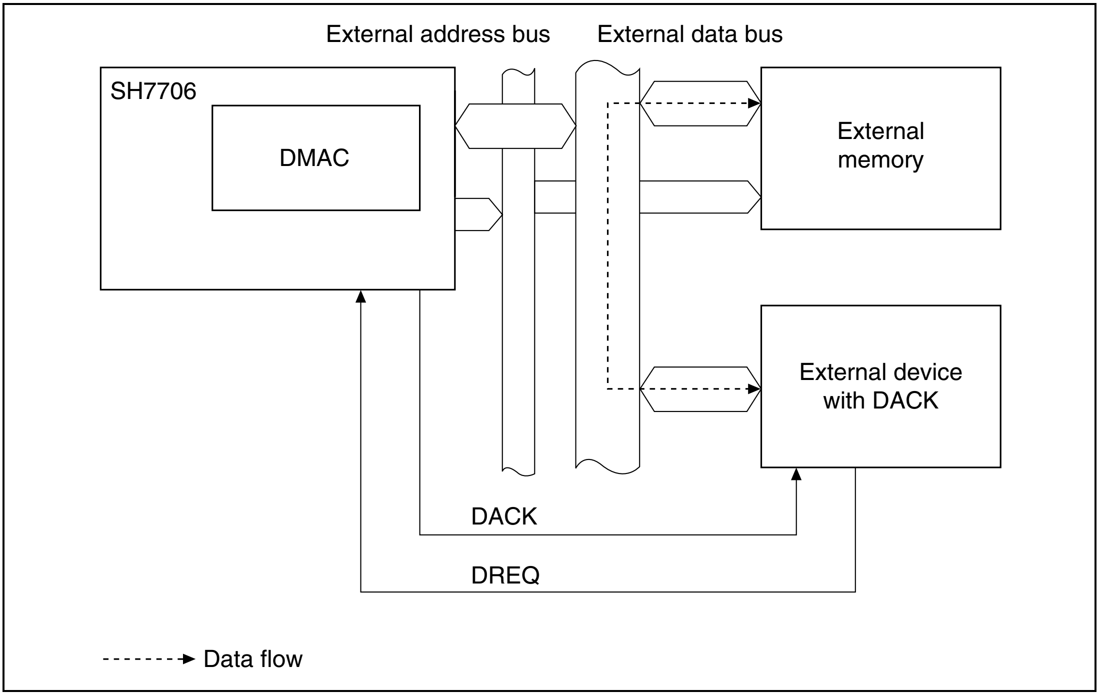
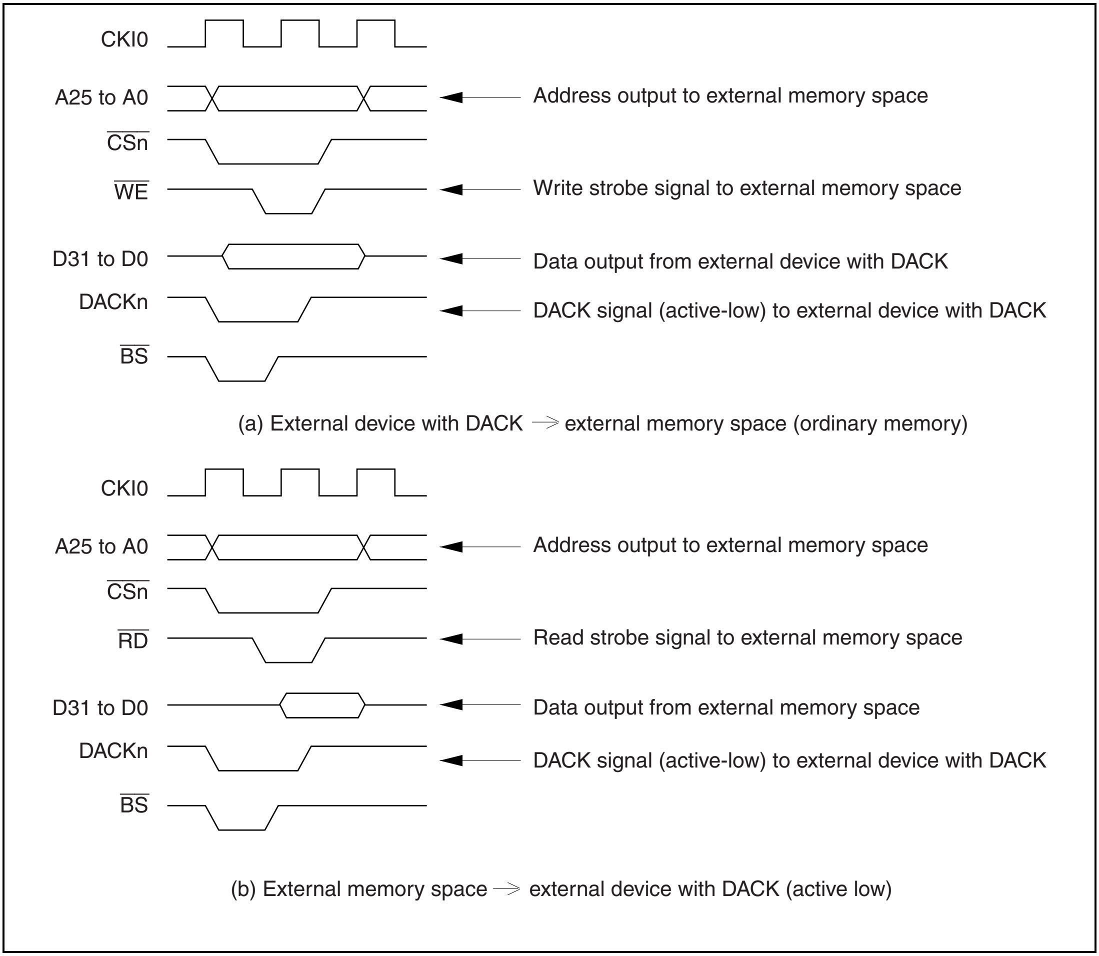


Figure 9.11   Data Flow in the Single Address Mode

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

Figures 9.12 to 9.14 show examples of DMA transfer timing in the single address mode.


CKI0
A25 to A0
CSn
WE
D31 to D0
DACKn
BS
Address output to external memory space
Write strobe signal to external memory space
Data output from external device with DACK
DACK signal (active-low) to external device with DACK
CKI0
A25 to A0
CSn
RD
D31 to D0
DACKn
BS
Address output to external memory space
Read strobe signal to external memory space
Data output from external memory space
DACK signal (active-low) to external device with DACK

(a) External device with DACK → external memory space (ordinary memory)

(b) External memory space → external device with DACK (active low)

Figure 9.12 Example of DMA Transfer Timing in the Single Address Mode

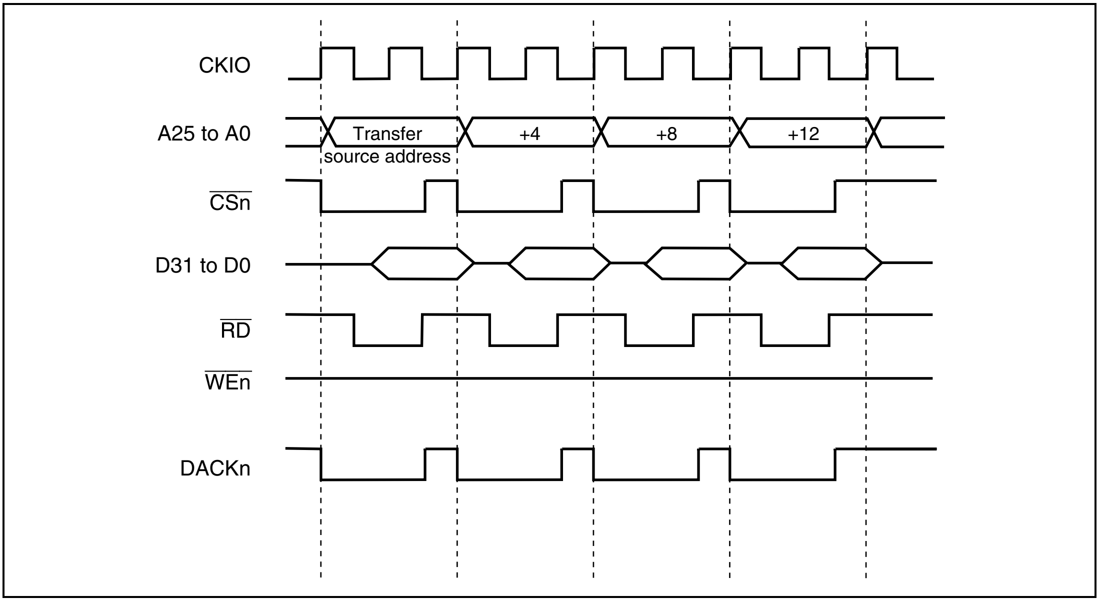


Figure 9.13 Example of DMA Transfer Timing in the Single Address Mode (16- Byte Transfer, External Memory Space (Ordinary Memory) -> External Device with DACK)

Bus Modes: There are two bus modes: cycle-steal and burst. Select the mode in the TM bits of CHCR_0–CHCR_3 (one byte, word, or longword, or 16-byte data).

- Cycle-Steal Mode

  In the cycle-steal mode, the bus right is given to another bus master after a one-transfer-unit (8-, 16-, or 32-bit unit) DMA transfer. When another transfer request occurs, the bus rights are obtained from the other bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus right is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

  In the cycle-steal mode, transfer areas are not affected regardless of settings of the transfer request source, transfer source, and transfer destination. Figure 9.14 shows an example of DMA transfer timing in the cycle steal mode. Transfer conditions shown in the figure are:

  — Dual address mode

  — DREQ level detection

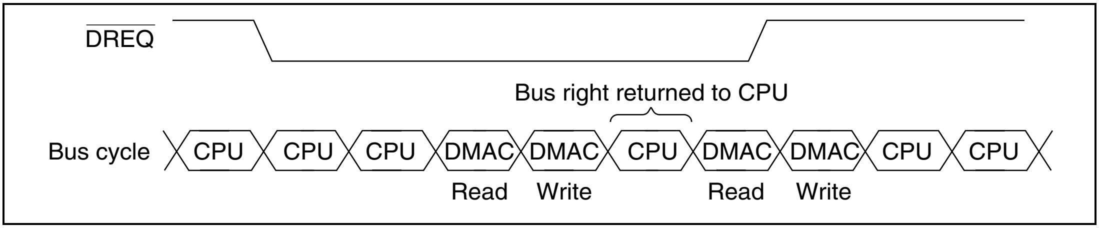


Figure 9.14 DMA Transfer Example in the Cycle-Steal Mode

- Burst Mode

  In the burst mode, once the bus right is obtained, the transfer is performed continuously without passing it until the transfer end conditions are satisfied. In the external request mode with low level detection of the $\overline{\text{DREQ}}$ pin, however, when the $\overline{\text{DREQ}}$ pin is driven high, the bus is passed to the other bus master after the DMAC transfer request that has already been accepted ends, even if the transfer end conditions have not been satisfied.

  The burst mode cannot be used when the serial communications interface (SCIF) and A/D converter are the transfer request sources. Figure 9.15 shows a timing at this point.

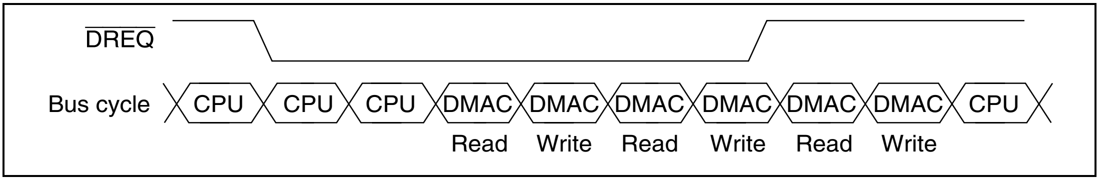


Figure 9.15 DMA Transfer Example in the Burst Mode

Relationship between Request Modes and Bus Modes by DMA Transfer Category: Table 9.5 shows the relationship between request modes and bus modes by DMA transfer category.

Table 9.5 Relationship of Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (bits)	Usable Channels
Dual	External device with DACK and external memory	External	B/C	8/16/32/128	0,1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0, 1
	External memory and external memory	All*[1]	B/C	8/16/32/128	0–3*[5]
	External memory and memory-mapped external device	All*[1]	B/C	8/16/32/128	0–3*[5]
	Memory-mapped external device and memory-mapped external device	All*[1]	B/C	8/16/32/128	0–3*[5]
	External memory and on-chip peripheral module	All*[2]	B/C*[3]	8/16/32*[4]	0–3*[5]
	Memory-mapped external device and on-chip peripheral module	All*[2]	B/C*[3]	8/16/32*[4]	0–3*[5]
	On-chip peripheral module and on-chip peripheral module	All*[2]	B/C*[3]	8/16/32*[4]	0–3*[5]
Single	External device with DACK and external memory	External	B/C	8/16/32/128	0, 1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0, 1

B: Burst, C: Cycle steal

Notes: 1. External requests, auto requests and on-chip peripheral module requests are all available. For on-chip peripheral module requests, however, SCIF, and A/D converter cannot be specified as the transfer request source.

2. External requests, auto requests and on-chip peripheral module requests are all available. When the SCIF, or A/D converter is also the transfer request source, however, the transfer destination or transfer source must be the SCIF, or A/D converter, respectively.

3. If the transfer request source is the SCIF, the cycle-steal mode is only available.

4. The access size permitted when the transfer destination or source is an on-chip peripheral module register.

5. If the transfer request is an external request, channels 0 and 1 are only available.

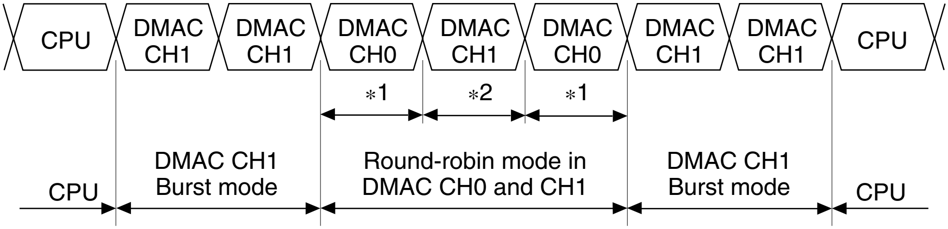
6. If the transfer request source is the SDRAM, the transfer size should be set smaller than the bus width.

Bus Mode and Channel Priority Order: When a given channel 1 is transferring in the burst mode and there is a transfer request to a channel 0 with a higher priority, the transfer of channel 0 will begin immediately.

At this time, if the priority is set in the fixed mode (CH0 > CH1), the channel 1 transfer will continue when the channel 0 transfer has completely finished, even if channel 0 is operating in the cycle-steal mode or in the burst mode.

If the priority is set in the round-robin mode, channel 1 will begin operating again after channel 0 completes the transfer of one transfer unit, even if channel 0 is in the cycle-steal mode or in the burst mode. The bus will then switch between the two in the order channel 1, channel 0, channel 1, channel 0.

Even if the priority is set in the fixed mode or in the round-robin mode, it will not give the bus to the CPU since channel 1 is in the burst mode. This example is illustrated in figure 9.16.



Notes: 1. Cycle-steal mode
2. Burst mode

Figure 9.16 Bus State when Multiple Channels are Operating (Priority Level is Round-robin Mode)

9.4.5 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sampling Timing

Number of Bus Cycle States: When the DMAC is the bus master, the number of bus cycles is controlled by the bus state controller (BSC) in the same way as when the CPU is the bus master. For details, see section 8, Bus State Controller (BSC).

$\overline{\text{DREQ}}$ Pin Sampling Timing: In the external request mode, the $\overline{\text{DREQ}}$ pin is sampled by clock pulse (CKIO) falling edge or low level detection. When $\overline{\text{DREQ}}$ input is detected, a DMAC bus cycle is generated and DMA transfer is performed, at the earliest, three states later.

The second and subsequent $\overline{\text{DREQ}}$ sampling operations are started two cycles after the first sample.

Operation

- Cycle-Steal Mode

  In the cycle-steal mode, the DREQ sampling timing is the same regardless of whether level or edge detection is used.

  For example, in figure 9.17 (cycle-steal mode, level detection), DMAC transfer begins, at the earliest, three cycles after the first sampling is performed. The second sampling is started two cycles after the first. If DREQ is not detected at this time, sampling is performed in each subsequent cycle.

  Thus, DREQ sampling is performed one step in advance. The third sampling operation is not performed until the idle cycle following the end of the first DMA transfer.

  The above conditions are the same whatever the number of CPU transfer cycles, as shown in figure 9.18, and whatever the number of DMA transfer cycles, as shown in figure 9.19.

  DACK is output in a read in the example in figure 9.17, and in a write in the example in figure 9.18. In both cases, DACK is output for the same duration as CSn.

  Figure 9.20 illustrates the case where DREQ is not detected and sampling is subsequently executed every cycle.

  Figure 9.21 shows an example of edge detection in the cycle-steal mode.

- Burst Mode, Level Detection

  In the case of burst mode with level detection, the DREQ sampling timing is the same as in the cycle-steal mode.

  For example, in figure 9.22, DMAC transfer begins, at the earliest, three cycles after the first sampling is performed. The second sampling is started two cycles after the first. Subsequent sampling operations are performed in the idle cycle following the end of the DMA transfer cycle.

  In the burst mode, also, the DACK output period is the same as in the cycle-steal mode.

- Burst Mode, Edge Detection

  In the case of burst mode with edge detection, $\overline{\text{DREQ}}$ sampling is only performed once.

  For example, in figure 9.23, DMAC transfer begins, at the earliest, three cycles after the first sampling is performed. After this, DMAC transfer is executed continuously until the number of data transfers set in the DMATCR register have been completed. $\overline{\text{DREQ}}$ is not sampled during this time.

  To restart DMA transfer after it has been suspended by an NMI, first clear NMIF, then input an edge request again.

  In the burst mode, also, the DACK output period is the same as in the cycle-steal mode.

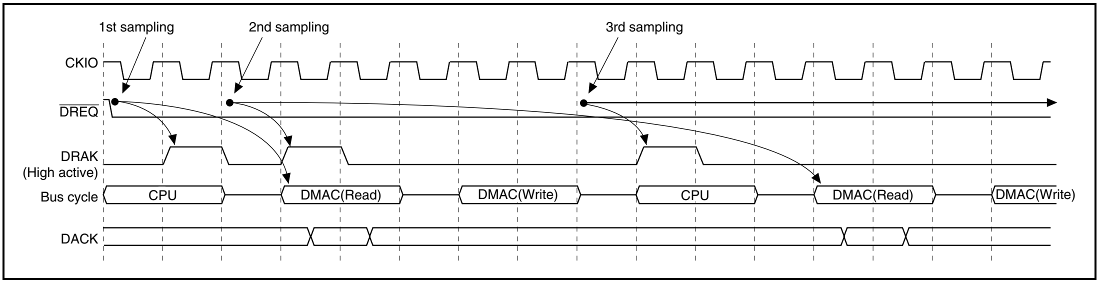


Figure 9.17 Cycle-Steal Mode, Level Input (CPU Access: 2 Cycles)

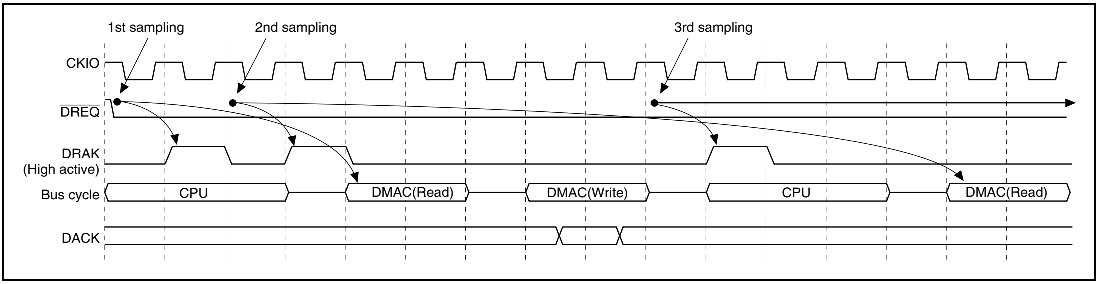


Figure 9.18 Cycle-Steal Mode, Level Input (CPU Access: 3 Cycles)

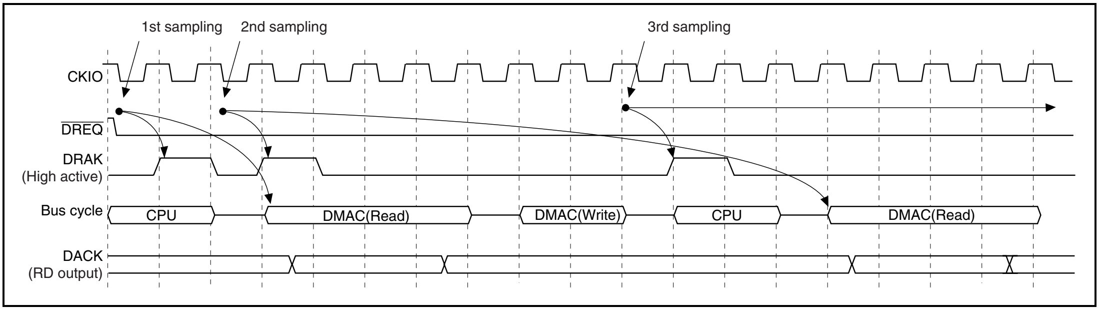


Figure 9.19 Cycle-Steal Mode, Level input (CPU Access: 2 Cycles, DMA RD Access: 4 Cycles)

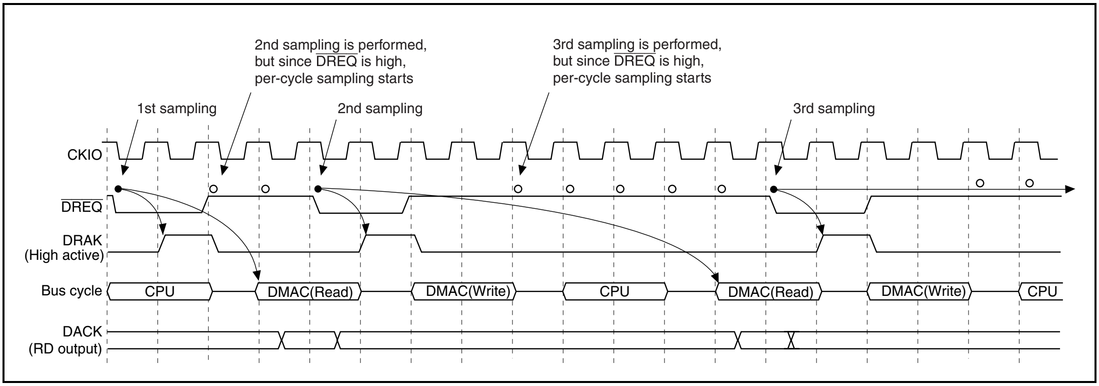
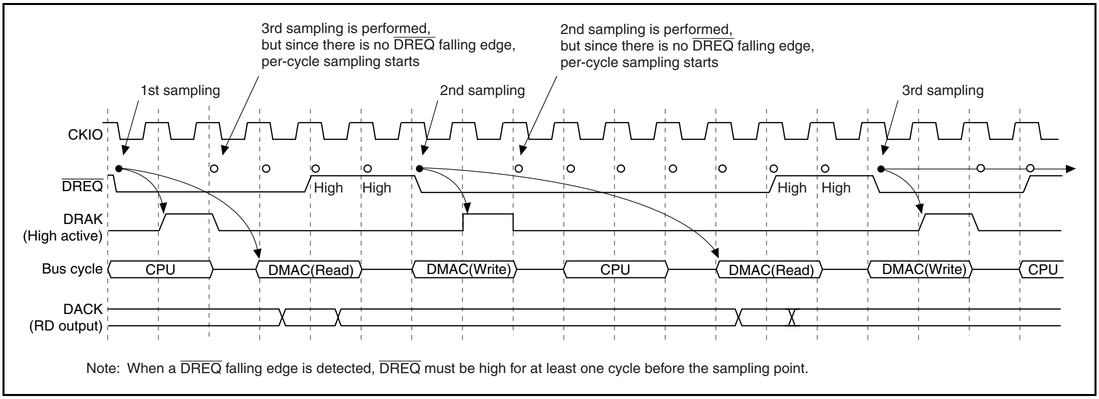

2nd sampling is performed, but since DREQ is high, per-cycle sampling starts
3rd sampling is performed, but since DREQ is high, per-cycle sampling starts
1st sampling
2nd sampling
3rd sampling
CKIO
DREQ
DRAK (High active)
Bus cycle
CPU
DMAC(Read)
DMAC(Write)
CPU
DMAC(Read)
DMAC(Write)
CPU
DACK (RD output)

Figure 9.20 Cycle-Steal Mode, Level input (CPU Access: 2 Cycles, DREQ Input Delayed)


3rd sampling is performed, but since there is no DREQ falling edge, per-cycle sampling starts
2nd sampling is performed, but since there is no DREQ falling edge, per-cycle sampling starts
1st sampling
2nd sampling
3rd sampling
CKIO
DREQ
High
High
High
High
DRAK (High active)
Bus cycle
CPU
DMAC(Read)
DMAC(Write)
CPU
DMAC(Read)
DMAC(Write)
CPU
DACK (RD output)

Note: When a DREQ falling edge is detected, DREQ must be high for at least one cycle before the sampling point.

Figure 9.21 Cycle-Steal Mode, Edge input (CPU Access: 2 Cycles)

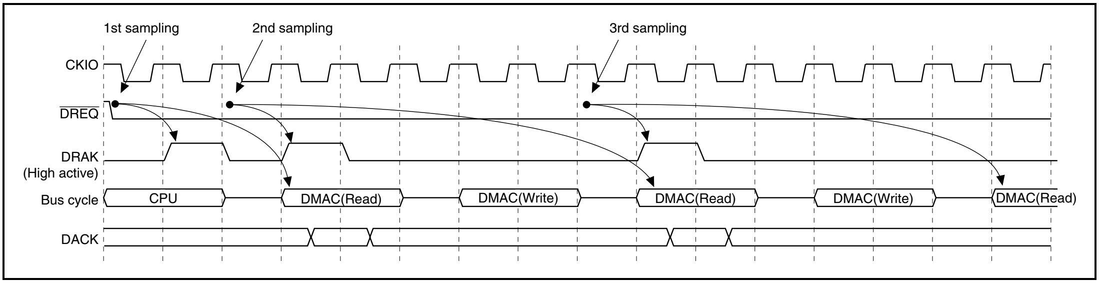


Figure 9.22 Burst Mode, Level Input

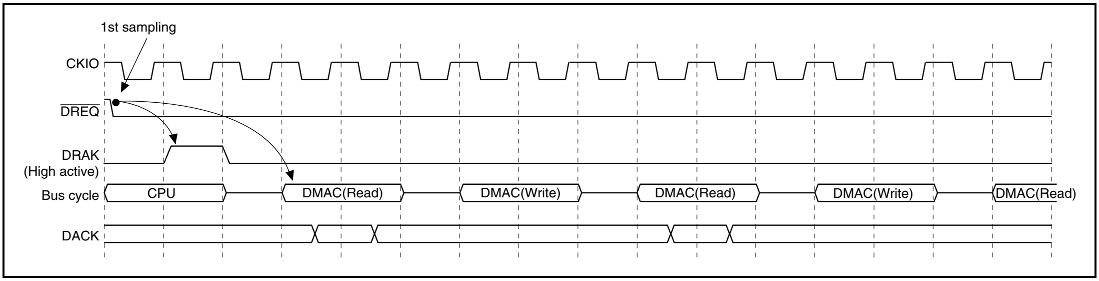


Figure 9.23 Burst Mode, Edge Input

9.4.6 Source Address Reload Function

Channel 2 includes a reload function, in which the value returns to the value set in the SAR_2 for each four transfers by setting the RO bit in CHCR_2 to 1. 16-byte transfer cannot be used. Figure 9.24 shows this operation. Figure 9.25 shows the timing chart of the source address reload function, which is under the following conditions: burst mode, auto request, 16-bit transfer data size, SAR_2 count-up, DAR_2 fixed, reload function on, and usage of only channel 2.

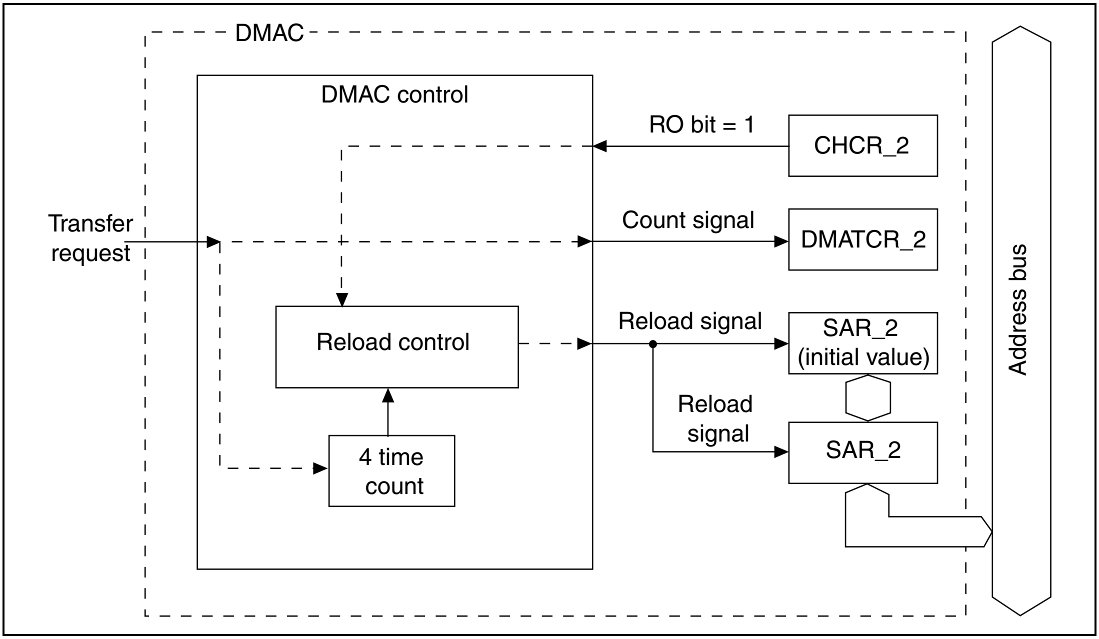


Figure 9.24 Source Address Reload Function Diagram

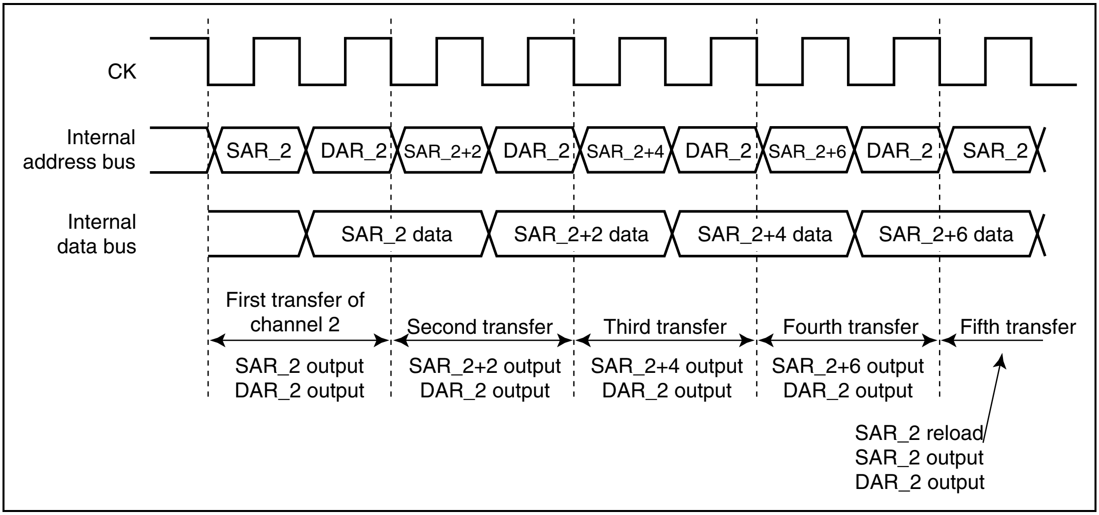


Figure 9.25 Timing Chart of Source Address Reload Function

Even if the transfer data size is 8, 16, or 32 bits, a reload function can be executed.

DMATCR_2, which specifies a transfer count, decrements 1 each time a transfer ends regardless of whether a reload function is on or off. Consequently, be sure to specify the value multiple of four in DMATCR_2 when the reload function is on. Specifying other values does not guarantee the operation.

Though the counters that count transfers of four times for the reload function are reset by clearing the DME bit in DMAOR or the DE bit in CHCR_2, by setting the transfer end flag (TE bit in CHCR_2) by a DMAC address error, or by inputting NMI, besides by resets, the SAR_2, DAR_2, DMATCR_2 registers are not reset. Therefore, if these sources are generated, the counters that are initialized and are not initialized exist in the DMAC; malfunction will be caused by restarting the DMAC in that state. Consequently, if these sources occur except for setting the TE bit during the usage of the reload function, set SAR_2, DAR_2, and DMATCR_2 again.

9.4.7 DMA Transfer Ending Conditions

The DMA transfer ending conditions vary for individual channels ending and all channels ending together. At transfer end, the following conditions are applied except the case where the value set in the DMATCR reaches 0.

(a) Cycle-steal mode (external request, internal request, and auto request)

When the transfer ending conditions are satisfied, DMAC transfer request acceptance is suspended. The DMAC stops operating after completing the number of transfers that it has accepted until the ending conditions are satisfied.

In the cycle-steal mode, the operation is the same regardless of whether the transfer request is detected by the level or at the edge.

(b) Burst mode, edge detection (external request, internal request, and auto request)

The timing from the point where the ending conditions are satisfied to the point where the DMAC stops operating differs from that in cycle steal mode. In the edge detection in the burst mode, though only one transfer request is generated to start up the DMAC, stop request sampling is performed in the same timing as transfer request sampling in the cycle-steal mode. As a result, the period when stop request is not sampled is regarded as the period when transfer request is generated, and after performing the DMA transfer for this period, the DMAC stops operating.

(c) Burst mode, level detection (external request)

Same as described in (a).

(d) Bus timing when transfers are suspended

The transfer is suspended when one transfer ends. Even if transfer ending conditions are satisfied during read in the direct address transfer in the dual address mode, the subsequent write process is executed, and after the transfer in (a) to (c) above has been executed, DMAC operation suspends.

Individual Channel Ending Conditions: There are two ending conditions. A transfer ends when the value of the channel’s DMATCR is 0, or when the DE bit of the channel’s CHCR is cleared to 0.

- When DMATCR is 0: When the DMATCR value becomes 0 and the corresponding channel's DMA transfer ends, the transfer end flag bit (TE) is set in the CHCR. If the IE (interrupt enable) bit has been set, a DMAC interrupt (DEI) is requested to the CPU. This transfer ending does not apply to conditions in (a) to (d) described above.
- When DE of CHCR is 0: Software can halt a DMA transfer by clearing the DE bit in the channel’s CHCR. The TE bit is not set when this happens. This transfer ending does not apply to conditions in (a) to (d) described above.

Conditions for Ending All Channels Simultaneously: Transfers on all channels end when 1) the NMIF or AE bit in the DMAOR is set to 1, or 2) when the DME bit in the DMAOR is cleared to 0.

- Transfers ending when the AE bit or NMIF bit is set to 1 in DMAOR: When an NMI interrupt occurs, the AE bit or NMIF bit is set to 1 in the DMAOR and all channels stop their transfers according to the conditions in (a) to (d) described above, and pass the bus right to other bus masters. Consequently, even if the AE bit or NMI bit is set to 1 during transfer, the SAR, DAR, DMATCR are updated. The TE bit is not set. To resume the transfers after DMAC address error exception handling or NMI interrupt exception handling, clear the AE or NMIF bit to 0. At this time, if there are channels that should not be restarted, clear the corresponding DE bit in the CHCR.
- Transfers ending when DME is cleared to 0 in DMAOR: Clearing the DME bit to 0 in the DMAOR forcibly stops the transfers on all channels. The TE bit is not set. All channels stop their transfers according to the conditions in (a) to (d) in 9.4.7, DMA Transfer Ending Condition, as in DMAC address error occurrence or NMI interrupt generation. In this case, the values in SAR, DAR, and DMATCR are also updated.

9.5 Compare Match Timer (CMT)

DMAC has an on-chip compare match timer (CMT) to generate DMA transfer request. The CMT has 16-bit counter. Figure 9.26 shows a CMT block diagram.

9.5.1 Features

The CMT has the following features:

- 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.
- Generate DMA transfer request when compare match occurs.

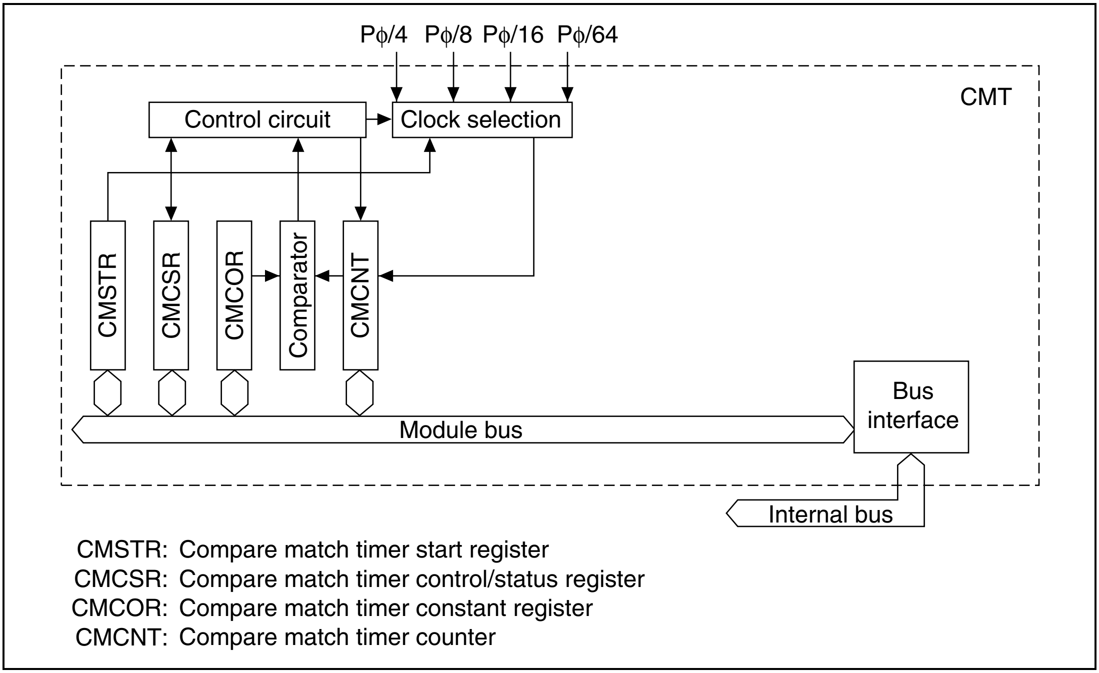

Pϕ/4
Pϕ/8
Pϕ/16
Pϕ/64
CMT
Control circuit
Clock selection
CMSTR
CMCSR
CMCOR
Comparator
CMCNT
Module bus
Bus interface
Internal bus
CMSTR: Compare match timer start register
CMCSR: Compare match timer control/status register
CMCOR: Compare match timer constant register
CMCNT: Compare match timer counter

Figure 9.26 CMT Block Diagram

9.5.2 Description of Registers

The CMT has the following registers. Refer to section 23, Control Registers Table, for more detail of the addresses and access size.

- Compare match timer start register (CMSTR)
- Compare match timer control/status register (CMCSR)
- Compare match counter (CMCNT)
- Compare match constant register (CMCOR)

- Compare Match Timer Start Register (CMSTR)

The compare match timer start register (CMSTR) is a 16-bit register that selects whether to operate or halt the channel 0 and channel 1 counter (CMCNT).

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	—	0	R/W	Reserved This bit can be read or written. Write 0 when writing.
0	STR0	0	R/W	Count start 0 Selects whether to operate or halt compare match timer counter 0. 0: CMCNT0 count operation halted 1: CMCNT0 count operation

- Compare Match Timer Control/Status Register (CMCSR)

The compare match timer control/status register (CMCSR) is a 16-bit register that indicates the occurrence of compare matches, sets the enable/disable of interrupts, and establishes the clock used for incrementation.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	0	R	Reserved These bits always read as 0. The write value should always be 0.
7	CMF	0	R/(W)*	Compare match flag This flag indicates whether CMCNT and CMCOR values have matched or not. 0: CMCNT and CMCOR values have not matched Clearing condition: Write 0 to CMF after reading CMF = 1 1: CMCNT and CMCOR values have matched
6	—	0	R/W	Reserved Both read and write are available. The write value should always be 0.
5 to 2	—	0	R	Reserved These bits always read as 0. The write value should always be 0.
1	CKS1	0	R/W	Clock select 1 and 0
0	CKS0	0	R/W	These bits select the clock input to the CMCNT from among the four internal clocks obtained by dividing the system clock (Pϕ). When the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with the clock selected by CKS1 and CKS0. 00: P ϕ/4 01: P ϕ/8 10: P ϕ/16 11: P ϕ/64

Note: * The only value that can be written is 0 to clear the flag.

- **Compare Match Counter (CMCNT)**

The compare match counter (CMCNT) is a 16-bit register used as an up-counter.

When an internal clock is selected with the CKS1 and CKS0 bits of the CMCSR and the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with the selected clock. When the CMCNT value matches that of the CMCOR, the CMCNT is cleared to H'0000 and the CMF flag of the CMCSR is set to 1.

The CMCNT0 is initialized to H'0000 by resets. It retains its previous value in standby mode.

- **Compare Match Constant Register (CMCOR)**

The compare match constant register (CMCOR) is a 16-bit register that sets the compare match period with the CMCNT.

The CMCOR is initialized to H'FFFF by resets. It retains its previous value in standby mode.

9.5.3 Description of Operation

- **Period Count Operation**

When an internal clock is selected with the CKS1 and CKS0 bits of the CMCSR register and the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with the selected clock. When the CMCNT counter value matches that of the CMCOR, the CMCNT counter is cleared to H'0000 and the CMF flag of the CMCSR register is set to 1. The CMCNT counter begins counting up again from H'0000.

Figure 9.27 shows the compare match counter operation.

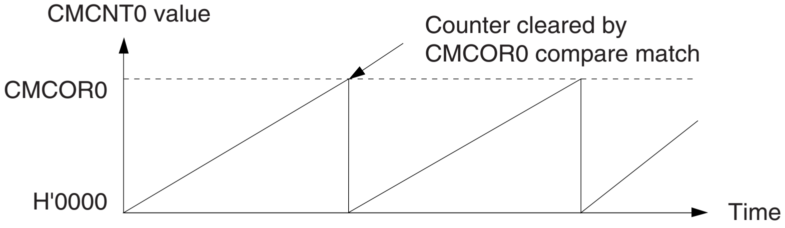


Figure 9.27 Counter Operation

- **CMCNT Count Timing**

One of four clocks (Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/64) obtained by dividing the the system clock (Pϕ) can be selected by the CKS1 and CKS0 bits of the CMCSR. Figure 9.28 shows the timing.

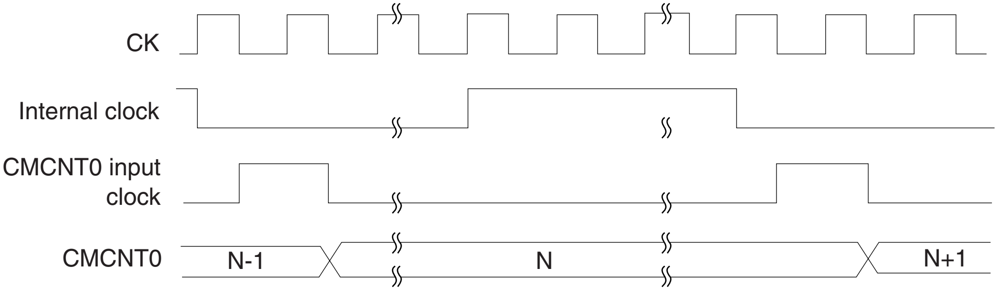


Figure 9.28 Count Timing

- **Compare Match Flag Set Timing**

The CMF bit of the CMCSR register is set to 1 by the compare match signal generated when the CMCOR register and the CMCNT counter match. The compare match signal is generated upon the final state of the match (timing at which the CMCNT counter matching count value is updated). Consequently, after the CMCOR register and the CMCNT counter match, a compare match signal will not be generated until a CMCNT counter input clock occurs. Figure 9.29 shows the CMF bit set timing.

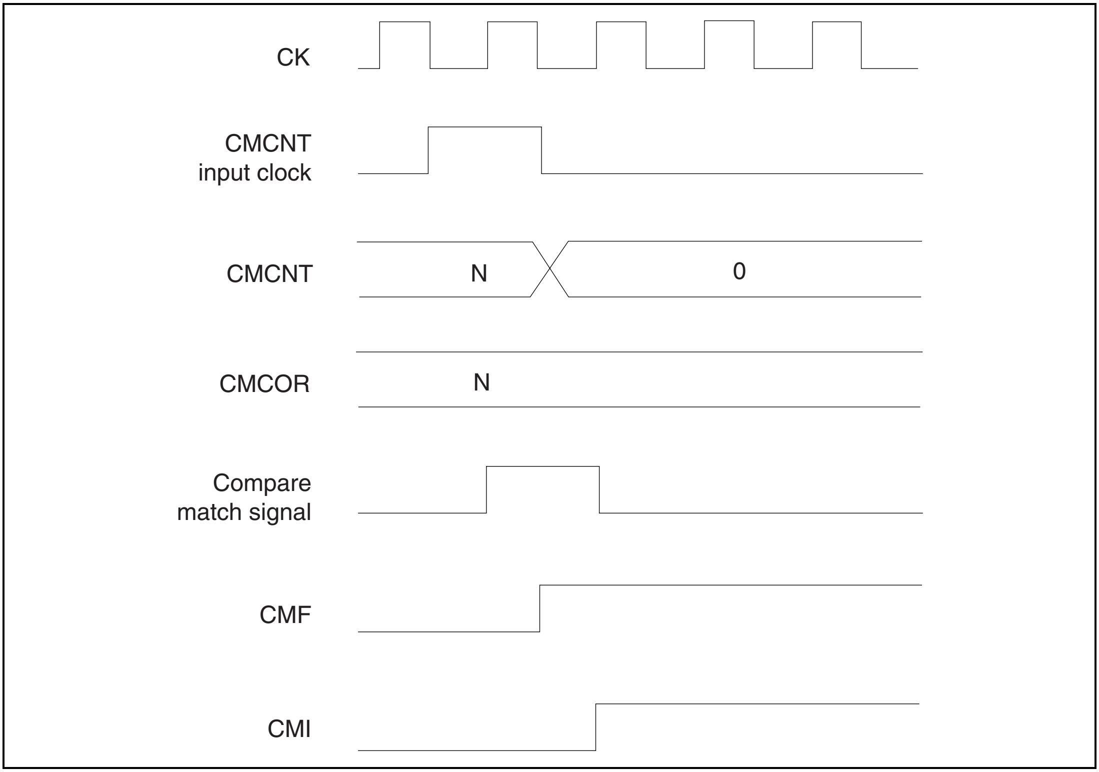


Figure 9.29 CMF Set Timing

- **Compare Match Flag Clear Timing**

The CMF bit of the CMCSR register is cleared by writing 0 to it after reading 1. Figure 9.30 shows the timing when the CMF bit is cleared by the CPU.

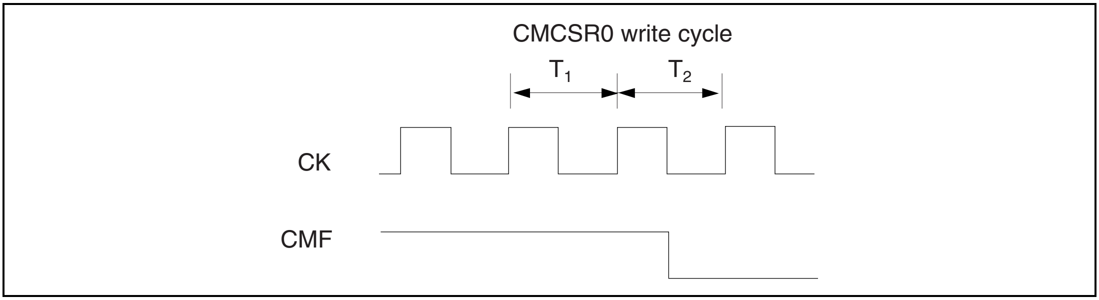


Figure 9.30 Timing of CMF Clear by the CPU

9.6 Examples of Use

9.6.1 Example of DMA Transfer between A/D Converter and External Memory (Address Reload on)

In this example, DMA transfer is performed between the on-chip A/D converter (transfer source) and the external memory (transfer destination) with address reload function on. Table 9.6 shows the transfer conditions and register settings.

Table 9.6 Transfer Conditions and Register Settings for Transfer between On-Chip A/D converter and External Memory

Transfer Conditions	Register	Setting
Transfer source: on-chip A/D converter	SAR_2	H'04000080
Transfer destination: external memory	DAR_2	H'00400000
Number of transfers: 128 (reloading 32 times)	DMATCR_2	H'00000080
Transfer source address: incremented	CHCR_2	H'00089E35
Transfer destination address: decremented		
Transfer request source: A/D converter		
Bus mode: burst		
Transfer unit: long word		
Interrupt request generated at end of transfer		
Channel priority order: 0 > 2 > 3 > 1	DMAOR	H'0101

When the address reload function is on, the values set in SAR_0 to SAR_3 returns to the initially set value at each four transfers. In this example, when an interrupt request is generated from A/D converter, longword data is read from the register in address H'04000080 in A/D converter , and it is written to external memory address H'00400000. Since longword data has been transferred, the values in SAR_2 and DAR_2 are H'04000084 and H'003FFFFC, respectively. The bus right is maintained and data transfers are successively performed because this transfer is in the burst mode.

After four transfers end, fifth and sixth transfers are performed if the address reload function is off, and the value in SAR_2 is incremented from H'0400008C, H'04000090, H'04000094,.... If the address reload function is on, the DMA transfer stops after the fourth transfer ends, the bus request signal to the CPU is cleared. At this time, the value stored in SAR_2 is not incremented from H'0400008C to H'04000090, but returns to the initially set value H'04000080. The value in DAR_2 continues being decremented regardless of whether the address reload function is on or off.

As a result, the values in the DMAC are as shown in table 9.7 when the fourth transfer ends, depending on whether the address reload function is on or off.

Table 9.7 Values in the DMAC after the Fourth Transfer Ends

Items	Address reload on	Address reload off
SAR_2	H'04000080	H'04000090
DAR_2	H'003FFFFC	H'003FFFFC
DMATCR_2	H'0000007C	H'0000007C
Bus right	Released	Held
DMAC operation	Stops	Keeps operating
Interrupt	Not generated	Not generated
Transfer request source flag clear	Executed	Not executed

Notes: 1. An interrupt is generated regardless of whether the address reload function is on or off, if transfers are executed until the value in DMATCR_2 reaches 0 and the IE bit in CHCR_2 has been set to 1.

2. The transfer request source flag is cleared regardless of whether the address reload function is on or off, if transfers are executed until the value in DMATCR_2 reaches 0.

3. Specify the burst mode to use the address reload function. This function may not be correctly executed in the cycle steal mode.

4. Set the value multiple of four in DMATCR_2 to use the address reload function. This function may not be correctly executed if other values are specified.

9.6.2 Example of DMA Transfer between External Memory and SCIF Transmitter (Indirect Address on)

In this example, DMA transfer is performed between the external memory specified with the indirect address (transfer source) and the SCIF transmitter (transfer destination) using DMAC channel 3. Table 9.8 shows the transfer conditions and register settings. In addition, the trigger of the number of transmit FIFO data is set to 1 (TTRG1 = TTRG0 = 1 in SCFCR).

Table 9.8 Transfer Conditions and Register Settings for Transfer between External Memory and SCIF Transmitter

Transfer Conditions	Register	Setting
Transfer source: external memory	SAR_3	H'00400000
Value stored in address H'00400000	—	H'00450000
Value stored in address H'04500000	—	H'55
Transfer destination: On-chip SCIF TDR2	DAR_3	H'04000156
Number of transfers: 10	DMATCR_3	H'0000000A
Transfer source address: incremented	CHCR_3	H'00011C01
Transfer destination address: fixed		
Transfer request source: SCIF (TXI2)		
Bus mode: cycle steal		
Transfer unit: byte		
No interrupt request generated at end of transfer		
Channel priority order: 0 > 1 > 2 > 3	DMAOR	H'0001

If the indirect address is on, data stored in the address set in SAR_0 to SAR_3 is not used as transfer source data. In the indirect address, after the value stored in the address set in SAR_0 to SAR_3 is read, that read value is used as an address again, and the value stored in that address is read and stored in the corresponding address set in DAR_0 to DAR_3.

In the example shown in table 9.3, when an SCIF transfer request is generated, the DMAC reads the value in address H'00400000 set in SAR_3. Since the value H'00450000 is stored in that address, the DMAC reads the value H'00450000. Next, the DMAC uses that read value as an address again, and reads the value H'55 stored in that address. Then, the DMAC writes the value H'55 to address H'04000156 set in DAR_3; this completes one indirect address transfer.

In the indirect address, when data is read first from the address set in SAR_3, the data transfer size is always longword regardless of the settings of the TS0 and the TS1 bits that specify the transfer data size. However, whether the transfer source address is fixed, incremented, or decremented is specified according to the SM0 and the SM1 bits. Therefore, in this example, though the transfer data size is specified as byte, the value in SAR_3 is H'00400004 when one transfer ends. Write operation is the same as that in the normal dual address transfer.

9.7 Cautions

1. CHCR_0–CHCR_3 can be accessed in any data size. The DMA operation register (DMAOR) must be accessed in byte (eight bits) or word (16 bits); other registers must be accessed in word (16 bits) or longword (32 bits).
2. Before rewriting the RS0–RS3 bits of CHCR_0–CHCR_3, first clear the DE bit to 0 (when rewriting CHCR, be sure to clear the DE bit to 0 in advance).
3. Even when the NMI interrupt is input when the DMAC is not operating, the NMIF bit of the DMAOR will be set.
4. When entering the standby mode, the DME bit in DMAOR must be cleared to 0 and the transfers accepted by the DMAC must end.
5. The on-chip peripherals which DMAC can access are SCIF, A/D converter, D/A converter, and I/O ports. Do not access the other peripherals by DMAC.
6. When starting up the DMAC, set CHCR_0 −CHCR_3 or DMAOR last. Specifying other registers last does not guarantee normal operation.
7. Even if the maximum number of transfers is performed in the same channel after the DMATCR_0–DMATCR_3 count reaches 0 and the DMA transfer ends normally, write 0 to DMATCR_0–DMATCR_3. Otherwise, normal DMA transfer may not be performed.
8. When using the address reload function, specify the burst mode as a transfer mode. In the cycle-steal mode, normal DMA transfer may not be performed.
9. When using the address reload function, set the value multiple of four in DMATCR_0 to DMATCR_3. Specifying other values does not guarantee normal operation.
10. When detecting an external request at the falling edge, keep the external request pin high when setting the DMAC.
11. Do not access the space ranging from H'4000062 to H'400006F, which is not used in the DMAC. Accessing that space may cause malfunctions.
12. The $\overline{\text{WAIT}}$ signal is ignored in the following cases:
    a. In 16-byte DMA transfer or dual addressing mode, or when writing data to the external address area
    b. In 16-byte DMA transfer or single addressing mode, or when transferring data from an external device with DACK to the external address area

Section 10 Clock Pulse Generator (CPG)

The clock pulse generator (CPG) supplies all clocks to the processor and controls the power-down modes. A block diagram of the clock pulse generator is shown in figure 10.1.

10.1 Features

The CPG has the following features:

- Four clock modes: Selection of 4 clock modes for different frequency ranges, power consumption, direct crystal input, and external clock input are available.
- Three clocks generated independently: An internal clock for the CPU, cache, and TLB (Iϕ); a peripheral clock (Pϕ) for the on-chip supporting modules; and a bus clock (CKIO) for the external bus interface.
- Frequency change function: Internal and peripheral clock frequencies can be changed independently using the PLL circuit and divider circuit within the CPG. Frequencies are changed by software using frequency control register (FRQCR) settings.
- Power-down mode control: The clock can be stopped for sleep mode and software standby mode and specific modules can be stopped using the module standby function.

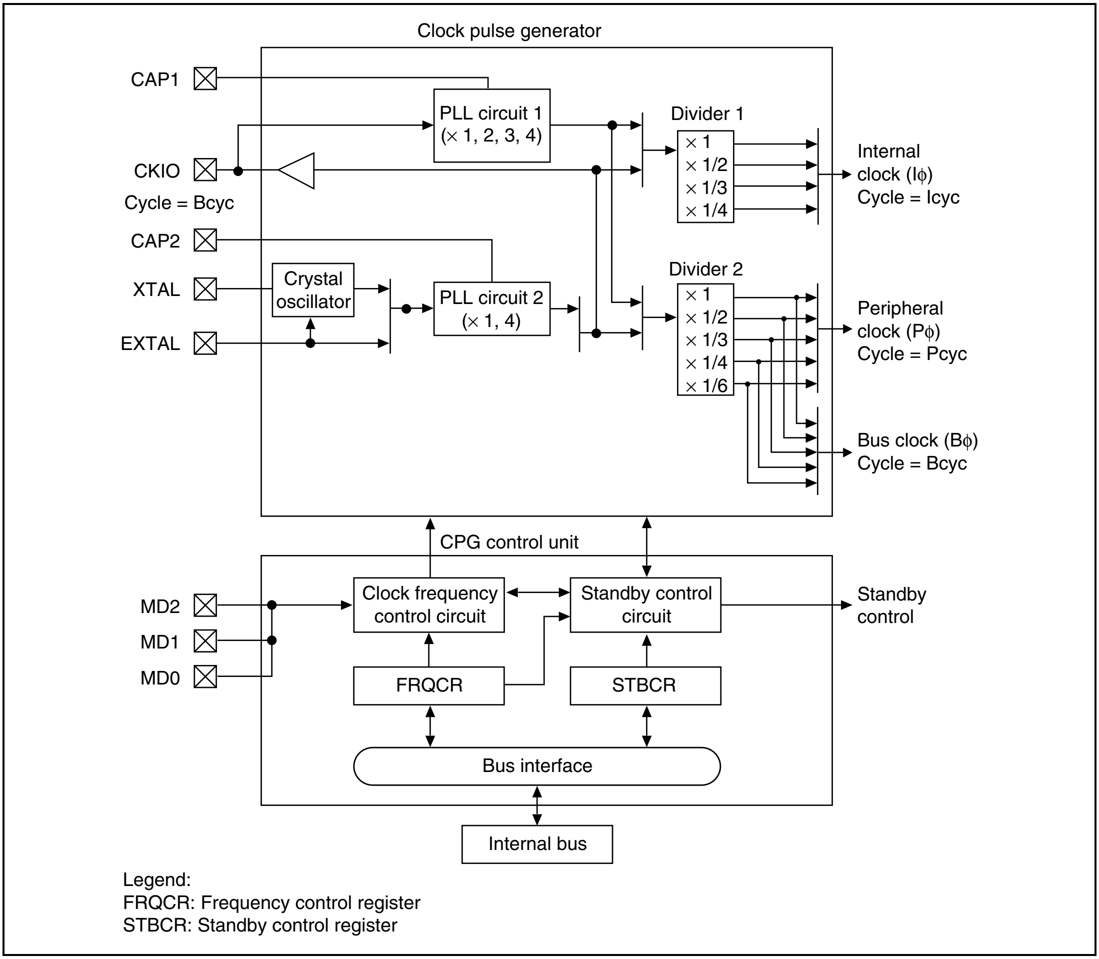


Figure 10.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

1. PLL Circuit 1: PLL circuit 1 doubles, triples, quadruples, or leaves unchanged the input clock frequency from the CKIO terminal. The multiplication rate is set by the frequency control register. When this is done, the phase of the leading edge of the internal clock (Iϕ, Bϕ, Pϕ) is controlled so that it will agree with the phase of the leading edge of the CKIO pin.
2. PLL Circuit 2: PLL circuit 2 leaves unchanged or quadruples the frequency of the crystal oscillator or the input clock frequency coming from the EXTAL pin. The multiplication ratio is fixed by the clock operation mode. The clock operation mode is set by pins MD0, MD1, and MD2. See table 10.3 for more information on clock operation modes.
3. Crystal Oscillator: This oscillator is used when a crystal oscillator element is connected to the XTAL and EXTAL pins. It operates according to the clock operating mode setting.
4. Divider 1: Divider 1 generates a clock at the operating frequency used by the internal clock. The operating frequency can be 1, 1/2, 1/3, or 1/4 times the output frequency of PLL circuit 1, as long as it stays at or above the clock frequency of the CKIO pin. The division ratio is set in the frequency control register.
5. Divider 2: Divider 2 generates a clock at the operating frequency used by the bus clock (Bϕ) and peripheral clock (Pϕ). The operating frequencies can be 1, 1/2, 1/3, 1/4 , or 1/6 times the output frequency of PLL Circuit 1 or the clock frequency of the CKIO pin, as long as it stays at or below the clock frequency of the CKIO pin. The division ratio is set in the frequency control register.
6. Clock Frequency Control Circuit: The clock frequency control circuit controls the clock frequency using the MD2–MD0 pins and the frequency control register.
7. Standby Control Circuit: The standby control circuit controls the state of the clock pulse generator and other modules during clock switching and sleep/standby modes.
8. Frequency Control Register: The frequency control register has control bits assigned for the following functions: clock output/non-output from the CKIO pin, on/off control of PLL circuit 1, PLL standby, the frequency multiplication ratio of PLL 1, and the frequency division ratio of the internal clock and the peripheral clock.
9. Standby Control Register: The standby control register has bits for controlling the power-down modes. See section 22, Power-Down Modes, for more information.

10.2 I/O Pins

Table 10.1 lists the CPG pins and their functions.

Table 10.1 Clock Pulse Generator Pins and Functions

Pin Name	Symbol	I/O	Description
Mode control pins	MD0	I	Set the clock operating mode.
	MD1	I	
	MD2	I	
Crystal I/O pins (clock input pins)	XTAL	O	Connects a crystal oscillator.
	EXTAL	I	Connects a crystal oscillator. Also used to input an external clock.
Clock I/O pin	CKIO	I/O	Inputs or outputs an external clock.
Capacitor connection pins for PLL	CAP1	I	Connects capacitor for PLL circuit 1 operation (recommended value 470 pF).
	CAP2	I	Connects capacitor for PLL circuit 2 operation (recommended value 470 pF).

10.3 Clock Operating Modes

Table 10.2 shows the relationship between the mode control pin (MD2–MD0) combinations and the clock operating modes. Table 10.3 shows the usable frequency ranges in the clock operating modes and frequency ranges of the input clock (crystal oscilation).

Table 10.2 Clock Operating Modes

	Pin Values			Clock I/O						
Mode	MD2	MD1	MD0	Source	Output	PLL2 On/Off	PLL1 On/Off	Divider 1 Input	Divider 2 Input	CKIO Frequency
0	0	0	0	EXTAL	CKIO	On, multi-plication ratio: 1	On	PLL1 output	PLL1	(EXTAL)
1	0	0	1	EXTAL	CKIO	On, multi-plication ratio: 4	On	PLL1 output	PLL1	(EXTAL) × 4
2	0	1	0	Crystal oscillato r	CKIO	On, multi-plication ratio: 4	On	PLL1 output	PLL1	(Crystal) × 4
7	1	1	1	CKIO	–	Off	On	PLL1 output	PLL1	(CKIO)
—	Other than the above			Reserved (setting disabled)						

Mode 0: An external clock is input from the EXTAL pin and undergoes waveform shaping by PLL circuit 2 before being supplied inside this LSI. The frequency ratio between EXTAL input clock and CKIO output clock is 1:1. An input clock frequency of 25 MHz to 66.67 MHz can be used, and the CKIO frequency range is 25 MHz to 66.67 MHz.

Mode 1: An external clock is input from the EXTAL pin and its frequency is multiplied by 4 by PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency external clock to be used. The frequency ratio between EXTAL input clock and CKIO output clock is 1:4. An input clock frequency of 6.25 MHz to 16.67 MHz can be used, and the CKIO frequency range is 25 MHz to 66.67 MHz.

Mode 2: The on-chip crystal oscillator operates, with the oscillation frequency being multiplied by 4 by PLL circuit 2 before being supplied inside this LSI, allowing a low crystal frequency to be used. The frequency ratio between cystal oscilation and CKIO output clock is 1:4. A crystal oscillation frequency of 6.25 MHz to 16.67 MHz can be used, and the CKIO frequency range is 25 MHz to 66.67 MHz.

Mode 7: In this mode, the CKIO pin is an input, an external clock is input to this pin, and undergoes waveform shaping, and also frequency multiplication according to the setting, by PLL circuit 1 before being supplied to this LSI. In modes 0 to 2, the system clock is generated from the output of this LSI's CKIO pin. Consequently, if a large number of Ics are operating synchronized with the clock, the CKIO pin load will be large. This mode, however, assumes a comparatively large-scale system. If a large number of ICs are operating on the clock cycle, a clock generator with a number of low-skew clock outputs can be provided, so that the ICs can operate synchronously by distributing the clocks to each one.

As PLL circuit 1 compensates for fluctuations in the CKIO pin load, this mode is suitable for connection of synchronous DRAM.

Table 10.3 Available Combination of Clock Mode and FRQCR Values

Clock Mode	FRQCR	PLL1	PLL2	Clock Rate* (I:B:P)	Input Frequency Range	CKIO Frequency Range
0	H'0100	ON (× 1)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'0101	ON (× 1)	ON (× 1)	1:1:1/2	25 MHz to 66.67 MHz	25 MHz to 66.67 MHz
	H'0102	ON (× 1)	ON (× 1)	1:1:1/4	25 MHz to 66.67 MHz	25 MHz to 66.67 MHz
	H'0111	ON (× 2)	ON (× 1)	2:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'0112	ON (× 2)	ON (× 1)	2:1:1/2	25 MHz to 66.67 MHz	25 MHz to 66.67 MHz
	H'0115	ON (× 2)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'0116	ON (× 2)	ON (× 1)	1:1:1/2	25 MHz to 66.67 MHz	25 MHz to 66.67 MHz
	H'0122	ON (× 4)	ON (× 1)	4:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'0126	ON (× 4)	ON (× 1)	2:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'012A	ON (× 4)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'A100	ON (× 3)	ON (× 1)	3:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'A101	ON (× 3)	ON (× 1)	3:1:1/2	25 MHz to 44.44 MHz	25 MHz to 44.44 MHz
	H'E100	ON (× 3)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'E101	ON (× 3)	ON (× 1)	1:1:1/2	25 MHz to 44.44 MHz	25 MHz to 44.44 MHz
1, 2	H'0100	ON (× 1)	ON (× 4)	4:4:4	6.25 MHz to 8.34 MHz	25 MHz to 33.34 MHz
	H'0101	ON (× 1)	ON (× 4)	4:4:2	6.25 MHz to 16.67 MHz	25 MHz to 66.67 MHz
	H'0102	ON (× 1)	ON (× 4)	4:4:1	6.25 MHz to 16.67 MHz	25 MHz to 66.67 MHz
	H'0111	ON (× 2)	ON (× 4)	8:4:4	6.25 MHz to 8.34 MHz	25 MHz to 33.34 MHz
	H'0112	ON (× 2)	ON (× 4)	8:4:2	6.25 MHz to 16.67 MHz	25 MHz to 66.67 MHz
	H'0115	ON (× 2)	ON (× 4)	4:4:4	6.25 MHz to 8.34 MHz	25 MHz to 33.34 MHz
	H'0116	ON (× 2)	ON (× 4)	4:4:2	6.25 MHz to 16.67 MHz	25 MHz to 66.67 MHz
	H'0122	ON (× 4)	ON (× 4)	16:4:4	6.25 MHz to 8.34 MHz	25 MHz to 33.34 MHz
	H'0126	ON (× 4)	ON (× 4)	8:4:4	6.25 MHz to 8.34 MHz	25 MHz to 33.34 MHz
	H'012A	ON (× 4)	ON (× 4)	4:4:4	6.25 MHz to 8.34 MHz	25 MHz to 33.34 MHz
	H'A100	ON (× 3)	ON (× 4)	12:4:4	6.25 MHz to 8.34 MHz	25 MHz to 33.34 MHz
	H'A101	ON (× 3)	ON (× 4)	12:4:2	6.25 MHz to 11.11 MHz	25 MHz to 44.44 MHz
	H'E100	ON (× 3)	ON (× 4)	4:4:4	6.25 MHz to 8.34 MHz	25 MHz to 33.34 MHz
	H'E101	ON (× 3)	ON (× 4)	4:4:2	6.25 MHz to 11.11 MHz	25 MHz to 44.44 MHz

Clock Mode	FRQCR	PLL1	PLL2	Clock Rate* (I:B:P)	Input Frequency Range	CKIO Frequency Range
7	H'0100	ON (× 1)	OFF	1:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'0101	ON (× 1)	OFF	1:1:1/2	25 MHz to 66.67 MHz	25 MHz to 66.67 MHz
	H'0102	ON (× 1)	OFF	1:1:1/4	25 MHz to 66.67 MHz	25 MHz to 66.67 MHz
	H'0111	ON (× 2)	OFF	2:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'0112	ON (× 2)	OFF	2:1:1/2	25 MHz to 66.67 MHz	25 MHz to 66.67 MHz
	H'0115	ON (× 2)	OFF	1:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'0116	ON (× 2)	OFF	1:1:1/2	25 MHz to 66.67 MHz	25 MHz to 66.67 MHz
	H'0122	ON (× 4)	OFF	4:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'0126	ON (× 4)	OFF	2:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'012A	ON (× 4)	OFF	1:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'A100	ON (× 3)	OFF	3:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'A101	ON (× 3)	OFF	3:1:1/2	25 MHz to 44.44 MHz	25 MHz to 44.44 MHz
	H'E100	ON (× 3)	OFF	1:1:1	25 MHz to 33.34 MHz	25 MHz to 33.34 MHz
	H'E101	ON (× 3)	OFF	1:1:1/2	25 MHz to 44.44 MHz	25 MHz to 44.44 MHz

Note: * Taking input clock as 1

Max. frequency: Iφ = 133.34 MHz, Bφ (CKIO) = 66.67 MHz, Pφ = 33.34 MHz

Cautions:

1. The input to divider 1 is the output of the PLL circuit 1:
   - When PLL circuit 1 is on.
2. The input of divider 2 is the output of the PLL circuit 1:
3. The frequency of the internal clock (Iφ) is:
   - The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 1 when PLL circuit 1 is on.
   - Do not set the internal clock frequency lower than the CKIO pin frequency.
4. The frequency of the peripheral clock (Pφ) becomes:
   - The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 2 when the clock operating mode is 0–2 or 7.
   - The peripheral clock frequency should not be set higher than the frequency of the CKIO pin, higher than 33 MHz, or lower than 1/8 the internal clock (Iφ).
5. The output frequency of PLL circuit 1 is the product of the CKIO frequency and the multiplication ratio of PLL circuit 1.
6. × 1, × 2, × 3, or × 4 can be used as the multiplication ratio of PLL circuit 1. × 1, × 1/2, × 1/3, and × 1/4 can be selected as the division ratios of dividers 1 and 2. Set the rate in the frequency control register. The on/off state of PLL circuit 2 and the multiplication ratio are determined by the mode.

10.4 Descriptions of Registers

The CPG includes the following register. Refer to section 23, Control Registers Table for more details of the address and access sizes.

- Frequency control register (FRQCR)

10.4.1 Frequency Control Register (FRQCR)

The frequency control register (FRQCR) is a 16-bit read/write register used to specify, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock and the peripheral clock. Only word access can be used on the FRQCR register.

Bit	Bit Name	Initial Value	R/W	Description
15 5 4	STC2 STC1 STC0	0 0 0	R/W R/W R/W	Frequency Multiplication Ratio These bits specify the frequency multiplication ratio of PLL circuit 1. 000: × 1 001: × 2 100: × 3 010: × 4 Other than the above: Reserved (Setting prohibited) Note: Do not set the output frequency of PLL circuit 1 higher than 133 MHz.
14 3 2	IFC2 IFC1 IFC0	0 0 0	R/W R/W R/W	Internal Clock Frequency Division Ratio These bits specify the frequency division ratio (Divider 1)of the internal clock with respect to the output frequency of PLL circuit 1. 000: × 1 001: × 1/2 100: × 1/3 010: × 1/4 Other than the above: Reserved (Setting prohibited) Note: Do not set the internal clock frequency lower than the CKIO frequency.

Bit	Bit Name	Initial Value	R/W	Description
13	PFC2	0	R/W	Peripheral Clock Frequency Division Ratio
1	PFC1	0	R/W	These bits specify the division ratio (Divider 2)of the peripheral clock frequency with respect to the frequency of the output frequency of PLL circuit 1 or the frequency of the CKIO pin. 000: × 1 001: × 1/2 100: × 1/3 010: × 1/4 101: × 1/6 Other than the above: Reserved (Setting prohibited) Note: Do not set the peripheral clock frequency higher than the frequency of the CKIO pin.
0	PFC0	0	R/W	
12 to 9, 7, 6	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	0	R	Reserved This bit is always read as 1. The write value should always be 1.

Note: Take enough care because the positions of the bits are not continuous.

10.5 Description of Operation

The frequency of the internal clock and peripheral clock can be changed either by changing the multiplication rate of PLL circuit 1 or by changing the division rates of dividers 1 and 2. All of these are controlled by software through the frequency control register. The methods are described below.

10.5.1 Changing the Multiplication Rate

A PLL settling time is required when the multiplication rate of PLL circuit 1 is changed. The on-chip WDT counts the settling time. Refer the section 11, Watch Dog Timer (WDT) for more details.

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 stop the WDT. The following must be set:
   WTCSR register TME bit = 0: WDT stops
   WTCSR register CKS2–CKS0 bits: Division ratio of WDT count clock
   WTCNT counter: Initial counter value
3. Set the desired value in the STC2, STC1 and STC0 bits. The division ratio can also be set in the IFC2–IFC0 bits and PFC2–PFC0 bits.
4. The processor pauses internally and the WDT starts incrementing. At this time, the internal (Iϕ) and peripheral clocks (Pϕ) both stop, and the clock is continuously output to the CKIO pin in clock modes 0 to 2.
5. Supply of the clock that has been set begins at WDT count overflow, and the processor begins operating again. The WDT stops after it overflows.

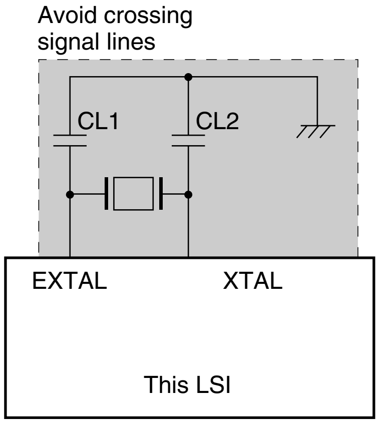
10.5.2 Changing the Division Ratio

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

1. In the initial state, IFC2–IFC0 = 000 and PFC2–PFC0 = 010.
2. Set the IFC2, IFC1, IFC0, PFC2, PFC1, and PFC0 bits to the new division ratio. The values that can be set are limited by the clock mode and the multiplication rate of PLL circuit 1. Note that if the wrong value is set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.

10.6 Notes for Usage

When Using an External Crystal Oscillator: Place the crystal oscillator, capacitors CL1 and CL2, close to the EXTAL and XTAL pins. To prevent induction from interfering with correct oscillation, use a common grounding point for the capacitors connected to the oscillator, and do not locate a wiring pattern near these components.



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

Figure 10.2 Points for Attention when Using Crystal Oscillator

Decoupling Capacitors: As far as possible, insert a laminated ceramic capacitor of 0.1 to 1 µF as a passive capacitor for each V_{SS}/V_{CC} pair. Mount the passive capacitors as close as possible to the SH7706 power supply pins, and use components with a frequency characteristic suitable for the chip's operating frequency, as well as a suitable capacitance value.

Digital system V_{SS}/V_{CC} pairs: 11-13, 19-21, 25-27, 37-39, 49-51, 61-63, 84-86, 93-95, 115-117, 137-139, 148-150, 156-158

On-chip oscillator V_{SS}/V_{CC} pairs: 1-4, 123-125, 126-128

When Using a PLL Oscillator Circuit: Keep the wiring from the PLL V_{CC} and V_{SS} connection pattern to the power supply pins short, and make the pattern width large, to minimize the inductance component. Ground the oscillation stabilization capacitors C1 and C2 to V_{SS} (PLL1) and V_{SS} (PLL2), respectively. Place C1 and C2 close to the CAP1 and CAP2 pins and do not locate a wiring pattern in the vicinity. In clock mode 7, connect the EXTAL pin to V_{CC} or V_{SS} and leave the XTAL pin open.

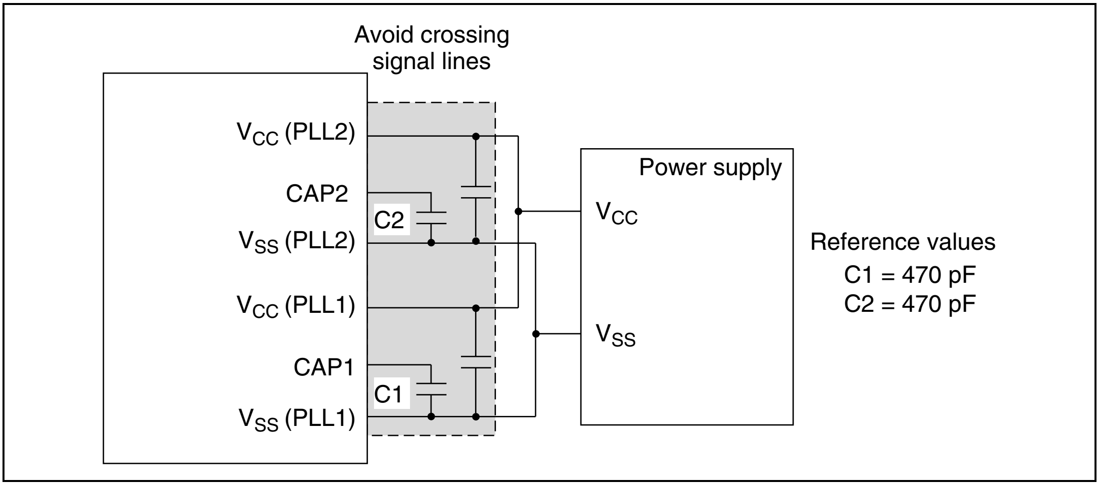


Figure 10.3 Points for Attention when Using PLL Oscillator Circuit

Notes on Wiring Power Supply Pins: To avoid crossing signal lines, wire V_{cc}–PLL1, V_{cc}–PLL2, 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}.

Section 11 Watchdog Timer (WDT)

The WDT is a single-channel timer that counts the clock settling time and is used when clearing software standby mode and temporary standbys, such as frequency changes. It can also be used as an ordinary watchdog timer or interval timer.

Figure 11.1 shows a block diagram of the WDT.

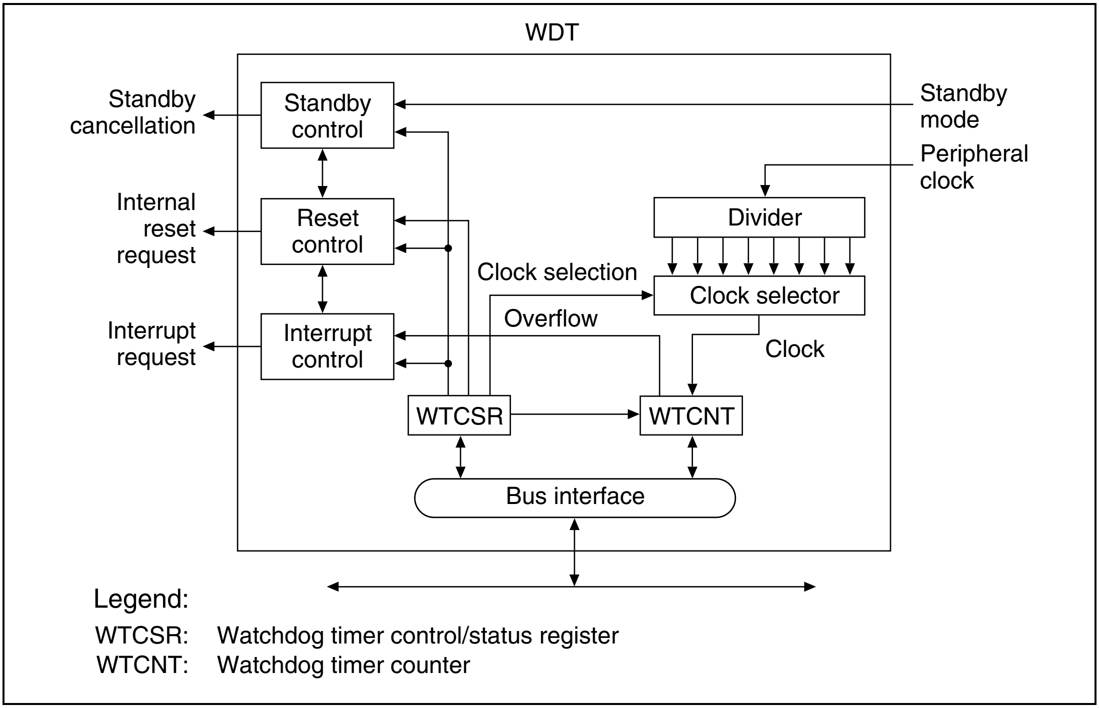


Figure 11.1 Block Diagram of the WDT

11.1 Features

The WDT has the following features:

- Can be used to ensure the clock setting time: Use the WDT to cancel software standby mode and the temporary standbys that occur when 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. Selection of power-on reset or manual reset.
- Generates interrupts in interval timer mode: Internal timer interrupts occur after counter overflow.

- Selection of eight counter input clocks. Eight clocks (×1 to × 1/4096) can be obtained by dividing the peripheral clock.

11.2 Description of Registers

The WDT has two registers that select the clock, switch the timer mode, and perform other functions. Refer to section 23, Control Registers Table, for more detail of the addresses and access size.

- Watchdog timer counter (WTCNT)
- Watchdog timer control/status register (WTCSR)

11.2.1 Watchdog Timer Counter (WTCNT)

The watchdog timer counter (WTCNT) is an 8-bit read/write register that increments on the selected clock. When an overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval time mode. The WTCNT is initialized to H'00 only by a power-on reset through the $\overline{\text{RESET}}$ pin. Use a word access to write to the WTCNT, with H'5A in the upper byte. Use a byte access to read WTCNT.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	0	R/W	8-bit counter

Note: The watchdog timer counter (WTCNT) is more difficult to write to than other registers to prevent from the erroneous writing to the register. Refer to section 11.2.3 Notes on Register Access.

11.2.2 Watchdog Timer Control/Status Register (WTCSR)

The watchdog timer control/status register (WTCSR) is an 8-bit read/write register composed of bits to select the clock used for the count, bits to select the timer mode, and overflow flags. The WTCSR is initialized to H'00 only by a power-on reset through the $\overline{\text{RESET}}$ pin. When a WDT overflow causes an internal reset, the WTCSR retains its value. When used to count the clock settling time for canceling a software standby, it retains its value after counter overflow. Use a word access to write to the WTCSR, with H'A5 in the upper byte. Use a byte access to read WTCSR.

Bit	Bit Name	Initial Value	R/W	Description
7	TME	0	R/W	Timer Enable Starts and stops timer operation. Clear this bit to 0 when using the WDT in software standby mode or when changing the clock frequency. 0: Timer disabled: Count-up stops and WTCNT value is retained 1: Timer enabled
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether to use the WDT as a watchdog timer or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: If WT/$\overline{\text{IT}}$ is modified when the WDT is running, the up-count may not be performed correctly.
5	RSTS	0	R/W	Reset Select Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Power-on reset 1: Manual reset
4	WOVF	0	R/W	Watchdog Timer Overflow Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode. 0: No overflow 1: WTCNT has overflowed in watchdog timer mode
3	IOVF	0	R/W	Interval Timer Overflow Indicates that the WTCNT has overflowed in interval timer mode. This bit is not set in watchdog timer mode. 0: No overflow 1: WTCNT has overflowed in interval timer mode

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>2 to 0</td><td>CKS2 to CKS0</td><td>0</td><td>R/W</td><td>Clock Select 2 to 0
These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock. The overflow period in the table is the value when the peripheral clock (Pϕ) is 15 MHz.
<table>
<tr><th>Clock Division Ratio</th><th>Overflow Period (when Pø = 15 MHz)</th></tr>
<tr><td>000: 1</td><td>17 µs</td></tr>
<tr><td>001: 1/4</td><td>68 µs</td></tr>
<tr><td>010: 1/16</td><td>273 µs</td></tr>
<tr><td>011: 1/32</td><td>546 µs</td></tr>
<tr><td>100: 1/64</td><td>1.09 ms</td></tr>
<tr><td>101: 1/256</td><td>4.36 ms</td></tr>
<tr><td>110: 1/1024</td><td>17.48 ms</td></tr>
<tr><td>111: 1/4096</td><td>69.91 ms</td></tr>
</table>
Note: If bits CKS2–CKS0 are modified when the WDT is running, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not running.</td></tr>
</table>

Note: The watchdog timer control/status register (WTCSR) is more difficult to write to than other registers to prevent from the erroneous writing to the register. Refer to section 11.2.3 Notes on Register Access.

11.2.3 Notes on Register Access

The WTCNT and WTCSR are more difficult to write to than other registers. The procedure for writing to these registers are given below.

Writing to WTCNT and WTCSR: These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction. When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 11.2. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

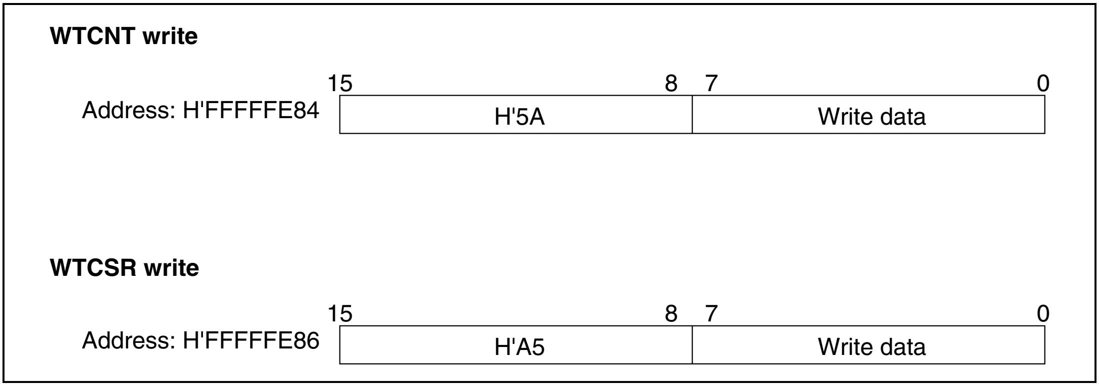


Figure 11.2 Writing to WTCNT and WTCSR

11.3 Description of Operation

11.3.1 Canceling Software Standbys

The WDT can be used to cancel software standby mode with an NMI or other interrupts. The procedure is described below. (The WDT does not run when resets are used for canceling, so keep the $\overline{\text{RESETP}}$ pin or $\overline{\text{RESETM}}$ pin low until the clock stabilizes.)

1. Before transitioning to software standby mode, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2–CKS0 bits in WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. Move to software standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting the edge change of the NMI signal or detecting interrupts.
5. When the WDT count overflows, the CPG starts supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
6. Since the WDT continues counting from H’00, set the STBY bit in the STBCR register to 0 in the interrupt processing program and this will stop the WDT. When the STBY bit remains 1, the SH7706 again enters the standby mode when the WDT has counted up to H’80. This standby mode can be canceled by power-on resets.

11.3.2 Changing the Frequency

To change the frequency used by the PLL, use the WDT. When changing the frequency only by switching the divider, do not use the WDT.

1. Before changing the frequency, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2–CKS0 bits of WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. When the frequency control register (FRQCR) is written, the clock stops and the processor enters standby mode temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
5. The counter stops at the values H'00–H'01. The stop value depends on the clock ratio.
6. Confirm that the value of WTCNT is H’00 before writing WTCNT, when WTCNT is written after the frequency change.

11.3.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in the WTCSR register to 1, set the reset type in the RSTS bit, set the type of count clock in the CKS2–CKS0 bits, and 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 watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and generates the type of reset specified by the RSTS bit. The counter then resumes counting.

   When a reset occurs, and a high level is output from the STATUS0 and STATUS1 pins. The signal output period is about one cycle of the count clock for power-on reset, and about five cycles of the peripheral clock for manual reset.

11.3.4 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{\text{IT}}$ bit in the WTCSR register to 0, set the type of count clock in the CKS2–CKS0 bits, and 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 interval timer interrupt request is sent to INTC. The counter then resumes counting.

Section 12 Timer Unit(TMU)

This LSI uses a three-channel (channel 0 to 2) 32-bit timer unit (TMU).

Figure 12.1 shows a block diagram of the TMU.

12.1 Features

The TMU has the following features:

- Each channel is provided with an auto-reload 32-bit down counter
- Channel 2 is provided with an input capture function
- All channels are provided with 32-bit constant registers and 32-bit down counters that can be read or written to at any time
- All channels generate interrupt requests when the 32-bit down counter underflows (H'00000000 → H'FFFFFFFF)
- Allows selection between 6 counter input clocks: External clock (TCLK), on-chip RTC output clock (16 kHz), Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256. (Pϕ is the internal clock for peripheral modules.) See section 10, Clock Pulse Generator (CPG), for more information.
- All channels can operate when this LSI is in software standby mode: When the RTC output clock is being used as the counter input clock, this LSI is still able to count in software standby mode.
- Synchronized read: TCNT is a sequentially changing 32-bit register. Since the peripheral module used has an internal bus width of 16 bits, a time lag can occur between the time when the upper 16 bits and lower 16 bits are read. To correct the discrepancy in the counter read value caused by this time lag, a synchronization circuit is built into the TCNT so that the entire 32-bit data in the TCNT can be read at once.
- The maximum operating frequency of the 32-bit counter is 2 MHz on all channels: Operate the SH7706 so that the clock input to the timer counters of each channel (obtained by dividing the external clock and internal clock with the prescaler) does not exceed the maximum operating frequency.

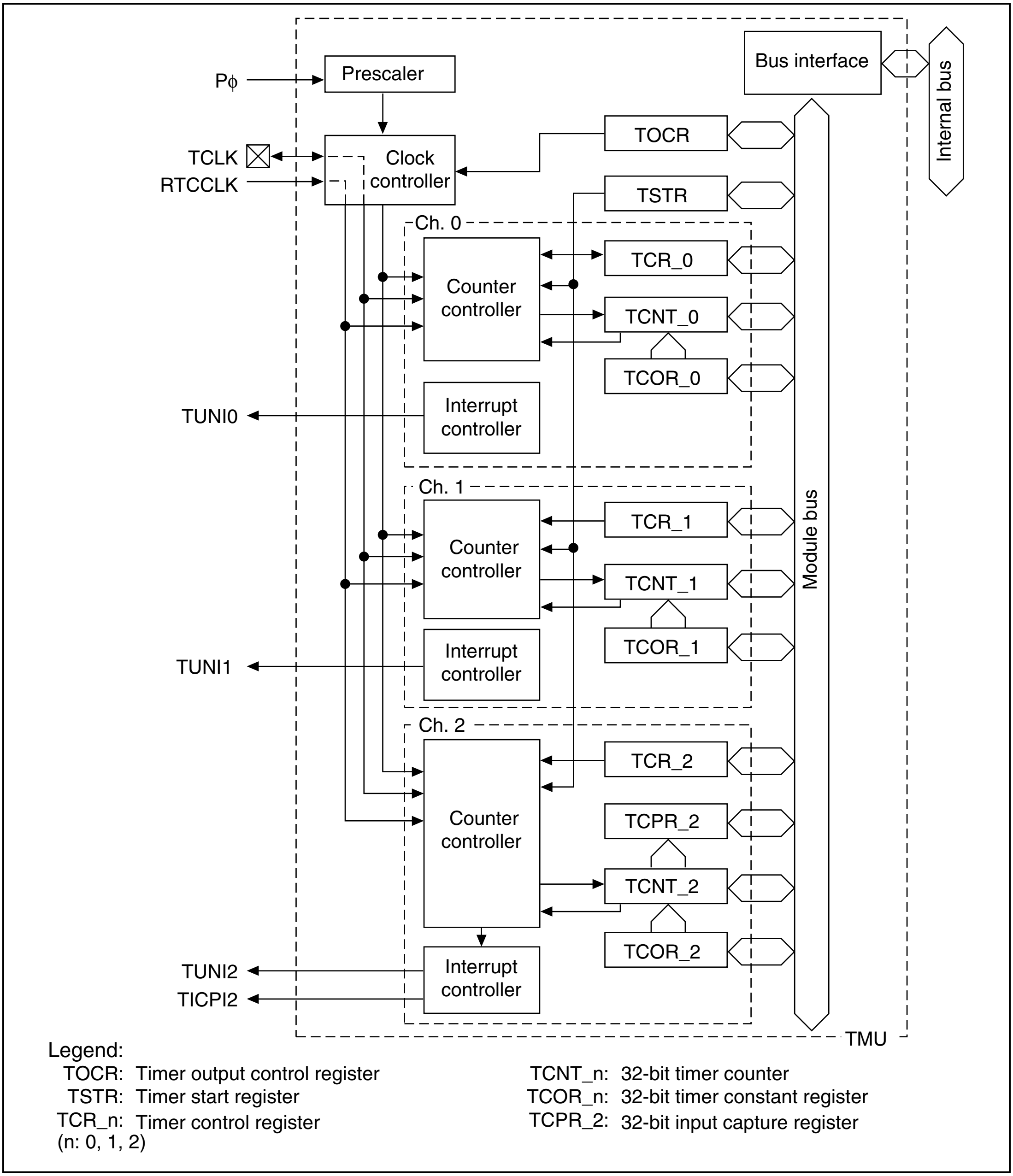


Figure 12.1 TMU Block Diagram

12.2 I/O Pins

Table 12.1 shows the pin configuration of the TMU.

Table 12.1 Pin Configuration

Channel	Pin	I/O	Description
Clock input/clock output	TCLK	I/O	External clock input pin/input capture control input pin/realtime clock (RTC) output pin

12.3 Description of Registers

The TMU has the following registers. For the addresses and access size of these registers, see section 23, Control Register Table.

- Timer output control register (TOCR)
- Timer start register (TSTR)
- Timer constant register 0 (TCOR_0)
- Timer counter 0 (TCNT_0)
- Timer control register 0 (TCR_0)
- Timer constant register 1 (TCOR_1)
- Timer counter 1 (TCNT_1)
- Timer control register 1 (TCR_1)
- Timer constant register 2 (TCOR_2)
- Timer counter 2 (TCNT_2)
- Timer control register 2 (TCR_2)
- Input capture register 2 (TCPR_2)

12.3.1 Timer Output Control Register (TOCR)

TOCR is an 8-bit read/write register that selects whether to use the external TCLK pin as an external clock or an input capture control usage input pin, or an output pin for the on-chip RTC output clock. TOCR is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TCOE	0	R/W	Timer Clock Pin Control Selects use of the timer clock pin (TCLK) as an external clock output pin or input pin for input capture control for the on-chip timer, or as an output pin for the on-chip RTC output clock. As the TCLK pin is multiplexed as the PTE6 pin, when the TCLK pin is used, bits PE6MD1 and PH7MD0 in the PECR register should be set to 00 (Other function). 0: Timer clock pin (TCLK) used as external clock input or input capture control input pin for the on-chip timer 1: Timer clock pin (TCLK) used as output pin for on-chip RTC output clock

12.3.2 Timer Start Register (TSTR)

TSTR is an 8-bit read/write register that selects whether to run or halt the timer counters (TCNT_0 to TCNT_2) for channels 0–2. TSTR is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode when the input clock selected for the channel is the on-chip RTC clock (RTCCLK). It is initialized in standby mode, changing the multiplying ratio of PLL circuit 1 or MSTP2 bit in STBCR is set to a logic one only when an external clock (TCLK) or the peripheral clock (Pϕ) is used as the input clock.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	0	R	Reserved These bits are always read 0. The write value should always be 0.
2	STR2	0	R/W	Counter Start 2 Selects whether to run or halt timer counter 2 (TCNT_2). 0: Halt TCNT_2 count 1: Start TCNT_2 counting
1	STR1	0	R/W	Counter Start 1 Selects whether to run or halt timer counter 1 (TCNT_1). 0: Halt TCNT_1 count 1: Start TCNT_1 counting
0	STR0	0	R/W	Counter Start 0 Selects whether to run or halt timer counter 0 (TCNT_0). 0: Halt TCNT_0 count 1: Start TCNT_0 counting

12.3.3 Timer Control Register 0–2 (TCR_0–TCR_2)

The timer control registers (TCR_0–TCR_2) control the timer counters (TCNT_0–TCNT_2) and interrupts. The TMU has three TCR_0–TCR_2 registers for each channel.

The TCR_0–TCR_2R registers are 16-bit read/write registers that control the issuance of interrupts when the flag indicating timer counter (TCNT_0–TCNT_2) underflow has been set to 1, and also carry out counter clock selection. When the external clock has been selected, they also select its edge. Additionally, TCR_2 controls the channel 2 input capture function and the issuance of interrupts during input capture. The TCR_0–TCR_2 are initialized to H'0000 by a power-on reset and manual reset. They are not initialized in standby mode.

In cases of Channel 0 and 1:

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	UNF	0	R/W	Underflow Flag Status flag that indicates occurrence of a TCNT_0 and TCNT_1 underflow. 0: TCNT has not underflowed. Clearing condition: When 0 is written to UNF 1: TCNT has underflowed Setting condition: When TCNT_0 and TCNT_1 underflows* Note: * Contents do not change when 1 is written to UNF.
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write value should always be 0.
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling of interrupt generation when the status flag (UNF) indicating TCNT_0 and TCNT_1 underflow has been set to 1. 0: Interrupt due to UNF (TUNI) is not enabled. 1: Interrupt due to UNF (TUNI) is enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the external clock edge when the external clock is selected, or when the input capture function is used. 00: Count/capture register set on rising edge 01: Count/capture register set on falling edge 1X: Count/capture register set on both rising and falling edge Note: X: Don't care
2	TPSC2	0	R/W	Timer Prescalers 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT_0 and TCNT_1 count clock.
0	TPSC0	0	R/W	000: Internal clock: count on Pϕ/4 001: Internal clock: count on Pϕ/16 010: Internal clock: count on Pϕ/64 011: Internal clock: count on Pϕ/256 100: Internal clock: count on clock output of on-chip RTC (RTCCLK) 101: External clock: count on TCLK pin input 110: Reserved (Setting prohibited) 111: Reserved (Setting prohibited)

In case of Channel 2:

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	ICPF	0	R	Input Capture Interrupt Flag A function of channel 2 only: the flag is set when input capture is requested via the TCLK pin. 0: No input capture request has been issued. Clearing condition: When 0 is written to ICPF 1: Input capture has been requested via the TCLK pin. Setting condition: When an input capture is requested via the TCLK pin* Note: * Contents do not change when 1 is written to ICPF.
8	UNF	0	R/W	Underflow Flag Status flag that indicates occurrence of a TCNT_2 underflow. 0: TCNT has not underflowed. Clearing condition: When 0 is written to UNF 1: TCNT has underflowed. Setting condition: When TCNT_2 underflows* Note: * Contents do not change when 1 is written to UNF.

Bit	Bit Name	Initial Value	R/W	Description
7	ICPE1	0	R/W	Input Capture Control
6	ICPE0	0	R/W	A function of channel 2 only: determines whether the input capture function can be used, and when used, whether or not to enable interrupts. When using this input capture function it is necessary to set the TCLK pin to input mode with the TCOE bit in the TOCR register. Additionally, use the CKEG bit to designate use of either the rising or falling edge of the TCLK pin to set the value of TCNT_2 in TCPR_2. 00: Input capture function is not used. 01: Reserved (Setting prohibited) 10: Input capture function is used. Interrupt due to ICPF (TICPI2) are not enabled. 11: Input capture function is used. Interrupt due to ICPF (TICPI2) are enabled.
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling of interrupt generation when the status flag (UNF) indicating TCNT_2 underflow has been set to 1. 0: Interrupt due to UNF (TUNI) is not enabled. 1: Interrupt due to UNF (TUNI) is enabled.
4	CKEG1	0	R/W	Clock Edge
3	CKEG0	0	R/W	These bits select the external clock edge when the external clock is selected, or when the input capture function is used. 00: Count/capture register set on rising edge 01: Count/capture register set on falling edge 1X: Count/capture register set on both rising and falling edge Note: X: Don’t care.

Bit	Bit Name	Initial Value	R/W	Description
2	TPSC2	0	R/W	Timer Prescalers
1	TPSC1	0	R/W	These bits select the TCNT_2 count clock.
0	TPSC0	0	R/W	000: Internal clock: count on Pϕ/4 001: Internal clock: count on Pϕ/16 010: Internal clock: count on Pϕ/64 011: Internal clock: count on Pϕ/256 100: Internal clock: count on clock output of on-chip RTC (RTCCLK) 101: External clock: count on TCLK pin input 110: Reserved (Setting prohibited) 111: Reserved (Setting prohibited)

12.3.4 Timer Constant Register 0–2 (TCOR_0–TCOR_2)

TCOR_0–TCOR_2 are specified the setting value for TCNT_0–TCNT_2 when TCNT_0–TCNT_2 are underflowed. TMU has 3 timer constant registers, one for each channel.

TCOR_0–TCOR_2 is a 32-bit read/write register. TCOR is initialized to H'FFFFFFFF by a power-on reset or manual reset; it is not initialized in standby mode, and retains its contents.

12.3.5 Timer Counters 0-2 (TCNT_0-TCNT_2)

TCNT counts down according to the input of a clock. The timer counters are 32-bit read/write registers. The TMU has three timer counters, one for each channel.The clock input is selected using the TPSC2–TPSC0 bits in the TCR_0-TCR_2.

When a TCNT count-down results in an underflow (H'00000000 → H'FFFFFFFF), the underflow flag (UNF) in the timer control register (TCR) of the relevant channel is set. The TCOR value is simultaneously set in TCNT itself and the count-down continues from that value.

Because the internal bus for this LSI on-chip supporting modules is 16 bits wide, a time lag can occur between the time when the upper 16 bits and lower 16 bits are read. Since TCNT counts sequentially, this time lag can create discrepancies between the data in the upper and lower halves. To correct the discrepancy, a buffer register is connected to TCNT so that upper and lower halves are not read separately. The entire 32-bit data in TCNT can thus be read at once.

TCNT is initialized to H'FFFFFFFF by a power-on reset or manual reset; it is not initialized in standby mode, and retains its contents.

12.3.6 Input Capture Register 2(TCPR_2)

The input capture register (TCPR_2) is a read-only 32-bit register built only into timer 2. Control of TCPR_2 setting conditions due to the TCLK pin is affected by the input capture function bits (ICPE1/ICPE2 and CKEG1/CKEG0) in TCR2. When a TCPR_2 setting indication due to the TCLK pin occurs, the value of TCNT_2 is copied into TCPR_2.

TCNT_2 is not initialized by a power-on reset or manual reset, or in standby mode.

12.4 Description of Operation

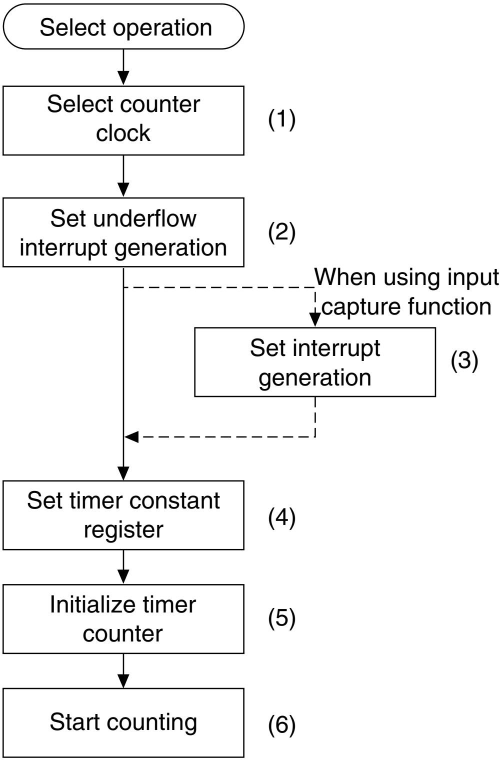
Each of three channels has a 32-bit timer counter (TCNT_0–TCNT_2) and a 32-bit timer constant register (TCOR_0–TCOR_2). The TCNT counts down. The auto-reload function enables synchronized counting and counting by external events. Channel 2 has an input capture function.

12.4.1 Counter Operation

When the STR0–STR2 bits in TSTR are set to 1, the corresponding 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 is 1, an interrupt request is sent to the CPU. Also at this time, the value is copied from TCOR to TCNT and the down-count operation is continued.

- **An example of the count operation setting flow**

The count operation is shown in figure 12.2.



1. Select the counter clock with the TPSC0-TPSC2 bits in the timer control register. If the external clock is selected, set the TCLK pin to input mode with the TOCE bit in TOCR, and select its edge with the CKEG1 and CKEG0 bits in the timer control register.
2. Use the UNIE bit in the timer control register to set whether to generate an interrupt when timer counter underflows.
3. When using the input capture function, set the ICPE bits in the timer control register, including the choice of whether or not to use the interrupt function (channel 2 only).
4. Set a value in the timer constant register (the cycle is the set value plus 1).
5. Set the initial value in the timer counter.
6. Set the STR bit in the timer start register to 1 to start operation.

Note: When an interrupt has been generated, clear the flag in the interrupt handler that caused it. If interrupts are enabled without clearing the flag, another interrupt will be generated.

Figure 12.2 Setting the Count Operation

- **Auto-Reload Count Operation:** Figure 12.3 shows the TCNT auto-reload operation.

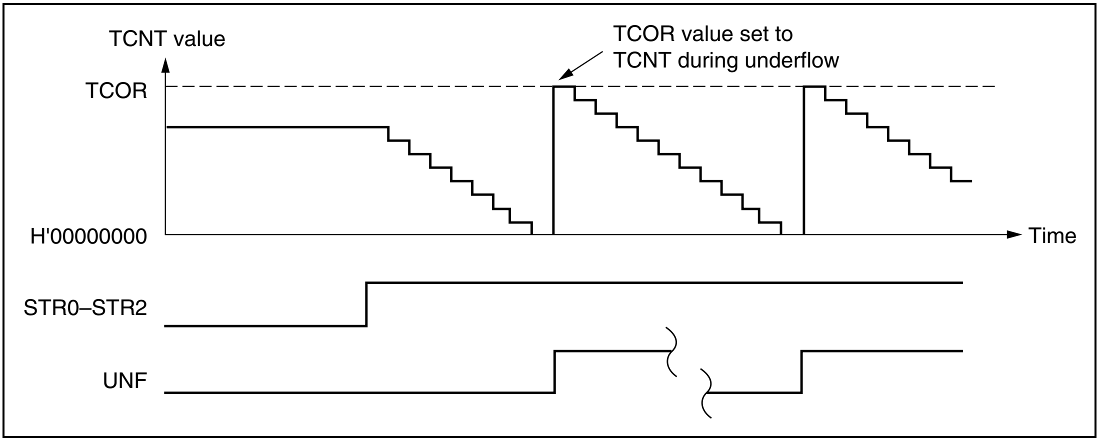


Figure 12.3 Auto-Reload Count Operation

- **TCNT Count Timing:**

1. Internal Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select whether peripheral module clock Pϕ or one of the four internal clocks created by dividing it is used (Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256). Figure 12.4 shows the timing.

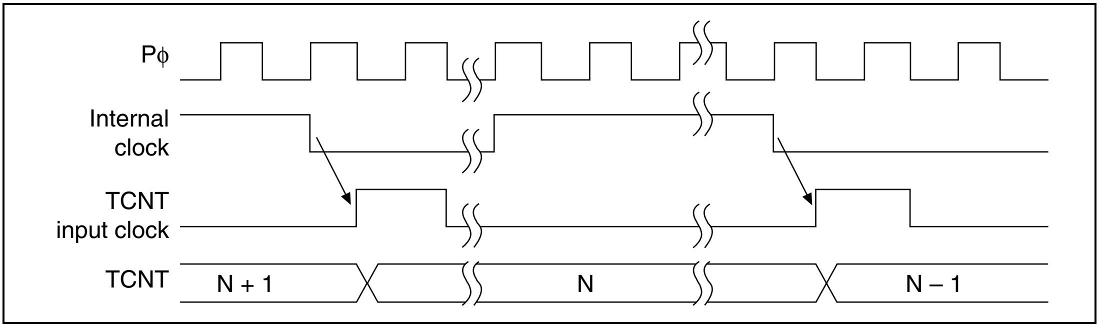


Figure 12.4 Count Timing when Internal Clock Is Operating

2. External Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select the external clock (TCLK) as the timer clock. Use the CKEG1 and CKEG0 bits in TCR to select the detection edge. Rise, fall or both may be selected. The pulse width of the external clock must be at least 1.5 peripheral module clock cycles for single edges or 2.5 peripheral module clock cycles for both edges. A shorter pulse width will result in accurate operation. Figure 12.5 shows the timing for both-edge detection.

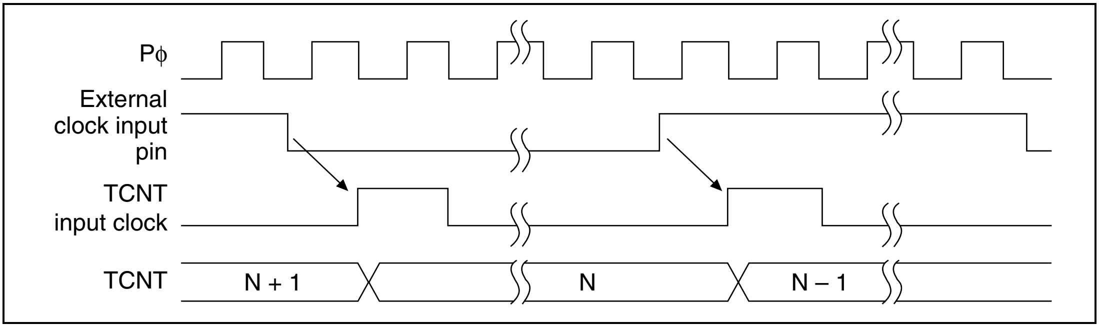


Figure 12.5 Count Timing when External Clock is Operating (Both Edges Detected)

3. On-Chip RTC Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select the on-chip RTC clock as the timer clock. Figure 12.6 shows the timing.

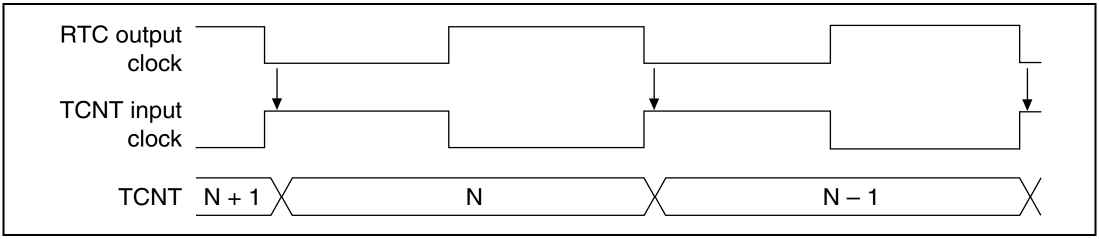


Figure 12.6 Count Timing when On-Chip RTC Clock Is Operating

12.4.2 Input Capture Function

Channel 2 has an input capture function (figure 12.7). When using the input capture function, set the TCLK pin to input mode with the TCOE bit in the timer output control register (TOCR) and set the timer operation clock to internal clock or on-chip RTC clock with the TPCS2–TPCS0 bits in the timer control register (TCR_2). Also, designate use of the input capture function and whether to generate interrupts on using it with the IPCE1–IPCE0 bits in TCR_2, and designate the use of either the rising or falling edge of the TCLK pin to set the timer counter (TCNT_2) value into the input capture register (TCPR_2) with the CKEG1–CKEG0 bits in TCR_2.

The input capture function cannot be used in standby mode.

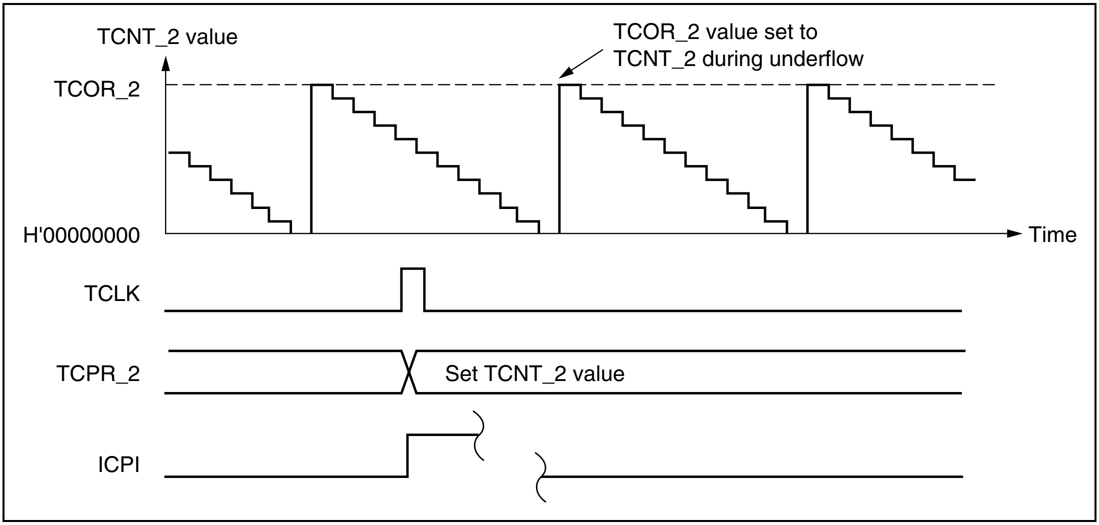


Figure 12.7 Operation Timing when Using the Input Capture Function (Using TCLK Rising Edge)

12.5 Interrupts

There are two sources of TMU interrupts: underflow interrupts (TUNI) and interrupts when using the input capture function (TICPI2).

12.5.1 Status Flag Set Timing

UNF is set to 1 when the TCNT underflows. Figure 12.8 shows the timing.

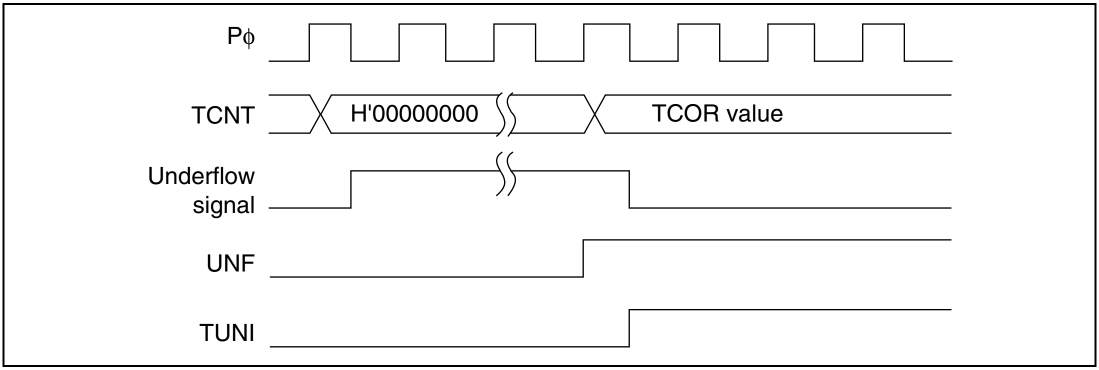


Figure 12.8 UNF Set Timing

12.5.2 Status Flag Clear Timing

The status flag can be cleared by writing 0 from the CPU. Figure 12.9 shows the timing.

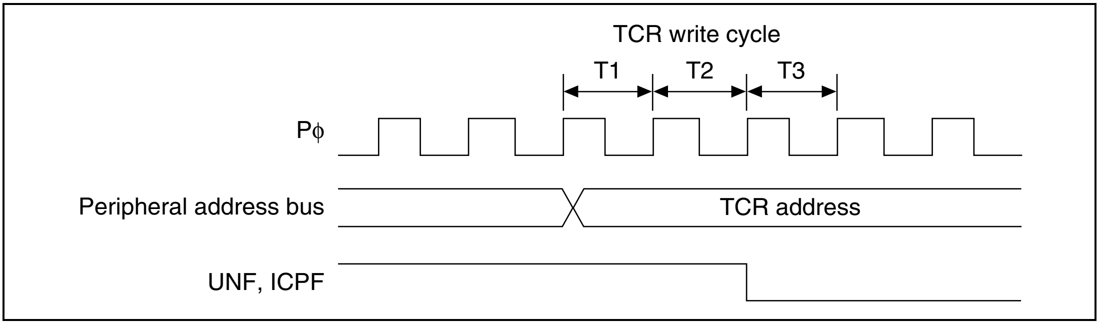


Figure 12.9 Status Flag Clear Timing

12.5.3 Interrupt Sources and Priorities

The TMU produces underflow interrupts for each channel. When the interrupt request flag and interrupt enable bit are both set to 1, the interrupt is requested. Codes are set in the exception event register (INTEVT, INTEVT2) for these interrupts and interrupt processing occurs according to the codes.

The relative priorities of channels can be changed using the interrupt controller (see section 4, Exception Processing, and section 6, Interrupt Controller (INTC)). 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 Notes for Usage

12.6.1 Writing to Registers

Synchronization processing is not performed for timer counting during register writes. When writing to registers, always clear the appropriate start bits for the channel (STR2−STR0) in the timer start register (TSTR) to halt timer counting.

12.6.2 Reading Registers

Synchronization processing is performed for timer counting during register reads. When timer counting and register read processing are performed simultaneously, the register value before TCNT counting down (with synchronization processing) is read.

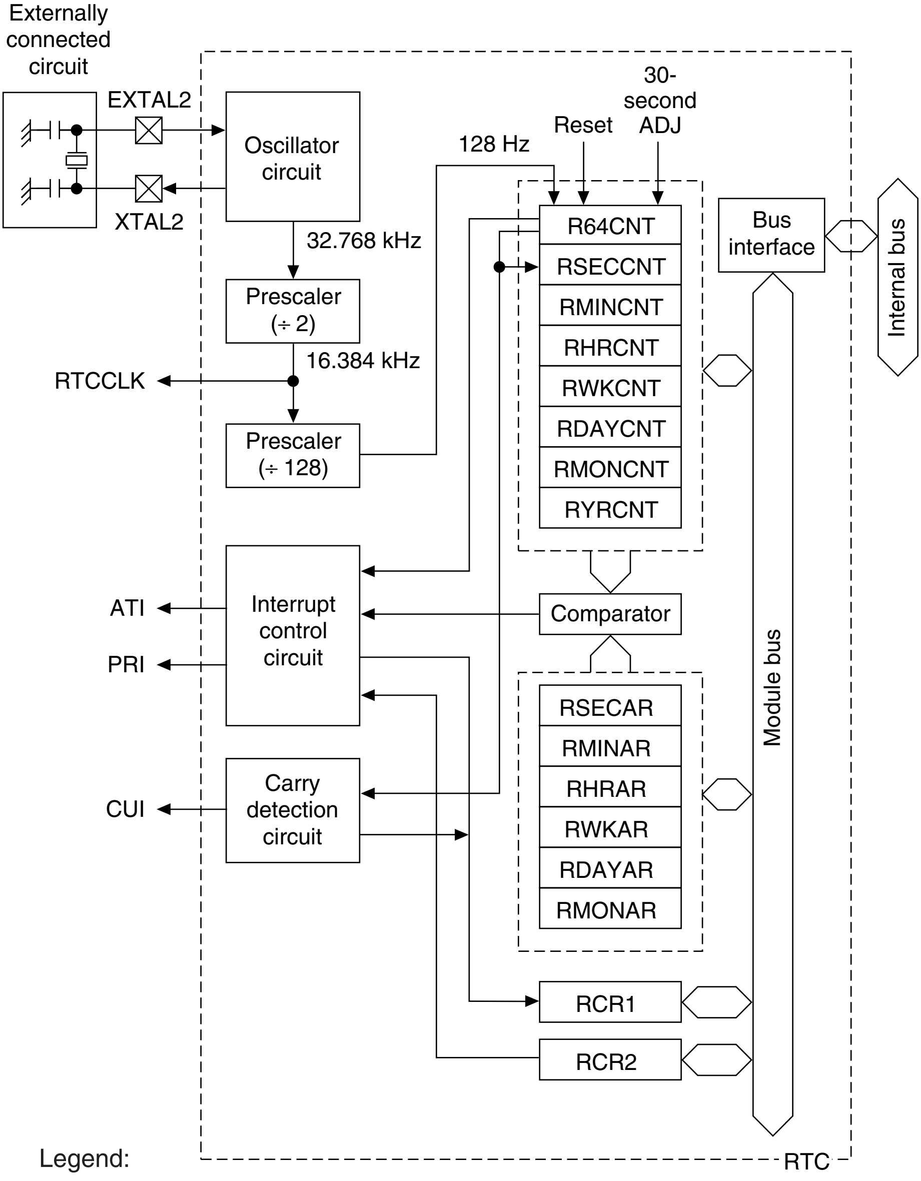
Section 13 Realtime Clock (RTC)

This LSI has a realtime clock (RTC) with its own 32.768-kHz crystal oscillator. A block diagram of the RTC is shown in figure 13.1.

13.1 Features

The RTC has following features:

- Clock and calendar functions (BCD display): seconds, minutes, hours, date, day of the week, month, and year
- 1-Hz to 64-Hz timer (binary display)
- Start/stop function
- 30-second adjust function
- Alarm interrupt: frame comparison of seconds, minutes, hours, date, day of the week, and month can be used as conditions for the alarm interrupt
- Cyclic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year correction



Legend:

R64CNT:	64-Hz counter	RSECAR:	Second alarm register
RSECCNT:	Second counter	RHRAR:	Minute alarm register
RMINCNT:	Minute counter	RMINAR:	Hour alarm register
RHRCNT:	Hour counter	RWKAR:	Day of the week alarm register
RWKCNT:	Day of the week counter	RDAYAR:	Date alarm register
RDAYCNT:	Date counter	RMONAR:	Month alarm register
RMONCNT:	Month counter	RCR1:	RTC control register 1
RYRCNT:	Year counter	RCR2:	RTC control register 2

Figure 13.1 RTC Block Diagram

13.2 I/O Pins

Table 13.1 shows the RTC pin configuration.

Table 13.1 RTC Pin Configuration

Pin	Abbreviation	I/O	Description
RTC oscillator crystal pin	EXTAL2	I	Connects crystal to RTC oscillator*[2]
RTC oscillator crystal pin	XTAL2	O	Connects crystal to RTC oscillator*[2]
Clock input/clock output	TCLK	I/O	External clock input pin/input capture control input pin/realtime clock (RTC) output pin (shared by TMU)
Dedicated power-supply pin for RTC	V_{CC}-RTC	—	Dedicated power-supply pin for RTC*[1]
Dedicated GND pin for RTC	V_{SS}-RTC	—	Dedicated GND pin for RTC*[1]

Note: 1. Power must be supplied to the RTC power supply pins even when the RTC is not used. Even if only the RTC is used, power must be supplied to all power supply pins, including these pins.

In software standby mode, also, power must be supplied to all power supply pins, including these pins.

In hardware standby mode, it is possible to supply no power to the other power supply pins then RTC pins.

2. Pull-up(Vcc) EXTAL2, and supply NC to XTAL2 when the RTC is not used.

13.3 Description of Registers

RTC has the registers listed below. Refer to section 23, Control Registers Table, for more detail of the 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)
- RTC control register 1 (RCR1)
- RTC control register 2 (RCR2)

13.3.1 64-Hz Counter (R64CNT)

The 64-Hz counter (R64CNT) is an 8-bit read-only register that indicates the state of the RTC divider circuit between 64 Hz and 1 Hz.

R64CNT is reset to H'00 by setting the RESET bit in RCR2 or the ADJ bit in RCR2 to 1.

R64CNT 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	Always read as 0.
6 to 0	—	—	R	64Hz counter Each bit (bits 6 to 0) indicates the state of the RTC divider circuit between 64 and 1Hz. Bit Frequency 6: 1Hz 5: 2Hz 4: 4Hz 3: 8Hz 2: 16Hz 1: 32Hz 0: 64Hz

13.3.2 Second Counter (RSECCNT)

The second counter (RSECCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded second section of the RTC. The count operation is performed by a carry for each second of the 64-Hz counter.

The range of second can be set is 00–59 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2.

RSECCNT 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	Always read as 0.
6 to 4	—	—	R/W	Counter for 10-unit of second in the BCD-code. The range can be set from 0 to 5 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of second in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.3 Minute Counter (RMINCNT)

The minute counter (RMINCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded minute section of the RTC. The count operation is performed by a carry for each minute of the second counter.

The range of minute can be set is 00–59 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in 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	Always read as 0.
6 to 4	—	—	R/W	Counter for 10-unit of minute in the BCD-code. The range can be set from 0 to 5 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of minute in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.4 Hour Counter (RHRCNT)

The hour counter (RHRCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded hour section of the RTC. The count operation is performed by a carry for each 1 hour of the minute counter.

The range of hour can be set is 00–23 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2.

RHRCNT 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	Always read as 0.
5, 4	—	—	R/W	Counter for 10-unit of hour in the BCD-code. The range can be set from 0 to 2 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of hour in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.5 Day of the Week Counter (RWKCNT)

The day of the week counter (RWKCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded day of week section of the RTC. The count operation is performed by a carry for each day of the date counter.

The range for day of the week can be set is 0–6 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START 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	Always read as 0.
2 to 0	—	—	R/W	Counter for the day of week in the BCD-code. The range can be set from 0 to 6 (decimal). Code Day of Week 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday

13.3.6 Date Counter (RDAYCNT)

The date counter (RDAYCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded date section of the RTC. The count operation is performed by a carry for each day of the hour counter.

The range of date can be set is 01–31 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2.

RDAYCNT is not initialized by a power-on reset or manual reset, or in standby mode.

The RDAYCNT range that can be set changes with each month and in leap years. Please confirm the correct setting.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	0	R	Always read as 0.
5, 4	—	—	R/W	Counter for 10-unit of date in the BCD-code. The range can be set from 0 to 3 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of date in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.7 Month Counter (RMONCNT)

The month counter (RMONCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded month section of the RTC. The count operation is performed by a carry for each month of the date counter.

The range of month can be set is 00–12 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in 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	Always read as 0.
4	—	—	R/W	Counter for 10-unit of month in the BCD-code. The range can be set from 0 to 1 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of month in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.8 Year Counter (RYRCNT)

The year counter (RYRCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded year section of the RTC. The least significant 2 digits of the western calendar year are displayed. The count operation is performed by a carry for each year of the month counter.

The range for year can be set is 00–99 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2 or using a carry flag.

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 fractional result of 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	R/W	Counter for 10-unit of year in the BCD-code. The range can be set from 0 to 9 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of year in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.9 Second Alarm Register (RSECAR)

The second alarm register (RSECAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded second section counter RSECCNT of the RTC. When the ENB bit is set to 1, a comparison with the RSECCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of second can be set is 00–59 (decimal). Errant operation will result if any other value is set.

The ENB bit in RSECAR is initialized to 0 by a power-on reset. The remaining RSECAR fields are 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	Second Alarm Enable 0: No compared 1: Compared
6 to 4	—	—	R/W	Setting value for 10-unit of second alarm in the BCD-code. The range can be set from 0 to 5 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of second alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.10 Minute Alarm Register (RMINAR)

The minute alarm register (RMINAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded minute section counter RMINCNT of the RTC. When the ENB bit is set to 1, a comparison with the RMINCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of minute can be set is 00–59 (decimal). Errant operation will result if any other value is set.

The ENB bit in RMINAR is initialized by a power-on reset. The remaining RMINAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Minute Alarm Enable 0: No compared 1: Compared
6 to 4	—	—	R/W	Setting value for 10-unit of minute alarm in the BCD-code. The range can be set from 0 to 5 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of minute alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.11 Hour Alarm Register (RHRAR)

The hour alarm register (RHRAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded hour section counter RHRCNT of the RTC. When the ENB bit is set to 1, a comparison with the RHRCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of hour can be set is 00–23 (decimal). Errant operation will result if any other value is set.

The ENB bit in RHRAR is initialized by a power-on reset. The remaining RHRAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Hour Alarm Enable 0: No compared 1: Compared
6	—	0	R	Always read as 0.
5, 4	—	—	R/W	Setting value for 10-unit of hour alarm in the BCD-code. The range can be set from 0 to 2 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of hour alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.12 Day of the Week Alarm Register (RWKAR)

The day of the week alarm register (RWKAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded day of week section counter RWKCNT of the RTC. When the ENB bit is set to 1, a comparison with the RWKCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of day of the week can be set 0–6 (decimal). Errant operation will result if any other value is set.

The ENB bit in RWKAR is initialized by a power-on reset. The remaining RWKAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Day of the week Alarm Enable 0: No compared 1: Compared
6 to 3	—	0	R	Always read as 0.
2 to 0	—	—	R/W	Setting value for day of the week alarm in the BCD-code. The range can be set from 0 to 6 (decimal). Code Day of the Week 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday

13.3.13 Date Alarm Register (RDAYAR)

The date alarm register (RDAYAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded date section counter RDAYCNT of the RTC. When the ENB bit is set to 1, a comparison with the RDAYCNT value is performed. From among the registers RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of date can be set 01–31 (decimal). Errant operation will result if any other value is set. The RDAYCNT range that can be set changes with some months and in leap years. Please confirm the correct setting.

The ENB bit in RDAYAR is initialized by a power-on reset. The remaining RDAYAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Date Alarm Enable 0: No compared 1: Compared
6	—	0	R	Always read as 0.
5, 4	—	—	R/W	Setting value for 10-unit of date alarm in the BCD-code. The range can be set from 0 to 3 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of date alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.14 Month Alarm Register (RMONAR)

The month alarm register (RMONAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded month section counter RMONCNT of the RTC. When the ENB bit is set to 1, a comparison with the RMONCNT value is performed. From among the registers RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of month can be set 01–12 (decimal). Errant operation will result if any other value is set.

The ENB bit in RMONAR is initialized by a power-on reset. The remaining RMONAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Month Alarm Enable 0: No compared 1: Compared
6, 5	—	0	R	Always read as 0.
4	—	—	R/W	Setting value for 10-unit of month alarm in the BCD-code. The range can be set from 0 to 1 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of month alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

13.3.15 RTC Control Register 1 (RCR1)

The RTC control register 1 (RCR1) is an 8-bit read/write register that affects carry flags and alarm flags. It also selects whether to generate interrupts for each flag. Because flags are sometimes set after an operand read, do not use this register in read-modify-write processing.

RCR1 is initialized to H'00 by a power-on reset. In a manual reset, all bits are initialized to 0 except for the CF flag, which is undefined. When using the CF flag, it must be initialized beforehand. This register is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CF	0	R/W	Carry Flag Status flag that indicates that a carry has occurred. CF is set to 1 when a count-up to R64CNT or RSECCNT occurs. A count register value read at this time cannot be guaranteed; another read is required. 0: No count up of R64CNT or RSECCNT. Clearing condition: When 0 is written to CF 1: Count up of R64CNT or RSECCNT. Setting condition: When 1 is written to CF
6	—	0	R	Reserved
5	—	0	R	These bits are always read as 0. The write value should always be 0.
4	CIE	0	R/W	Carry Interrupt Enable Flag When the carry flag (CF) is set to 1, the CIE bit enables interrupts. 0: A carry interrupt is not generated when the CF flag is set to 1 1: A carry interrupt is generated when the CF flag is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag When the alarm flag (AF) is set to 1, the AIE bit allows interrupts. 0: An alarm interrupt is not generated when the AF flag is set to 1 1: An alarm interrupt is generated when the AF flag is set to 1

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R	Reserved
1	—	0	R	These bits are always read as 0. The write value should always be 0.
0	AF	0	R/W	Alarm Flag The AF flag is set to 1 when the alarm time set in an alarm register (only registers with ENB bit set to 1) matches the clock and calendar time. This flag is cleared to 0 when 0 is written, but holds the previous value when 1 is to be written. 0: Clock/calendar and alarm register have not matched since last reset to 0. Clearing condition: When 0 is written to AF 1: Setting condition: Clock/calendar and alarm register have matched (only registers that ENB bit is 1)

13.3.16 RTC Control Register 2 (RCR2)

The RTC control register 2 (RCR2) is an 8-bit read/write register for periodic interrupt control, 30-second adjustment ADJ, divider circuit RESET, and RTC count start/stop control. It is initialized to H'09 by a power-on reset. It is initialized except for RTCEN and START by a 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 period designated by the PES bits. When set to 1, PEF generates periodic interrupts. 0: Interrupts not generated with the period designated by the PES bits. Clearing condition: When 0 is written to PEF 1: Interrupts generated with the period designated by the PES bits. Setting condition: When 1 is written to PEF

Bit	Bit Name	Initial Value	R/W	Description
6	PES2	0	R/W	Periodic Interrupt Flags
5	PES1	0	R/W	These bits specify the periodic interrupt.
4	PES0	0	R/W	000: No periodic interrupts generated 001: Periodic interrupt generated every 1/256 second 010: Periodic interrupt generated every 1/64 second 011: Periodic interrupt generated every 1/16 second 100: Periodic interrupt generated every 1/4 second 101: Periodic interrupt generated every 1/2 second 110: Periodic interrupt generated every 1 second 111: Periodic interrupt generated every 2 seconds
3	RTCEN	1	R/W	Controls the operation of the crystal oscillator for the RTC. 0: Halts the crystal oscillator for the RTC. 1: Runs the crystal oscillator for the RTC.
2	ADJ	0	R/W	30 Second Adjustment When 1 is written to the ADJ bit, times of 29 seconds or less will be rounded to 00 seconds and 30 seconds or more to 1 minute. The divider circuit will be simultaneously reset. This bit always reads 0. 0: Runs normally. 1 : 30-second adjustment.
1	RESET	0	R/W	Reset When 1 is written, initializes the divider circuit (RTC prescaler and R64CNT). This bit always reads 0. 0: Runs normally. 1: Divider circuit is reset.

Bit	Bit Name	Initial Value	R/W	Description
0	START	1	R/W	Start Bit Halts and restarts the counter (clock). 0: Second/minute/hour/day/week/month/year counter halts. 1: Second/minute/hour/day/week/month/year counter runs normally. Note: The 64-Hz counter always runs unless stopped with the RTCEN bit.

13.4 RTC Operation

13.4.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

13.4.2 Setting the Time

Figure 13.2 shows how to set the time when the clock is stopped. This works when the entire calendar or clock is to be set.

It is easy to set the time using a software program.

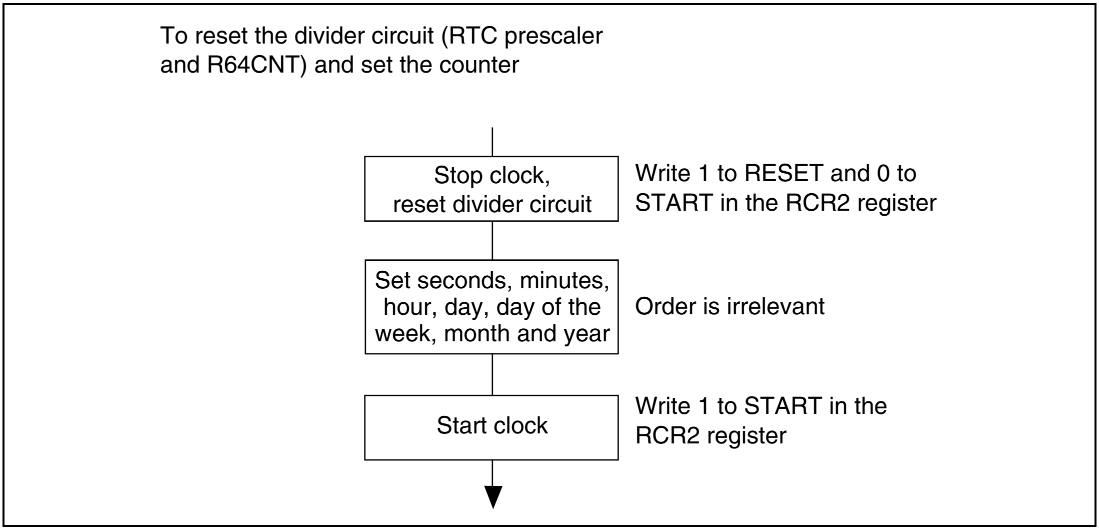

To reset the divider circuit (RTC prescaler and R64CNT) and set the counter
Stop clock, reset divider circuit
Write 1 to RESET and 0 to START in the RCR2 register
Set seconds, minutes, hour, day, day of the week, month and year
Order is irrelevant
Start clock
Write 1 to START in the RCR2 register

Figure 13.2 Setting the Time

13.4.3 Reading the Time

Figure 13.3 shows how to read the time. If a carry occurs while reading the time, the correct time will not be obtained, so it must be read again. Part (a) in figure 13.3 shows the method of reading the time without using interrupts; part (b) in figure 13.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

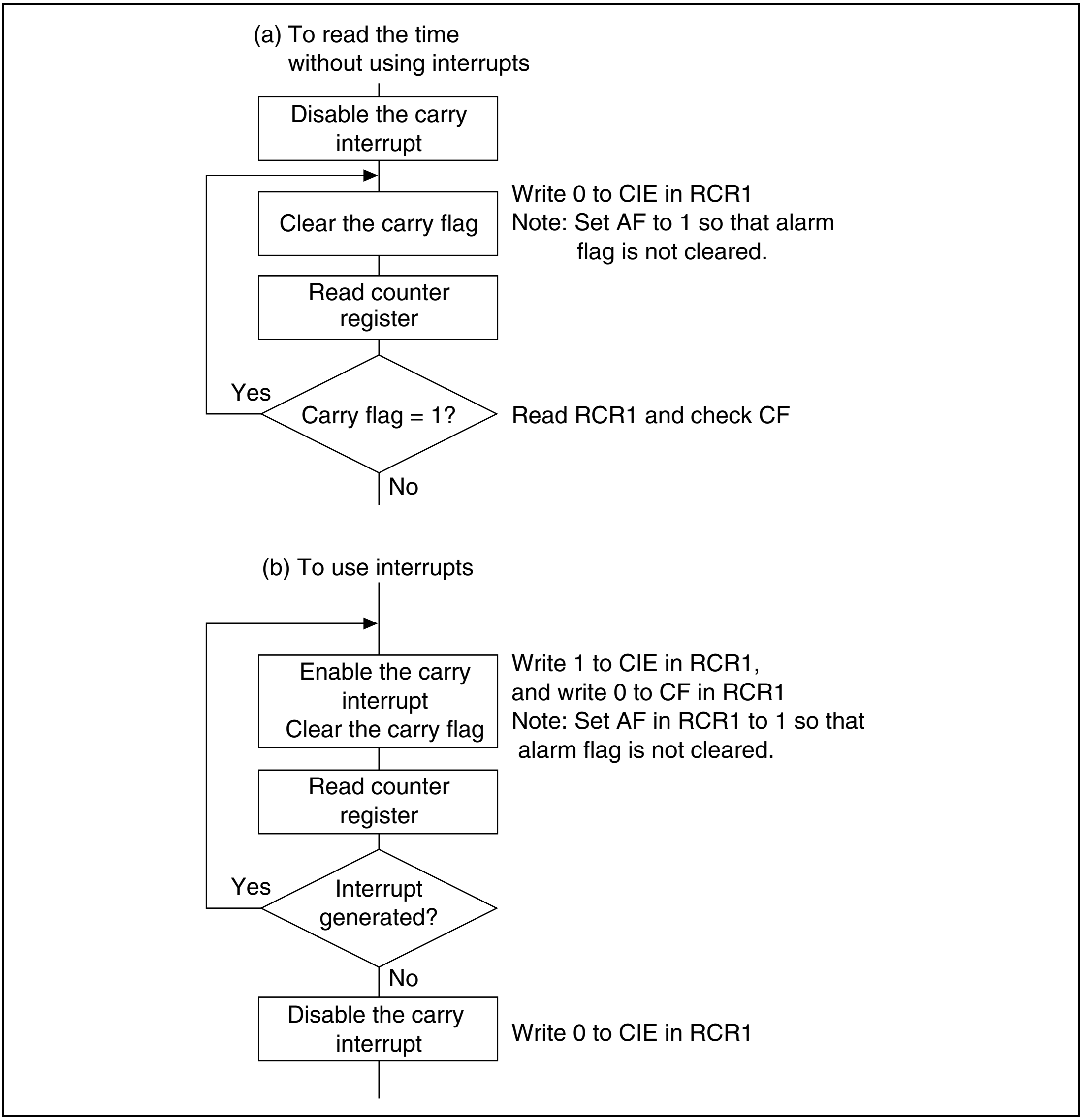

(a) To read the time without using interrupts
Disable the carry interrupt
Clear the carry flag
Write 0 to CIE in RCR1
Note: Set AF to 1 so that alarm flag is not cleared.
Read counter register
Yes
Carry flag = 1?
Read RCR1 and check CF
No
(b) To use interrupts
Enable the carry interrupt
Clear the carry flag
Write 1 to CIE in RCR1, and write 0 to CF in RCR1
Note: Set AF in RCR1 to 1 so that alarm flag is not cleared.
Read counter register
Yes
Interrupt generated?
No
Disable the carry interrupt
Write 0 to CIE in RCR1

Figure 13.3 Reading the Time

13.4.4 Alarm Function

Figure 13.4 shows how to use the alarm function.

Alarms can be generated using seconds, minutes, hours, day of the week, date, month, or any combination of these. Set the ENB bit (bit 7) in the register on which the alarm is placed to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register on which the alarm is placed to 0.

When the clock and alarm times match, 1 is set in the AF bit (bit 0) in RCR1. Alarm detection can be checked by reading this bit, but normally it is done by interrupt. If 1 is placed in the AIE bit (bit 3) in RCR1, an interrupt is generated when an alarm occurs.

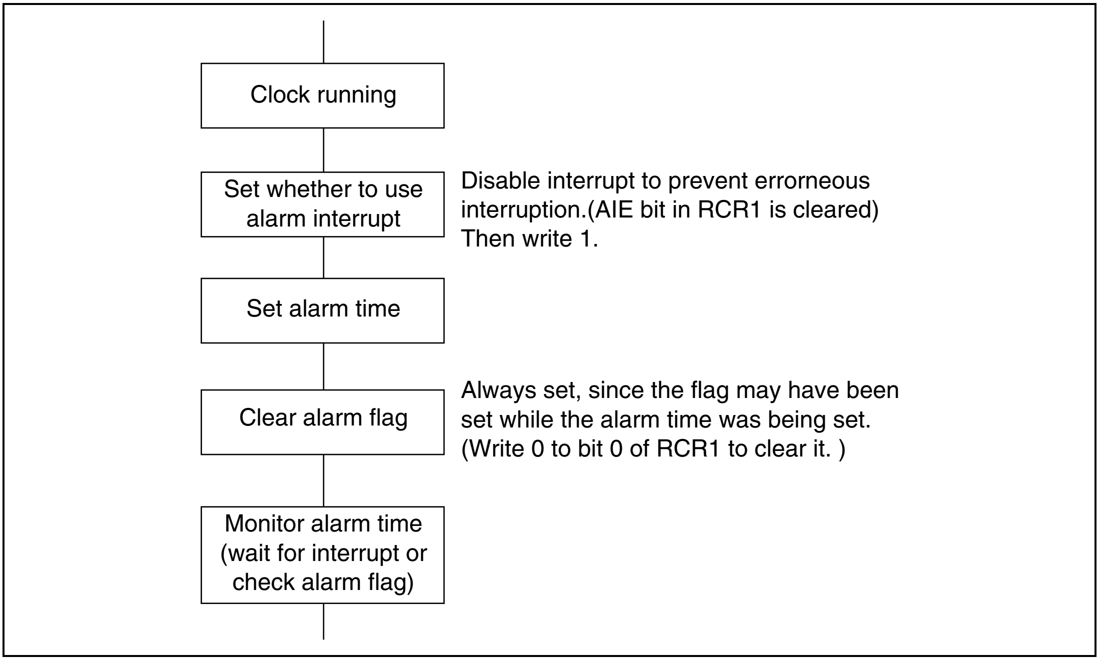

Clock running
Set whether to use alarm interrupt
Disable interrupt to prevent errorneous interruption.(AIE bit in RCR1 is cleared) Then write 1.
Set alarm time
Clear alarm flag
Always set, since the flag may have been set while the alarm time was being set. (Write 0 to bit 0 of RCR1 to clear it.)
Monitor alarm time (wait for interrupt or check alarm flag)

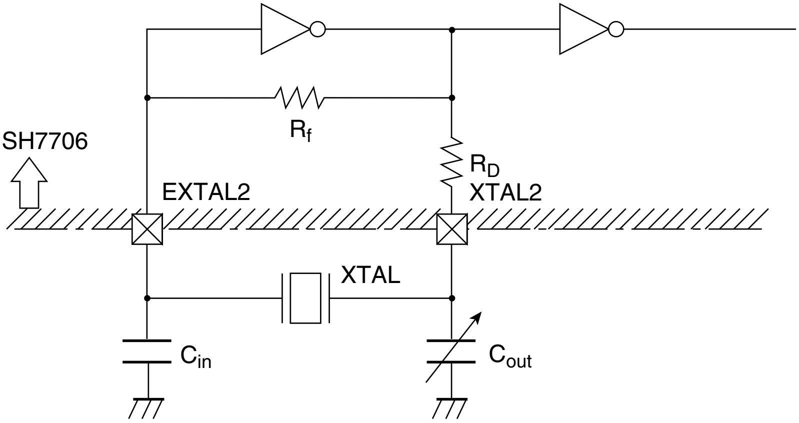
Figure 13.4 Using the Alarm Function

13.4.5 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in table 13.2, and the RTC crystal oscillator circuit in figure 13.5.

Table 13.2 Recommended Oscillator Circuit Constants (Recommended Values)

fosc	Cin	Cout
32.768 kHz	10 to 22 pF	10 to 22 pF



Notes: 1. Select either the C_{in} or C_{out} side for frequency adjustment variable capacitor according to requirements such as frequency range, degree of stability, etc.

2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 kΩ

3. C_{in} and C_{out} values include floating capacitance due to the wiring. Take care when using a ground plane.

4. The crystal oscillation settling time depends on the mounted circuit constants, floating 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 13.5 Example of Crystal Oscillator Circuit Connection

13.5 Notes for Usage

13.5.1 Register Writing during RTC Count

The following RTC registers cannot be written to during an RTC count (while bit 0 = 1 in RCR2).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCNT

The RTC count must be halted before writing to any of the above registers.

13.5.2 Use of Realtime Clock (RTC) Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 13.6.

A periodic interrupt can be generated periodically at the interval set by the periodic interrupt enable flag (PES0–PES2) in RCR2. When the time set by the PES0–PES2 has elapsed, the PEF is set to 1.

The PEF is cleared to 0 upon periodic interrupt generation when the periodic interrupt enable flag (PES0 to PES2) is set. Periodic interrupt generation can be confirmed by reading this bit, but normally the interrupt function is used.

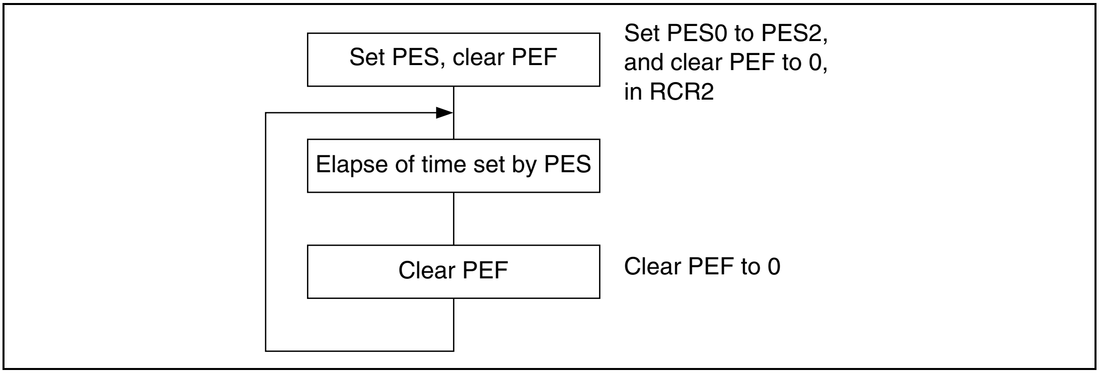

Set PES, clear PEF
Set PES0 to PES2, and clear PEF to 0, in RCR2
Elapse of time set by PES
Clear PEF
Clear PEF to 0

Figure 13.6 Using Periodic Interrupt Function

Section 14 Serial Communication Interface (SCI)

This LSI has an on-chip serial communication interface (SCI) that supports both asynchronous and clock synchronous serial communication. It also has a multiprocessor communication function for serial communication among two or more processors. A block diagram of SCI is shown in figure 14.1, and the I/O ports are shown in figures 14.2 to 14.4.

14.1 Features

- **Selectable from asynchronous or clock synchronous as the serial communications mode.**
- Asynchronous mode:
  - Serial data communications are synched by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. It can also communicate with two or more other processors using the multiprocessor communication function. There are 12 selectable serial data communication formats.
  - Data length: Seven or eight bits
  - Stop bit length: One or two bits
  - Parity: Even, odd, or none
  - Multiprocessor bit: 1 or 0
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: By reading the RxD0 pin level directly from the port Serial communication port data register (SCPDR) when a framing error occurs
- Clock synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a clock synchronous communication function. One serial data communication format is available.
  - Data length: Eight bits
  - Receive error detection: Overrun errors
- **Full duplex communication**

  The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so continuous data transfer is possible in both the transmit and receive directions.
- **On-chip baud rate generator with selectable bit rates**
- **Internal or external transmit/receive clock source**

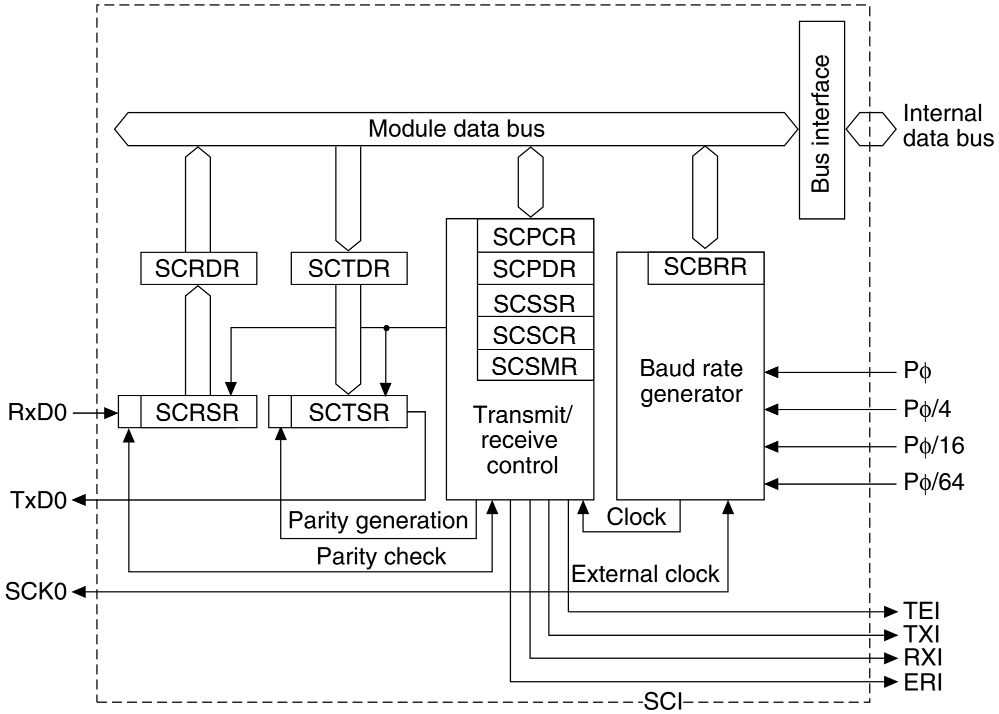
  From either baud rate generator (internal) or SCK0 pin (external)
- **Four types of interrupts**

Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently.

- **Saving power:**

  When the SCI is not in use, it can be stopped by halting the clock supply for the saving power.

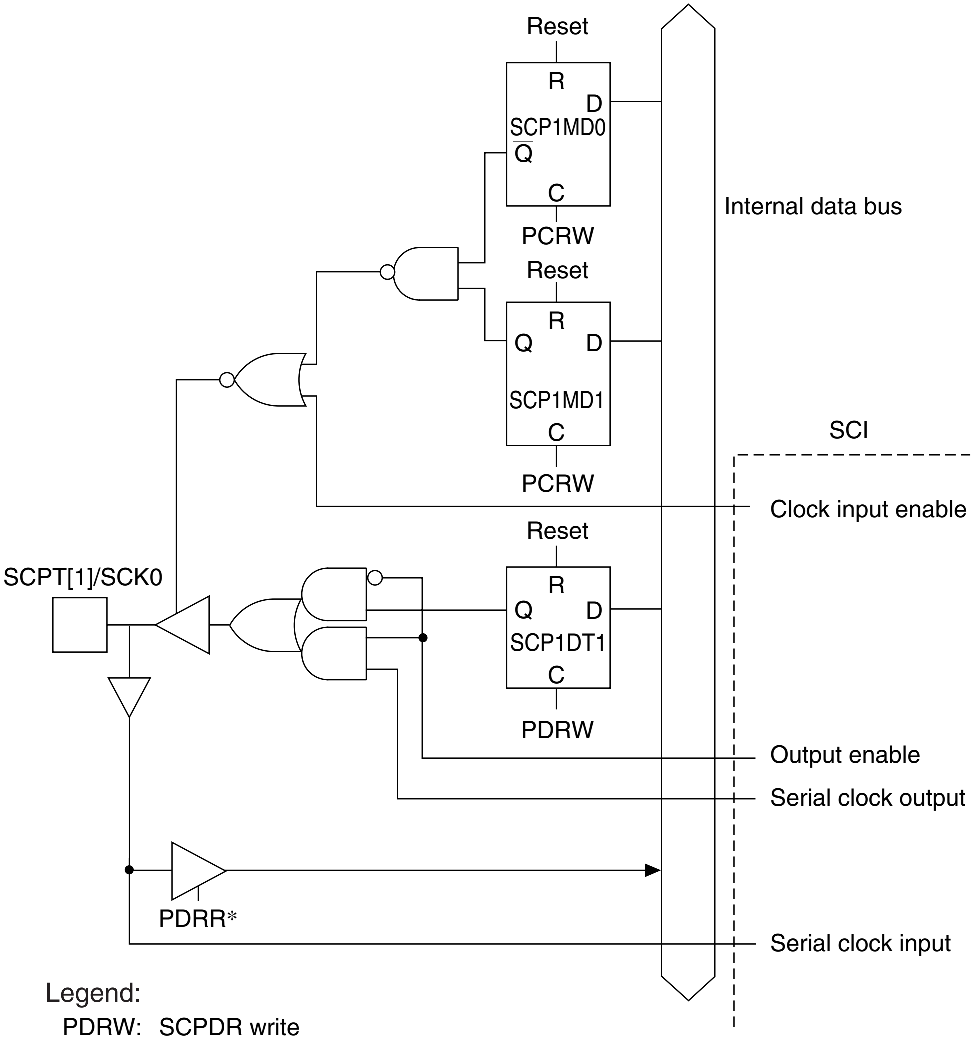
Figure 14.1 shows a SCI block diagram.



Legend:

SCRSR:	Receive shift register	SCSCR:	Serial control register
SCRDR:	Receive data register	SCSSR:	Serial status register
SCTSR:	Transmit shift register	SCBRR:	Bit rate register
SCTDR:	Transmit data register	SCPDR:	SC port data register
SCSMR:	Serial mode register	SCPCR:	SC port control register

Figure 14.1 SCI Block Diagram



Legend:

PDRW: SCPDR write
PDRR: SCPDR read
PCRW: SCPCR write

Note: * When reading the SCK0 pin, clear the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0, and set the SCP1MD1 bit in SCPCR to 1.

Figure 14.2 SCPT[1]/SCK0 Pin

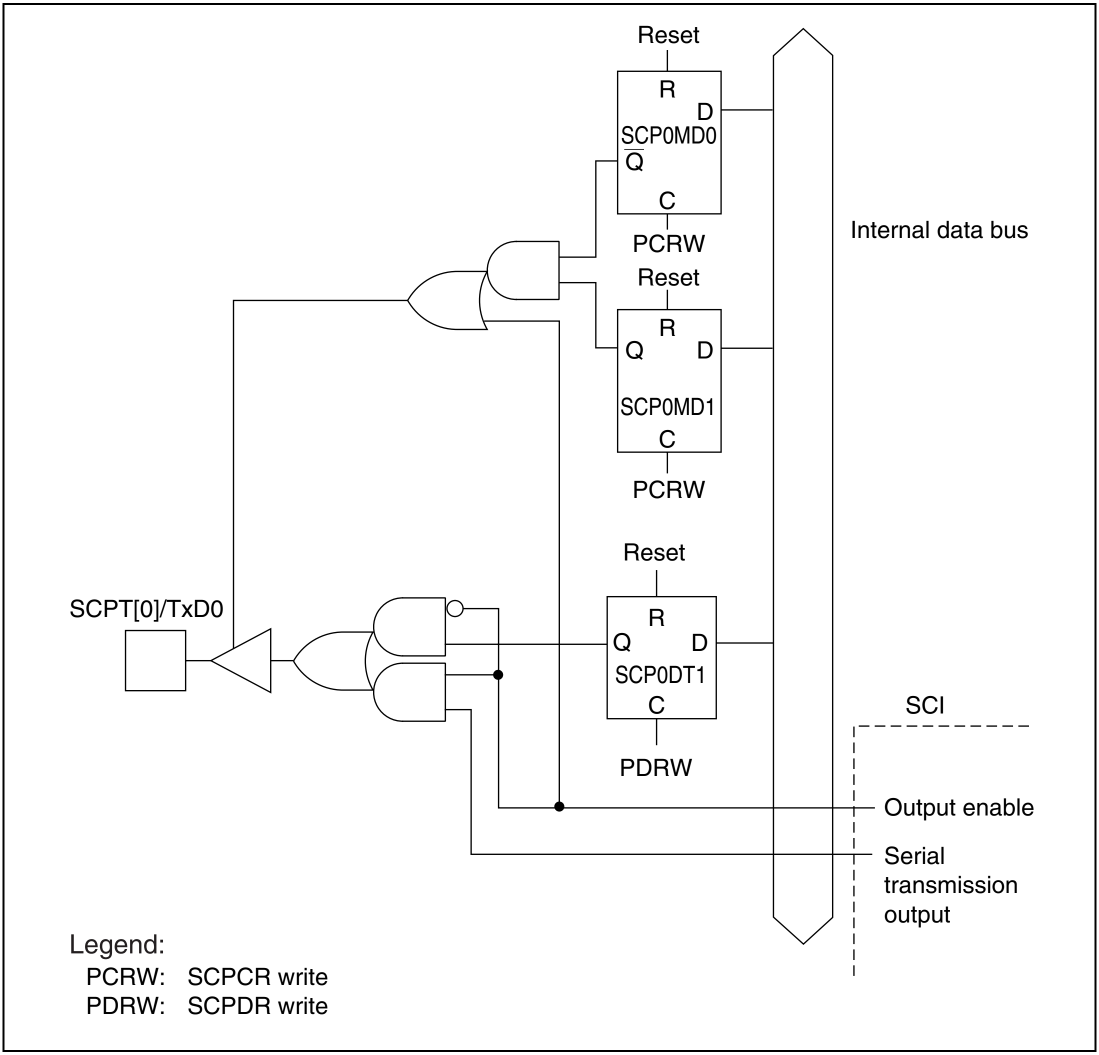


Figure 14.3 SCPT[0]/TxD0 Pin

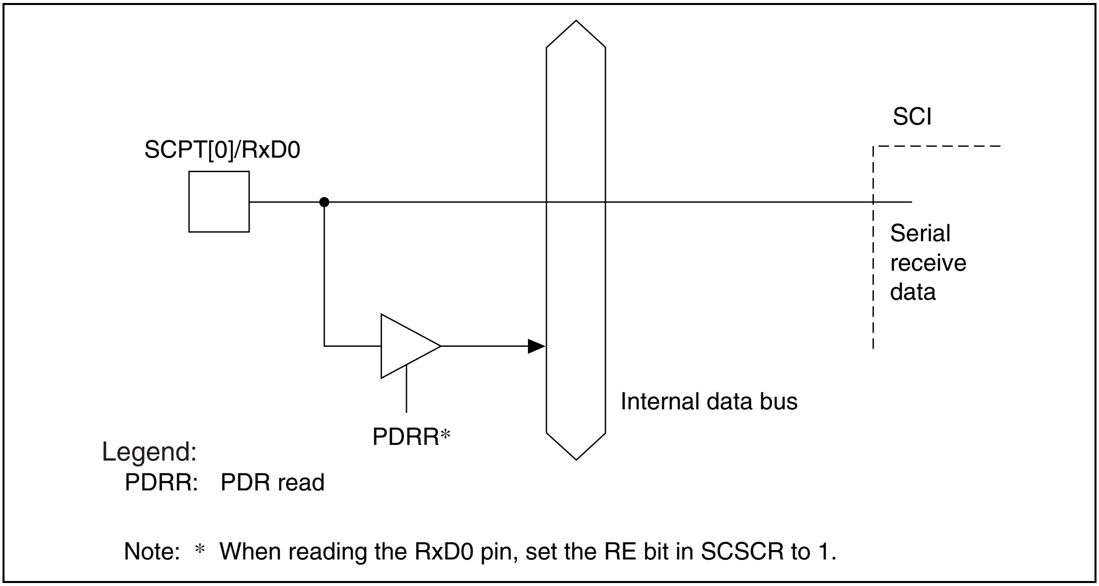


Figure 14.4 SCPT[0]/RxD0 Pin

14.2 I/O Ports

The SCI has the serial pins summarized in table 14.1.

Table 14.1 SCI Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK0	I/O	Clock I/O
Receive data pin	RxD0	Input	Receive data input
Transmit data pin	TxD0	Output	Transmit data output

Note: They are made to function as serial pins by performing SCI operation settings with the TE, RE, CKEI, and CKEO bits in SCSCR and the C/$\overline{A}$ bit in SCSMR. Break state transmission and detection can be performed by means of the SCI's SCPDR.

14.3 Description of Registers

The SCI has the registers listed below. These registers select the communication mode (asynchronous or clock synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

SCI has the registers listed below. Refer to section 23, Control Registers Table, for more detail of the addresses and access size.

- Serial mode register (SCSMR)
- Bit rate register (SCBRR)
- Serial control register (SCSCR)
- Transmit data register (SCTDR)
- Serial status register (SCSSR)
- Receive data register (SCRDR)
- SC port control register (SCPCR)
- SC port data register (SCPDR)

14.3.1 Receive Shift Register (SCRSR)

The receive shift register (SCRSR) is an 8-bit register that receives serial data. Data input at the RxD pin is loaded into the SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the SCRDR. The CPU cannot read or write the SCRSR directly.

14.3.2 Receive Data Register (SCRDR)

The receive data register (SCRDR) is an 8-bit register that stores serial receive data. The SCI completes the reception of one byte of serial data by moving the received data from the SCRSR into the SCRDR for storage. The SCRSR is then ready to receive the next data. This double buffering allows the SCI to receive data continuously.

The CPU can read but not write the SCRDR. The SCRDR is initialized to H'00 by a reset or in standby or module standby modes.

14.3.3 Transmit Shift Register (SCTSR)

The transmit shift register (SCTSR) transmits serial data. The SCI loads transmit data from the SCTDR into the SCTSR, then transmits the data serially to the TxD0 pin, LSB (bit 0) first. After transmitting one-byte data, the SCI automatically loads the next transmit data from the SCTDR into the SCTSR and starts transmitting again. If the TDRE bit of the SCSSR is 1, however, the

SCI does not load the SCTDR contents into the SCTSR. The CPU cannot read or write the SCTSR directly.

14.3.4 Transmit Data Register (SCTDR)

The transmit data register (SCTDR) is an eight-bit register that stores data for serial transmission. When the SCI detects that the SCTSR is empty, it moves transmit data written in the SCTDR into the SCTSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in the SCTDR during serial transmission from the SCTSR.

The CPU can always read and write the SCTDR. The SCTDR is initialized to H'FF by a reset or in standby and module standby modes.

14.3.5 Serial Mode Register (SCSMR)

The serial mode register (SCSMR) is an eight-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write the SCSMR.

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{\text{A}}$	0	R/W	Communication Mode Selects whether the SCI operates in the asynchronous or clock synchronous mode. 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects seven-bit or eight-bit data length in the asynchronous mode. In the clock synchronous mode, the data length is always eight bits, regardless of the CHR setting. 0: Eight-bit data 1: Seven-bit data When seven-bit data is selected, the MSB (bit 7) in the SCTDR is not transmitted.

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to the transmit data or to check the parity of receive data in asynchronous mode. In the clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting. 0: Parity bit not added and not checked 1: Parity bit added and checked Note: When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/E) setting. Receive data parity is checked according to the even/odd (O/E) mode setting.
4	O/$\overline{\text{E}}$	0	R/W	Parity Mode Selects even or odd parity when parity bits are added and checked. The O/$\overline{\text{E}}$ setting is available only when the PE is set to 1 to enable parity addition and check in asynchronous mode. The O $\overline{\text{E}}$ setting is ignored in the clock synchronous mode, or in the asynchronous mode when parity addition and check is disabled. 0: Even parity Note: If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. 1: Odd parity Note: If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length in the asynchronous mode. This setting is used only in the asynchronous mode. It is ignored in the clock synchronous mode because no stop bits are added. In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. 0: One stop bit Note: In transmitting, a single bit of 1 is added at the end of each transmitted character. 1: Two stop bits Note: In transmitting, two bits of 1 are added at the end of each transmitted character.
2	MP	0	R/W	Multiprocessor Mode Selects multiprocessor format. When multiprocessor format is selected, settings of the PE and O/$\overline{\text{E}}$ bits are ignored. The MP setting is used available in the asynchronous mode; it is ignored in the clock synchronous mode. For the multiprocessor communication function, see section 14.4.2, Multiprocessor Communication. 0: Multiprocessor function disabled 1: Multiprocessor format selected
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 14.3.10, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

14.3.6 Serial Control Register (SCSCR)

The serial control register (SCSCR) operates the SCI transmitter/receiver, selects the serial clock output in the asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write the SCSCR.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the TXI request when the serial transmit data is transferred from SCTDR to SCTCR and the TDRE in SCSSR is set to 1. 0: Transmit-data-empty interrupt request (TXI) is disabled Note: The TXI interrupt request can be cleared by reading TDRE after it has been set to 1, then clearing TDRE to 0, or by clearing TIE to 0. 1: Transmit-data-empty interrupt request (TXI) is enabled
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive-data-full interrupt (RXI) request when the serial receive data is transferred from SCRSR to SCRDR and the receive data register full bit (RDRF) in SCSSR is set to 1. It also enables or disables receive-error interrupt (ERI) requests. 0: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled Note: RXI and ERI interrupt requests can be cleared by reading 1 from the RDRF flag or error flag (FER, PER, or ORER) then clearing the flag to 0, or by clearing RIE to 0. 1: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled

Bit	Bit Name	Initial Value	R/W	Description
5	TE	0	R/W	Transmit Enable Enables or disables the SCI serial transmitter. 0: Transmission disabled The TDRE in SCSSR is fixed to 1. 1: Transmission enabled Note: Serial transmission starts when TDRE bit in SCSSR is cleared to 0 after writing of transmit data into the SCTDR. Specify the transmit format to the SCSMR before setting TE to 1.
4	RE	0	R/W	Receive Enable Enables or disables the SCI serial receiver. 0: Reception disabled Note: Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, ORER). These flags retain their previous values. 1: Reception enabled Note: Serial reception starts when a start bit is detected in the asynchronous mode, or synchronous clock input is detected in the clock synchronous mode. Specify the receive format to the SCSMR before setting RE to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	MPIE	0	R/W	Multiprocessor Interrupt Enable Enables or disables multiprocessor interrupts. The MPIE setting is used only in the asynchronous mode, and only if the multiprocessor mode bit (MP) in the serial mode register (SCSMR) is set to 1 during reception. The MPIE setting is ignored in the clock synchronous mode or when the MP bit is cleared to 0. 0: Multiprocessor interrupts are disabled (normal receive operation) [Clearing condition] 1. MPIE is cleared to 0. 2. MPB = 1 is in received data. 1: Multiprocessor interrupts are enabled Receive-data-full interrupt requests (RXI), receive-error interrupt requests (ERI), and setting of the RDRF, FER, and ORER status flags in the serial status register (SCSSR) are disabled until data with a multiprocessor bit of 1 is received. Note: The SCI does not transfer receive data from the SCRSR to the SCRDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in the serial status register (SCSSR). When it receives data that includes MPB = 1, the SCSSR's MPB flag is set to 1, and the SCI automatically clears MPIE to 0, generates RXI and ERI interrupts (if the TIE and RIE bits in the SCSCR are set to 1), and allows the FER and ORER bits to be set.
2	TEIE	0	R/W	Transmit-End Interrupt Enable Enables or disables the transmit-end interrupt (TEI) requested if SCTDR does not contain new transmit data when the MSB is transmitted. 0: Transmit-end interrupt (TEI) requests are disabled* 1: Transmit-end interrupt (TEI) requests are enabled* Note: * The TEI request can be cleared by reading the TDRE bit in SCSSR after it has been set to 1, then clearing TDRE to 0 and clearing the TEND bit to 0, or by clearing the TEIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	These bits select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output or serial clock input. The CKE0 setting is valid only in the asynchronous mode, and only when the SCI is internally clock (CKE1 = 0). The CKE0 setting is ignored in the clock synchronous mode, or when an external clock source is selected (CKE1 = 1). Before selecting the SCI operating mode in the serial mode register (SCSMR), set CKE1 and CKE0. For further details on selection of the SCI clock source, see table 14.9. 00: Asynchronous mode Internal clock, SCK0 pin used for input pin (input signal is ignored) • Asynchronous mode 00: Internal clock; SCK0 pin is used for input pin (input signal is ignored).*1 01: Internal clock; SCK0 pin is used for clock output. *2 01: External clock; SCK0 pin is used for clock input. *3 11: External clock; SCK0 pin is used for clock input. *3 • Clock synchronous mode 00: Internal clock; SCK0 pin is used for synchronous clock output.*1 01: Internal clock; SCK0 pin is used for synchronous clock output. 01: External clock; SCK0 pin is used for synchronous clock input. 11: External clock; SCK0 pin is used for synchronous clock input. Notes: 1. Initial value 2. The output clock frequency is the same as the bit rate. 3. The input clock frequency is 16 times the bit rate.

14.3.7 Serial Status Register (SCSSR)

The serial status register (SCSSR) is an 8-bit register containing multiprocessor bit values, and status flags that indicate SCI operating state.

The CPU can always read and write the SCSSR, but cannot write 1 in the status flags (TDRE, RDRF, ORER, PER, and FER). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 2 (TEND) and 1 (MPB) are read-only bits that cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates that the SCI has loaded transmit data from the SCTDR into the SCTSR and new serial transmit data can be written in the SCTDR. 0: SCTDR contains valid transmit data [Clearing condition] TDRE is read as 1, then written to with 0. 1: SCTDR does not contain valid transmit data [Setting condition] 1. The chip is reset or enters standby mode. 2. TE bit in the serial control register (SCSCR) is 0. 3. SCTDR contents are loaded into SCTSR, so new data can be written in SCTDR.

Bit	Bit Name	Initial Value	R/W	Description
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that SCRDR contains received data. 0: SCRDR does not contain valid received data [Clearing condition] The chip is reset or enters standby mode. RDRF is read as 1, then written to with 0. 1: SCRDR contains valid received data [Setting condition] Serial data is received normally and transferred from SCRSR to SCRDR. Note: The SCRDR and RDRF are not affected by detection of receive errors or by clearing of the RE bit to 0 in the serial control register. They retain their previous contents. If RDRF is still set to 1 when reception of the next data ends, an overrun error (ORER) occurs and the received data is lost.
5	ORER	0	R/(W)*	Overrun Error Indicates that data reception aborted due to an overrun error. 0: Receiving is in progress or has ended normally*[1] [Clearing condition] 1. The chip is reset or enters standby mode. 2. ORER is read as 1, then written to with 0. 1: A receive overrun error occurred*[2] [Setting condition] Reception of the next serial data has ended when RDRF is set to 1. Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the ORER bit, which retains its previous value. 2. SCRDR continues to hold the data received before the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while ORER is set to 1. In the clock synchronous mode, serial transmitting is also disabled.

Bit	Bit Name	Initial Value	R/W	Description
4	FER	0	R/(W)*	Framing Error Indicates that data reception aborted due to a framing error in the asynchronous mode. 0: Receiving is in progress or has ended normally [Clearing condition] 1. The chip is reset or enters standby mode. 2. FER is read as 1, then written to with 0. Note: Clearing the RE bit to 0 in the serial control register does not affect the FER bit, which retains its previous value. 1: A receive framing error occurred [Setting condition] When the SCI has completed receiving, the stop bit at the end of receive data is checked and found to be 0. Note: When the stop bit length is two bits, only the first bit is checked. The second stop bit is not checked. When a framing error occurs, the SCI transfers the receive data into the SCRDR but does not set RDRF. Serial receiving cannot continue while FER is set to 1. In the clock synchronous mode, serial transmitting is also disabled.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*	Parity Error Indicates that data reception (with parity) aborted due to a parity error in the asynchronous mode. 0: Receiving is in progress or has ended normally [Clearing condition] 1. The chip is reset or enters standby mode. 2. PER is read as 1, then written to with 0. Note: Clearing the RE bit to 0 in the SCSCR does not affect the PER bit, which retains its previous value. 1: A receive parity error occurred [Setting condition] The number of 1s in receive data, including the parity bit, does not match the even or odd parity setting of the parity mode bit (O/$\overline{\text{E}}$) in SCSMR. When a parity error occurs, the SCI transfers the receive data into the SCRDR but does not set RDRF. Serial receiving cannot continue while PER is set to 1. In the clock synchronous mode, serial transmitting also cannot continue.
2	TEND	1	R	Transmit End Indicates that when the last bit of a serial character was transmitted, the SCTDR did not contain valid data, so transmission has ended. TEND is a read-only bit and cannot be written. [Clearing condition] TDRE is read as 1, then written to with 0. [Setting condition] 1. The chip is reset or enters standby mode. 2. TE bit in SCSCR is 0. 3. TDRE is 1 when the last bit of a one-byte serial character is transmitted.

Bit	Bit Name	Initial Value	R/W	Description
1	MPB	0	R	Multiprocessor Bit Stores the value of the multiprocessor bit in receive data when a multiprocessor format is selected for receiving in the asynchronous mode. The MPB is a read-only bit and cannot be written. 0: Multiprocessor bit value in receive data is 0 If RE is cleared to 0 when a multiprocessor format is selected, the MPB retains its previous value. 1: Multiprocessor bit value in receive data is 1 Note: Clearing the RE bit to 0 in the maltiprocessor format, which retain its previous value.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in the asynchronous mode. The MPBT setting is ignored in the clock synchronous mode, when a multiprocessor format is not selected, or when the SCI is not transmitting. 0: Multiprocessor bit value in transmit data is 0 1: Multiprocessor bit value in transmit data is 1

Note: * The only value that can be written is a 0 to clear the flag.

14.3.8 SC Port Control Register (SCPCR)

The SC port control register (SCPCR) controls the direction of I/O signals on the SCI and SCIF pins. SCPCR settings are used to perform I/O direction control, enabling data written in SCPDR to be output to the TxD0 pin, data read from the RxD0 pin to be input, and the breaking of serial transmission/reception. It is also possible to read data on and write output data to the SCK0 pin. The I/O controls on the SCI and SCIF pins are performed using bits 3 to 0, and bits 11 to 4 in SCPCR, respectively.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	0	R	Reserved These bits are always read as 0; only 0 should be written here.
11	SCP5MD1	1	R/W	See section 17.1.0, SC Port Control Register.
10	SCP5MD0	0	R/W	
9	SCP4MD1	1	R/W	
8	SCP4MD0	0	R/W	
7	SCP3MD1	1	R/W	
6	SCP3MD0	0	R/W	
5	SCP2MD1	1	R/W	
4	SCP2MD0	0	R/W	
3	SCP1MD1	1	R/W	Serial clock port I/O
2	SCP1MD0	0	R/W	These bits specify serial port SCK0 pin I/O. When the SCK0 pin is actually used as a port I/O pin, clear the $C/\overline{A}$ bit of SCSMR and bits CKE1 and CKE0 of SCSCR to 0. 00: SCP1DT bit value is not output to SCK0 pin. 01: SCP1DT bit value is output to SCK0 pin. 10: SCK0 pin value is read from SCP1DT bit. 11: SCK0 pin value is read from SCP1DT bit.
1	SCP0MD1	0	R/W	Serial port break I/O
0	SCP0MD0	0	R/W	These bits specify the serial port TxD0 pin output condition. When the TxD0 pin is actually used as a port output pin and outputs the value set with the SCP0DT bit, clear the TE bit of SCSCR to 0. 00: SCP0DT bit value is not output to TxD0 pin. 01: SCP0DT bit value is output to TxD0 pin.

14.3.9 SC Port Data Register (SCPDR)

The SC port data register (SCPDR) controls data on the SCI and SCIF pins. The data controls on the SCI and SCIF pins are performed using bits 1 and 0, and bits 5 and 2 in SCPDR, respectively.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	R	Reserved:
6	—	—	R	These bits are always read as 0; only 0 should be written here.
5	SCP5DT	—	R	See section 18.10.2, SC Port Data Register.
4	SCP4DT	0	R/W	
3	SCP3DT	0	R/W	
2	SCP2DT	0	R/W	
1	SCP1DT	0	R/W	Serial clock port data: Specifies the serial port SCK0 pin I/O data. Input or output is specified by the SCP1MD0 and SCP1MD1 bits. In output mode, the value of the SCP1DT bit is output to the SCK0 pin. 0: I/O data is low (0). 1: I/O data is high (1).
0	SCP0DT	0	R/W	Serial port break data: Specifies the serial port RxD0 pin input data and TxD0 pin output data. The TxD0 pin output condition is specified by the SCP0MD0 and SCP0MD1 bits. When the TxD0 pin is set to output mode, the value of the SCP0DT bit is output to the TxD0 pin. The RxD0 pin value is read from the SCP0DT bit regardless of the values of the SCP0MD0 and SCP0MD1 bits, if RE in the SCSCR is set to 1. The initial value of this bit after a power-on reset is undefined. 0: I/O data is low (0). 1: I/O data is high (1).

14.3.10 Bit Rate Register (SCBRR)

The bit rate register (SCBRR) is an eight-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in SCSMR, determines the serial transmit/receive bit rate.

The CPU can always read and write the SCBRR. The SCBRR is initialized to H'FF by a reset or in module standby or standby mode. Each channel has independent baud rate generator control, so different values can be set in two channels.

The SCBRR setting is calculated as follows:

Asynchronous mode: $N = [P\phi/(64 \times 2^{2n-1} \times B)] \times 10^6 - 1$

Clock synchronous mode: $N = [P\phi/(8 \times 2^{2n-1} \times B)] \times 10^6 - 1$

B: Bit rate (bit/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 14.2.)

Table 14.2 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: Find the bit rate error for the asynchronous mode by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^{6}}{(N + 1) \times B \times 64 \times 2^{2n-1}} -1 \right\} \times 100$$

Table 14.3 lists examples of SCBRR settings in the asynchronous mode; table 14.4 lists examples of SCBRR settings in the clock synchronous mode.

Table 14.3 Bit Rates and SCBRR Settings in Asynchronous Mode

	Pφ (MHz)								
	7.3728			8			9.8304		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	130	–0.07	2	141	0.03	2	174	–0.26
150	2	95	0.00	2	103	0.16	2	127	0.00
300	1	191	0.00	1	207	0.16	1	255	0.00
600	1	95	0.00	1	103	0.16	1	127	0.00
1200	0	191	0.00	0	207	0.16	0	255	0.00
2400	0	95	0.00	0	103	0.16	0	127	0.00
4800	0	47	0.00	0	51	0.16	0	63	0.00
9600	0	23	0.00	0	25	0.16	0	31	0.00
19200	0	11	0.00	0	12	0.16	0	15	0.00
31250	0	6	5.33	0	7	0.00	0	9	–1.70
38400	0	5	0.00	0	6	–6.99	0	7	0.00

	Pφ (MHz)								
	10			12			12.288		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	129	0.16	2	155	0.16	2	159	0.00
300	2	64	0.16	2	77	0.16	2	79	0.00
600	1	129	0.16	1	155	0.16	1	159	0.00
1200	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	1.73	0	19	0.16	0	19	0.00
31250	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bits/s)	Pϕ (MHz)											
	14.7456			16			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	64	0.70	3	70	0.03	3	86	0.31	3	88	–0.25
150	2	191	0.00	2	207	0.16	2	255	0.00	3	64	0.16
300	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
600	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
1200	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
2400	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
4800	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
9600	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
19200	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
31250	0	14	–1.70	0	15	0.00	0	19	–1.70	0	19	0.00
38400	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73

Bit Rate (bits/s)	Pϕ (MHz)											
	24			24.576			28.7			30		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	106	–0.44	3	108	0.08	3	126	0.31	3	132	0.13
150	3	77	0.16	3	79	0.00	3	92	0.46	3	97	–0.35
300	2	155	0.16	2	159	0.00	2	186	–0.08	2	194	0.16
600	2	77	0.16	2	79	0.00	2	92	0.46	2	97	–0.35
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	194	0.16
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	97	–0.35
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	194	0.16
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	97	–0.35
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	48	–0.35
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	29	0.00
38400	0	19	–2.34	0	19	0.00	0	22	1.55	0	23	1.73

Bit Rate (bits/s)	Pϕ (MHz)		
	33.34		
	n	N	Error (%)
110	3	147	0.00
150	3	108	–0.43
300	2	216	0.03
600	2	108	–0.43
1200	1	216	0.03
2400	1	108	–0.43
4800	0	216	0.03
9600	0	108	–0.43
19200	0	53	0.49
31250	0	32	1.03
38400	0	26	0.49

Table 14.4 Bit Rates and SCBRR Settings in Clock Synchronous Mode

Bit Rate (bits/s)	Pϕ (MHz)							
	8		16		28.7		30	
	n	N	n	N	n	N	n	N
110	—	—	—	—	—	—	—	—
250	3	124	3	249	—	—	—	—
500	2	249	3	124	3	223	3	233
1k	2	124	2	249	3	111	3	116
2.5k	1	199	2	99	2	178	2	187
5k	1	99	1	199	2	89	2	93
10k	0	199	1	99	1	178	1	187
25k	0	79	0	159	1	71	1	74
50k	0	39	0	79	0	143	0	149
100k	0	19	0	39	0	71	0	74
250k	0	7	0	15	—	—	0	29
500k	0	3	0	7	—	—	0	14
1M	0	1	0	3	—	—	—	—
2M	0	0*	0	1	—	—	—	—

Note: Settings with an error of 1% or less are recommended.
Blank: No setting possible
— : Setting possible, but error occurs
*: Continuous transmit/receive not possible

Table 14.5 indicates the maximum bit rates in the asynchronous mode when the baud rate generator is used. Tables 14.6 and 14.7 list the maximum rates for external clock input.

Table 14.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pφ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0

Table 14.6 Maximum Bit Rates during External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750

Table 14.7 Maximum Bit Rates during External Clock Input (Clock Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
8	1.3333	1333333.3
16	2.6667	2666666.7
24	4.0000	4000000.0
28.7	4.7833	4783333.3
30	5.0000	5000000.0

14.4 Description of Operation

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses. Asynchronous/clock synchronous mode and the transmission format are selected in SCSMR, as listed in table 14.8. The SCI clock source is selected by the combination of the C/$\overline{\text{A}}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR, as listed in table 14.9.

Asynchronous Mode:

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (one or two bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors , overrun errors and breaks.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates on the clock of the on-chip baud rate generator, and can output a serial clock signal with a frequency matching the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Clock Synchronous Mode:

- The transmission/reception format has a fixed eight-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates on the clock of the on-chip baud rate generator, and outputs a synchronous clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input synchronous clock. The on-chip baud rate generator is not used.

Table 14.8 Serial Mode Register Settings and SCI Communication Formats

	SCSMR Settings					SCI Communication Format			
Mode	Bit 7 C/$\overline{\text{A}}$	Bit 6 CHR	Bit 5 PE	Bit 2 MP	Bit 3 STOP	Data Length	Parity Bit	Multiprocessor Bit	Stop Bit Length
Asynchronous	0	0	0	0	0	8-bit	Not set	Not set	1 bit
					1				2 bits
			1		0		Set		1 bit
					1				2 bits
		1	0		0	7-bit	Not set		1 bit
					1				2 bits
			1		0		Set		1 bit
					1				2 bits
Asynchronous (multiprocessor format)		0	*	1	0	8-bit	Not set	Set	1 bit
			*		1				2 bits
		1	*		0	7-bit			1 bit
			*		1				2 bits
Clock synchronous	1	*	*	*	*	8-bit		Not set	None

Note: Asterisks (*) indicate don't-care bits.

Table 14.9 SCSMR and SCSCR Settings and SCI Clock Source Selection

	SCSMR	SCSCR Settings		SCI Transmit/Receive Clock	
Mode	Bit 7 C/$\overline{\text{A}}$	Bit 1 CKE1	Bit 0 CKE0	Clock Source	SCK Pin Function
Asynchronous mode	0	0	0	Internal	SCI does not use the SCK pin
			1		Outputs a clock with frequency matching the bit rate
		1	0	External	Inputs a clock with frequency 16 times the bit rate
			1		
Clock synchronous mode	1	0	0	Internal	Outputs the synchronous clock
			1		
		1	0	External	Inputs the synchronous clock
			1		

14.4.1 Operation in Asynchronous Mode

In the asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 14.5 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in the asynchronous mode, the SCI synchronizes on the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

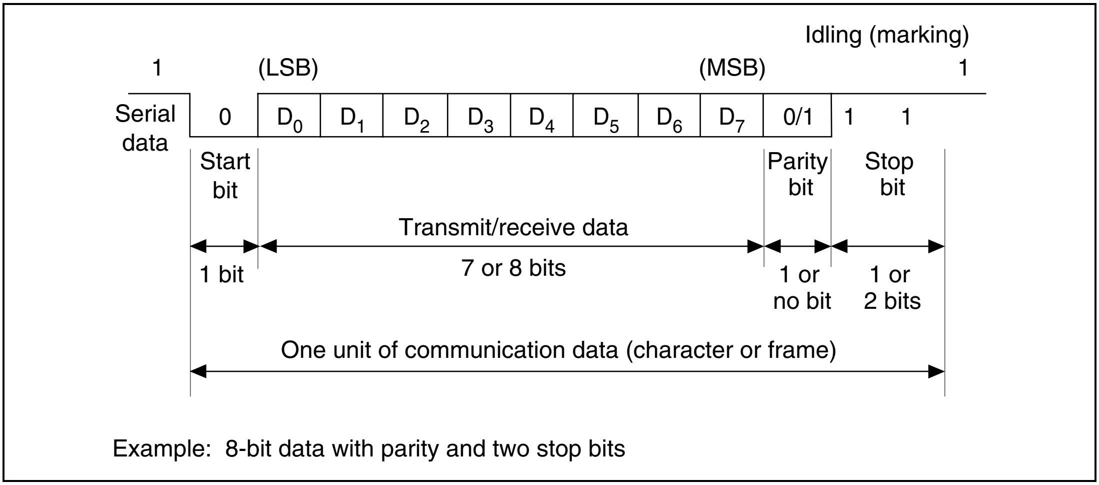


Figure 14.5 Data Format in Asynchronous Communication

Transmit/Receive Formats: Table 14.10 lists the 11 communication formats that can be selected in the asynchronous mode. The format is selected by settings in the SCSMR.

Table 14.10 Serial Communication Formats (Asynchronous Mode)

SCSMR Bits				Serial Transmit/Receive Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	START	8-Bit data								STOP		
0	0	0	1	START	8-Bit data								STOP	STOP	
0	1	0	0	START	8-Bit data								P	STOP	
0	1	0	1	START	8-Bit data								P	STOP	STOP
1	0	0	0	START	7-Bit data							STOP			
1	0	0	1	START	7-Bit data							STOP	STOP		
1	1	0	0	START	7-Bit data							P	STOP		
1	1	0	1	START	7-Bit data							P	STOP	STOP	
0	—	1	0	START	8-Bit data								MPB	STOP	
0	—	1	1	START	8-Bit data								MPB	STOP	STOP
1	—	1	0	START	7-Bit data							MPB	STOP		
1	—	1	1	START	7-Bit data							MPB	STOP	STOP	

— :	Don't care bits
START:	Start bit
STOP:	Stop bit
P:	Parity bit
MPB:	Multiprocessor bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK0 pin can be selected as the SCI transmit/receive clock. The clock source is selected by the $C/\overline{A}$ bit in SCSMR and bits CKE1 and CKE0 in the SCSCR (table 14.9).

When an external clock is input on the SCK0 pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal on the SCK0 pin. The frequency of this output clock is equal to the bit rate. The phase is aligned as in figure 14.6 so that the rising edge of the clock occurs at the center of each transmit data bit.

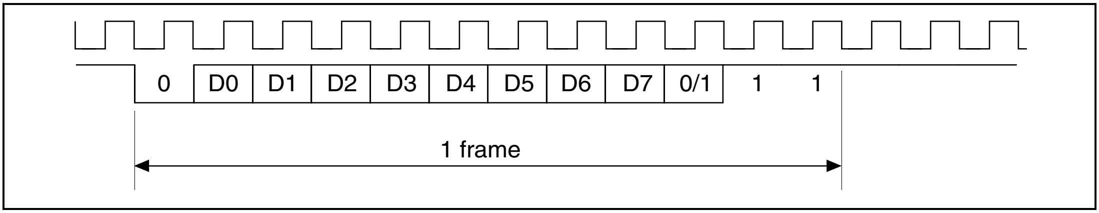


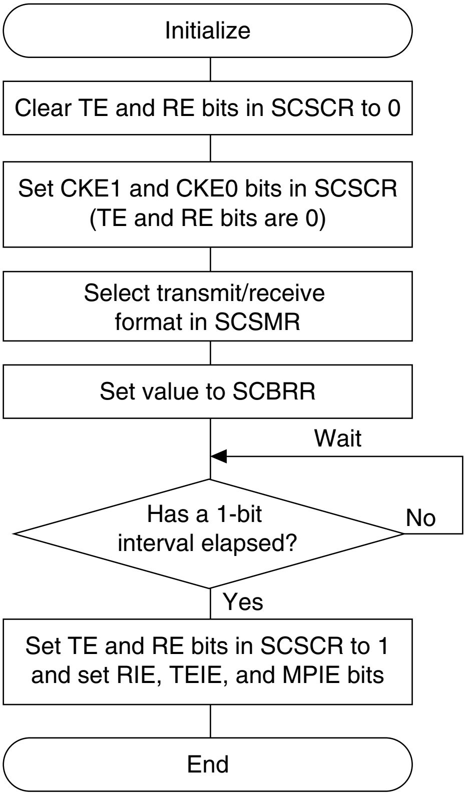
Figure 14.6 Output Clock and Serial Data Timing (Asynchronous Mode)

Transmitting and Receiving Data (SCI Initialization (Asynchronous Mode)): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI as follows.

When changing the operation mode or communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags or receive data register (SCRDR), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

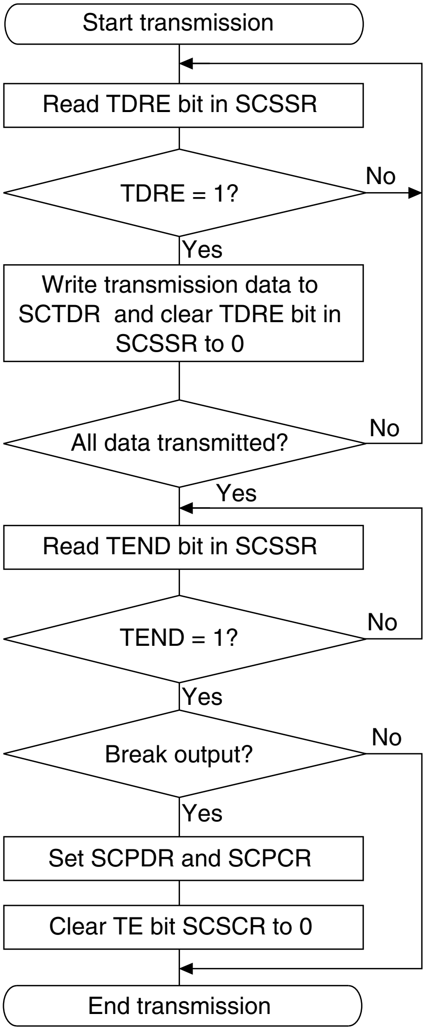
Figure 14.7 is a sample flowchart for initializing the SCI.


Initialize
Clear TE and RE bits in SCSCR to 0
Set CKE1 and CKE0 bits in SCSCR (TE and RE bits are 0)
Select transmit/receive format in SCSMR
Set value to SCBRR
Wait
Has a 1-bit interval elapsed?
No
Yes
Set TE and RE bits in SCSCR to 1 and set RIE, TEIE, and MPIE bits
End

1. Select the clock source in the SCSCR. Leave RIE, TIE, TEIE, MPIE, TE, and RE cleared to 0. If clock output is selected in asynchronous mode, clock output starts immediately after the setting is made to SCSCR.
2. Select the communication format in the SCSMR.
3. Write the value corresponding to the bit rate in SCBRR unless an external clock is used.
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the SCSCR to 1. Also set RIE, TIE, TEIE, and MPIE as necessary. Setting TE or RE enables the SCI to use the TxD0 or RxD0 pin. The initial states are the mark transmit state, and the idle receive state (waiting for a start bit).

Figure 14.7 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 14.8 shows a sample flowchart for transmitting serial data. Serial data transmission should be carried out in the following procedure after setting the SCI in a transmission-enabled state.



1. SCI status check and transmit data write: read the serial status register (SCSSR), check that the TDRE bit is 1, then write transmit data in the SCTDR and clear TDRE to 0.
2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0.
3. To output a break at the end of serial t ransmission: Set the SCPDR and SCPCR, then clear the TE bit to 0 in SCSCR. For SCPCR and SCPDR settings, see section 14.3.8, SC Port Control Register (SCPCR), and section 14.3.9, SC Port Data Register (SCPDR).

Figure 14.8 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SCSSR. When TDRE is cleared to 0, the SCI recognizes that the transmit data register (SCTDR) contains new data, and loads this data from the SCTDR into the SCTSR.
2. After loading the data from the SCTDR into the SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) is set to 1 in the SCSCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time. Serial transmit data is transmitted in the following order from the TxD0 pin:
   a. Start bit: One 0 bit is output.
   b. Transmit data: Seven or eight bits of data are output, LSB first.
   c. Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.
   d. Stop bit: One or two 1 bits (stop bits) are output.
   e. Marking: Output of 1 bits continues until the start bit of the next transmit data.
3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads new data from the SCTDR into the SCTSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit to 1 in the SCSSR, outputs the stop bit, then continues output of 1 bits (marking). If the transmit-end interrupt enable bit (TEIE) in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested.

Figure 14.9 shows an example of SCI transmit operation in the asynchronous mode.

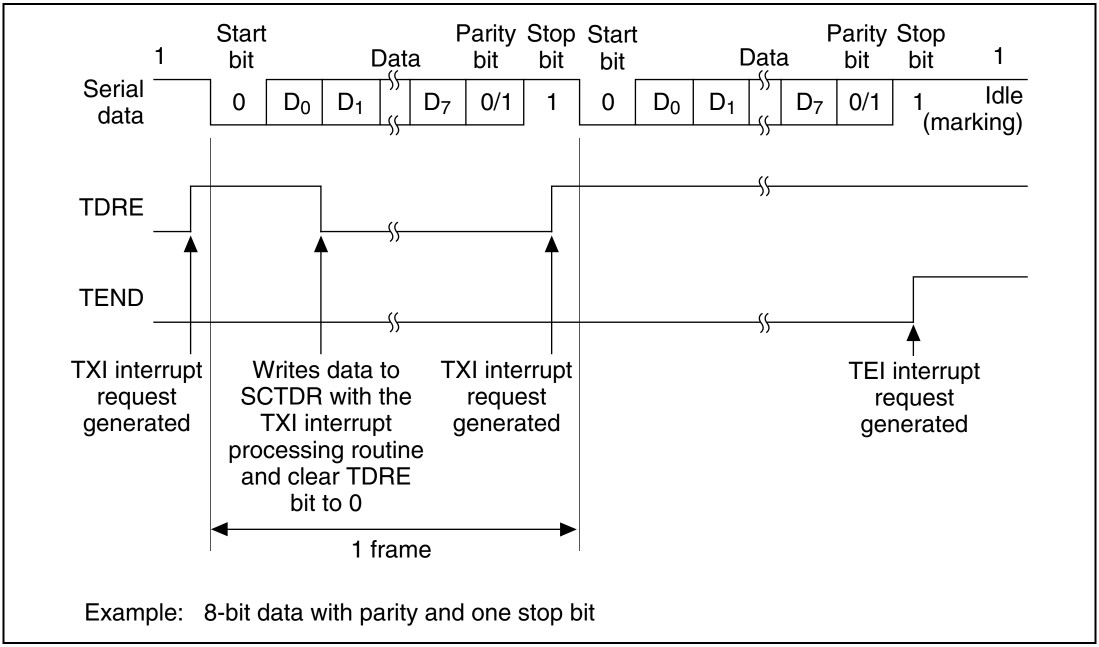
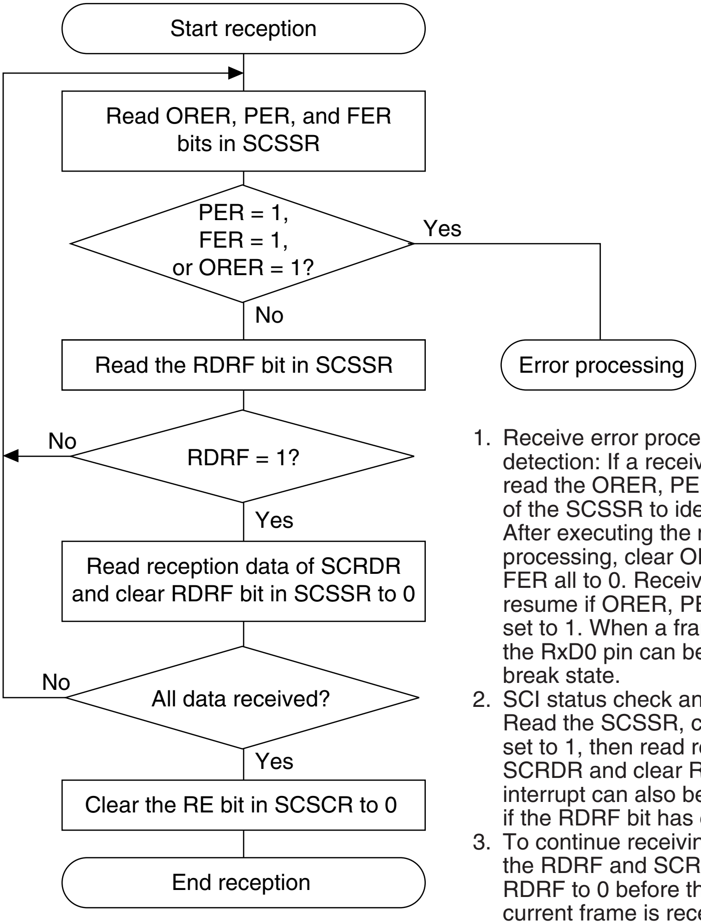

1
Start bit
Data
Parity bit
Stop bit
Start bit
Data
Parity bit
Stop bit
1
Serial data
0
D0
D1
D7
0/1
1
0
D0
D1
D7
0/1
1
Idle (marking)
TDRE
TEND
TXI interrupt request generated
Writes data to SCTDR with the TXI interrupt processing routine and clear TDRE bit to 0
TXI interrupt request generated
TEI interrupt request generated
1 frame
Example: 8-bit data with parity and one stop bit

Figure 14.9 SCI Transmit Operation in Asynchronous Mode

Receiving Serial Data (Asynchronous Mode): Figure 14.10 shows a sample flowchart for receiving serial data. Serial data reception should be carried out in the following procedure after setting the SCI in a reception-enabled state.


Start reception
Read ORER, PER, and FER bits in SCSSR
PER = 1, FER = 1, or ORER = 1?
Yes
No
Error processing
Read the RDRF bit in SCSSR
No
RDRF = 1?
Yes
Read reception data of SCRDR and clear RDRF bit in SCSSR to 0
No
All data received?
Yes
Clear the RE bit in SCSCR to 0
End reception

1. Receive error processing and break detection: If a receive error occurs, read the ORER, PER and FER bits of the SCSSR to identify the error. After executing the necessary error processing, clear ORER, PER and FER all to 0. Receiving cannot resume if ORER, PER or FER remain set to 1. When a framing error occurs, the RxD0 pin can be read to detect the break state.
2. SCI status check and receive-data read: Read the SCSSR, check that RDRF is set to 1, then read receive data from the SCRDR and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
3. To continue receiving serial data: Read the RDRF and SCRDR bits and clear RDRF to 0 before the stop bit of the current frame is received.

Note:

Figure 14.10 Sample Flowchart for Receiving Serial Data

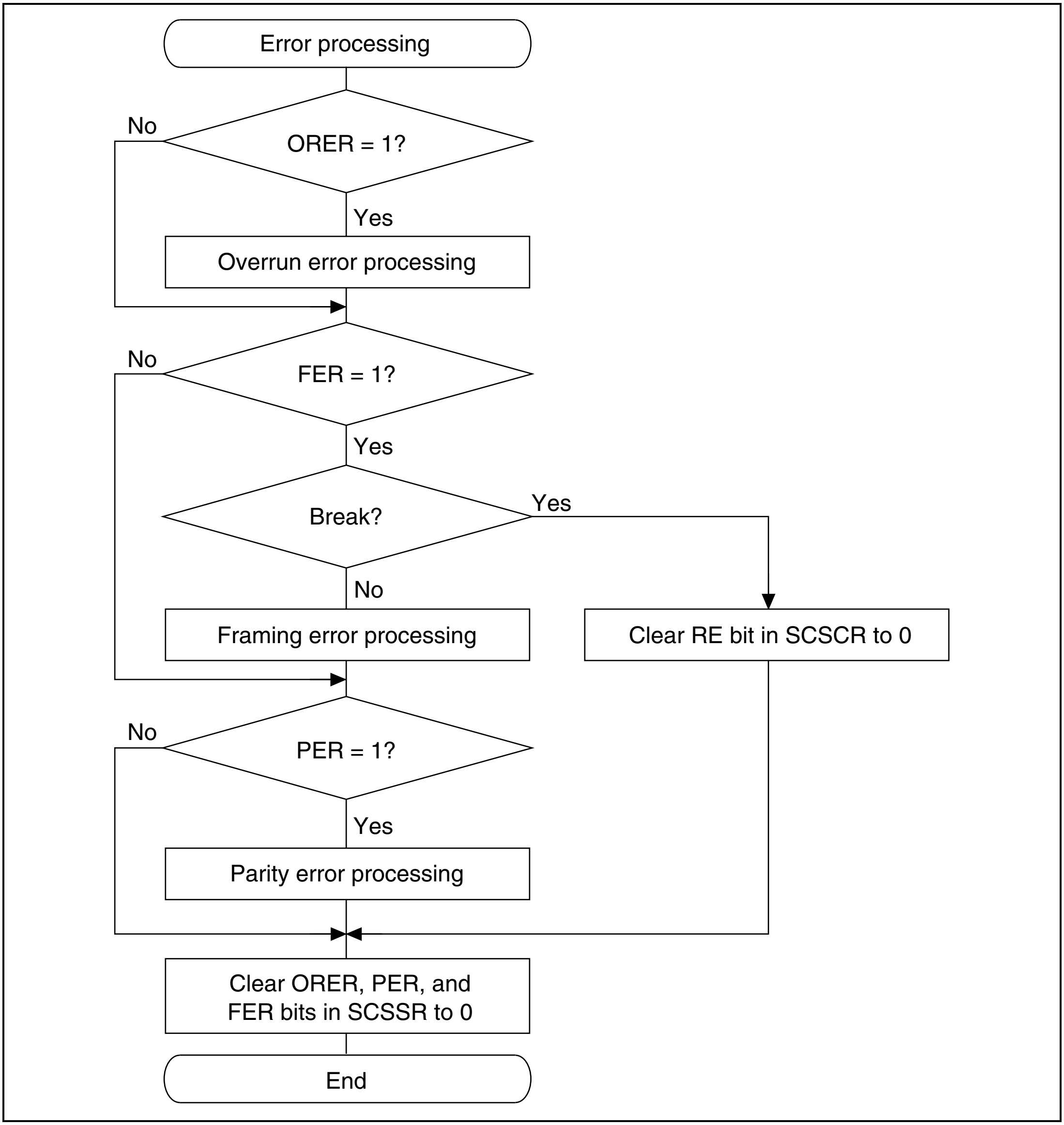


Figure 14.10 Sample Flowchart for Receiving Serial Data (cont)

In receiving, the SCI operates as follows:

1. The SCI monitors the communication line. When it detects a start bit (0), the SCI synchronizes internally and starts receiving.
2. Receive data is stored into the SCRSR in order from the LSB to the MSB.
3. The parity bit and stop bit are received. After receiving these bits, the SCI makes the following checks:
   a. Parity check: The number of 1s in the receive data must match the even or odd parity setting of the O/$\overline{\text{E}}$ bit in the SCSMR.
   b. Stop bit check: The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.
   c. Status check: RDRF must be 0 so that receive data can be loaded from the SCRSR into the SCRDR.

   If these checks all pass, the SCI sets RDRF to 1 and stores the received data in the SCRDR. If one of the checks fails (receive error), the SCI operates as indicated in table 14.11.

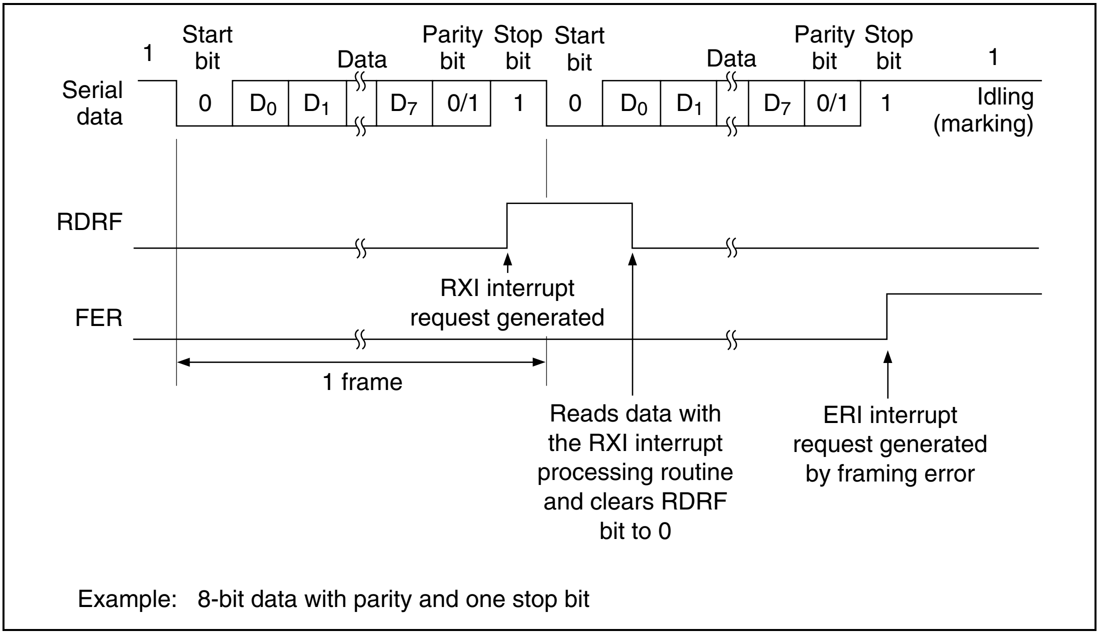
Note: When a receive error flag is set, further receiving is disabled. The RDRF bit is not set to 1. Be sure to clear the error flags.

4. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in the SCSCR, the SCI requests a receive-data-full interrupt (RXI). If one of the error flags (ORER, PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in the SCSCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Table 14.11 Receive Error Conditions and SCI Operation

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SCSSR	Receive data not loaded from SCRSR into SCRDR
Framing error	FER	Stop bit is 0	Receive data loaded from SCRSR into SCRDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SCSMR	Receive data loaded from SCRSR into SCRDR

Figure 14.11 shows an example of SCI receive operation in the asynchronous mode.



1
Start bit
Data
Parity bit
Stop bit
Start bit
Data
Parity bit
Stop bit
1
Serial data
0
D_0
D_1
D_7
0/1
1
0
D_0
D_1
D_7
0/1
1
Idling (marking)
RDRF
FER
RXI interrupt request generated
1 frame
Reads data with the RXI interrupt processing routine and clears RDRF bit to 0
ERI interrupt request generated by framing error
Example: 8-bit data with parity and one stop bit

Figure 14.11 SCI Receive Operation

14.4.2 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line. The processors communicate in the asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by a unique ID. A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles. The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1. When they receive data with the multiprocessor bit set to 1, receiving processors compare the data with their IDs. The receiving processor with a matching ID continues to receive further incoming data. Processors with IDs not matching the received data skip further incoming data until they again receive data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 14.12 shows an example of communication among processors using the multiprocessor format.

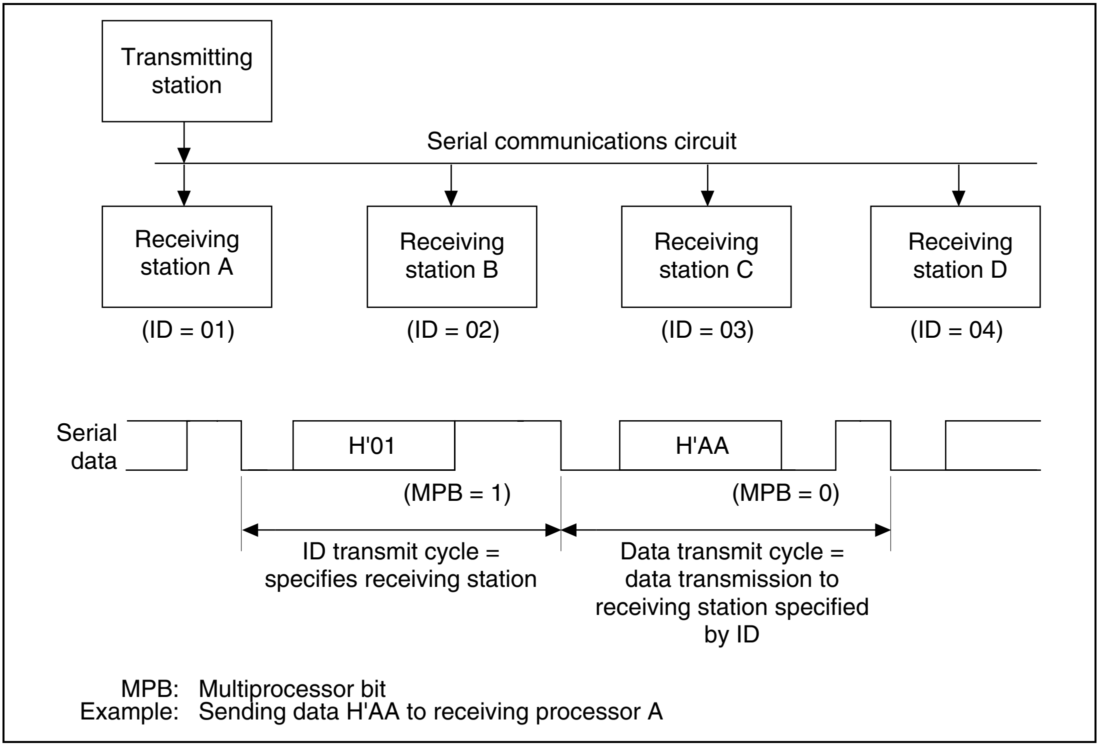
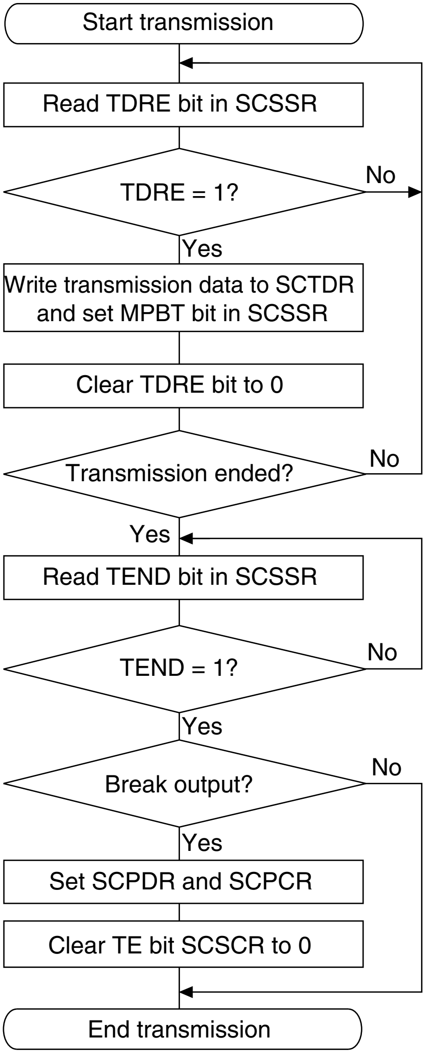

Transmitting station
Serial communications circuit
Receiving station A
Receiving station B
Receiving station C
Receiving station D
(ID = 01)
(ID = 02)
(ID = 03)
(ID = 04)
Serial data
H'01
H'AA
(MPB = 1)
(MPB = 0)
ID transmit cycle = specifies receiving station
Data transmit cycle = data transmission to receiving station specified by ID
MPB: Multiprocessor bit
Example: Sending data H'AA to receiving processor A

Figure 14.12 Communication Among Processors Using Multiprocessor Format

Communication Formats: Four formats are available. Parity-bit settings are ignored when the multiprocessor format is selected. For details see table 14.10.

Clock: See the description in the asynchronous mode section.

Transmitting Multiprocessor Serial Data: Figure 14.13 shows a sample flowchart for transmitting multiprocessor serial data. Transmission of multiprocessor serial data should be carried out in the following procedure after setting the SCI in a transmission-enabled state.



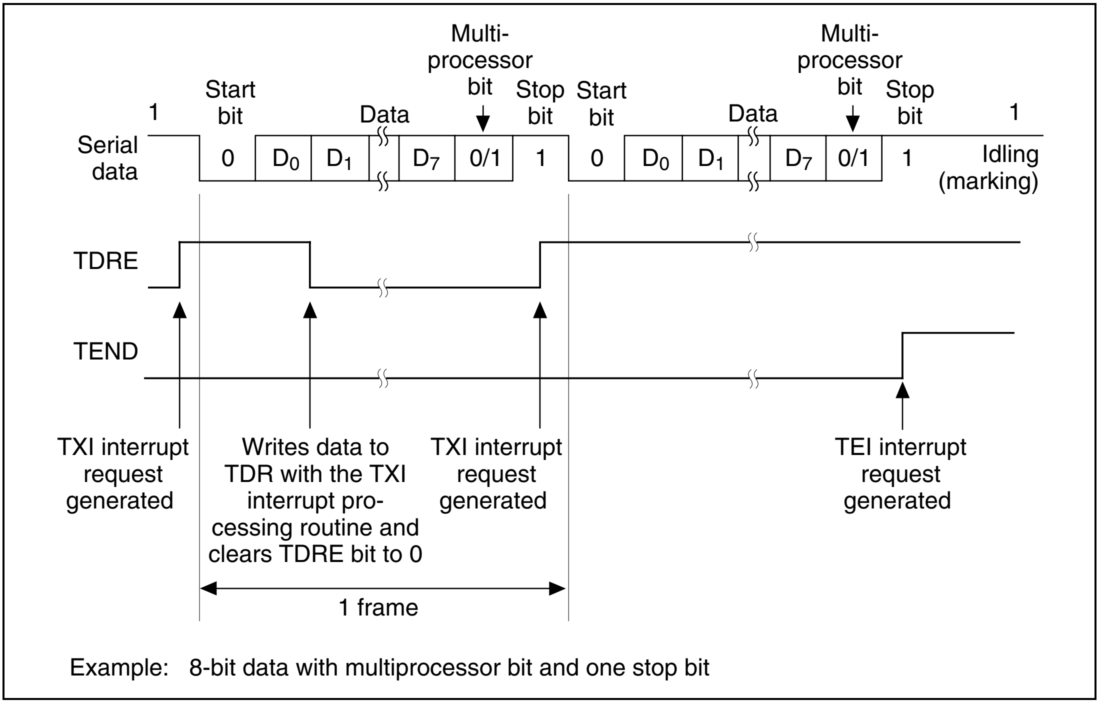
1. SCI status check and transmit data write: Read the SCSSR, check that the TDRE bit is 1, then write transmit data in the SCTDR. Also set MPBT (multiprocessor bit transfer) to 0 or 1 in SCSSR. Finally, clear TDRE to 0.
2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0.
3. To output a break at the end of serial transmission: Set the SCPDR and SCPCR, then clear the TE bit to 0 in SCSCR. For SCPCR and SCPDR settings, see section 14.3.8, SC Port Control Register (SCPCR), and section 14.3.9, SC Port Data Register (SCPDR).

Figure 14.13 Sample Flowchart for Transmitting Multiprocessor Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SCSSR. When TDRE is cleared to 0 the SCI recognizes that the SCTDR contains new data, and loads this data from the SCTDR into the SCTSR.
2. After loading the data from the SCTDR into the SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCSCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time. Serial transmit data is transmitted in the following order from the TxD0 pin:
   a. Start bit: One 0 bit is output.
   b. Transmit data: Seven or eight bits are output, LSB first.
   c. Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
   d. Stop bit: One or two 1 bits (stop bits) are output.
   e. Marking: Output of 1 bits continues until the start bit of the next transmit data.
3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads data from the SCTDR into the SCTSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in the SCSSR to 1, outputs the stop bit, then continues output of 1 bits in the marking state. If the transmit-end interrupt enable bit (TEIE) in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

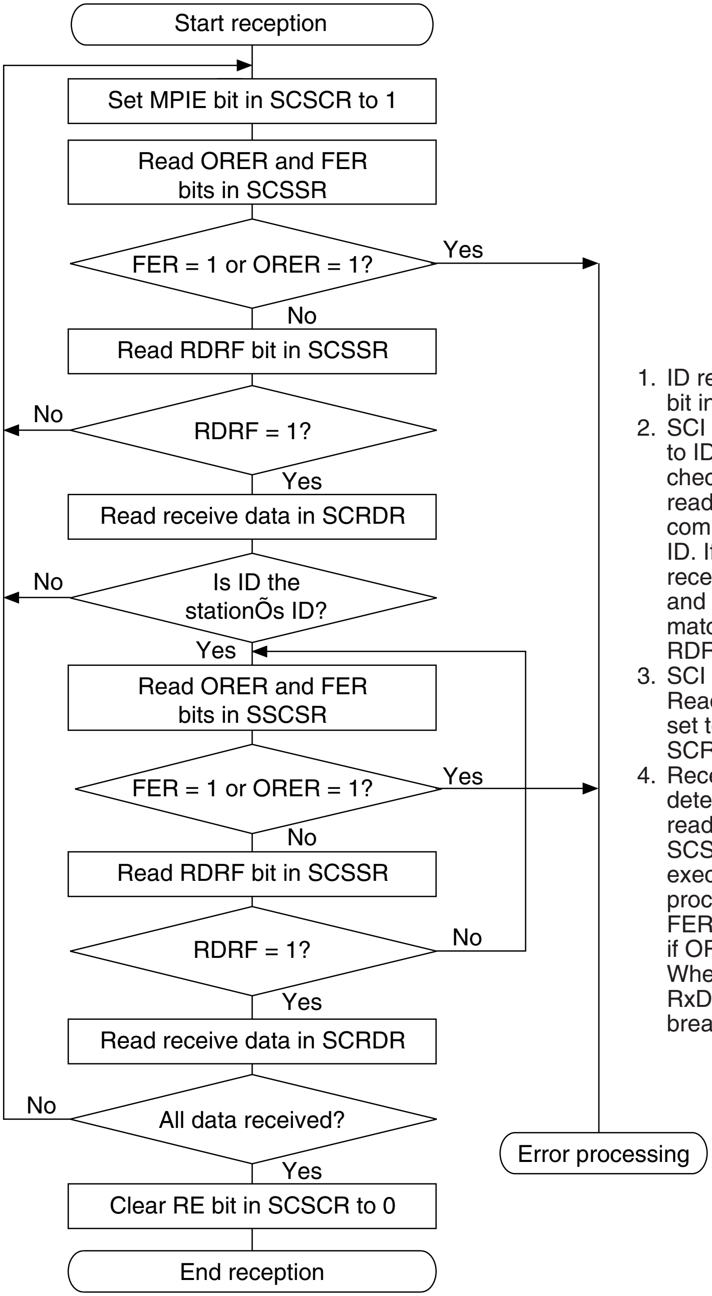
Figure 14.14 shows SCI transmission in the multiprocessor format.



Multi-processor bit
Start bit
Data
Stop bit
1
Serial data
0
D_0
D_1
D_7
0/1
1
Idling (marking)
TDRE
TEND
TXI interrupt request generated
Writes data to TDR with the TXI interrupt processing routine and clears TDRE bit to 0
TXI interrupt request generated
TEI interrupt request generated
1 frame
Example: 8-bit data with multiprocessor bit and one stop bit

Figure 14.14 SCI Multiprocessor Transmit Operation

Receiving Multiprocessor Serial Data: Figure 14.15 shows a sample flowchart for receiving multiprocessor serial data. Reception of multiprocessor serial data should be carried out in the following procedure after setting the SCI in a reception-enabled state.



1. ID receive cycle: Set the MPIE bit in SCSCR to 1.
2. SCI status check and compare to ID reception: Read the SCSSR, check that RDRF is set to 1, then read data from the SCRDR and compare with the processor's own ID. If the ID does not match the receive data, set MPIE to 1 again and clear RDRF to 0. If the ID matches the receive data, clear RDRF to 0.
3. SCI status check and data receiving: Read SCSSR, check that RDRF is set to 1, then read data from the SCRDR.
4. Receive error processing and break detection: If a receive error occurs, read the ORER and FER bits in SCSSR to identify the error. After executing the necessary error processing, clear both ORER and FER to 0. Receiving cannot resume if ORER or FER remain set to 1. When a framing error occurs, the RxD0 pin can be read to detect the break state.

Figure 14.15 Sample Flowchart for Receiving Multiprocessor Serial Data

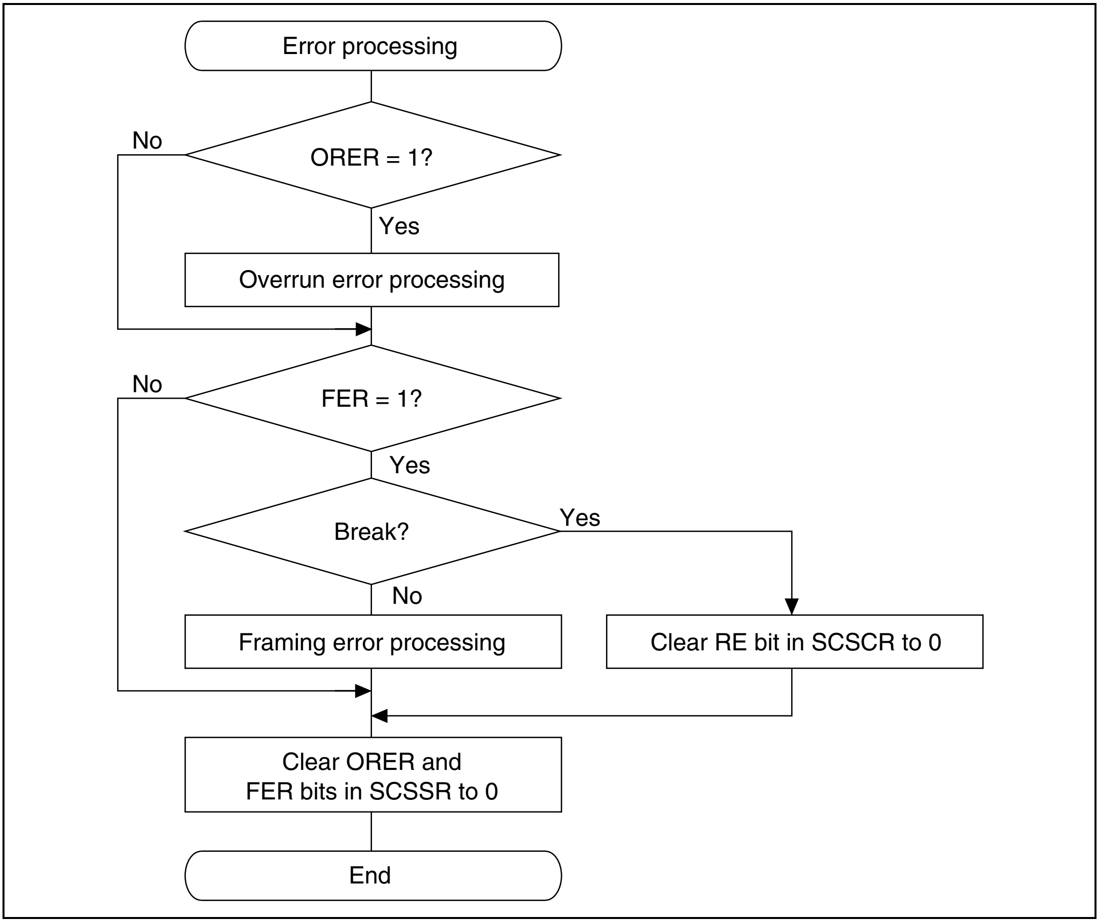


Figure 14.15 Sample Flowchart for Receiving Multiprocessor Serial Data (cont)

Figure 14.16 shows an example of SCI receive operation using a multiprocessor format.

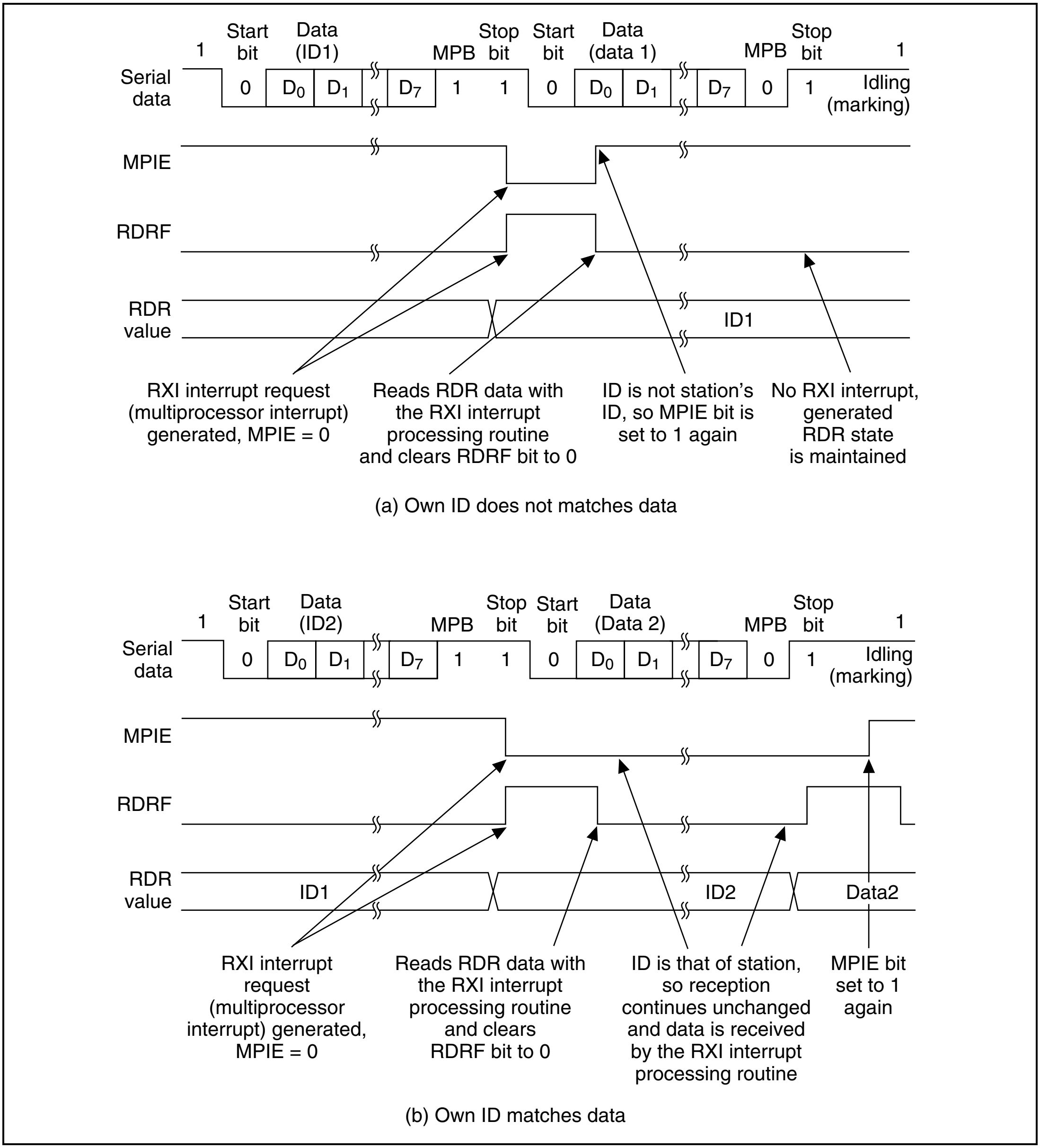

1
Start bit
Data (ID1)
MPB
Stop bit
Start bit
Data (data 1)
MPB
Stop bit
1
Serial data
0
D0
D1
D7
1
1
0
D0
D1
D7
0
1
Idling (marking)
MPIE
RDRF
RDR value
ID1
RXI interrupt request (multiprocessor interrupt) generated, MPIE = 0
Reads RDR data with the RXI interrupt processing routine and clears RDRF bit to 0
ID is not station's ID, so MPIE bit is set to 1 again
No RXI interrupt, generated RDR state is maintained
(a) Own ID does not matches data
1
Start bit
Data (ID2)
MPB
Stop bit
Start bit
Data (Data 2)
MPB
Stop bit
1
Serial data
0
D0
D1
D7
1
1
0
D0
D1
D7
0
1
Idling (marking)
MPIE
RDRF
RDR value
ID1
ID2
Data2
RXI interrupt request (multiprocessor interrupt) generated, MPIE = 0
Reads RDR data with the RXI interrupt processing routine and clears RDRF bit to 0
ID is that of station, so reception continues unchanged and data is received by the RXI interrupt processing routine
MPIE bit set to 1 again
(b) Own ID matches data

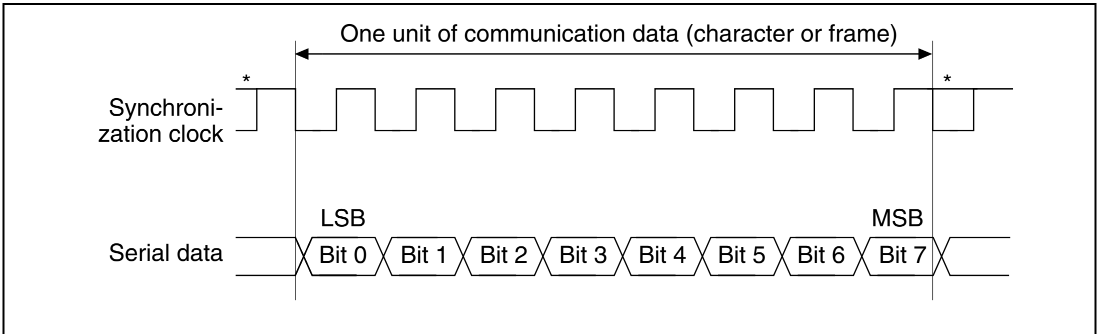
Figure 14.16 Example of SCI Receive Operation

14.4.3 Clock Synchronous Operation

In the clock synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 14.17 shows the general format in clock synchronous serial communication.



Note: * High except in continuous transmitting or receiving

Figure 14.17 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data are guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In the clock synchronous mode, the SCI transmits or receives data by synchronizing with the rising edge of the serial clock.

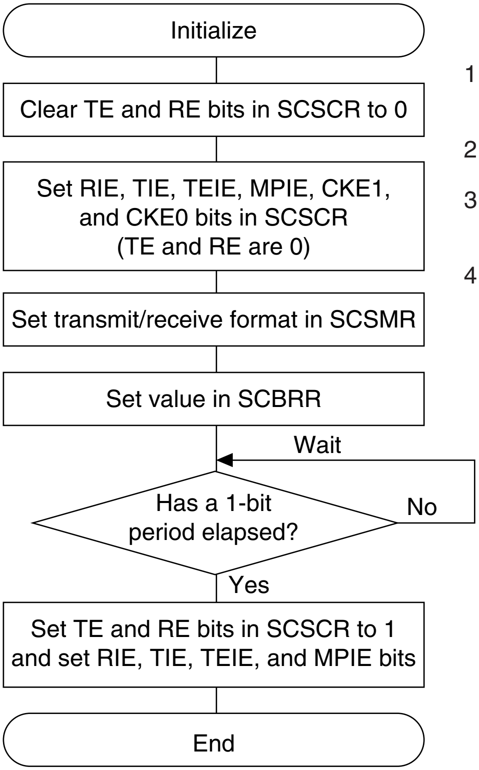
Communication Format: The data length is fixed at eight bits. No parity bit or multiprocessor bit can be added.

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK0 pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{\text{A}}$ bit in SCSMR and bits CKE1 and CKE0 in the SCSCR. See table 14.9.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK0 pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state. When only receiving, the SCI receives in 2-character units, so a 16 pulse synchronization clock is output. To receive in 1-character units, select an external clock source.

Transmitting and Receiving Data (SCI Initialization (clock synchronous mode)):. Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in SCSCR, then initialize the SCI. Clearing TE to 0 sets TDRE to 1 and initializes the SCTSR. Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and SCRDR, which retain their previous contents.

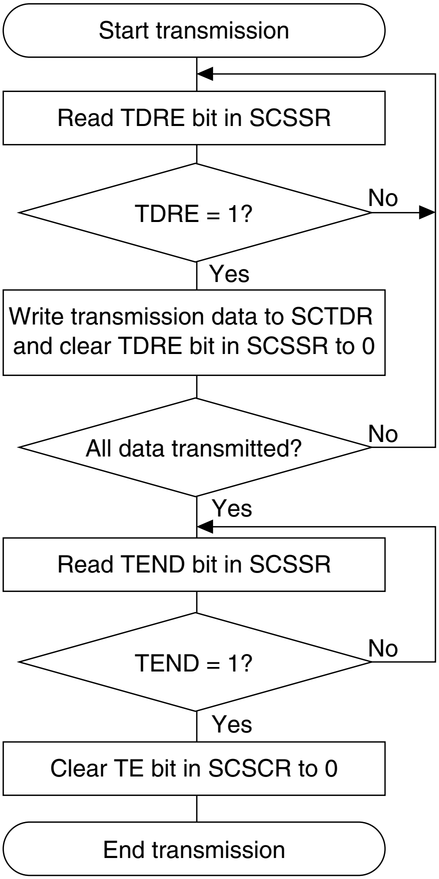
Figure 14.18 is a sample flowchart for initializing the SCI.


Initialize
Clear TE and RE bits in SCSCR to 0
Set RIE, TIE, TEIE, MPIE, CKE1, and CKE0 bits in SCSCR (TE and RE are 0)
Set transmit/receive format in SCSMR
Set value in SCBRR
Wait
Has a 1-bit period elapsed?
No
Yes
Set TE and RE bits in SCSCR to 1 and set RIE, TIE, TEIE, and MPIE bits
End

1. Select the clock source in the SCSCR. Leave RIE, TIE, TEIE, MPIE, TE and RE cleared to 0.
2. Select the communication format in the SCSMR.
3. Write the value corresponding to the bit rate in SCBRR unless an external clock is used.
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the SCSCR to 1. Also set RIE, TIE, TEIE and MPIE. Setting TE and RE allows use of the TxD0 and RxD0 pins.

Figure 14.18 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Clock Synchronous Mode): Figure 14.19 shows a sample flowchart for transmitting serial data. Transmission of serial data should be carried out in the following procedure after setting the SCI in a transmission-enabled state.



1. SCI status check and transmit data write: Read the serial status register (SCSSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR) and clear TDRE to 0.
2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0.

Figure 14.19 Sample Flowchart for Serial Transmitting

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SCSSR. When TDRE is cleared to 0 the SCI recognizes that the SCTDR contains new data and loads this data from the SCTDR into the SCTSR.
2. After loading the data from the SCTDR into the SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCSCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

   If clock output mode is selected, the SCI outputs eight synchronous clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data are output from the TxD0 pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the SCI loads data from the SCTDR into the SCTSR, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in the SCSSR to 1, transmits the MSB, then holds the transmit data pin (TxD0) in the MSB state. If the TEIE in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.
4. After the end of serial transmission, the SCK0 pin is held in the high state.

Figure 14.20 shows an example of SCI transmit operation.

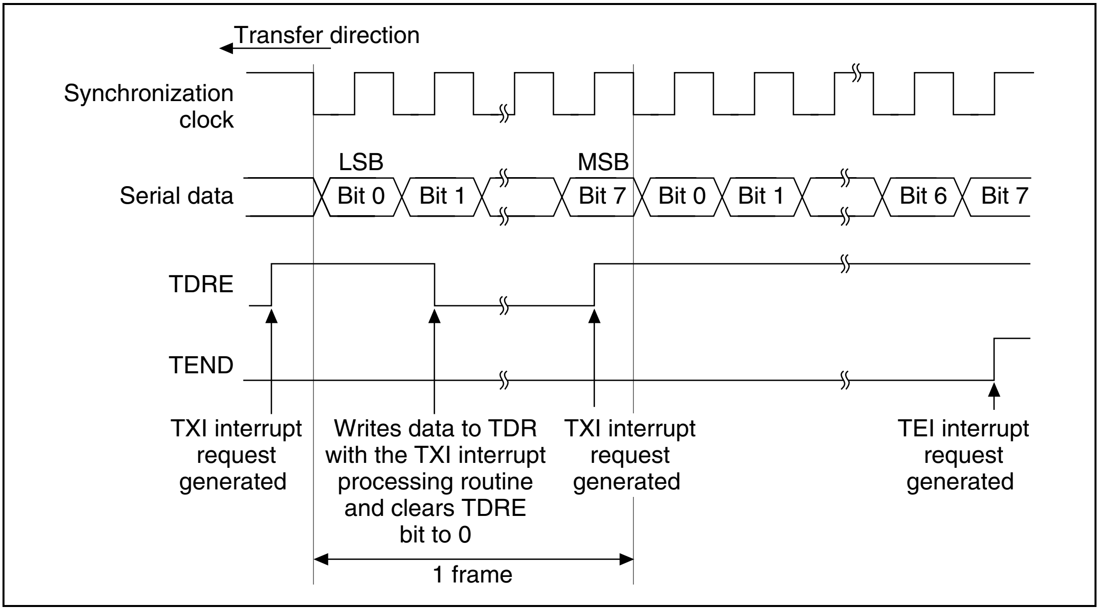
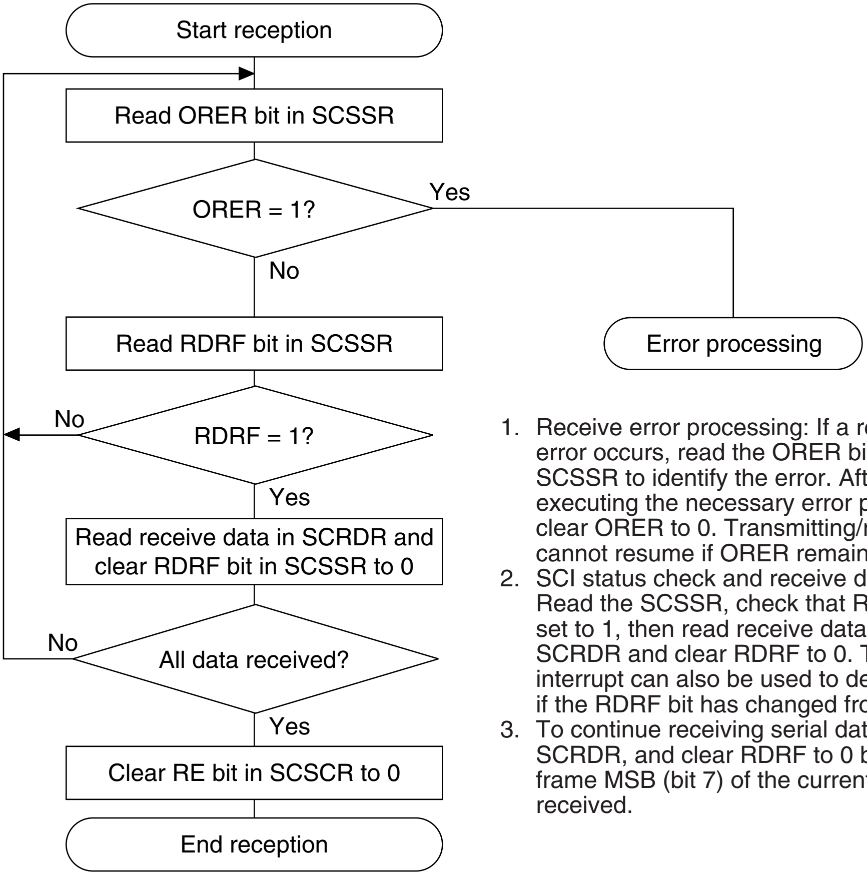

Transfer direction
Synchronization clock
LSB
MSB
Serial data
Bit 0
Bit 1
Bit 7
Bit 0
Bit 1
Bit 6
Bit 7
TDRE
TEND
TXI interrupt request generated
Writes data to TDR with the TXI interrupt processing routine and clears TDRE bit to 0
TXI interrupt request generated
TEI interrupt request generated
1 frame

Figure 14.20 Example of SCI Transmit Operation

Receiving Serial Data (Clock Synchronous Mode): Figure 14.21 shows a sample flowchart for receiving serial data. Serial data reception should be carried out in the procedure described below after setting the SCI in a reception-enabled state. When switching from the asynchronous mode to the clock synchronous mode, make sure that ORER, PER, and FER are cleared to 0. If PER or FER is set to 1, the RDRF bit will not be set and both transmitting and receiving will be disabled.



1. Receive error processing: If a receive error occurs, read the ORER bit in SCSSR to identify the error. After executing the necessary error processing, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
2. SCI status check and receive data read: Read the SCSSR, check that RDRF is set to 1, then read receive data from the SCRDR and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
3. To continue receiving serial data: Read SCRDR, and clear RDRF to 0 before the frame MSB (bit 7) of the current frame is received.

Figure 14.21 Sample Flowchart for Serial Data Receiving

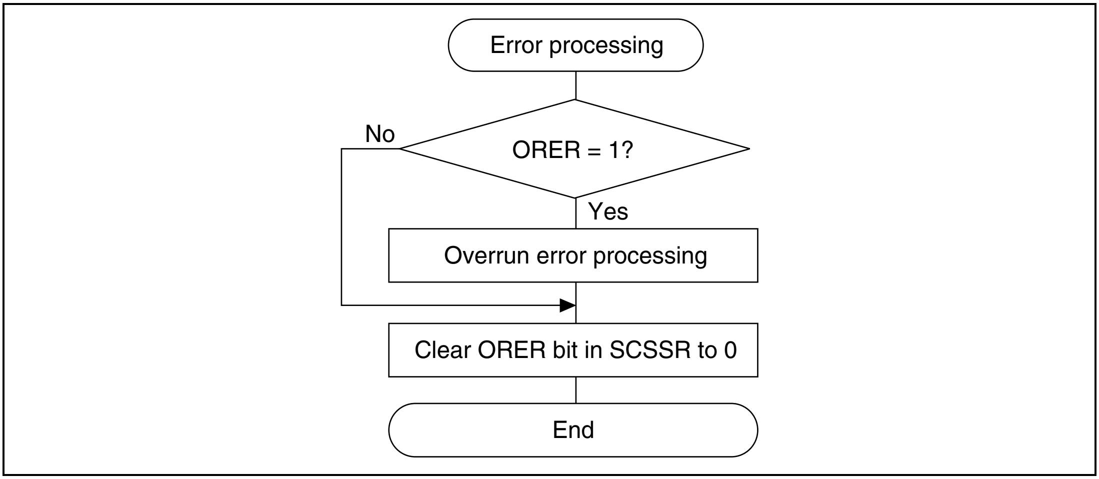


Figure 14.21 Sample Flowchart for Serial Data Receiving (cont)

In receiving, the SCI operates as follows:

1. The SCI synchronizes with serial clock input or output and initializes internally.
2. Receive data is stored into the SCRSR in order from the LSB to the MSB. After receiving the data, the SCI checks that RDRF is 0 so that receive data can be loaded from the SCRSR into the SCRDR. If this check is passed, the SCI sets RDRF to 1 and stores the received data in the SCRDR. If the check is not passed (receive error), the SCI operates as indicated in table 14.11. This state prevents further transmission or reception. While receiving, the RDRF bit is not set to 1. Be sure to clear the error flag.
3. After setting RDRF to 1, if the RIE is set to 1 in the SCSCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the RIE in the SCSCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Figure 14.22 shows an example of the SCI receive operation.

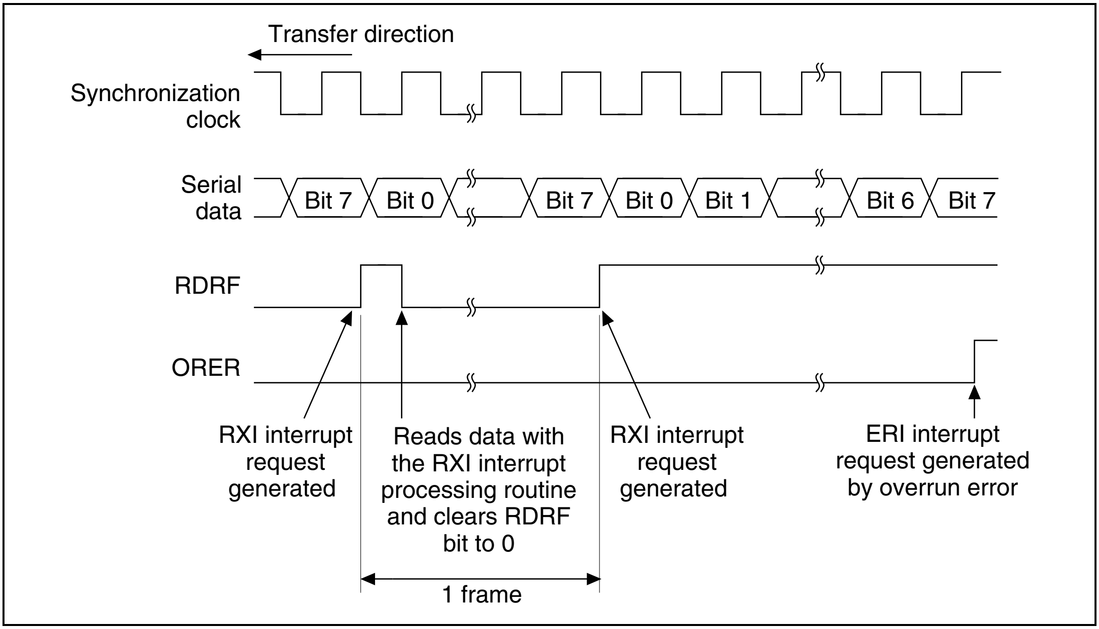
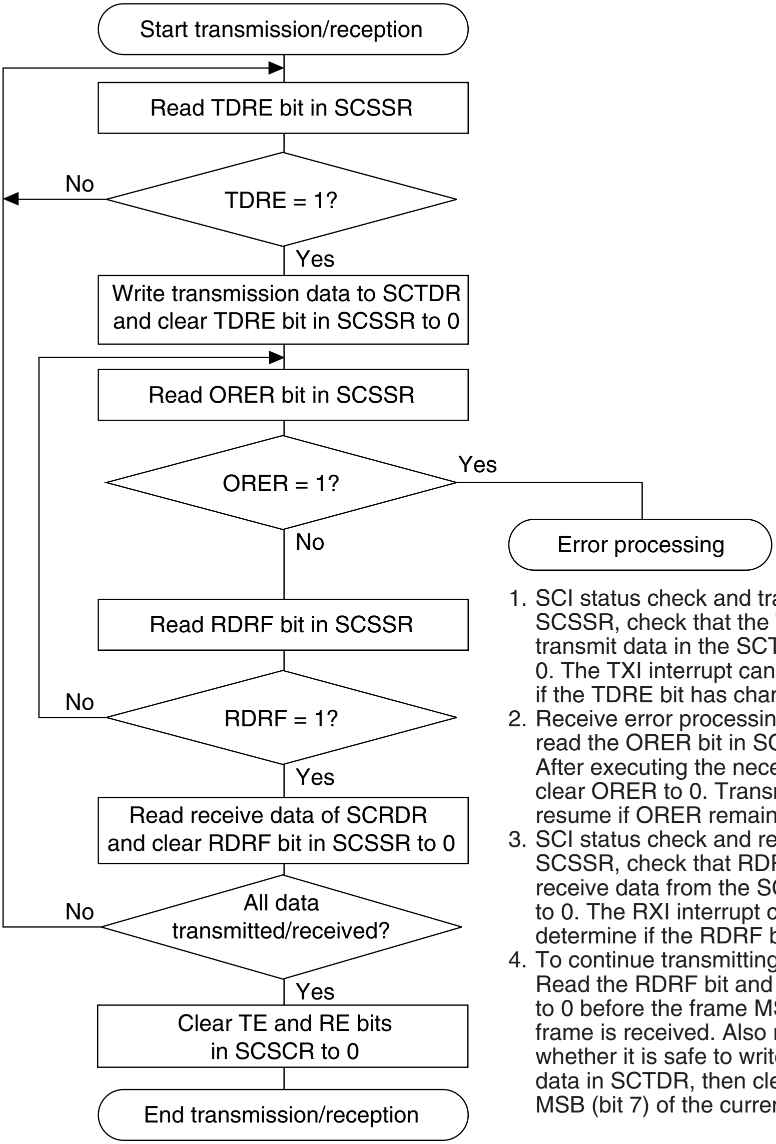

Transfer direction
Synchronization clock
Serial data
Bit 7
Bit 0
Bit 7
Bit 0
Bit 1
Bit 6
Bit 7
RDRF
ORER
RXI interrupt request generated
Reads data with the RXI interrupt processing routine and clears RDRF bit to 0
RXI interrupt request generated
ERI interrupt request generated by overrun error
1 frame

Figure 14.22 Example of SCI Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode): Figure 14.23 shows a sample flowchart for transmitting and receiving serial data simultaneously. Simultaneous transmission and reception of serial data should be carried out in the following procedure after setting the SCI in a transmission/reception-enabled state.


Start transmission/reception
Read TDRE bit in SCSSR
TDRE = 1?
No
Yes
Write transmission data to SCTDR and clear TDRE bit in SCSSR to 0
Read ORER bit in SCSSR
ORER = 1?
Yes
No
Error processing
Read RDRF bit in SCSSR
RDRF = 1?
No
Yes
Read receive data of SCRDR and clear RDRF bit in SCSSR to 0
All data transmitted/received?
No
Yes
Clear TE and RE bits in SCSCR to 0
End transmission/reception

1. SCI status check and transmit data write: Read the SCSSR, check that the TDRE bit is 1, then write transmit data in the SCTDR and clear TDRE to 0. The TXI interrupt can also be used to determine if the TDRE bit has changed from 0 to 1.
2. Receive error processing: If a receive error occurs, read the ORER bit in SCSSR to identify the error. After executing the necessary error processing, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
3. SCI status check and receive data read: Read the SCSSR, check that RDRF is set to 1, then read receive data from the SCRDR and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
4. To continue transmitting and receiving serial data: Read the RDRF bit and SCRDR, and clear RDRF to 0 before the frame MSB (bit 7) of the current frame is received. Also read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0 before the MSB (bit 7) of the current frame is transmitted.

Note: In switching from transmitting or receiving to simultaneous transmitting and receiving, clear both TE and RE to 0, then set both TE and RE to 1.

Figure 14.23 Sample Flowchart for Serial Data Transmitting/Receiving

14.5 SCI Interrupt Sources

The SCI has four interrupt sources in each channel: Transmit-end (TEI), receive-error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI). Table 14.12 lists the interrupt sources and indicates their priority. These interrupts can be enabled and disabled by the TIE, RIE, and TEIE bits in SCSCR. Each interrupt request is sent separately to the interrupt controller.

TXI is requested when the TDRE bit in the SCSSR is set to 1.

RXI is requested when the RDRF bit in the SCSSR is set to 1.

ERI is requested when the ORER, PER, or FER bit in the SCSSR is set to 1.

TEI is requested when the TEND bit in the SCSSR is set to 1. Where the TXI interrupt indicates that transmit data writing is enabled, the TEI interrupt indicates that the transmit operation is complete.

Table 14.12 SCI Interrupt Sources

Interrupt Source	Description	Priority When Reset Is Cleared
ERI	Receive error (ORER, PER, or FER)	High
RXI	Receive data full (RDRF)	↕
TXI	Transmit data empty (TDRE)	
TEI	Transmit end (TEND)	Low

See section 4, Exception Processing, for information on the priority order and relationship to non-SCI interrupts.

14.6 Notes for Usage

Note the following points when using the SCI.

SCTDR Writing to and TDRE Flag: The TDRE bit in SCSSR is a status flag indicating loading of transmit data from the SCTDR into the SCTSR. The SCI sets TDRE to 1 when it transfers data from the SCTDR to the SCTSR. Data can be written to the SCTDR regardless of the TDRE bit state. If new data is written in the SCTDR when TDRE is 0, however, the old data stored in the SCTDR will be lost because the data has not yet been transferred to the SCTSR. Before writing transmit data to the SCTDR, be sure to check that TDRE is set to 1.

Simultaneous Multiple Receive Errors: Table 14.13 indicates the state of the SCSSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs, the SCRSR contents cannot be transferred to the SCRDR, so receive data is lost.

Table 14.13 SCSSR Status Flags and Transfer of Receive Data

	SCSSR Status Flags				Receive Data Transfer
Receive Error Status	**RDRF**	**ORER**	**FER**	**PER**	**SCRSR → SCRDR**
Overrun error	1	1	0	0	X
Framing error	0	0	1	0	O
Parity error	0	0	0	1	O
Overrun error + framing error	1	1	1	0	X
Overrun error + parity error	1	1	0	1	X
Framing error + parity error	0	0	1	1	O
Overrun error + framing error + parity error	1	1	1	1	X

Notes: X: Receive data is not transferred from SCRSR to SCRDR.
O: Receive data is transferred from SCRSR to SCRDR.

Break Detection and Processing: Break signals can be detected by reading the RxD0 pin directly when a framing error (FER) is detected. In the break state, the input from the RxD0 pin consists of all 0s, so FER is set and the parity error flag (PER) may also be set. In the break state, the SCI receiver continues to operate, so if the FER bit is cleared to 0, it will be set to 1 again.

Sending a Break Signal: The TxD0 pin I/O condition and level can be determined by means of the SCP0DT bit of the SCPDR and bits SCP0MD0 and SCP0MD1 of the SCPCR. These bits can be used to send breaks. To send a break during serial transmission, clear the SCP0DT bit to 0 (designating low level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TxD0 pin.

TEND Flag and TE Bit Processing: The TEND flag is set to 1 during transmission of the stop bit of the last data. Consequently, if the TE bit is cleared to 0 immediately after setting of the TEND flag has been confirmed, the stop bit will be in the process of transmission and will not be transmitted normally. Therefore, the TE bit should not be cleared to 0 for at least 0.5 serial clock cycles (or 1.5 cycles if two stop bits are used) after setting of the TEND flag setting is confirmed.

Receive Error Flags and Transmitter Operation (Clock Synchronous Mode Only): When a receive error flag (ORER, PER, or FER) is set to 1, the SCI will not start transmitting even if TDRE is set to 1. Be sure to clear the receive error flags to 0 before starting to transmit. Note that clearing RE to 0 does not clear the receive error flags.

Receive Data Sampling Timing and Receive Margin in the Asynchronous Mode: In the asynchronous mode, the SCI operates on a base clock of 16 times the transfer rate frequency. In receiving, the SCI synchronizes internally with the falling edge of the start bit, which it samples on the base clock. Receive data is latched on the rising edge of the eighth base clock pulse (figure 14.24).

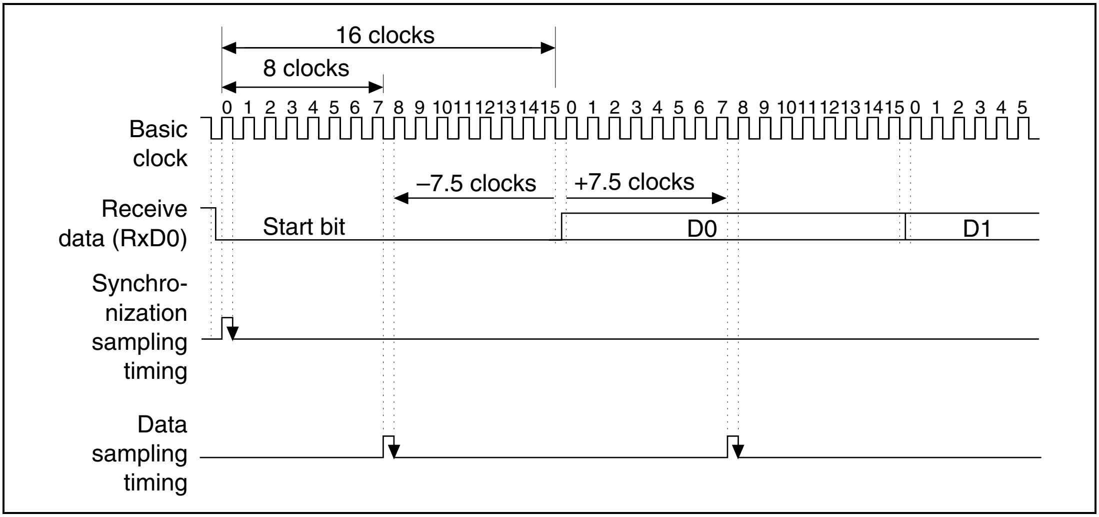


Figure 14.24 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in the asynchronous mode can therefore be expressed as in equation 1.

Equation 1:

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where:

M = Receive margin (%)
N = Ratio of clock frequency to bit rate (N = 16)

D = Clock duty cycle (D = 0–1.0)
L = Frame length (L = 9–12)
F = Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as in equation 2.

Equation 2:

$$\begin{aligned} M &= (0.5 - 1/(2 \times 16)) \times 100\% \\ &= 46.875\% \end{aligned}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20–30%.

Cautions for Clock Synchronous External Clock Mode:

- Set TE = RE = 1 only when the external clock SCK0 is 1.
- Do not set TE = RE = 1 until at least four clocks after the external clock SCK0 has changed from 0 to 1.
- When receiving, RDRF is 1 when RE is set to zero 2.5–3.5 clocks after the rising edge of the SCK0 input of the D7 bit in RxD0, but it cannot be copied to SCRDR.

Caution for Clock Synchronous Internal Clock Mode: In the receiving, RDRF become 1 when RE is set to 0, 1.5 clocks after the rising edge of the SCK0 output of the D7 bit in RxD0, but it cannot be copied to SCRDR.

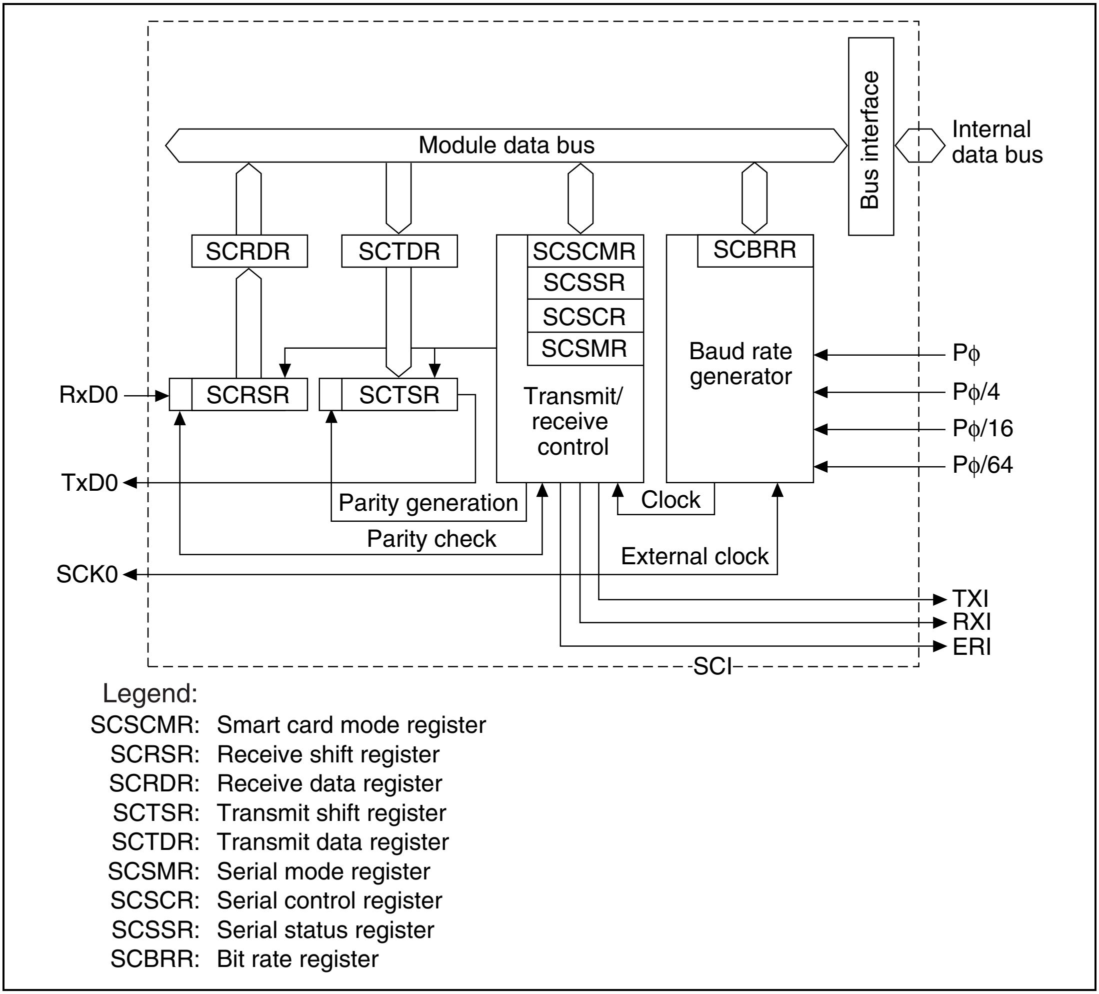
Section 15 Smart Card Interface

As an added serial communications interface function, the SCI supports an IC card (smart card) interface that conforms to the data transfer protocol (asynchronous half-duplex character transmission protocol) of the ISO/IEC standard 7816-3 for identification of cards. Register settings are used to switch between the ordinary serial communication interface and the smart card interface. Figure 15.1 is the block diagram of the smart card interface.

15.1 Features

The smart card interface has the following features:

- Asynchronous mode
  - Data length: Eight bits
  - Parity bit generation and check
  - Receive mode error signal detection (parity error)
  - Transmit mode error signal detection and automatic re-transmission of data
  - Supports both direct convention and inverse convention
- Bit rate can be selected using on-chip baud rate generator.
- Three types of interrupts: Transmit-data-empty, receive-data-full, and communication-error interrupts are requested independently.



Legend:

SCSCMR:	Smart card mode register
SCRSR:	Receive shift register
SCRDR:	Receive data register
SCTSR:	Transmit shift register
SCTDR:	Transmit data register
SCSMR:	Serial mode register
SCSCR:	Serial control register
SCSSR:	Serial status register
SCBRR:	Bit rate register

Figure 15.1 Smart Card Interface Block Diagram

15.2 I/O Pins

Table 15.1 summarizes the smart card interface pins.

Table 15.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK0	Output	Clock output
Receive data pin	RxD0	Input	Receive data input
Transmit data pin	TxD0	Output	Transmit data output

15.3 Description of Registers

The smart card interface has the following registers.

The SCSMR, SCBRR, SCSCR, SCTDR, and SCRDR registers are the same as those of the SCI. So see the description of registers in section 14, Serial Communication Interface.

And for the addresses and access size of these registers, see section 23, Control Register Table.

- Smart card mode register (SCSCMR)
- Serial status register (SCSSR)
- Serial mode register (SCSMR)
- Bit rate register (SCBRR)
- Serial control register (SCSCR)
- Transmit data register (SCTDR)
- Receive data register (SCRDR)

15.3.1 Smart Card Mode Register (SCSCMR)

The smart card mode register (SCSCMR) is an 8-bit read/write register that selects smart card interface functions.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	R	Reserved An undefined value are read from these bits.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: Contents of SCTDR are transferred as LSB first, receive data is stored in SCRDR as LSB first. 1: Contents of SCTDR are transferred as MSB first, receive data is stored in SCRDR as MSB first.
2	SINV	0	R/W	Smart Card Data Inversion Specifies whether to invert the logic level of the data. This function is used in combination with bit 3 for transmitting and receiving with an inverse convention card. SINV does not affect the logic level of the parity bit. See section 15.4.4, Register Settings, for information on how parity is set. 0: Contents of SCTDR are transferred unchanged, receive data is stored in SCRDR unchanged. 1: Contents of SCTDR are inverted before transfer, receive data is inverted before storage in SCRDR.
1	—	—	R	Reserved An undefined value is read from this bit.
0	SMIF	0	R/W	Smart Card Interface Mode Select Enables the smart card interface function. 0: Smart card interface function disabled 1: Smart card interface function enabled

15.3.2 Serial Status Register (SCSSR)

In the smart card interface mode, the function of bit 4 in SCSSR of the SCI is changed as shown blow. Relating to this, the setting conditions for bit 2, the TEND bit, are also changed.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty
6	RDRF	0	R/(W)*	Receive Register Full
5	ORER	0	R/(W)*	Overrun Error These bits have the same function as in the ordinary SCI. See section 14, Serial Communication Interface (SCI), for more information.
4	ERS	0	R/(W)*	Error Signal Status In the smart card interface mode, bit 4 indicates the state of the error signal returned from the receiving side during transmission. The smart card interface cannot detect framing errors. 0: Receiving ended normally with no error signal. [Clearing conditions] 1. The chip is reset or enters standby mode. 2. ERS is read as 1, then written to with 0. 1: An error signal indicating a parity error was transmitted from the receiving side. [Setting condition] The error signal sampled is low. Note: The ERS flag maintains its state even when the TE bit in SCSCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*	Parity error
2	TEND	1	R	Transmission end
1	MPB	0	R	Multiprocessor bit
0	MPBT	0	R/W	Multiprocessor bit transfer These bits have the same function as in the ordinary SCI. See section 14, Serial Communication Interface (SCI), for more information. The setting conditions for bit 2, the transmit end bit (TEND), are changed as follows. 0: Transmission is in progress. [Clearing condition] TDRE is read as 1, then written to with 0. 1: End of transmission. [Setting conditions] 1. The chip is reset or enters standby mode. 2. TE bit in SCSCR is 0 and the FER/ERS bit is also 0. 3. C/$\overline{\text{A}}$ bit in SCSMR is 0, and TDRE = 1 and FER/ERS = 0 (normal transmission) 2.5 etu after a one-byte serial character is transmitted. 4. C/$\overline{\text{A}}$ bit in SCSMR is 1, and TDRE = 1 and FER/ERS = 0 (normal transmission) 1.0 etu after a one-byte serial character is transmitted. Note: etu is an abbreviation of elementary time unit, which is the period for the transfer of 1 bit.

Note: * Only 0 can be written, to clear the flag.

15.4 Description of Operation

15.4.1 Overview

The primary functions of the smart card interface are described below.

1. Each frame consists of 8-bit data and a parity bit.
2. During transmission, the card leaves a guard time of at least 2 etu (elementary time units: the period for 1 bit to transfer) from the end of the parity bit to the start of the next frame.
3. During reception, the card outputs an error signal low level for 1 etu after 10.5 etu has elapsed from the start bit if a parity error was detected.
4. During transmission, it automatically transmits the same data after allowing at least 2 etu from the time the error signal is sampled.
5. Only start-stop type asynchronous communication functions are supported; no synchronous communication functions are available.

15.4.2 Pin Connections

Figure 15.2 shows the pin connection diagram for the smart card interface. During communication with an IC card, transmission and reception are both carried out over the same data transfer line, so connect the TxDϕ and RxDϕ pins on the chip. Pull up the data transfer line to the power supply V_{CC} side with a resistor.

When using the clock generated by the smart card interface on an IC card, input the SCKϕ pin output to the IC card’s CLK pin. This connection is not necessary when the internal clock is used on the IC card.

Use the chip’s port output as the reset signal. Apart from these pins, the power and ground pin connections are usually also required.

Note: When the IC card is not connected and both RE and TE are set to 1, closed communication is possible and self-diagnosis can be performed.

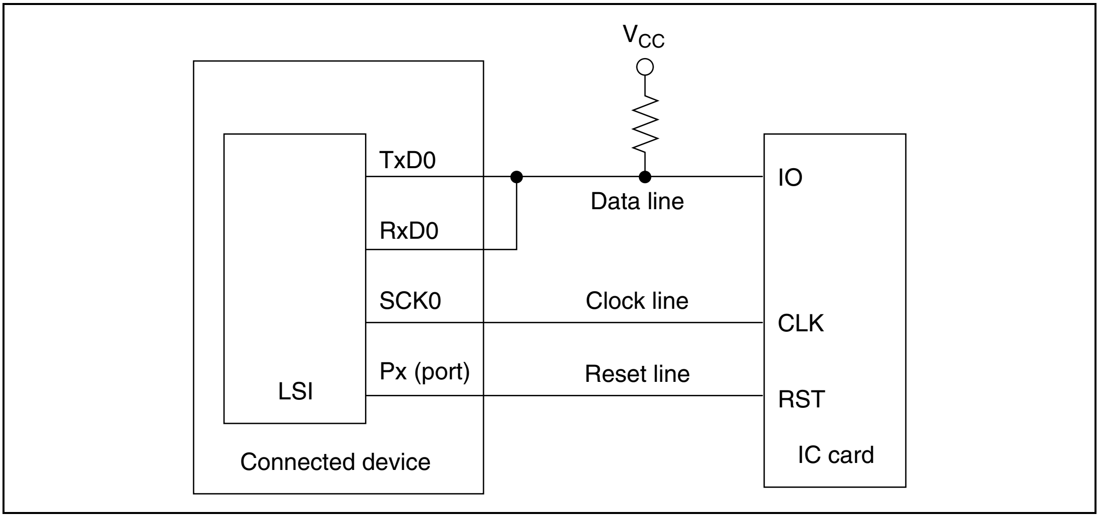


Figure 15.2 Pin Connection Diagram for the Smart Card Interface

15.4.3 Data Format

Figure 15.3 shows the data format for the smart card interface. In this mode, parity is checked every frame while receiving and error signals sent to the transmitting side whenever an error is detected so that data can be re-transmitted. During transmission, if an error signal is sampled, the same data is re-transmitted.

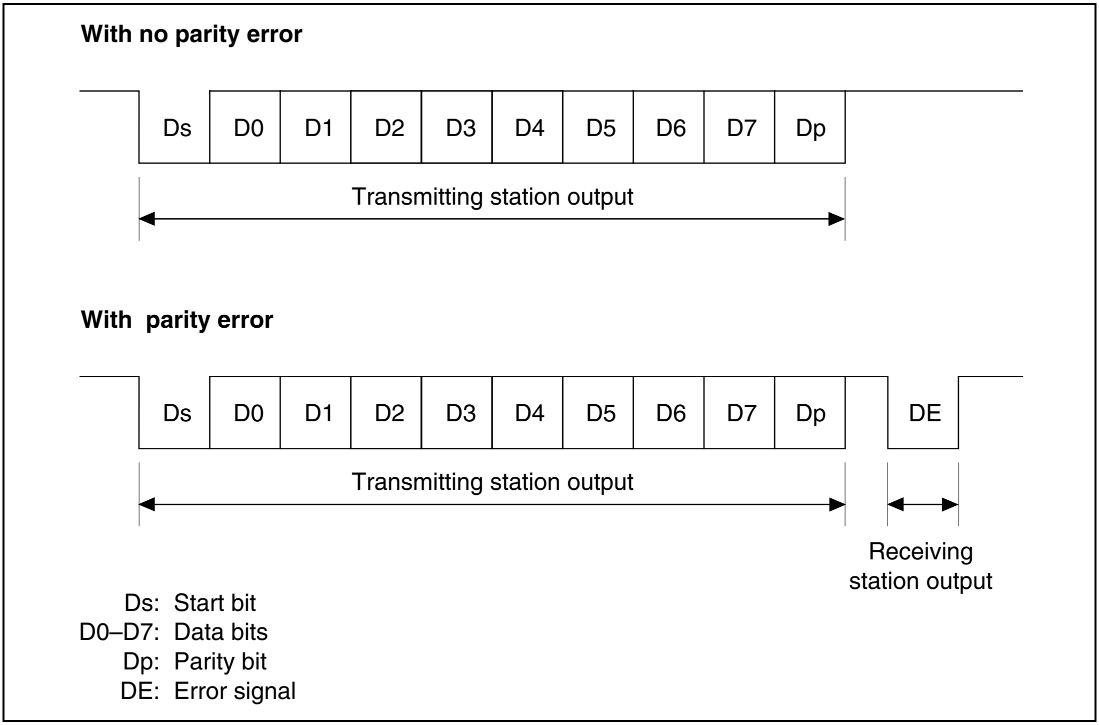


Figure 15.3 Data Format for Smart Card Interface

The operating sequence is:

1. The data line is high impedance when not in use and is fixed high with a pull-up resistor.
2. The transmitting side starts one frame of data transmission. The data frame starts with a start bit (Ds, low level). The start bit is followed by eight data bits (D0–D7) and a parity bit (Dp).
3. On the smart card interface, the data line returns to high impedance after this. The data line is pulled high with a pull-up resistor.
4. The receiving side checks parity. When the data is received normally with no parity errors, the receiving side then waits to receive the next data. When a parity error occurs, the receiving side outputs an error signal (DE, low level) and requests re-transfer of data. The receiving station returns the signal line to high impedance after outputting the error signal for a specified period. The signal line is pulled high with a pull-up resistor.

5. The transmitting side transmits the next frame of data unless it receives an error signal. If it does receive an error signal, it returns to step 2 to re-transmit the erroneous data.

15.4.4 Register Settings

Table 15.2 shows the bit map of the registers that the smart card interface uses. Bits shown as 1 or 0 must be set to the indicated value. The settings for the other bits are described below.

Table 15.2 Register Settings for the Smart Card Interface

Register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCSMR	H'FFFFFE80	C/$\overline{\text{A}}$	0	1	O/$\overline{\text{E}}$	1	0	CKS1	CKS0
SCBRR	H'FFFFFE82	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCSCR	H'FFFFFE84	TIE	RIE	TE	RE	0	0	CKE1	CKE0
SCTDR	H'FFFFFE86	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SCSSR	H'FFFFFE88	TDRE	RDRF	ORER	FER/ERS	PER	TEND	0	0
SCRDR	H'FFFFFE8A	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
SCSCMR	H'FFFFFE8C	—	—	—	—	SDIR	SINV	—	SMIF

Note: Dashes indicate unused bits.

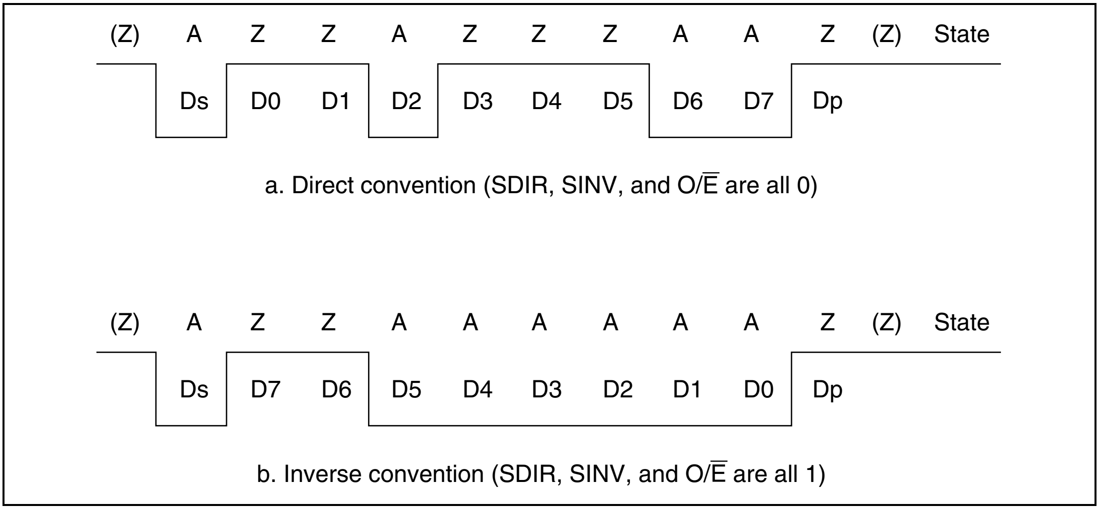
1. Setting the serial mode register (SCSMR): The C/$\overline{\text{A}}$ bit selects the set timing of the TEND flag, and selects the clock output state with the combination of bits CKE1 and CKE0 in the SCSCR. Set the O/$\overline{\text{E}}$ bit to 0 when the IC card uses the direct convention or to 1 when it uses the inverse convention. Select the on-chip baud rate generator clock source with the CKS1 and CKS0 bits (see section 15.4.5, Clock).
2. Setting the bit rate register (SCBRR): Set the bit rate. See section 15.4.5, Clock, to see how to calculate the set value.
3. Setting the serial control register (SCSCR): The TIE, RIE, TE and RE bits function as they do for the ordinary SCI. See section 14, Serial Communication Interface (SCI), for more information. The CKE0 bit specifies the clock output. When no clock is output, set 0; when a clock is output, set 1.
4. Setting the smart card mode register (SCSCMR): The SDIR and SINV bits are both set to 0 for IC cards that use the direct convention and both to 1 when the inverse convention is used. The SMIF bit is set to 1 for the smart card interface.

Figure 15.4 shows sample waveforms for register settings of the two types of IC cards (direct convention and inverse convention) and their start characters.

In the direct convention type, the logical 1 level is state Z, the logical 0 level is state A, and communication is LSB first. The start character data is H'3B. The parity bit is even (as specified in the smart card standards), and thus 1.

In the inverse convention type, the logical 1 level is state A, the logical 0 level is state Z, and communication is MSB first. The start character data is H'3F. The parity bit is even (as specified in the smart card standards), and thus 0, which corresponds to state Z.

Only data bits D7–D0 are inverted by the SINV bit. To invert the parity bit, set the O/$\overline{\text{E}}$ bit in SCSMR to odd parity mode. This applies to both transmission and reception.


(Z) A Z Z A Z Z Z A A Z (Z) State
Ds D0 D1 D2 D3 D4 D5 D6 D7 Dp
(Z) A Z Z A A A A A A Z (Z) State
Ds D7 D6 D5 D4 D3 D2 D1 D0 Dp

a. Direct convention (SDIR, SINV, and O/$\overline{\text{E}}$ are all 0)

b. Inverse convention (SDIR, SINV, and O/$\overline{\text{E}}$ are all 1)

Figure 15.4 Waveform of Start Character

15.4.5 Clock

Only the internal clock generated by the on-chip baud rate generator can be used as the communication clock in the smart card interface. The bit rate for the clock is set by the SCBRR and the CKS1 and CKS0 bits in the SCSMR, and is calculated using the equation below. Table 15.4 shows sample bit rates. If clock output is then selected by setting CKE0 to 1, a clock with a frequency 372 times the bit rate is output from the SCK0 pin.

$$B = \frac{P\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

Where:

N = Value set in SCBRR ($0 \le N \le 255$)
B = Bit rate (bit/s)
Pφ = Peripheral module operating frequency (MHz)
n = 0–3 (table 15.3)

Table 15.3 Relationship of n to CKS1 and CKS0

n	CKS1	CKS0
0	0	0
1	0	1
2	1	0
3	1	1

Table 15.4 Examples of Bit Rate B (Bit/s) for SCBRR Settings (n = 0)

	Pϕ (MHz)						
N	7.1424	10.00	10.7136	13.00	14.2848	16.00	18.00
0	9600.0	13440.9	14400.0	17473.1	19200.0	21505.4	24193.5
1	4800.0	6720.4	7200.0	8736.6	9600.0	10752.7	12096.8
2	3200.0	4480.3	4800.0	5824.4	6400.0	7168.5	8064.5

Note: The bit rate is rounded to two decimal places.

Calculate the value to be set in the bit rate register (SCBRR) from the operating frequency and the bit rate. N is an integer in the range 0 ≤ N ≤ 255, specifying a smallish error.

$$N = \frac{P\phi}{1488 \times 2^{2n-1} \times B} \times 10^{6} - 1$$

Table 15.5 Examples of SCBRR Settings for Bit Rate B (Bit/s) (n = 0)

ϕ (MHz) (9600 Bits/s)													
7.1424		10.00		10.7136		13.00		14.2848		16.00		18.00	
N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error
0	0.00	1	30.00	1	25.00	1	8.99	1	0.00	1	12.01	2	15.99

Table 15.6 Maximum Bit Rates for Frequencies (Smart Card Interface Mode)

Pφ (MHz)	Maximum Bit Rate (Bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0

The bit rate error is found as follows:

$$\text{Error (\%)} = \left(\frac{P\phi}{1488 \times 2^{2n-1} \times B \times (N+1)} \times 10^{6} - 1\right) \times 100$$

Table 15.5 shows example settings of SCBRR, and table 15.6 shows the maximum bit rate for each frequency.

Table 15.7 shows the relationship between transmit/receive clock register set values and output states on the smart card interface.

Table 15.7 Register Set Values and SCKφ Pin

	Register Value				SCK Pin	
Setting	**SMIF**	**C/$\overline{\text{A}}$**	**CKE1**	**CKE0**	**Output**	**State**
1*[1]	1	0	0	0	Port	Determined by setting of port register SCP1MD1 and SCP1MD0 bits
	1	0	0	1	⊓⊓⊓	SCK0 (serial clock) output state
2*[2]	1	1	0	0	Low output	Low output state
	1	1	0	1	⊓⊓⊓	SCK0 (serial clock) output state
3*[2]	1	1	1	0	High output	High output state
	1	1	1	1	⊓⊓⊓	SCK0 (serial clock) output state

Notes: 1. The SCK0 output state changes as soon as the CKE0 bit is modified. The CKE1 bit should be cleared to 0.

2. The clock duty remains constant despite stopping and starting of the clock by modification of the CKE0 bit.

15.4.6 Data Transmission and Reception

Initialization: Initialize the SCI using the following procedure before sending or receiving data. Initialization is also required for switching from transmit mode to receive mode or from receive mode to transmit mode. Figure 15.5 shows an example of initialization process flowchart.

1. Clear TE and RE in SCSCR to 0.
2. Clear error flags FER/ERS, PER, and ORER to 0 in SCSSR.
3. Set the C/$\overline{\text{A}}$ bit, parity bit (O/$\overline{\text{E}}$ bit), and baud rate generator select bits (CKS1 and CKS0 bits) in SCSMR. At this time also clear the CHR and MP bits to 0 and set the STOP and PE bits to 1.
4. Set the SMIF, SDIR, and SINV bits in SCSCMR. When the SMIF bit is set to 1, the TxD and RxD pins both switch from ports to SCI pins and become high impedance.
5. Set the value corresponding to the bit rate in SCBRR.
6. Set the clock source select bits (CKE1 and CKE0 bits) in SCSCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0. When the CKE0 bit is set to 1, a clock is output from the SCKϕ pin.
7. After waiting at least 1 bit, set the TIE, RIE, TE, and RE bits in SCSCR. Do not set the TE and RE bits simultaneously unless performing self-diagnosis.

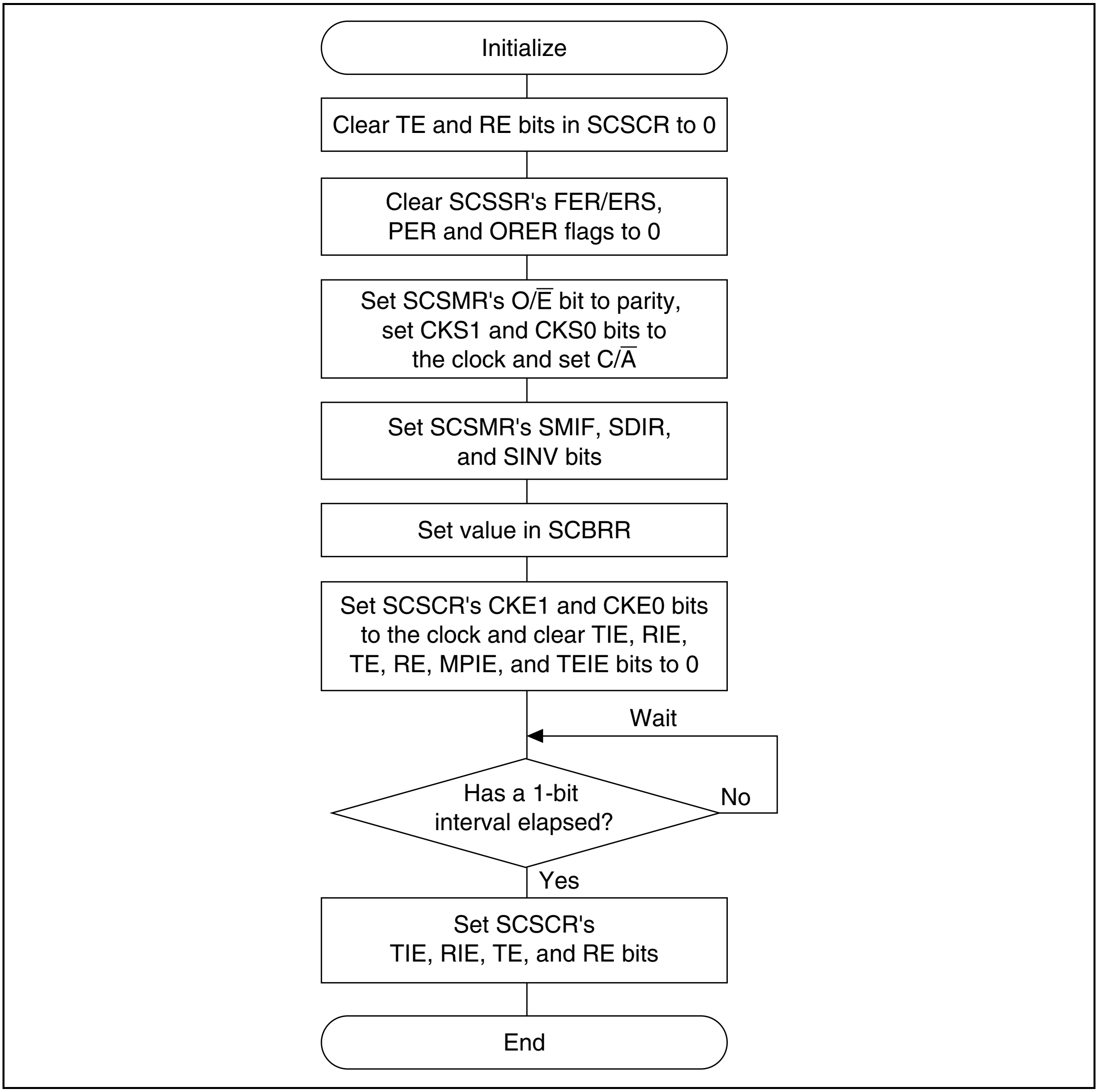

Initialize
Clear TE and RE bits in SCSCR to 0
Clear SCSSR's FER/ERS, PER and ORER flags to 0
Set SCSMR's O/$\overline{E}$ bit to parity, set CKS1 and CKS0 bits to the clock and set C/$\overline{A}$
Set SCSMR's SMIF, SDIR, and SINV bits
Set value in SCBRR
Set SCSCR's CKE1 and CKE0 bits to the clock and clear TIE, RIE, TE, RE, MPIE, and TEIE bits to 0
Wait
Has a 1-bit interval elapsed?
No
Yes
Set SCSCR's TIE, RIE, TE, and RE bits
End

Figure 15.5 Initialization Flowchart (Example)

Serial Data Transmission: The processing procedures in the smart card mode differ from ordinary SCI processing because data is retransmitted when an error signal is sampled during a data transmission. An example of transmission processing flowchart is shown in figure 15.6.

1. Initialize the smart card interface mode as described in Initialization above.
2. Check that the FER/ERS bit in SCSSR is cleared to 0.
3. Repeat steps 2 and 3 until the TEND flag in SCSSR is set to 1.
4. Write the transmit data into SCTDR, clear the TDRE flag to 0 and start transmitting. The TEND flag will be cleared to 0.
5. To transmit more data, return to step 2.
6. To end transmission, clear the TE bit to 0.

This processing can be interrupted. When the TIE bit is set to 1 and interrupt requests are enabled, a transmit-data-empty interrupt (TXI) will be requested when the TEND flag is set to 1 at the end of the transmission. When the RIE bit is set to 1 and interrupt requests are enabled, a communication error interrupt (ERI) will be requested when the ERS flag is set to 1 when an error occurs in transmission. See Interrupt Operation below for more information.

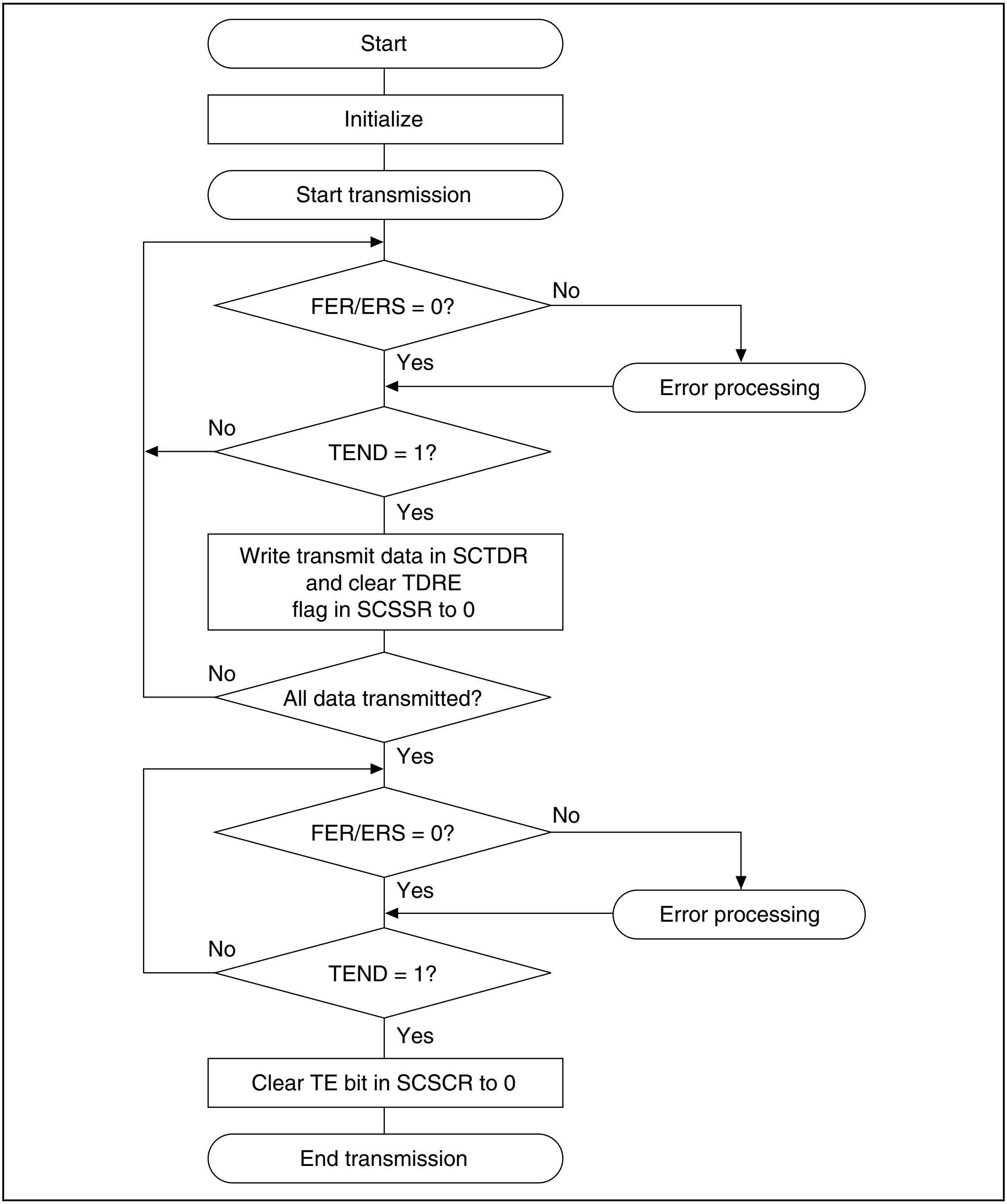


Figure 15.6 Transmission Flowchart

Serial Data Reception: The processing procedures in the smart card mode are the same as in ordinary SCI processing. The reception processing flowchart is shown in figure 15.7.

1. Initialize the smart card interface mode as described above in Initialization and in figure 15.5.
2. Check that the ORER and PER flags in SCSSR are cleared to 0. If either flag is set, clear both to 0 after performing the appropriate error processing procedures.
3. Repeat steps 2 and 3 until the RDRF flag is set to 1.
4. Read the receive data from SCRDR.
5. To receive more data, clear the RDRF flag to 0 and return to step 2.
6. To end reception, clear the RE bit to 0.

This processing can be interrupted. When the RIE bit is set to 1 and interrupt requests are enabled, a receive-data-full interrupt (RXI) will be requested when the RDRF flag is set to 1 at the end of the reception. When an error occurs during reception and either the ORER or PER flag is set to 1, a communication error interrupt (ERI) will be requested. See Interrupt Operation below for more information.

The received data will be transferred to SCRDR even when a parity error occurs during reception and PER is set to 1, so this data can still be read.

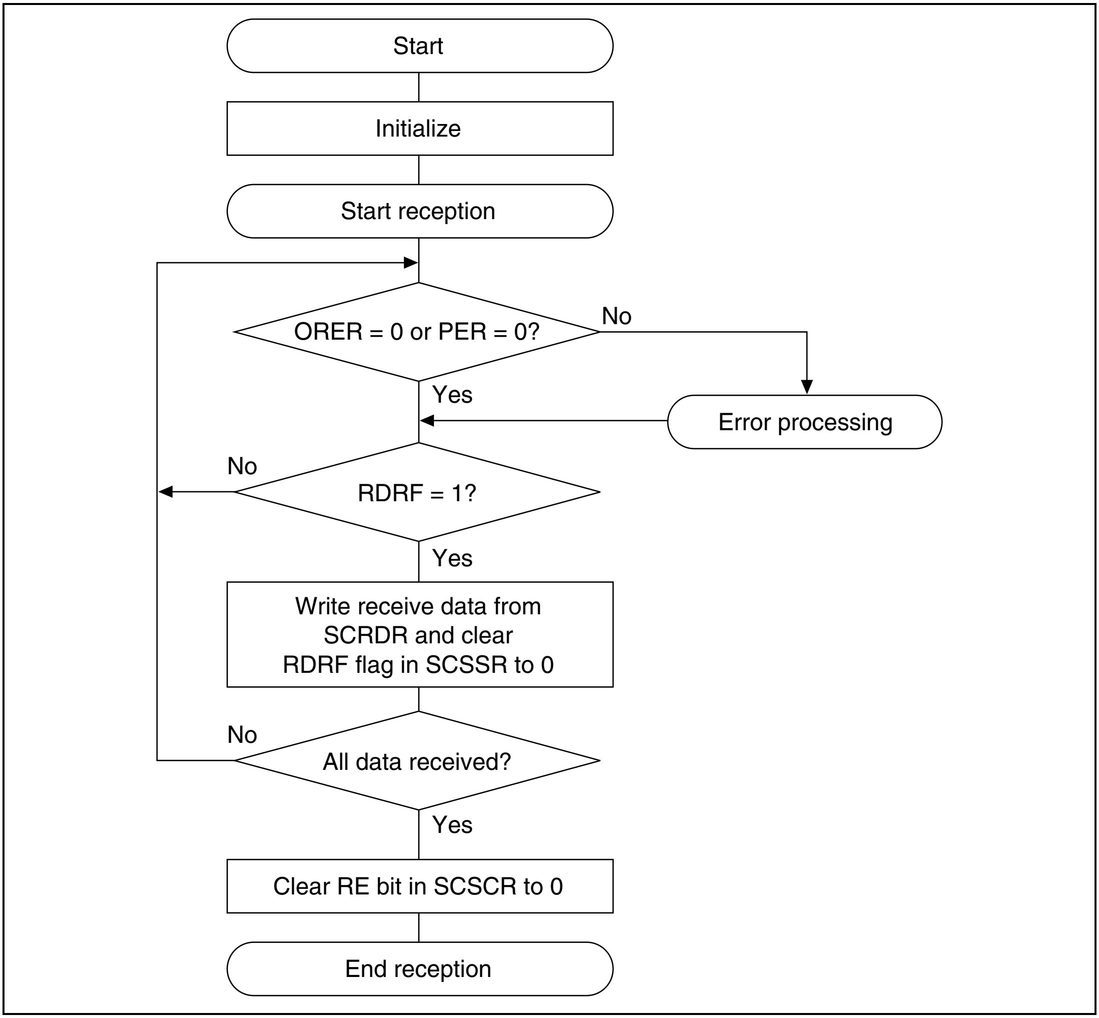


Figure 15.7 Reception Flowchart (Example)

Switching Modes: When switching from receive mode to transmit mode, check that the receive operation is completed before starting initialization and setting RE to 0 and TE to 1. The RDRF, PER, and ORER flags can be used to check if reception is completed. When switching from transmit mode to receive mode, check that the transmit operation is completed before starting initialization and setting TE to 0 and RE to 1. The TEND flag can be used to check if transmission is completed.

Interrupt Operation: In the smart card interface mode, there are three types of interrupts: transmit-data-empty (TXI), communication error (ERI) and receive-data-full (RXI). In this mode, the transmit-end interrupt (TEI) cannot be requested.

Set the TEND flag in SCSSR to 1 to request a TXI interrupt. Set the RDRF flag in SCSSR to 1 to request an RXI interrupt. Set the ORER, PER, or FER/ERS flag in SCSSR to 1 to request an ERI interrupt (table 15.8).

Table 15.8 Smart Card Mode Operating State and Interrupt Sources

Mode	State	Flag	Mask Bit	Interrupt Source
Transmit mode	Normal	TEND	TIE	TXI
	Error	FER/ERS	RIE	ERI
Receive mode	Normal	RDRF	RIE	RXI
	Error	PER, ORER	RIE	ERI

15.5 Notes for Usage

When the SCI is used as a smart card interface, be sure that all criteria in sections 15.4.1 and 15.4.2 are applied.

Receive Data Timing and Receive Margin in Asynchronous Mode:

In asynchronous mode, the SCI runs on a basic clock with a frequency of 372 times the transfer rate. During reception, the SCI samples the falling of the start bit using the base clock to achieve internal synchronization. Receive data is latched internally on the rising edge of the 186th basic clock cycle (figure 15.8).

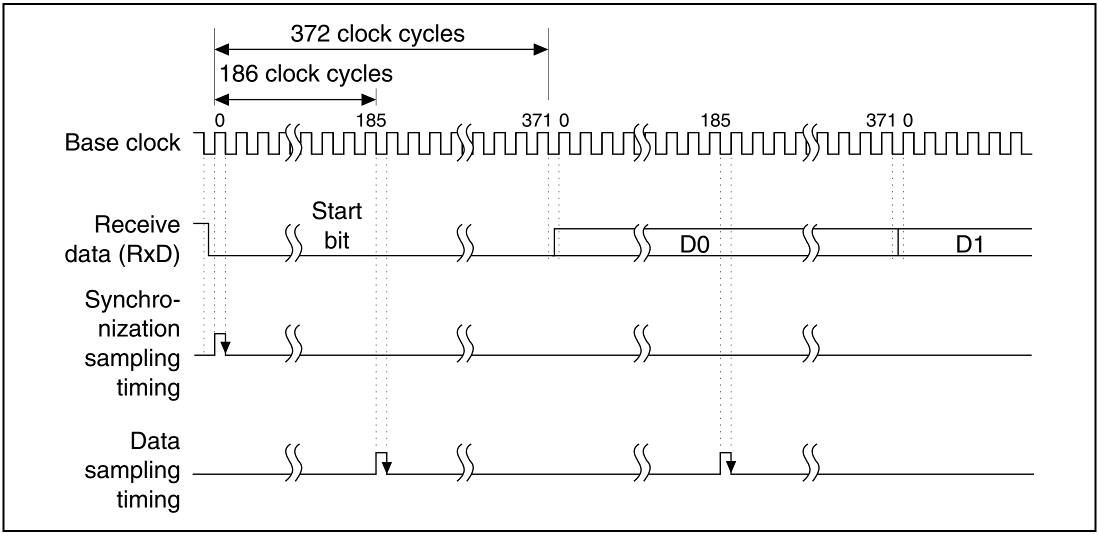


Figure 15.8 Receive Data Sampling Timing in Smart Card Mode

The receive margin is found from the following equation:

For smart card mode:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where:

M = Receive margin (%)
N = Ratio of bit rate to clock (N = 372)
D = Clock duty (D = 0 to 1.0)
L = Frame length (L = 10)
F = Absolute value of clock frequency deviation

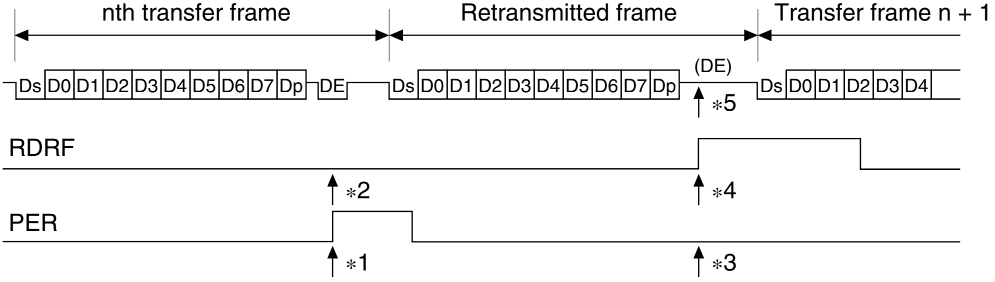
Using this equation, the receive margin when F = 0 and D = 0.5 is as follows:

$$M = (0.5 - 1/2 \times 372) \times 100\% = 49.866\%$$

Retransmission (Receive and Transmit Modes):

Retransmission by the SCI in Receive Mode: Figure 15.9 shows the retransmission operation in the SCI receive mode.

1. When the received parity bit is checked and an error is found, the PER bit in SCSSR is automatically set to 1. If the RIE bit in SCSCR is enabled at this time, an ERI interrupt is requested. Be sure to clear the PER bit before the next parity bit is sampled.
2. The RDRF bit in SCSSR is not set in the frame that caused the error.
3. When the received parity bit is checked and no error is found, the PER bit in SCSSR is not set.
4. When the received parity bit is checked and no error is found, reception is considered to have been completed normally and the RDRF bit in SCSSR is automatically set to 1. If the RIE bit in SCSCR is enabled at this time, an RXI interrupt is requested.
5. When a normal frame is received, the pin maintains a three-state state when it transmits the error signal.

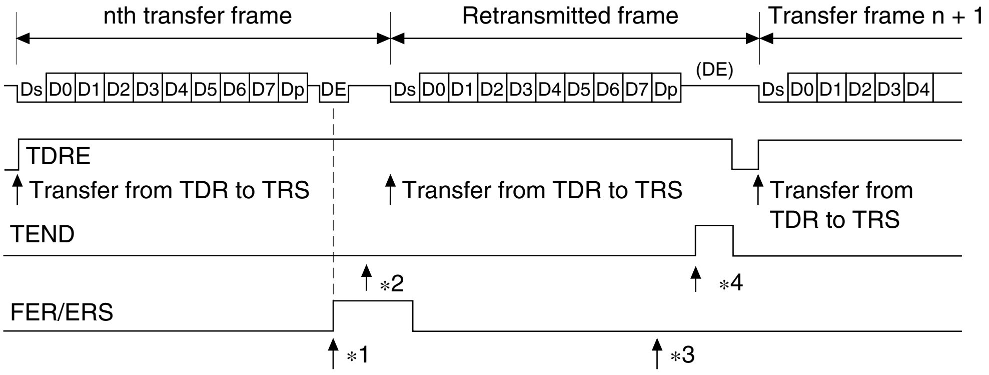


Notes: 1. This portion corresponds to the above explanation 1.
2. This portion corresponds to the above explanation 2.
3. This portion corresponds to the above explanation 3.
4. This portion corresponds to the above explanation 4.
5. This portion corresponds to the above explanation 5.

Figure 15.9 Retransmission in SCI Receive Mode

Retransmission by the SCI in Transmit Mode: Figure 15.10 shows the retransmission operation in the SCI transmit mode.

1. After transmission of one frame is completed, the FER/ERS bit in SCSSR is set to 1 when a error signal is returned from the receiving side. If the RIE bit in SCSCR is enabled at this time, an ERI interrupt is requested. Be sure to clear the FER/ERS bit before the next parity bit is sampled.
2. The TEND bit in SCSSR is not set in the frame that received the error signal that indicated the error.
3. The FER/ERS bit in SCSSR is not set when no error signal is returned from the receiving side.
4. When no error signal is returned from the receiving side, the TEND bit in SCSSR is set to 1 when the transmission of the frame that includes the retransmission is considered completed. If the TIE bit in SCSCR is enabled at this time, a TXI interrupt will be requested.



Notes: 1. This portion corresponds to the above explanation 1.
2. This portion corresponds to the above explanation 2.
3. This portion corresponds to the above explanation 3.
4. This portion corresponds to the above explanation 4.

Figure 15.10 Retransmission in SCI Transmit Mode

Support for Block Transfer Mode :

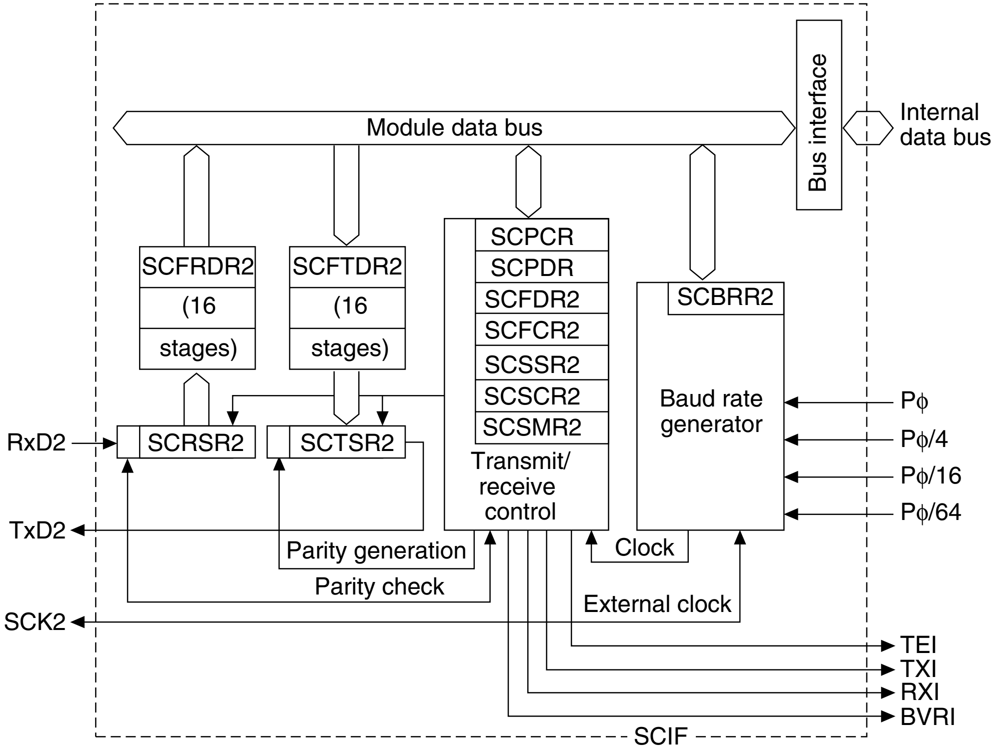
This smart card interface conforms to the T = 0 (character transfer) protocols of ISO/IEC7816-3. As a result, this smart card interface does not support block transfer, in which error signals are neither sent nor detected, and data is not automatically retransmitted.

Section 16 Serial Communication Interface with FIFO (SCIF)

This LSI has single-channel serial communication interface with FIFO (SCIF) that supports asynchronous serial communication. It also has 16-stage FIFO registers for both transfer and receive that enables this LSI efficient high-speed continuous communication. Figure 16.1 shows a diagram of the SCIF, and figures 16.2 to 16.4 show the I/O ports.

16.1 Features

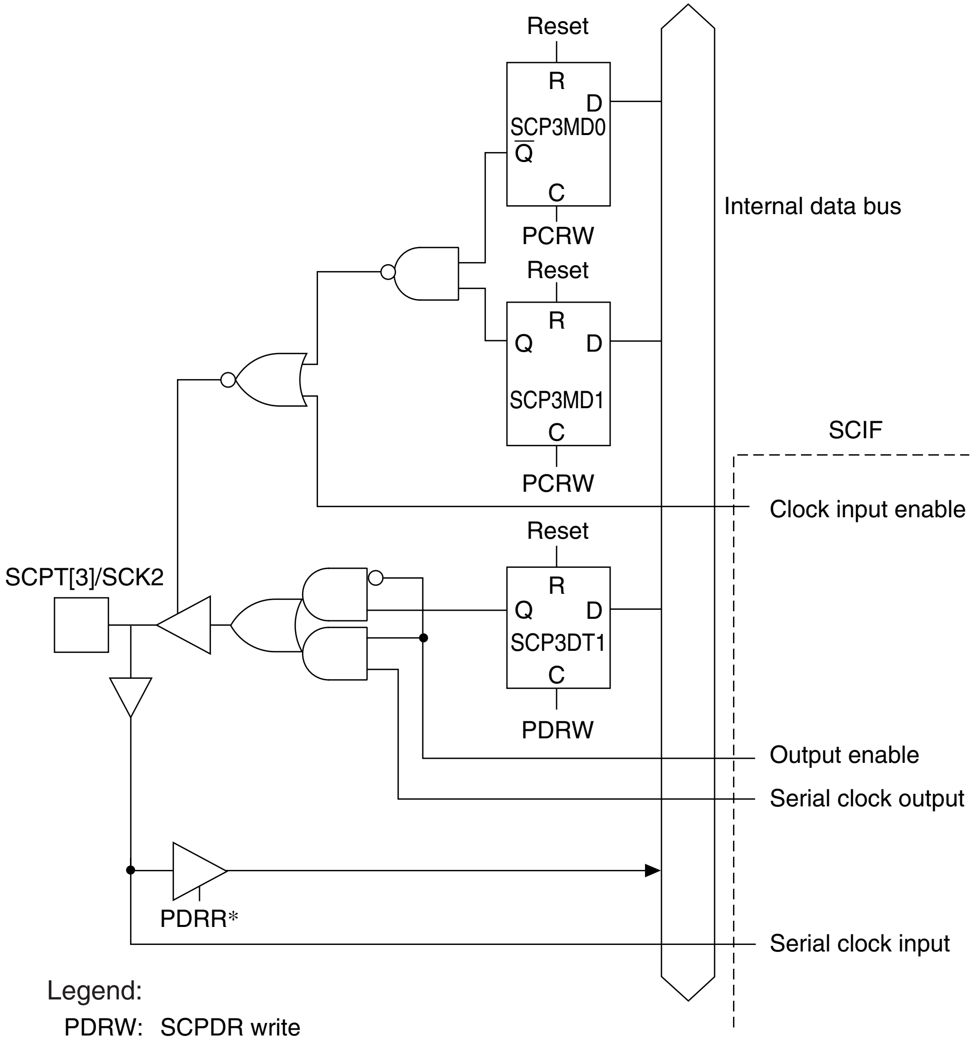
- Asynchronous serial communication:
  - Serial data communications are performed by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: Seven or eight bits
  - Stop bit length: One or two bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity and framing errors
  - Break detection:
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK2 pin (external)
- Four types of interrupts: Transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive-error interrupts are requested independently. The direct memory access controller (DMAC) can be activated to execute a data transfer by a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- On-chip modem control functions ($\overline{\text{RTS2}}$ and $\overline{\text{CTS2}}$)
- The quantity of data in the transmit and receive FIFO registers and the number of receive errors of the receive data in the receive FIFO register can be known.
- The time-out error (DR) can be detected in receiving.



Legend:

SCRSR2:	Receive shift register 2	SCSSR2:	Serial status register 2
SCFRDR2:	Receive FIFO data register 2	SCBRR2:	Bit rate register 2
SCTSR2:	Transmit shift register 2	SCFCR2:	FIFO control register 2
SCFTDR2:	Transmit FIFO data register 2	SCFDR2:	Number of FIFO data register 2
SCSMR2:	Serial mode register 2	SCPDR:	Port SC data register
SCSCR2:	Serial control register 2	SCPCR:	Port SC control register

Figure 16.1 SCIF Block Diagram



Legend:

PDRW:	SCPDR write
PDRR:	SCPDR read
PCRW:	SCPCR write

Note: * When reading the SCK2 pin, clear the CKE1 and CKE0 bits in SCSCR to 0, and set the SCP3MD1 bit in SCSPR to 1.

Figure 16.2 SCPT[3]/SCK2 Pin

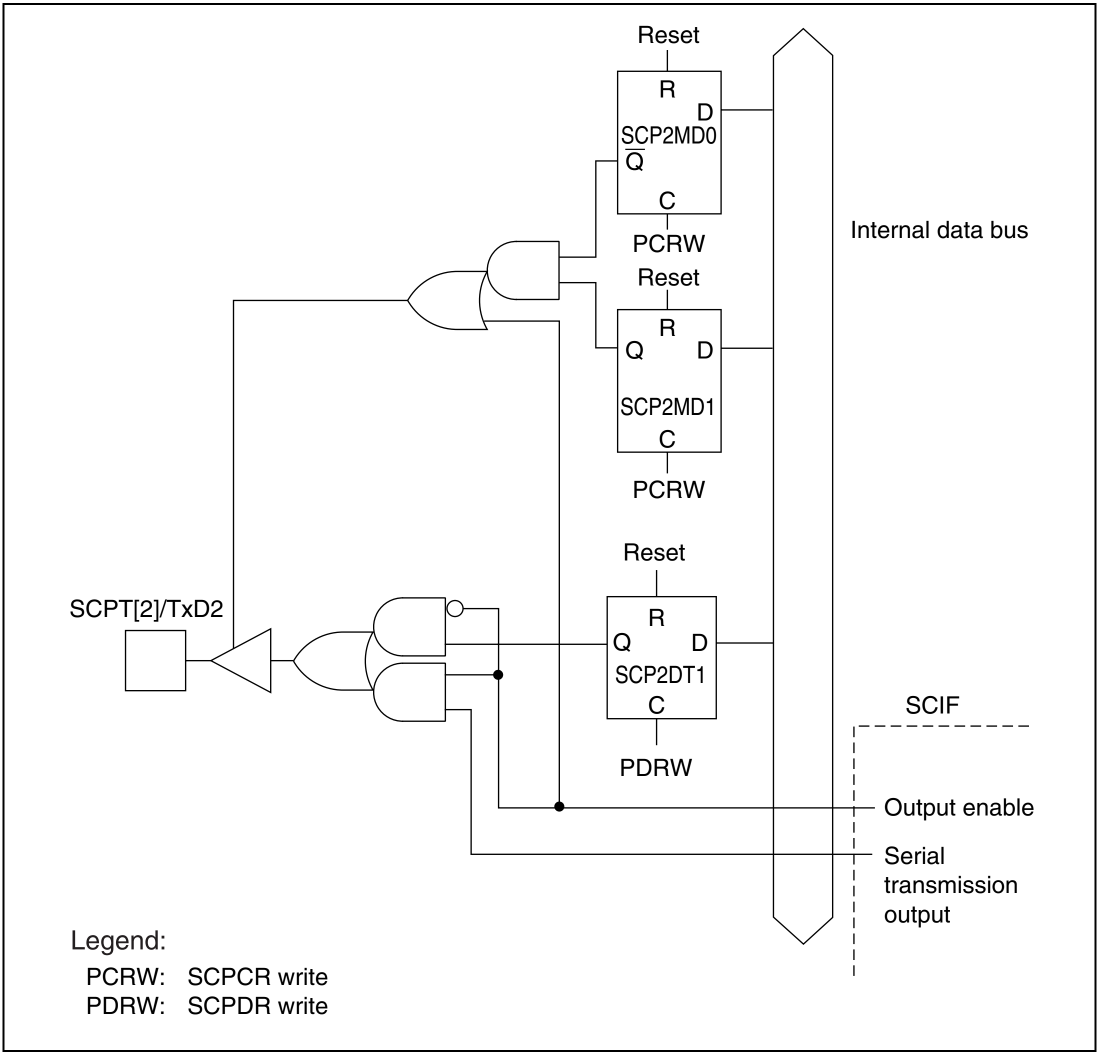
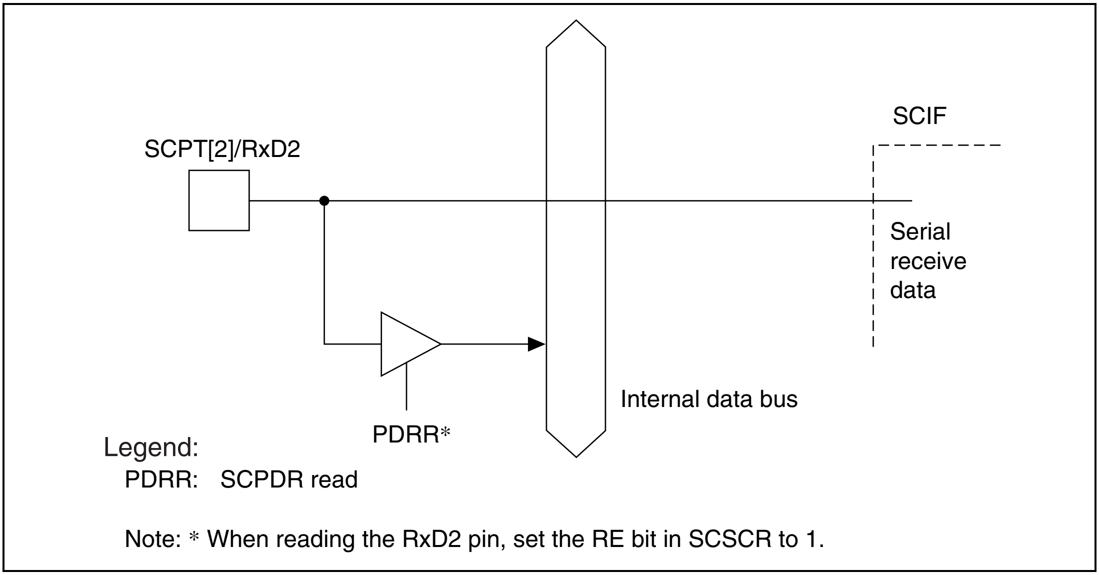


Figure 16.3 SCPT[2]/TxD2 Pin



Legend:
PDRR: SCPDR read

Note: * When reading the RxD2 pin, set the RE bit in SCSCR to 1.

Figure 16.4 SCPT[2]/RxD2 Pin

16.2 I/O Pins

The SCIF has the I/O pins summarized in table 16.1.

Table 16.1 SCIF Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK2	I/O	Clock I/O
Receive data pin	RxD2	Input	Receive data input
Transmit data pin	TxD2	Output	Transmit data output
Request to send pin	$\overline{\text{RTS2}}$	Output	Request to send
Clear to send pin	$\overline{\text{CTS2}}$	Input	Clear to send

16.3 Description of Registers

SCIF has the registers listed below. These registers specify the data format and bit rate, and control the transmitter and receiver sections.

Refer to section 23, Control Registers Table, for more detail of the addresses and access size.

- Serial mode register 2 (SCSMR2)
- Bit rate register 2 (SCBRR2)
- Serial control register 2 (SCSCR2)
- Transmit FIFO data register 2 (SCFTDR2)
- Serial status register 2 (SCSSR2)
- Receive data FIFO register 2 (SCFRDR2)
- FIFO control register 2 (SCFCR2)
- FIFO data count set register 2 (SCFDR2)
- SC port control register (SCPCR)
- SC port data register (SCPDR)

16.3.1 Receive Shift Register 2 (SCRSR2)

The receive shift register 2 (SCRSR2) is an eight-bit register taht receives serial data. The CPU cannot read from or write to the SCRSR2 directly. Data input at the RxD pin is loaded into the SCRSR2 in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the SCFRDR2, which is a receive FIFO register.

16.3.2 Receive FIFO Data Register 2 (SCFRDR2)

The 16-byte receive FIFO data register2(SCFRDR2) stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the SCRSR into the SCFRDR2 for storage. Continuous receive is possible until 16 bytes are stored.

The CPU can read but not write the SCFRDR2. When data is read without received data in the SCFRFR2, the value is undefined. When the received data in this register becomes full, the subsequent serial data is lost.

16.3.3 Transmit Shift Register 2 (SCTSR2)

The transmit shift register 2 (SCTSR2) is an eight-bit register that transmits serial data. The CPU cannot read from or write to the SCTSR2 directly. The SCI loads transmit data from the SCFTDR into the SCTSR2, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from the SCFTDR2 into the SCTSR2 and starts transmitting again.

16.3.4 Transmit FIFO Data Register 2 (SCFTDR2)

The transmit FIFO data register 2 (SCFTDR2) is a 16-byte FIFO register that stores data for serial transmission. When the SCIF detects that the SCTSR is empty, it moves transmit data written in the SCFTDR2 into the SCTSR2 and starts serial transmission. Continuous serial transmission is performed until the transmit data in the SCFTDR2 becomes empty. The CPU can always write to the SCFTDR2.

When the transmit data in the SCFTDR2 is full (16 bytes), next data cannot be written. If attempted to write, the data is ignored.

16.3.5 Serial Mode Register 2 (SCSMR2)

The serial mode register2 (SCSMR2) is an eight-bit register that specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write the SCSMR2.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read 0. The write value should always be 0.
6	CHR	0	R/W	Character Length Selects seven-bit or eight-bit data in the asynchronous mode. 0: Eight-bit data. 1: Seven-bit data. Note: When seven-bit data is selected, the MSB (bit 7) in SCFTPR2 is not transmitted.
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data. 0: Parity bit not added or checked. 1: Parity bit added and checked. Note: When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/E) setting. Receive data parity is checked according to the even/odd (O/E) mode setting.

Bit	Bit Name	Initial Value	R/W	Description
4	O/$\overline{\text{E}}$	0	R/W	Parity Mode Selects even or odd parity when parity bits are added and checked. The O/$\overline{\text{E}}$ setting is used only when the PE is set to 1 to enable parity addition and check. The O/$\overline{\text{E}}$ setting is ignored when parity addition and check is disabled. 0: Even parity. Note: If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. 1: Odd parity. Note: If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length. In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. 0: One stop bit. Note: In transmitting, a single bit of 1 is added at the end of each transmitted character. 1: Two stop bits. Note:In transmitting, two bits of 1 are added at the end of each transmitted character.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 16.3.8, Bit Rate Register2 (SCBRR2). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

16.3.6 Serial Control Register 2 (SCSCR2)

The serial control register 2 (SCSCR2) operates the SCI transmitter/receiver, selects the serial clock output in the asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write the SCSCR2.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-empty interrupt (TXI) requested when the serial transmit data is transferred from the SCFTDR2 to SCTSR2, and the quantity of data in the SCFTDR2 becomes less than the specified number of transmission triggers, and then the TDFE flag in the SCSSR2 is set to1. 0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled. Note: The TXI interrupt request can be cleared by writing the greater quantity of transmit data than the specified number of transmission triggers to SCFTDR2 and by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0. 1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive-data-full (RXI) and receive-error (ERI) interrupts requested when the serial receive data is transferred from the SCRSR2 to SCFRDR2, when the quantity of data in the SCFRDR2 becomes more than the specified number of receive triggers, and when the RDRF flag of SCSSR2 is set to1. 0: Receive-data-full interrupt (RXI), receive-error interrupt (ERI), and receive break interrupt (BRI) requests are disabled. Note: RXI and ERI interrupt requests can be cleared by reading the DR, ER, or RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. At RDF, read 1 from the RDF flag and clear it to 0, after reading the received data from SCFRDR2 until the quantity of received data becomes less than the specified number of the receive triggers. 1: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled.
5	TE	0	R/W	Transmit Enable Enables or disables the SCIF serial transmitter. 0: Transmitter disabled. 1: Transmitter enabled. Note: Serial transmission starts after writing of transmit data into the SCFTDR2. Select the transmit format in the SCSMR2 and SCFCR2 and reset the TFIFO before setting TE to 1.
4	RE	0	R/W	Receive Enable Enables or disables the SCIF serial receiver. 0: Receiver disabled. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, FER and PER). These flags retain their previous values. 1: Receiver enabled. Serial reception starts when a start bit is detected. Select the receive format in the SCSMR2 before setting RE to 1.
3	—	0	R/W	Reserved
2	—	0	R/W	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable
0	CKE0	0	R/W	These bits select the SCIF clock source and enable or disable clock output from the SCK2 pin. Depending on the combination of CKE1 and CKE0, the SCK2 pin can be used for serial clock output or serial clock input. The CKE0 setting is valid only when the SCI is operating with the internal clock (CKE1 = 0). The CKE0 setting is ignored when an external clock source is selected (CKE1 = 1). Always select the SCIF operating mode in the SCSMR2, before setting CKE1 and CKE0. For further details on selection of the SCIF clock source, see table 16.7 in section 16.3, Operation. 00: Internal clock, SCK pin used for input pin (input signal is ignored) 01: Internal clock, SCK2 pin used for clock output*1 10: External clock, SCK2 pin used for clock input*2 11: External clock, SCK2 pin used for clock input*2 Notes: 1. The output clock frequency is 16 times the bit rate. 2. The input clock frequency is 16 times the bit rate.

16.3.7 Serial Status Register 2 (SCSSR2)

The serial status register 2 (SCSSR2) is a 16-bit register. The upper 8 bits indicate the number of receive errors in the data of the SCFRDR2, and the lower 8 bits indicate SCIF operating state.

The CPU can always read and write the SCSSR2, but cannot write 1 to the status flags (ER, TEND, TDFE, BRK, OPER, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 3 (FER) and 2 (PER) are read-only bits and cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PER3 to PER0	0	R	Number of parity errors: These bits indicate the number of data items that contain a parity error in the receive data stored in the SCFRDR2. (The number of parity errors in the SCFRDR2)
11 to 8	FER3 to FER0	0	R	Number of framing errors: These bits indicate the number of data items that contain a framing error in the receive data stored in the SCFRDR2. (The number of framing errors in the SCFRDR2)

Bit	Bit Name	Initial Value	R/W	Description
7	ER	0	R/(W)*	Receive error: Indicates that a framing error or a parity error, when receiving data containing parity bits, has occurred. 0: Receive is in progress, or receive is normally completed.*1 [Clearing conditions] 1. The chip is power-on reset or enters standby mode. 2. ER is read as 1, then written to with 0. Note :1. Clearing the RE bit to 0 in SCSCR2 does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the received data is transferred to SCFRDR2 and the receive operation is continued. Whether or not the data read from SCRDR2 includes a receive error can be detected by the FER and PER bits of SCSSR2. 1: A framing error or a parity error has occurred during receiving. ER is set to 1 when the stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one-data receive*, or when the total number of 1's in the received data and in the parity bit does not match the even/odd parity specification specified by the $O/\overline{E}$ bit of the SCSMR. [Setting conditions] 1. The stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one-data receive.*2 2. The total number of 1's in the received data and in the parity bit does not match the even/odd parity specification specified by the $O/\overline{E}$ bit of the SCSMR2. Note: 2. n the stop mode, only the first stop bit is checked; the second stop bit is not checked.

Bit	Bit Name	Initial Value	R/W	Description
6	TEND	1	R/(W)*	Transmit End Indicates that when the last bit of a serial character was transmitted, the SCFTDR2 did not contain valid data, so transmission has ended. 0: Transmission is in progress. [Clearing condition] Data is written to SCFTDR2. 1: End of transmission [Setting conditions] 1. When the chip is reset or enters standby mode, TE in the SCSCR2 is cleared to 0. 2. SCFTDR2 contains no transmit data when the last bit of a one-byte serial character is transmitted.

Bit	Bit Name	Initial Value	R/W	Description
5	TDFE	1	R/(W)*	Transmit FIFO Data Empty Indicates that data is transferred from SCFTDR2 to SCTSR2, the quantity of data in SCFTDR2 becomes less than the number of transmission triggers specified by the TTRG1 and TTRG0 bits in SCFCR2, and writing the transmit data to SCFTDR is enabled. 0: The quantity of transmit data written to SCFTDR is greater than the specified number of transmission triggers. [Clearing condition] Data exceeding the specified number of transmission triggers is written to SCFTDR2, software reads TDFE after it has been set to 1, then writes 0 to TDFE. 1: End of transmission [Setting conditions] 1. The chip is power-on reset or enters standby mode. 2. The quantity of transmission data in SCFTDR2 becomes less than the specified number of transmission triggers as a result of transmission. Note: * Since SCFTDR2 is a 16-byte FIFO register, the maximum quantity of data which can be written when TDFE is 1 is "16 minus the specified number of transmission triggers". If attempted to write additional data, the data is ignored. The quantity of data in SCFTDR2 is indicated by the upper 8 bits of SCFTDR2.

Bit	Bit Name	Initial Value	R/W	Description
4	BRK	0	R/(W)*	Break Detection Indicates that a break signal is detected in received data. 0: No break signal is being received. [Clearing conditions] 1. The chip is power-on reset or enters standby mode. 2. BRK is read as 1, then written to with 0. 1: A break signal is received. [Setting conditions] 1. Data including a framing error is received. 2. A framing error with space 0 occurs in the subsequent received data. Note: When a break is detected, transfer of the received data (H'00) to SCFRDR2 stops after detection. When the break ends and the receive signal becomes mark 1, the transfer of the received data resumes. The received data of a frame in which a break signal is detected is transferred to SCFRDR2. After this, however, no received data is transferred until a break ends with the received signal being mark 1 and the next data is received.
3	FER	0	R	Framing Error Indicates a framing error in the data read from the SCFRDR2. 0: No framing error occurred in the data read from SCFRDR2. [Clearing conditions] 1. The chip is power-on reset or enters standby mode. 2. No framing error is present in the data read from SCFRDR2. 1: A framing error occurred in the data read from SCFRDR2. [Setting condition] A framing error is present in the data read from SCFRDR2

Bit	Bit Name	Initial Value	R/W	Description
2	PER	0	R	Parity Error Indicates a parity error in the data read from the SCFRDR2. 0: No parity error occurred in the data read from SCFRDR2. [Clearing conditions] 1. The chip is power-on reset or enters standby mode. 2. No parity error is present in the data read from SCFRDR2. 1: A parity error occurred in the data read from SCFRDR2. [Setting condition] A parity error is present in the data read from SCFRDR2

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/(W)*	Receive FIFO Data Full Indicates that received data is transferred to the SCFRDR2, the quantity of data in SCFRDR becomes more than the number of receive triggers specified by the RTRG1 and RTRG0 bits in SCFCR2. 0: The quantity of transmit data written to SCFRDR2 is less than the specified number of receive triggers. [Clearing conditions] 1. The chip is power-on reset or enters standby mode. 2. The SCFRDR2 is read until the quantity of receive data in SCFRDR2 becomes less than the specified number of receive triggers. 3. RDF is read as 1, then written to with 0. 1: The quantity of receive data in SCFRDR2 is more than the specified number of receive triggers. [Setting condition] The quantity of receive data which is greater than the specified number of receive triggers is being stored to SCFRDR2.* Note: * Since SCFTDR2 is a 16-byte FIFO register, the maximum quantity of data which can be read when RDF is 1 is the specified number of receive triggers. If attempted to read after all data in the SCFRDR2 have been read, the data is undefined. The quantity of receive data in SCFRDR2 is indicated by the lower 8 bits of SCFTDR2.

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/(W)*	Receive Data Ready Indicates that the SCFRDR2 stores the data which is less than the specified number of receive triggers, and that next data is not yet received after 15 etu has elapsed from the last stop bit. 0: Receive is in progress, or no received data remains in SCFRDR2 after the receive ended normally. [Clearing conditions] 1. The chip is power-on reset or enters standby mode. 2. DR is read as 1, then written to with 0. 1: Next receive data is not received. [Setting condition] SCFRDR2 stores the data which is less than the specified number of receive triggers, and that next data is not yet received after 15 etu has elapsed from the last stop bit.* Note: * This is equivalent to 1.5 frames with the 8-bit 1-stop-bit format. (etu: Element Time Unit)

Note: * The only value that can be written is 0 to clear the flag.

16.3.8 Bit Rate Register 2 (SCBRR2)

The bit rate register 2 (SCBRR2) is an eight-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the SCSMR2, determines the serial transmit/receive bit rate.

The CPU can always read and write the SCBRR2. The SCBRR2 is initialized to H'FF by a reset or in module standby or standby mode. Each channel has independent baud rate generator control, so different values can be set in two channels.

The SCBRR2 setting is calculated as follows:

Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bit/s)
N: SCBRR2 setting for baud rate generator ($0 \le N \le 255$)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 16.2.)

Table 16.2 SCSMR2 Settings

		SCSMR2 Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: Find the bit rate error by the following formula:

$$\text{Error (\%)} = \left\{ \frac{\text{Pø} \times 10^6}{(N+1) \times 64 \times 2^{2n-1} \times B} - 1 \right\} \times 100$$

Table 16.3 lists examples of SCBRR2 settings.

Table 16.3 Bit Rates and SCBRR2 Settings

	Pφ (MHz)								
	7.3728			8			9.8304		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	130	–0.07	2	141	0.03	1	174	–0.26
150	2	95	0.00	2	103	0.16	1	127	0.00
300	1	191	0.00	1	207	0.16	0	255	0.00
600	1	95	0.00	1	103	0.16	0	127	0.00
1200	0	191	0.00	0	207	0.16	0	255	0.00
2400	0	95	0.00	0	103	0.16	0	127	0.00
4800	0	47	0.00	0	51	0.16	0	63	0.00
9600	0	23	0.00	0	25	0.16	0	31	0.00
19200	0	11	0.00	0	12	0.16	0	15	0.00
31250	0	6	5.33	0	7	0.00	0	9	–1.70
38400	0	5	0.00	0	6	–6.99	0	1	0.00

	Pφ (MHz)								
	10			12			12.288		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	–0.25	1	212	0.03	2	217	0.08
150	2	129	0.16	1	155	0.16	2	159	0.00
300	2	64	0.16	1	77	0.16	2	79	0.00
600	1	129	0.16	0	155	0.16	1	159	0.00
1200	1	64	0.16	0	77	0.16	1	79	0.00
2400	0	129	0.16	0	38	0.16	0	159	0.00
4800	0	64	0.16	0	19	0.16	0	79	0.00
9600	0	32	–1.36	0	9	0.16	0	39	0.00
19200	0	15	1.73	0	4	0.16	0	19	0.00
31250	0	9	0.00	0	2	0.00	0	11	2.40
38400	0	7	1.73	0	9	–2.34	0	9	0.00

	Pφ (MHz)											
	14.7456			16			19.6608			20		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	64	0.70	3	70	0.03	3	86	0.31	3	88	–0.25
150	2	191	0.00	2	207	0.16	2	255	0.00	2	64	0.16
300	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
600	1	191	0.00	1	207	0.16	1	255	0.00	1	64	0.16
1200	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
2400	0	191	0.00	0	207	0.16	0	255	0.00	0	64	0.16
4800	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
9600	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
19200	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
31250	0	14	–1.70	0	15	0.00	0	19	–1.70	0	19	0.00
38400	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
115200	0	3	0.00	0	3	8.51	0	4	6.67	0	4	8.51
500000	0	0	–7.84	0	0	0.00	0	0	22.9	0	0	25.0

	Pφ (MHz)											
	24			24.576			28.7			30		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	106	–0.44	3	108	0.08	3	126	0.31	3	132	0.13
150	3	77	0.16	3	79	0.00	3	92	0.46	3	97	–0.35
300	2	155	0.16	2	159	0.00	2	186	–0.08	2	194	0.16
600	2	77	0.16	2	79	0.00	2	92	0.46	2	97	–0.35
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	194	0.16
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	97	–0.35
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	194	–1.36
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	97	–0.35
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	48	–0.35
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	29	0.00
38400	0	19	–2.34	0	19	0.00	0	22	1.55	0	23	1.73
115200	0	6	–6.99	0	6	–4.76	0	7	–2.68	0	7	1.73
500000	0	1	–25.0	0	1	–23.2	0	1	–10.3	0	1	–6.25

	Pϕ (MHz)		
	33.34		
Bit Rate (bits/s)	n	N	Error (%)
110	3	147	0.00
150	3	108	-0.43
300	2	216	0.03
600	2	108	-0.43
1200	1	216	0.03
2400	1	108	-0.43
4800	0	216	0.03
9600	0	108	-0.43
19200	0	53	0.49
31250	0	32	1.03
38400		26	0.49
11520	0	8	0.49
500000	0	1	4.19

Table 16.4 lists the maximum bit rates in the asynchronous mode when the baud rate generator is used. Table 16.5 lists the maximum bit rates when an external clock input is used.

Table 16.4 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	Maximum Bit Rate (bits/s)	n	N
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0

Table 16.5 Maximum Bit Rates during External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750

16.3.9 FIFO Control Register 2 (SCFCR2)

The FIFO control register 2 (SCFCR2) resets the number of data in the SCFTDR2 and SCFRDR2, sets the number of trigger data, and contains an enable bit for the loop back test. The SCFCR2 is always read and written by the CPU.

Bit	Bit Name	Initial Value	R/W	Description
7	RTRG1	0	R/W	Trigger of the Number of Receive FIFO Data
6	RTRG0	0	R/W	Set the reference number of the receive data full. The RDF in SCSSR2 is set to 1, when the receiving data count has exceeded the following trigger number. Trigger number of receive data. 00: 1 01: 4 10: 8 11: 14
5	TTRG1	0	R/W	Trigger of the Number of Transmit FIFO Data
4	TTRG0	0	R/W	Set the reference number of the send data empty. The TDFE in SCSSR2 is set to 1, when the transmitting data count has fallen the following trigger number. Trigger number of transmit data. 00: 8 (8) 01: 4 (12) 10: 2 (14) 11: 1 (15) Note: * Values in brackets mean the number of empty bytes in SCFTDR when the TDFE is set.
3	MCE	0	R/W	Modem Control Enable Enables the modem control signals $\overline{\text{CTS2}}$ and $\overline{\text{RTS2}}$. 0: Disables the modem signal* 1: Enables the modem signal Note: * The $\overline{\text{CTS2}}$ is fixed to active 0 regardless of the input value, and the $\overline{\text{RTS2}}$ is also fixed to 0.

Bit	Bit Name	Initial Value	R/W	Description
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Cancels the transmit data in the SCFTDR2 and resets the data to the empty state. 0: Disables reset operation* 1: Enables reset operation Note: * The reset is executed in a hardware reset or the standby mode.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Cancels the receive data in the SCFRDR2 and resets the data to the empty state. 0: Disables reset operation* 1: Enables reset operation Note: * The reset is executed in a hardware reset or the standby mode.
0	LOOP	0	R/W	Loop Back Test Internally connects the transmit output pin (TXD2) and receive input pin (RXD2) and enables the loop back test. 0: Disables the loop back test 1: Enables the loop back test

16.3.10 FIFO Data Count Set Register 2 (SCFDR2)

The SCFDR2 is a 16-bit register which indicates the number of data stored in the SCFTDR2 and SCFRDR2. The SCFDR2 is always read from the CPU.

The upper eight bits of this register indicate the number of transmit data items stored in the SCFTDR2 that have not yet been transmitted. The H'00 means no transmit data, and the H'10 means that the full of transmit data are stored in the SCFTDR2.

The lower eight bits of this register indicate the number of receive data items stored in the SCFRDR2. The H'00 means no receive data, and the H'10 means that the full of receive data are stored in the SCFRDR2.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	0	R	Reserved These bits are always read as 0.
12 to 8	T4 to T0	0	R	Number of non-transmitted data.
7 to 5	—	0	R	Reserved These bits are always read as 0.
4 to 0	R4 to R0	0	R	Number of received data.

16.3.11 SC Port Control Register (SCPCR)

For information about the SC port control register (SCPCR), see section 14.3.8, SC Port Control Register (SCPCR).

16.3.12 SC Port Data Register (SCPDR)

For information about the SC port data register (SCPDR), see section 14.3.9, SC Port Data Register (SCPDR).

16.4 Description of Operation

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually. Refer to section 14.4.1, Operation in Asynchronous Mode (SCI). The SCIF has the 16-byte FIFO buffer for both transmit and receive, reduces an overhead of the CPU, and enables continuous high-speed communication. Moreover, it has the $\overline{\text{RTS2}}$ and $\overline{\text{CTS2}}$ signals as the modem control signals. The transmission format is selected in the SCSMR2, as listed in table 16.6. The SCI clock source is selected by the combination of the CKE1 and CKE0 bits in SCSCR2, as listed in table 16.6.

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (one or two bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), receive FIFO data full, receive data ready, and breaks.
- In transmitting, it is possible to detect transmit FIFO data empty.
- The number of stored data for both the transmit and receive FIFO registers is displayed.
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the on-chip baud rate generator, and can output a serial clock signal with a frequency 16 times the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Table 16.6 SCSMR2 Settings and SCIF Communication Formats

	SCSMR2 Settings					SCIF Communication Format
Mode	**Bit 6 CHR**	**Bit 5 PE**	**Bit 3 STOP**	**Data Length**	**Parity Bit**	**Stop Bit Length**
Asynchronous	0	0	0	8-bit	Not set	1 bit
			1			2 bits
		1	0		Set	1 bit
			1			2 bits
	1	0	0	7-bit	Not set	1 bit
			1			2 bits
		1	0		Set	1 bit
			1			2 bits

Table 16.7 SCSCR2 and SCSCR2 Settings and SCIF Clock Source Selection

	SCSCR2 Settings		SCIF Transmit/Receive Clock	
Mode	Bit 1 CKE1	Bit 0 CKE0	Clock Source	SCK2 Pin Function
Asynchronous mode	0	0	Internal	SCIF does not use the SCK2 pin
		1		Outputs a clock with a frequency 16 times the bit rate
	1	0	External	Inputs a clock with frequency 16 times the bit rate
		1		

16.4.1 Serial Operation

Transmit/Receive Formats: Table 16.8 lists eight communication formats that can be selected. The format is selected by settings in the SCSMR2.

Table 16.8 Serial Communication Formats

SCSMR2 Bits			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-Bit data								STOP		
0	0	1	START	8-Bit data								STOP	STOP	
0	1	0	START	8-Bit data								P	STOP	
0	1	1	START	8-Bit data								P	STOP	STOP
1	0	0	START	7-Bit data							STOP			
1	0	1	START	7-Bit data							STOP	STOP		
1	1	0	START	7-Bit data							P	STOP		
1	1	1	START	7-Bit data							P	STOP	STOP	

START: Start bit

STOP: Stop bit

P: Parity bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK2 pin can be selected as the SCIF transmit/receive clock. The clock source is selected by bits CKE1 and CKE0 in the serial control register (SCSCR2) (table 16.7).

When an external clock is input at the SCK2 pin, it must have a frequency equal to 16 times the desired bit rate.

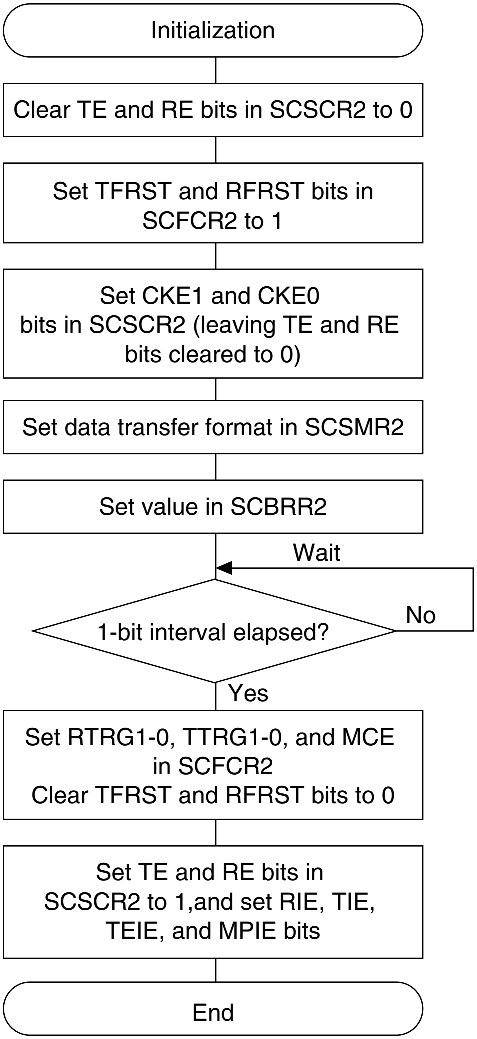
When the SCIF operates on an internal clock, it can output a clock signal at the SCK2 pin. The frequency of this output clock is 16 times the bit rate.

Transmitting and Receiving Data (SCIF Initialization): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR2), then initialize the SCIF as follows.

When changing the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 initializes the transmit shift register (SCTSR2). Clearing TE and RE to 0, however, does not initialize the serial status register (SCSSR2), transmit FIFO data register (SCFTDR2), or receive FIFO data register (SCFRDR2), which retain their previous contents. Clear TE to 0 after all transmit data are transmitted and the TEND flag in the SCSSR2 is set. The transmitting data enters the high impedance state after clearing to 0 although the bit can be cleared to 0 in transmitting. Set the TFRST bit in the SCFCR2 to 1 and reset the SCFTDR2 before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCIF operation becomes unreliable if the clock is stopped.

Figure 16.5 is a sample flowchart for initializing the SCIF. The procedure for initializing the SCIF is:


Initialization
Clear TE and RE bits in SCSCR2 to 0
Set TFRST and RFRST bits in SCFCR2 to 1
Set CKE1 and CKE0 bits in SCSCR2 (leaving TE and RE bits cleared to 0)
Set data transfer format in SCSMR2
Set value in SCBRR2
Wait
1-bit interval elapsed?
No
Yes
Set RTRG1-0, TTRG1-0, and MCE in SCFCR2
Clear TFRST and RFRST bits to 0
Set TE and RE bits in SCSCR2 to 1,and set RIE, TIE, TEIE, and MPIE bits
End

1. Set the clock selection in SCSCR2.
   Be sure to clear bits RIE TIE, TE, and RE to 0.
   When clock output is selected, it is output immediately after SCSCR2 settings are made.

2. Set the data transfer format in SCSMR2.

3. Write a value corresponding to the SCBRR2. (Not necessary if an external clock is used.)

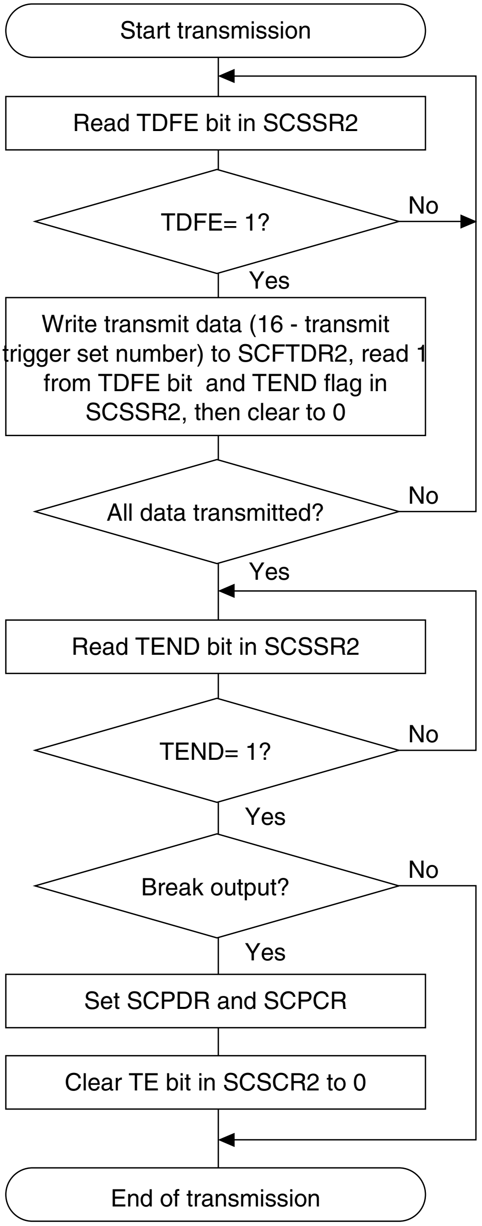
4. Wait at least one bit interval, then set the TE bit or RE bit in SCSR2 to 1. Also set the RIE and TIE bits.
   Setting the TE and RE bits enables the TxD2 and RxD2 pins to be used. When transmitting, the SCIF will go to the mark state; when receiving, it will go to the idle state, waiting for a start bit.

Figure 16.5 Sample SCIF Initialization Flowchart

- Serial data transmission

Figure 16.6 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.



1. SCIF status check and transmit data write: Read SCSSR2 and check that the TDFE flag is set to 1, then write transmit data to the SCFTDR2, read 1 from the TDFE and TEND flags, then clear these flags to 0. The number of transmit data bytes that can be written is 16 - (transmit trigger set number).

2. Serial transmission continuation procedure: To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR2, and then clear the TDFE flag to 0.

3. Break output at the end of serial transmission: To output a break in serial transmission, set the SCPDR and SCPCR, then clear the TE bit to 0 in the SCSCR2. For information on SCPDR and SCPCR, see sections 16.3.12 and 16.3.11.
   In steps 1 and 2, it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR2 indicated by the upper 8 bits of the SCFDR2.

Figure 16.6 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into the SCFTDR2, the SCIF transfers the data from SCFTDR2 to the transmit shift register (SCTSR2) and starts transmitting. Confirm that the TDFE flag in SCSSR2 is set to 1 before writing transmit data to SCFTDR2. The number of data bytes that can be written is (16 – transmit trigger setting).
2. When data is transferred from SCFTDR2 to SCTSR2 and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR2. When the number of transmit data bytes in SCFTDR2 falls below the transmit trigger number set in the SCFCR2, the TDFE flag is set. If the TIE bit in SCSCR2 is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TxD2 pin in the following order.

   a. Start bit: One-bit 0 is output.

   b. Transmit data: 8-bit or 7-bit data is output in LSB-first order.

   c. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)

   d. Stop bit(s): One- or two-bit 1s (stop bits) are output.

   e. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR2 transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR2 to SCTSR2, the stop bit is sent, and then serial transmission of the next frame is started.

   If there is no transmit data, the TEND flag in SCSSR2 is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output continuously.

Figure 16.7 shows an example of the operation for transmission.

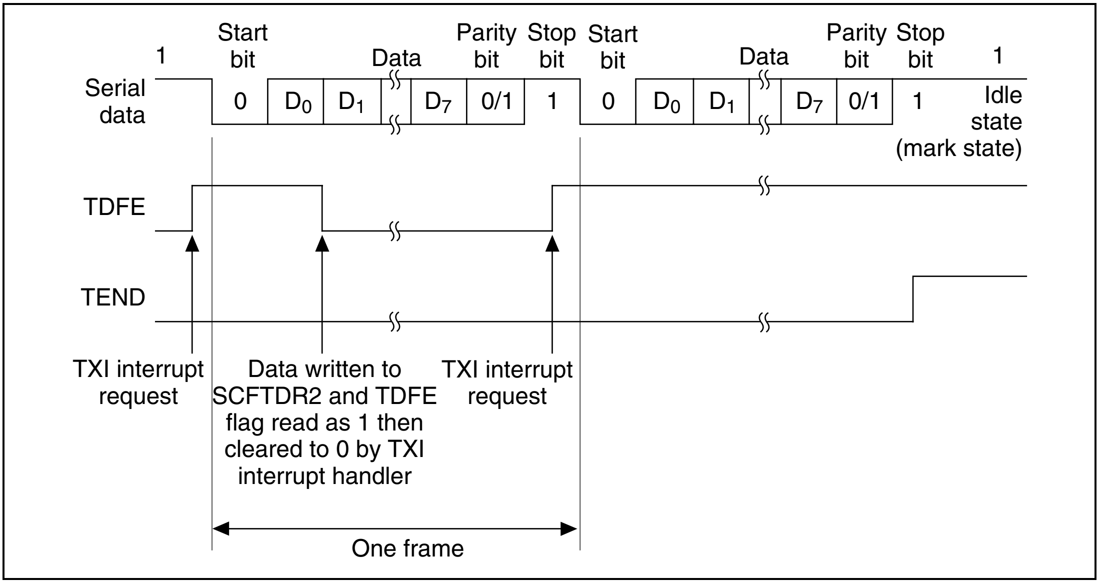


Figure 16.7 Example of Transmit Operation (Example with 8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the $\overline{\text{CTS2}}$ input value. When $\overline{\text{CTS2}}$ is set to 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When $\overline{\text{CTS2}}$ is set to 0, the next transmit data is output starting from the start bit.

   Figure 16.8 shows an example of the operation when modem control is used.

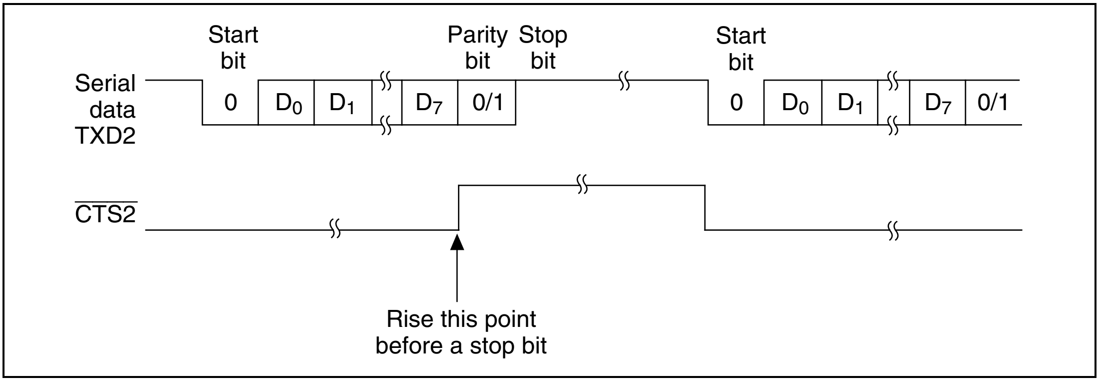
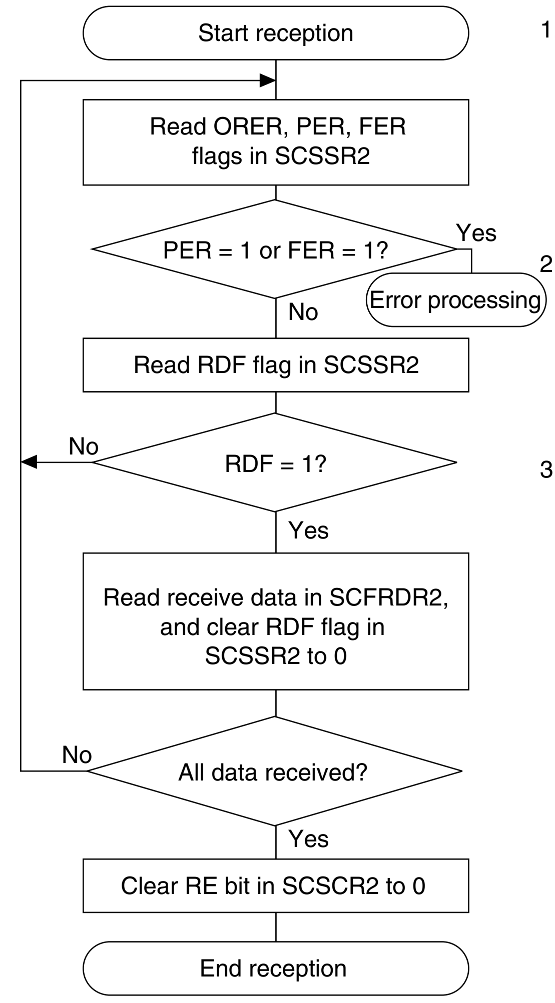


Figure 16.8 Example of Operation Using Modem Control ($\overline{\text{CTS2}}$)

- Serial data reception

Figure 16.9 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

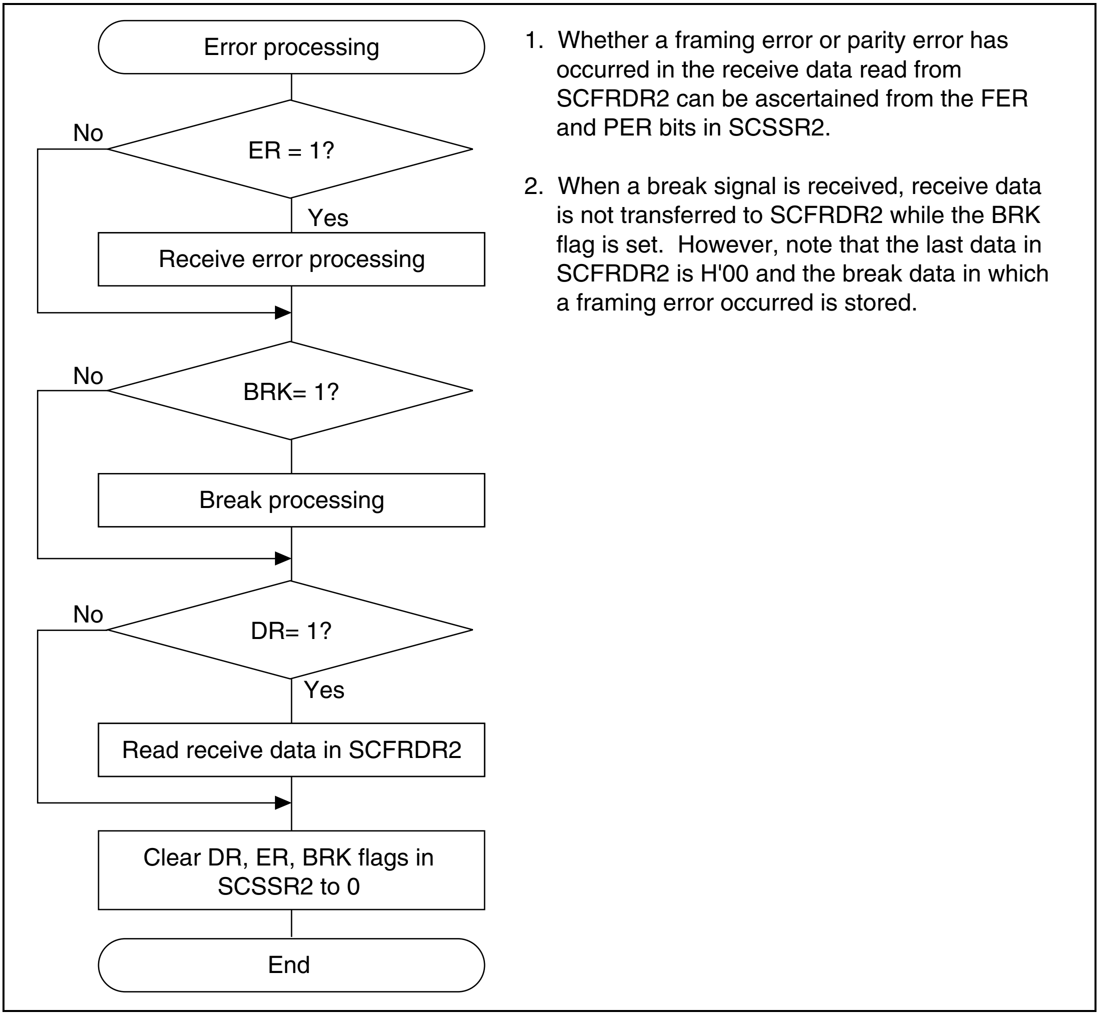


1. Receive error handling and break detection: Read the DR, ER, and BRK flags in SCSSR2 to identify any error, perform the appropriate error handling, then clear the DR, ER, and BRK flags to 0. In the case of a framing error, a break can also be detected by reading the value of the RxD2 pin.

2. SCIF status check and receive data read : Read the SCSSR2 and check that RDF = 1, then read the receive data in SCFRDR2, read 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can be identified by an RXI interrupt.

3. Serial reception continuation procedure: To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR2, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR2 can be ascertained by reading the lower bits of SCFDR2.

Figure 16.9 Sample Serial Reception Flowchart (1)



1. Whether a framing error or parity error has occurred in the receive data read from SCFRDR2 can be ascertained from the FER and PER bits in SCSSR2.

2. When a break signal is received, receive data is not transferred to SCFRDR2 while the BRK flag is set. However, note that the last data in SCFRDR2 is H'00 and the break data in which a framing error occurred is stored.

Figure 16.10 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR2 in LSB-to-MSB order.
3. The parity bit and stop bit are received.
   After receiving these bits, the SCIF carries out the following checks.
   a. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   b. The SCIF checks whether receive data can be transferred from the receive shift register (SCRSR2) to SCFRDR2.

c. Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.

If all the above checks are passed, the receive data is stored in SCFRDR2.

Note: Reception is not suspended when a receive error occurs.

4. If the RIE bit in SCSCR2 is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated.
   If the RIE bit in SCSCR2 is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated.

   If the RIE bit in SCSCR2 is set to 1 when the BRK flag changes to 1, a break reception interrupt (BRI) request is generated.

   Figure 16.11 shows an example of the operation for reception.

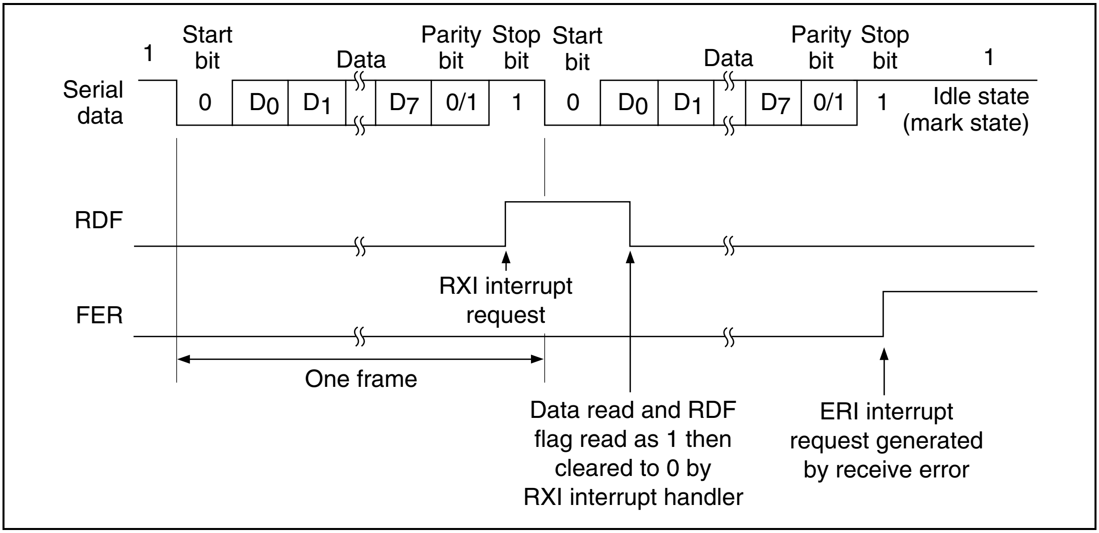


Figure 16.11 Example of SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{RTS2}}$ signal is output when SCFRDR2 is empty. When $\overline{\text{RTS2}}$ is 0, reception is possible. When $\overline{\text{RTS2}}$ is 1, this indicates that SCFRDR2 is full and reception is not possible.

   Figure 16.12 shows an example of the operation when modem control is used.

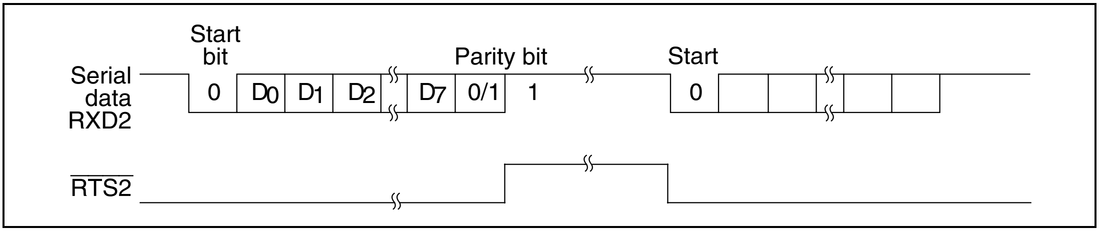


Figure 16.12 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

16.4.2 SCIF Interrupts

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive-data-full (RXI), and break (BRI).

Table 16.10 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE and RIE bits in SCSCR2. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When the TDFE flag in the SCSSR2 is set to 1, a TXI interrupt request is generated. The DMAC can be activated and data transfer performed when this interrupt is generated. The TDFE flag is automatically cleared to 0 when data is written to the SCFTDR2 by the DMAC.

When the RDF flag in SCSSR2 is set to 1, an RXI interrupt request is generated. The DMAC can be activated and data transfer performed when the RDF flag in SCSSR2 is set to 1. The RDF flag is automatically cleared to 0 when data is read from the SCFRDR2 by the DMAC.

When the ER flag in SCSSR2 is set to 1, an ERI interrupt request is generated.

When the BRK flag in SCSSR2 is set to 1, a BRI interrupt request is generated.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indicates that there is receive data in SCFRDR2.

Table 16.9 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error flag (ER)	Not possible	High
RXI	Interrupt initiated by receive data FIFO full flag (RDF) or data ready flag (DR)	Possible (RDF only)	↕
BRI	Interrupt initiated by break flag (BRK)	Not possible	
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	Low

See section 4, Exception Processing, for priorities and the relationship with non-SCIF interrupts.

16.5 Notes for Usage

Note the following when using the SCIF.

1. SCFTDR2 Writing and the TDFE Flag: The TDFE flag in SCSSR2 is set when the number of transmit data bytes written in the SCFTDR2 has fallen below the transmit trigger number set by bits TTRG1 and TTRG0 in the SCFCR2. After TDFE is set, transmit data up to the number of empty bytes in SCFTDR2 can be written, allowing efficient continuous transmission. However, if the number of data bytes written in SCFTDR2 is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to 0. TDFE clearing should therefore be carried out when SCFTDR2 contains more than the transmit trigger number of transmit data bytes.
   The number of transmit data bytes in SCFTDR2 can be found from the upper 8 bits of the SCFDR2.
2. SCFRDR2 Reading and the RDF Flag: The RDF flag in SCSSR2 is set when the number of receive data bytes in the SCFRDR2 has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in SCFCR2. After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR2, allowing efficient continuous reception. However, if the number of data bytes in SCFRDR2 is equal to or greater than the trigger number, the RDF flag will be set to 1 again if it is cleared to 0. RDF should therefore be cleared to 0 after being read as 1 after all the receive data has been read.
   The number of receive data bytes in SCFRDR2 can be found from the lower 8 bits of the FIFO data count register (SCFDR2).
3. Break Detection and Processing: Break signals can be detected by reading the RxD2 pin directly when a framing error (FER) is detected. In the break state the input from the RxD2 pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCFRDR2 is halted in the break state, the SCIF receiver continues to operate, so if the BRK flag is cleared to 0 it will be set to 1 again.
4. Sending a Break Signal: The I/O condition and level of the TxD2 pin are determined by the SCP4DT bit in SCPDR and bits SCP4MD0 and SCP4MD1 in the SCPCR. This feature can be used to send a break signal.
   To send a break signal during serial transmission, clear the CP4DT bit to 0 (designating low level), then set the SCP4MD0 and SCP4MD1 bits to 0 and 1, respectively, and finally clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TxD2 pin.
5. TEND Flag and TE Bit Processing: The TEND flag is set to 1 during transmission of the stop bit of the last data. Consequently, if the TE bit is cleared to 0 immediately after setting of the TEND flag has been confirmed, the stop bit will be in the process of transmission and will not be transmitted normally. Therefore, the TE bit should not be cleared to 0 for at least 0.5 serial

clock cycles (or 1.5 cycles if two stop bits are used) after setting of the TEND flag setting is confirmed.

6. Receive Data Sampling Timing and Receive Margin: The SCIF operates on a base clock with a frequency of 16 times the transfer rate. In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 16.13.

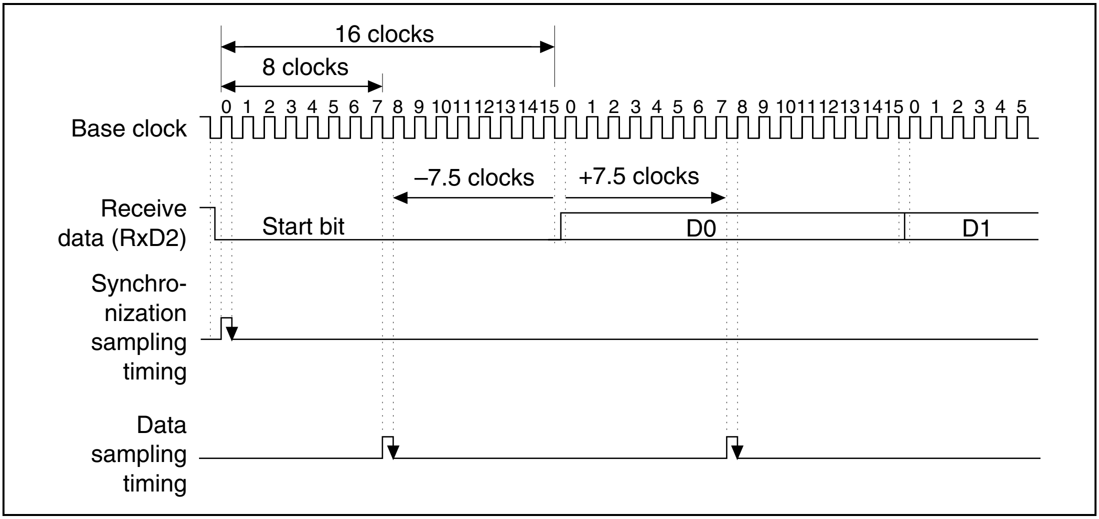


Figure 16.13 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}\,(1 + F) \right| \times 100\%$$

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 equation 2.

Equation 2:

When D = 0.5 and F = 0:

$$\begin{aligned} M &= (0.5 - 1/(2 \times 16)) \times 100\% \\ &= 46.875\% \end{aligned}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Section 17 Pin Function Controller

The pin function controller (PFC) is composed of registers for selecting the function of multiplexed pins and the direction of input/output. The pin function and I/O direction can be selected for each pin individually without regard to the operating mode of the LSI. Table 17.1 lists the multiplexed pins.

Table 17.1 List of Multiplexed Pins

Port	Port Function (Related Module)	Other Function (Related Module)
A	PTA7 I/O (port)	D23 I/O (data bus)
A	PTA6 I/O (port)	D22 I/O (data bus)
A	PTA5 I/O (port)	D21 I/O (data bus)
A	PTA4 I/O (port)	D20 I/O (data bus)
A	PTA3 I/O (port)	D19 I/O (data bus)
A	PTA2 I/O (port)	D18 I/O (data bus)
A	PTA1 I/O (port)	D17 I/O (data bus)
A	PTA0 I/O (port)	D16 I/O (data bus)
B	PTB7 I/O (port)	D31 I/O (data bus)
B	PTB6 I/O (port)	D30 I/O (data bus)
B	PTB5 I/O (port)	D29 I/O (data bus)
B	PTB4 I/O (port)	D28 I/O (data bus)
B	PTB3 I/O (port)	D27 I/O (data bus)
B	PTB2 I/O (port)	D26 I/O (data bus)
B	PTB1 I/O (port)	D25 I/O (data bus)
B	PTB0 I/O (port)	D24 I/O (data bus)

Port	Port Function (Related Module)	Other Function (Related Module)
C	PTC7 I/O (port)	$\overline{\text{CS6}}$ output (BSC) / $\overline{\text{CE1B}}$ output (BSC)
C	PTC6 I/O (port)	$\overline{\text{CS5}}$ output 9BSC) / $\overline{\text{CE1A}}$ output (BSC)
C	PTC5 I/O (port)	$\overline{\text{CS4}}$ output (BSC)
C	PTC4 I/O (port)	$\overline{\text{CS3}}$ output (BSC)
C	PTC3 I/O (port)	$\overline{\text{CS2}}$ output (BSC)
C	PTC2 I/O (port)	$\overline{\text{WE3}}$ output (BSC) / $\overline{\text{DQMUU}}$ output (BSC) / $\overline{\text{ICIOWR}}$ output (BSC)
C	PTC1 I/O (port)	$\overline{\text{WE2}}$ output (BSC) / $\overline{\text{DQMUL}}$ output (BSC) / $\overline{\text{ICIORD}}$ output (BSC)
C	PTC0 I/O (port)	$\overline{\text{BS}}$ output (BSC)
D	PTD7 I/O (port)	$\overline{\text{CE2B}}$ output (PCMCIA)
D	PTD6 I/O (port)	$\overline{\text{CE2A}}$ output (PCMCIA)
D	PTD5 I/O (port)	$\overline{\text{IOIS16}}$ input (PCMCIA)
D	PTD4 I/O (port)	CKE output (BSC)
D	PTD3 I/O (port)	$\overline{\text{CASU}}$ output (BSC)
D	PTD2 I/O (port)	$\overline{\text{CASL}}$ output (BSC)
D	PTD1 I/O (port)	$\overline{\text{RASU}}$ output (BSC)
D	PTD0 I/O (port)	$\overline{\text{RASL}}$ output (BSC)
E	PTE7 I/O (port)	$\overline{\text{IRQOUT}}$ output
E	PTE6 I/O (port)	TCLK I/O (Timer)
E	PTE5 I/O (port)	STATUS1 output (CPG)
E	PTE4 I/O (port)	STATUS0 output (CPG)
E	PTE3 I/O (port)	DRAK1 output (DMAC)
E	PTE2 I/O (port)	DRAK0 output (DMAC)
E	PTE1 I/O (port)	$\overline{\text{DACK1}}$ output (DMAC)
E	PTE0 I/O (port)	$\overline{\text{DACK0}}$ output (DMAC)

Port	Port Function (Related Module)	Other Function (Related Module)
F	PTF6 I/O (port)	$\overline{\text{ASEBRKAK}}$ output (AUD)
F	PTF5 I/O (port)	TDO output (H-UDI)
F	PTF4 I/O (port)	AUDSYNC output (AUD)
F	PTF3 I/O (port)	AUDATA[3] I/O (AUD)
F	PTF2 I/O (port)	AUDATA[2] I/O (AUD)
F	PTF1 I/O (port)	AUDATA[1] I/O (AUD)
F	PTF0 I/O (port)	AUDATA[0] I/O (AUD)
G	PTG5 input (port)	$\overline{\text{ADTRG}}$ input (ADC)
G	PTG4 input (port)	AUDCK input (AUD)
G	PTG3 input (port)	$\overline{\text{TRST}}$ input (AUD)/(H-UDI)
G	PTG2 input (port)	TMS input (H-UDI)
G	PTG1 input (port)	TCK input (H-UDI)
G	PTG0 input (port)	TDI input (H-UDI)
H	PTH6 I/O (port)	$\overline{\text{DREQ1}}$ input (DMAC)
H	PTH5 I/O (port)	$\overline{\text{DREQ0}}$ input (DMAC)
H	PTH4 I/O (port)	IRQ4 input (INTC)
H	PTH3 I/O (port)	IRQ3 input (INTC) / $\overline{\text{IRL3}}$ input (INTC)
H	PTH2 I/O (port)	IRQ2 input (INTC) / $\overline{\text{IRL2}}$ input (INTC)
H	PTH1 I/O (port)	IRQ1 input (INTC) / $\overline{\text{IRL1}}$ input (INTC)
H	PTH0 I/O (port)	IRQ0 input (INTC) / $\overline{\text{IRL0}}$ input (INTC)
J	PTJ3 I/O (port)	AN3 input (ADC)/ DA0 output (DAC)
J	PTJ2 I/O (port)	AN2 input (ADC)/ DA1 output (DAC)
J	PTJ1 I/O (port)	AN1 input (ADC)
J	PTJ0 I/O (port)	AN0 input (ADC)

Port	Port Function (Related Module)	Other Function (Related Module)
SCPT	SCPT5 input (port)	$\overline{\text{CTS2}}$ input (SCIF)/ IRQ5 input (INTC)
SCPT	SCPT4 I/O (port)	$\overline{\text{RTS2}}$ output (SCIF)
SCPT	SCPT3 I/O (port)	SCK2 I/O (SCIF)
SCPT	SCPT2 input (port)	RxD2 input (SCIF)
	SCPT2 output (port)	TxD2 output (SCIF)
SCPT	SCPT1 I/O (port)	SCK0 I/O (SCI)
SCPT	SCPT0 input (port)	RxD0 input (SCI)
	SCPT0 output (port)	TxD0 output (SCI)

Notes: SCPT0, and SCPT2 have the same data register to be accessed although they have different input pins and output pins.

17.1 Description of Registers

The pin function controller has the following registers. For the addresses and access size of these registers, see section 23, Control Register Table.

- Port A control register (PACR)
- Port B control register (PBCR)
- Port C control register (PCCR)
- Port D control register (PDCR)
- Port E control register (PECR)
- Port F control register (PFCR)
- Port G control register (PGCR)
- Port H control register (PHCR)
- Port J control register (PJCR)
- SC port control register (SCPCR)

17.1.1 Port A Control Register (PACR)

Port A Control Register (PACR) is a 16-bit read/write register that selects the pin functions and the input pullup MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PA7MD1	0	R/W	PA7 Mode
14	PA7MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
13	PA6MD1	0	R/W	PA6 Mode
12	PA6MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
11	PA5MD1	0	R/W	PA5 Mode
10	PA5MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
9	PA4MD1	0	R/W	PA4 Mode
8	PA4MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
7	PA3MD1	0	R/W	PA3 Mode
6	PA3MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

Bit	Bit Name	Initial Value	R/W	Description
5	PA2MD1	0	R/W	PA2 Mode
4	PA2MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
3	PA1MD1	0	R/W	PA1 Mode
2	PA1MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
1	PA0MD1	0	R/W	PA0 Mode
0	PA0MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)

17.1.2 Port B Control Register (PBCR)

Port B Control Register (PBCR) is a 16-bit read/write register that selects the pin functions and the input pullup MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PB7MD1	0	R/W	PB7 Mode
14	PB7MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
13	PB6MD1	0	R/W	PB6 Mode
12	PB6MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)

Bit	Bit Name	Initial Value	R/W	Description
11	PB5MD1	0	R/W	PB5 Mode
10	PB5MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
9	PB4MD1	0	R/W	PB4 Mode
8	PB4MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
7	PB3MD1	0	R/W	PB3 Mode
6	PB3MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
5	PB2MD1	0	R/W	PB2 Mode
4	PB2MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
3	PB1MD1	0	R/W	PB1 Mode
2	PB1MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
1	PB0MD1	0	R/W	PB0 Mode
0	PB0MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

17.1.3 Port C Control Register (PCCR)

Port C Control Register (PCCR) is a 16-bit read/write register that selects the pin functions and the input pullup MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PC7MD1	0	R/W	PC7 Mode
14	PC7MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
13	PC6MD1	0	R/W	PC6 Mode
12	PC6MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
11	PC5MD1	0	R/W	PC5 Mode
10	PC5MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
9	PC4MD1	0	R/W	PC4 Mode
8	PC4MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
7	PC3MD1	0	R/W	PC3 Mode
6	PC3MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
5	PC2MD1	0	R/W	PC2 Mode
4	PC2MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

Bit	Bit Name	Initial Value	R/W	Description
3	PC1MD1	0	R/W	PC1 Mode
2	PC1MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
1	PC0MD1	0	R/W	PC0 Mode
0	PC0MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

17.1.4 Port D Control Register (PDCR)

Port D Control Register (PDCR) is a 16-bit read/write register that selects the pin functions and the input pullup MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PD7MD1	0	R/W	PD7 Mode
14	PD7MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
13	PD6MD1	0	R/W	PD6 Mode
12	PD6MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
11	PD5MD1	0	R/W	PD5 Mode
10	PD5MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

Bit	Bit Name	Initial Value	R/W	Description
9	PD4MD1	0	R/W	PD4 Mode
8	PD4MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
7	PD3MD1	0	R/W	PD3 Mode
6	PD3MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
5	PD2MD1	0	R/W	PD2 Mode
4	PD2MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
3	PD1MD1	0	R/W	PD1 Mode
2	PD1MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
1	PD0MD1	0	R/W	PD0 Mode
0	PD0MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)

17.1.5 Port E Control Register (PECR)

Port E Control Register (PECR) is a 16-bit read/write register that selects the pin functions and the input pullup MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PE7MD1	0	R/W	PE7 Mode
14	PE7MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
13	PE6MD1	0	R/W	PE6 Mode
12	PE6MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
11	PE5MD1	0	R/W	PE5 Mode
10	PE5MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
9	PE4MD1	0	R/W	PE4 Mode
8	PE4MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
7	PE3MD1	0	R/W	PE3 Mode
6	PE3MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
5	PE2MD1	0	R/W	PE2 Mode
4	PE2MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

Bit	Bit Name	Initial Value	R/W	Description
3	PE1MD1	0	R/W	PE1 Mode
2	PE1MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
1	PE0MD1	0	R/W	PE0 Mode
0	PE0MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)

17.1.6 Port F Control Register (PFCR)

Port F Control Register (PFCR) is a 16-bit read/write register that selects the pin functions and the input pullup MOS control. PFCR is initialized to H'AAAA (in case of $\overline{\text{ASEMD0}}$ = 1) or H'0000 (in case of $\overline{\text{ASEMD0}}$ = 0) by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	1/0	R	Reserved
14	—	0	R	When $\overline{\text{ASEMDO}}$ = 0, these bits are always read as 0 and must only be written with 0. When $\overline{\text{ASEMDO}}$ = 1, these bits are always read as 1 and must only be written with 0.
13	PF6MD1	0	R/W	PF6 Mode
12	PF6MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
11	PF5MD1	0	R/W	PF5 Mode
10	PF5MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)

Bit	Bit Name	Initial Value	R/W	Description
9	PF4MD1	0	R/W	PF4 Mode
8	PF4MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
7	PF3MD1	0	R/W	PF3 Mode
6	PF3MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
5	PF2MD1	0	R/W	PF2 Mode
4	PF2MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
3	PF1MD1	0	R/W	PF1 Mode 1
2	PF1MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
1	PF0MD1	0	R/W	PF0 Mode 1
0	PF0MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

17.1.7 Port G Control Register (PGCR)

Port G Control Register (PGCR) is a 16-bit read/write register that selects the pin functions and the input pullup MOS control. PGCR is initialized to H'AAAA (in case of $\overline{\text{ASEMD0}}$ = 1) or H'A800 (in case of $\overline{\text{ASEMD0}}$ = 0) by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	1	R	Reserved
13	—	1	R	These bits are always read as 1. The write value should always be 0.
14	—	0	R	Reserved
12	—	0	R	These bits are always read as 0. The write value should always be 0.
11	PG5MD1	0	R/W	PG5 Mode
10	PG5MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
9	PG4MD1	0	R/W	PG4 Mode
8	PG4MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
7	PG3MD1	0	R/W	PG3 Mode
6	PG3MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
5	PG2MD1	0	R/W	PG2 Mode
4	PG2MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

Bit	Bit Name	Initial Value	R/W	Description
3	PG1MD1	0	R/W	PG1 Mode
2	PG1MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
1	PG0MD1	0	R/W	PG0 Mode
0	PG0MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

Note: The bit number are out of sequence.

17.1.8 Port H Control Register (PHCR)

Port H Control Register (PHCR) is a 16-bit read/write register that selects the pin functions and the input pullup MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved
14	—	0	R	These bits are always read as 0. The write value should always be 0.
13	PH6MD1	0	R/W	PH6 Mode
12	PH6MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
11	PH5MD1	0	R/W	PH5 Mode
10	PH5MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
9	PH4MD1	0	R/W	PH4 Mode
8	PH4MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

Bit	Bit Name	Initial Value	R/W	Description
7	PH3MD1	0	R/W	PH3 Mode
6	PH3MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
5	PH2MD1	0	R/W	PH2 Mode
4	PH2MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
3	PH1MD1	0	R/W	PH1 Mode
2	PH1MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)
1	PH0MD1	0	R/W	PH0 Mode
0	PH0MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pullup MOS: on)
				11: Port input (Pullup MOS: off)

17.1.9 Port J Control Register (PJCR)

Port J Control Register (PJCR) is a 16-bit read/write register that selects the pin functions.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PJ3MD1	0	R/W	PJ3 Mode
6	PJ3MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input 11: Port input
5	PJ2MD1	0	R/W	PJ2 Mode
4	PJ2MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input 11: Port input
3	PJ1MD1	0	R/W	PJ1 Mode
2	PJ1MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input 11: Port input
1	PJ0MD1	0	R/W	PJ0 Mode
0	PJ0MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input 11: Port input

17.1.10 SC Port Control Register (SCPCR)

SC Port Control Register (SCPCR) is a 16-bit read/write register that selects the pin functions and the input pullup MOS control. The setting of SCPCR is valid only when the transmit/receive operation is disabled in the setting of the SCSCR register. When the TE bit in SCSCR is set to 1, the other function output state has a higher priority than the SCPCR setting of the TxD2 or TxD0 pin. When the RE bit in SCSCR is set to 1, the input state has a higher priority than the SCPCR setting of the RxD2 or RxD0 pin.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	SCP5MD1	1	R/W	SCP5 Mode
10	SCP5MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
9	SCP4MD1	1	R/W	SCP4 Mode
8	SCP4MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
7	SCP3MD1	1	R/W	SCP3 Mode
6	SCP3MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)

Bit	Bit Name	Initial Value	R/W	Description
5	SCP2MD1	0	R/W	SCP2 Mode
4	SCP2MD0	0	R/W	00: Transmit data output 1 (TxD2) Receive data input 1 (RxD2) 01: General output (SCPT[2] output pin) Receive data input 1 (RxD2) 10: SCPT[2] input pin pullup (input pin) Transmit data output 1 (TxD2) 11: General input (SCPT[2] input pin) Transmit data output 1 (TxD2) Note: There is no combination of simultaneous I/O of SCPT[2] because one bit (SCP2DT) is accessed using two pins of TxD2 and RxD2. When the port input is set (bit SCPnMD1 is set to 1) and when the TE bit in SCSCR is set to 1, the TxD1 pin is in the output state. When the TE bit is cleared to 0, the TxD2 pin is in the high-impedance state.
3	SCP1MD1	1	R/W	SCP1 Mode
2	SCP1MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pullup MOS: on) 11: Port input (Pullup MOS: off)
1	SCP0MD1	0	R/W	SCP0 Mode 1 and 0
0	SCP0MD0	0	R/W	00: Transmit data output 0 (TxD0) Receive data input 0 (RxD0) 01: General output (SCPT[0] output pin) Receive data input 0 (RxD0) 10: SCPT[0] input pin pullup (input pin) Transmit data output 0 (TxD0) 11: General input (SCPT[0] input pin) Transmit data output 0 (TxD0) Note: There is no combination of simultaneous I/O of SCPT[0] because one bit (SCP0DT) is accessed using two pins of TxD0 and RxD0. When the port input is set (bit SCPnMD1 is set to 1) and when the TE bit in SCSCR is set to 1, the TxD0 pin is in the output state. When the TE bit is cleared to 0, the TxD0 pin is in the high-impedance state.

Section 18 I/O Ports

This LSI has 10 ports (ports A to J and SC). All port pins are multiplexed with other pin functions (Pin Function Controller (PFC) maintains the selection of the pin functions and pullup MOS control). Each port has a data register which stores the data to the pins.

18.1 Port A

Port A is an 8-bit I/O port with the pin configuration shown in figure 18.1. Each pin has an input pullup MOS, which is controlled by Port A Control Register (PACR) in PFC.

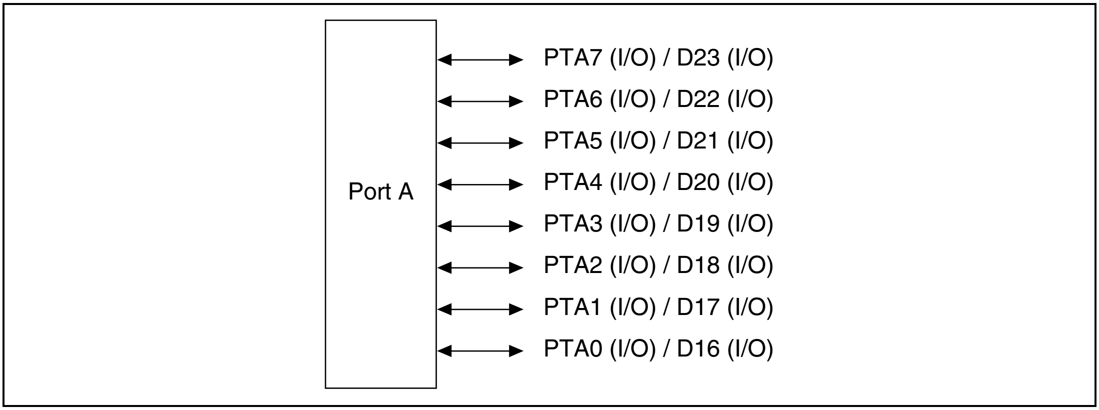


Figure 18.1 Port A

18.1.1 Description of Registers

Port A has the following register. For the addresses and access size of these registers, see section 23, Control Registers Table.

- Port A data register (PADR)

18.1.2 Port A Data Register (PADR)

Port A data Register (PADR) is an 8-bit read/write register that stores data for pins PTA7 to PTA0. PA7DT to PA0DT bit corresponds to PTA7 to PTA0 pin. When the pin function is general output port, if the port is read the value of the corresponding PADR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DT	0	R/W	Table 18.1 shows the function of PADR.
6	PA6DT	0	R/W	
5	PA5DT	0	R/W	
4	PA4DT	0	R/W	
3	PA3DT	0	R/W	
2	PA2DT	0	R/W	
1	PA1DT	0	R/W	
0	PA0DT	0	R/W	

Table 18.1 Read/Write Operation of the Port A Data Register (PADR)

PAnMD1	PAnMD0	Pin State	Read	Write
0	0	Other function	PADR value	Value is written to PADR, but does not affect pin state.
	1	Output	PADR value	Write value is output from pin.
1	0	Input (Pullup MOS on)	Pin state	Value is written to PADR, but does not affect pin state.
	1	Input (Pullup MOS off)	Pin state	Value is written to PADR, but does not affect pin state.

(n = 0 to 7)

18.2 Port B

Port B is an 8-bit I/O port with the pin configuration shown in figure 18.2. Each pin has an input pullup MOS, which is controlled by Port B Control Register (PBCR) in PFC.

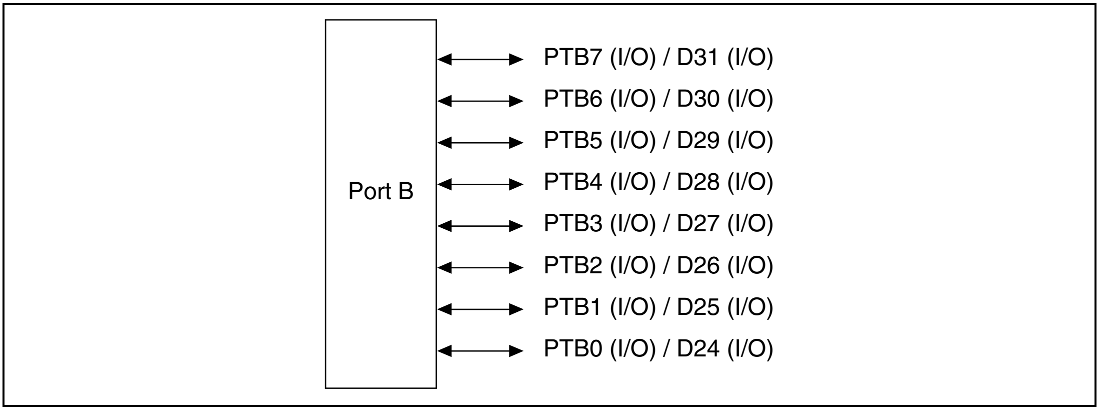


Figure 18.2 Port B

18.2.1 Description of Register

Port B has the following register. For the addresses and access size of these registers, see section 23, Control Registers Table.

- Port B data register (PBDR)

18.2.2 Port B Data Register (PBDR)

Port B data register (PBDR) is an 8-bit read/write register that stores data for pins PTB7 to PTB0. PB7DT to PB0DT bit corresponds to PTB7 to PTB0 pin. When the pin function is general output port, if the port is read the value of the corresponding PBDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DT	0	R/W	Table 18.2 shows the function of PBDR.
6	PB6DT	0	R/W	
5	PB5DT	0	R/W	
4	PB4DT	0	R/W	
3	PB3DT	0	R/W	
2	PB2DT	0	R/W	
1	PB1DT	0	R/W	
0	PB0DT	0	R/W	

Table 18.2 Read/Write Operation of the Port B Data Register (PBDR)

PBnMD1	PBnMD0	Pin State	Read	Write
0	0	Other function	PBDR value	Value is written to PBDR, but does not affect pin state.
	1	Output	PBDR value	Write value is output from pin.
1	0	Input (Pullup MOS on)	Pin state	Value is written to PBDR, but does not affect pin state.
	1	Input (Pullup MOS off)	Pin state	Value is written to PBDR, but does not affect pin state.

(n = 0 to 7)

18.3 Port C

Port C is an 8-bit I/O port with the pin configuration shown in figure 18.3. Each pin has an input pullup MOS, which is controlled by Port C Control Register (PCCR) in PFC.

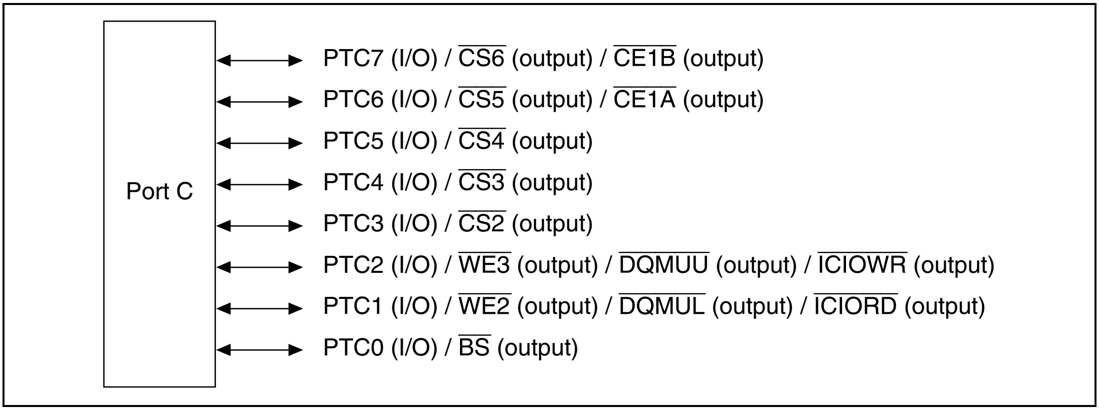

Port C
PTC7 (I/O) / $\overline{\text{CS6}}$ (output) / $\overline{\text{CE1B}}$ (output)
PTC6 (I/O) / $\overline{\text{CS5}}$ (output) / $\overline{\text{CE1A}}$ (output)
PTC5 (I/O) / $\overline{\text{CS4}}$ (output)
PTC4 (I/O) / $\overline{\text{CS3}}$ (output)
PTC3 (I/O) / $\overline{\text{CS2}}$ (output)
PTC2 (I/O) / $\overline{\text{WE3}}$ (output) / $\overline{\text{DQMUU}}$ (output) / $\overline{\text{ICIOWR}}$ (output)
PTC1 (I/O) / $\overline{\text{WE2}}$ (output) / $\overline{\text{DQMUL}}$ (output) / $\overline{\text{ICIORD}}$ (output)
PTC0 (I/O) / $\overline{\text{BS}}$ (output)

Figure 18.3 Port C

18.3.1 Description of Register

Port C has the following register. For the addresses and access size of these registers, see section 23, Control Registers Table.

- Port C data register (PCDR)

18.3.2 Port C Data Register (PCDR)

Port C data register (PCDR) is an 8-bit read/write register that stores data for pins PTC7 to PTC0. PC7DT to PC0DT bit corresponds to PTC7 to PTC0 pin. When the pin function is general output port, if the port is read, the value of the corresponding PCDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read. Table 18.3 shows the function of PCDR.

PCDR is initialized to H'00 by a power-on reset, after which the general input port function (pullup MOS on) is set as the initial pin function, and the corresponding pin levels are read. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DT	0	R/W	Table 18.3 shows the function of PCDR.
6	PC6DT	0	R/W	
5	PC5DT	0	R/W	
4	PC4DT	0	R/W	
3	PC3DT	0	R/W	
2	PC2DT	0	R/W	
1	PC1DT	0	R/W	
0	PC0DT	0	R/W	

Table 18.3 Read/Write Operation of the Port C Data Register (PCDR)

PCnMD1	PCnMD0	Pin State	Read	Write
0	0	Other function	PCDR value	Value is written to PCDR, but does not affect pin state.
	1	Output	PCDR value	Write value is output from pin.
1	0	Input (Pullup MOS: on)	Pin state	Value is written to PCDR, but does not affect pin state.
	1	Input (Pullup MOS: off)	Pin state	Value is written to PCDR, but does not affect pin state.

(n = 0 to 7)

18.4 Port D

Port D is an 8-bit I/O port with the pin configuration shown in figure 18.4. Each pin has an input pullup MOS, which is controlled by Port D Control Register (PDCR) in PFC.

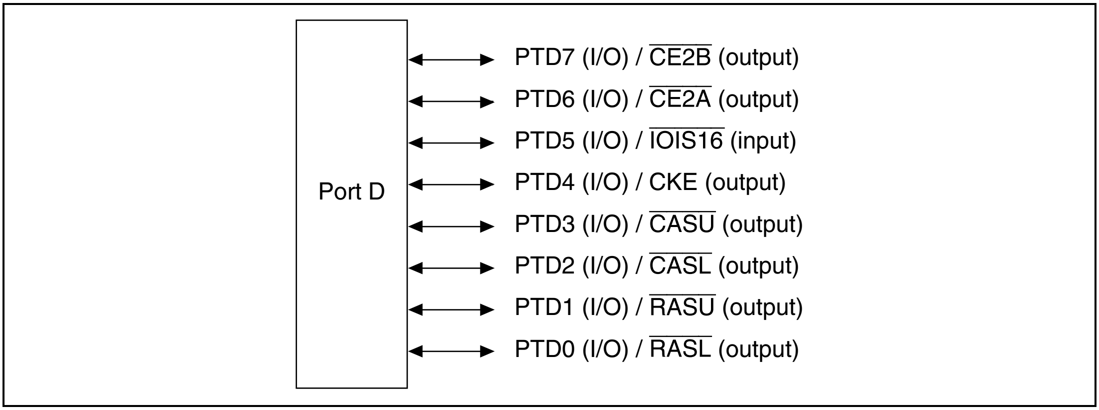

Port D
PTD7 (I/O) / $\overline{\text{CE2B}}$ (output)
PTD6 (I/O) / $\overline{\text{CE2A}}$ (output)
PTD5 (I/O) / $\overline{\text{IOIS16}}$ (input)
PTD4 (I/O) / CKE (output)
PTD3 (I/O) / $\overline{\text{CASU}}$ (output)
PTD2 (I/O) / $\overline{\text{CASL}}$ (output)
PTD1 (I/O) / $\overline{\text{RASU}}$ (output)
PTD0 (I/O) / $\overline{\text{RASL}}$ (output)

Figure 18.4 Port D

18.4.1 Description of Register

Port D has the following register. For the addresses and access size of these registers, see section 23, Control Registers Table.

- Port D data register (PDDR)

18.4.2 Port D Data Register (PDDR)

Port D data register (PDDR) is an 8-bit read/write register that stores data for pins PTD7 to PTD0. PD7DT to PD0DT bit corresponds to PTD7 to PTD0 pin. When the pin function is general output port, if the port is read, the value of the corresponding PDDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

PDDR is initialized to H'00 by a power-on reset, after which the general input port function (pullup MOS on) is set as the initial pin function, and the corresponding pin levels are read. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DT	0	R/W	Table 18.4 shows the function of PDDR.
6	PD6DT	0	R/W	
5	PD5DT	0	R/W	
4	PD4DT	0	R/W	
3	PD3DT	0	R/W	
2	PD2DT	0	R/W	
1	PD1DT	0	R/W	
0	PD0DT	0	R/W	

Table 18.4 Read/Write Operation of the Port D Data Register (PDDR)

PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function	PDDR value	Value is written to PDDR, but does not affect pin state.
	1	Output	PDDR value	Write value is output from pin.
1	0	Input (Pullup MOS: on)	Pin state	Value is written to PDDR, but does not affect pin state.
	1	Input (Pullup MOS: off)	Pin state	Value is written to PDDR, but does not affect pin state.

(n = 0 to 7)

18.5 Port E

Port E is an 8-bit I/O port with the pin configuration shown in figure 18.5. Each pin has an input pullup MOS, which is controlled by Port E Control Register (PECR) in PFC.

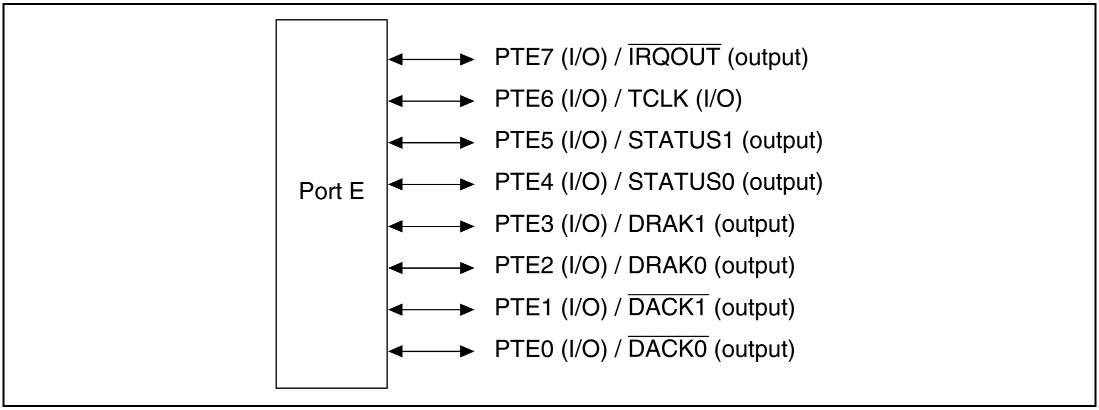

Port E
PTE7 (I/O) / IRQOUT (output)
PTE6 (I/O) / TCLK (I/O)
PTE5 (I/O) / STATUS1 (output)
PTE4 (I/O) / STATUS0 (output)
PTE3 (I/O) / DRAK1 (output)
PTE2 (I/O) / DRAK0 (output)
PTE1 (I/O) / DACK1 (output)
PTE0 (I/O) / DACK0 (output)

Figure 18.5 Port E

18.5.1 Description of Register

Port E has the following register. For the addresses and access size of these registers, see section 23, Control Registers Table.

- Port E data register (PEDR)

18.5.2 Port E Data Register (PEDR)

Port E data register (PEDR) is an 8-bit read/write register that stores data for pins PTE7 to PTE0. PE7DT to PE0DT bit corresponds to PTE7 to PTE0 pin. When the pin function is general output port, if the port is read the value of the corresponding PEDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read.

PEDR is initialized to H'00 by a power-on reset, after which the general input port function (pullup MOS on) is set as the initial pin function, and the corresponding pin levels are read. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DT	0	R/W	Table 18.5 shows the function of PEDR.
6	PE6DT	0	R/W	
5	PE5DT	0	R/W	
4	PE4DT	0	R/W	
3	PE3DT	0	R/W	
2	PE2DT	0	R/W	
1	PE1DT	0	R/W	
0	PE0DT	0	R/W	

Table 18.5 Read/Write Operation of the Port E Data Register (PEDR)

PEnMD1	PEnMD0	Pin State	Read	Write
0	0	Other function	PEDR value	Value is written to PEDR, but does not affect pin state.
	1	Output	PEDR value	Write value is output from pin.
1	0	Input (Pullup MOS: on)	Pin state	Value is written to PEDR, but does not affect pin state.
	1	Input (Pullup MOS: off)	Pin state	Value is written to PEDR, but does not affect pin state.

(n = 0 to 7)

18.6 Port F

Port F is a 7-bit input/output port with the pin configuration shown in figure 18.6. Each pin has an input pullup MOS, which is controlled by Port F Control Register (PFCR) in PFC.

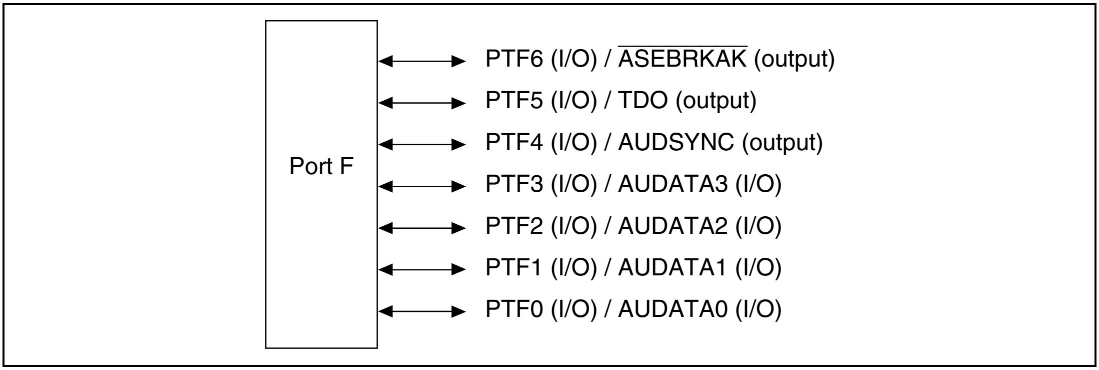

Port F
PTF6 (I/O) / $\overline{\text{ASEBRKAK}}$ (output)
PTF5 (I/O) / TDO (output)
PTF4 (I/O) / AUDSYNC (output)
PTF3 (I/O) / AUDATA3 (I/O)
PTF2 (I/O) / AUDATA2 (I/O)
PTF1 (I/O) / AUDATA1 (I/O)
PTF0 (I/O) / AUDATA0 (I/O)

Figure 18.6 Port F

18.6.1 Description of Register

Port F has the following register. For the addresses and access size of these registers, see section 23, Control Registers Table.

- Port F data register (PFDR)

18.6.2 Port F Data Register (PFDR)

Port F data register (PFDR) is an 8-bit register composed of a 1-bit readable register and a 7-bit readable/writable register. This register stores data for pins PTF6 to PTF0. PF6DT to PF0DT bit corresponds to PTF6 to PTF0 pin. When the function is general input port, if the port is read the corresponding pin level is read.

PFDR is initialized by a power-on reset, after which the general input port function (pullup MOS on) is set as the initial pin function, and the corresponding pin levels are read. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	PF6DT	0	R/W	Table 18.6 shows the function of PFDR.
5	PF5DT	0	R/W	
4	PF4DT	0	R/W	
3	PF3DT	0	R/W	
2	PF2DT	0	R/W	
1	PF1DT	0	R/W	
0	PF0DT	0	R/W	

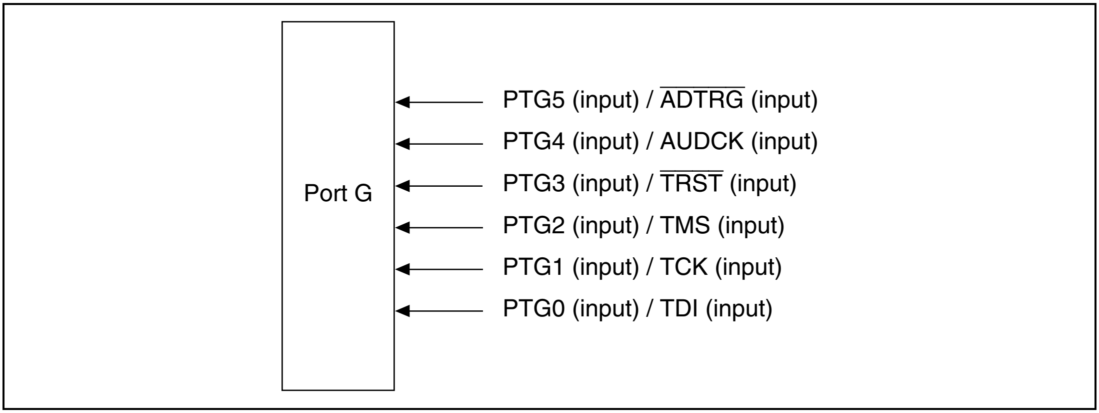
Table 18.6 Read/Write Operation of the Port F Data Register (PFDR)

PFnMD1	PFnMD0	Pin State	Read	Write
0	0	Other functions	PFDR value	Can be written to PFDR but does not affect the pin state.
	1	Output	PFDR value	A value to be written is output from the pin.
1	0	Input (Pullup MOS: on)	Pin state	Can be written to PFDR but does not affect the pin state.
	1	Input (Pullup MOS: off)	Pin state	Can be written to PFDR but does not affect the pin state.

(n = 0 to 6)

18.7 Port G

Port G is a 6-bit input port with the pin configuration shown in figure 18.7. Each pin has an input pullup MOS, which is controlled by Port G Control Register (PGCR) in PFC.



Port G
PTG5 (input) / $\overline{\text{ADTRG}}$ (input)
PTG4 (input) / AUDCK (input)
PTG3 (input) / $\overline{\text{TRST}}$ (input)
PTG2 (input) / TMS (input)
PTG1 (input) / TCK (input)
PTG0 (input) / TDI (input)

Figure 18.7 Port G

18.7.1 Description of Register

Port G has the following register. For the addresses and access size of these registers, see section 23, Control Registers Table.

- Port G data register (PGDR)

18.7.2 Port G Data Register (PGDR)

Bit	Bit Name	Initial Value	R/W	Description
7	—	*	R	Reserved
6	—	*	R	
5	PG5DT	*	R	Table 18.7 shows the function of PGDR.
4	PG4DT	*	R	
3	PG3DT	*	R	
2	PG2DT	*	R	
1	PG1DT	*	R	
0	PG0DT	*	R	

Note: * Undefined

Table 18.7 Read/Write Operation of the Port G Data Register (PGDR)

PGnMD1	PGnMD0	Pin State	Read	Write
0	0	Other function	Low level	Ignored (no affect on pin state)
	1	Reserved	—	Ignored (no affect on pin state)
1	0	Input (Pullup MOS: on)	Pin state	Ignored (no affect on pin state)
	1	Input (Pullup MOS: off)	Pin state	Ignored (no affect on pin state)

(n = 0 to 5)

18.8 Port H

Port H is a 7-bit I/O and port with the pin configuration shown in figure 18.8. Each pin has an input pullup MOS, which is controlled by Port H Control Register (PHCR) in PFC.

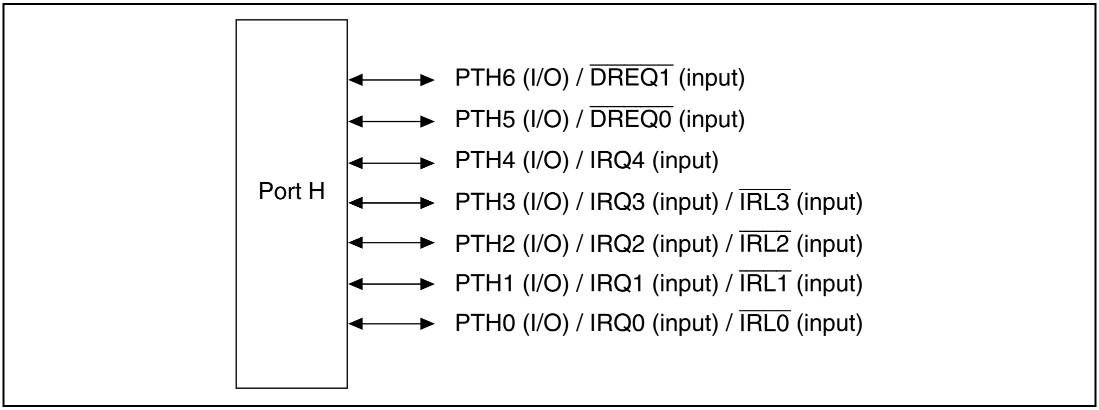

Port H
PTH6 (I/O) / $\overline{\text{DREQ1}}$ (input)
PTH5 (I/O) / $\overline{\text{DREQ0}}$ (input)
PTH4 (I/O) / IRQ4 (input)
PTH3 (I/O) / IRQ3 (input) / $\overline{\text{IRL3}}$ (input)
PTH2 (I/O) / IRQ2 (input) / $\overline{\text{IRL2}}$ (input)
PTH1 (I/O) / IRQ1 (input) / $\overline{\text{IRL1}}$ (input)
PTH0 (I/O) / IRQ0 (input) / $\overline{\text{IRL0}}$ (input)

Figure 18.8 Port H

18.8.1 Description of Register

Port H has the following register. For the addresses and access size of these registers, see section 23, Control Registers Table.

- Port H data register (PHDR)

18.8.2 Port H Data Register (PHDR)

Port H data register (PHDR) is a 7-bit read/write and 1-bit read register that stores data for pins PTH6 to PTH0. PH6DT to PH0DT bit corresponds to PTH6 to PTH0 pin. When the pin function is general output port, if the port is read, the value of the corresponding PHDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

PHDR is initialized to H'00 by a power-on reset, after which the general input port function (pullup MOS on) is set as the initial pin function, and the corresponding pin levels are read. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Note that the low level is read if bits 6 to 0 are read except in general-purpose input.

Bit	Bit Name	Initial Value	R/W	Description
7	—	*	R	Reserved
6	PH6DT	0	R/W	Table 18.8 shows the function of PHDR.
5	PH5DT	0	R/W	
4	PH4DT	0	R/W	
3	PH3DT	0	R/W	
2	PH2DT	0	R/W	
1	PH1DT	0	R/W	
0	PH0DT	0	R/W	

Note: * Undefined

Table 18.8 Read/Write Operation of the Port H Data Register (PHDR)

PHnMD1	PHnMD0	Pin State	Read	Write
0	0	Other function	PHDR value	Value is written to PHDR, but does not affect pin state.
	1	Output	PHDR value	Write value is output from pin.
1	0	Input (Pullup MOS: on)	Pin state	Value is written to PHDR, but does not affect pin state.
	1	Input (Pullup MOS: off)	Pin state	Value is written to PHDR, but does not affect pin state.

(n = 0 to 6)

18.9 Port J

Port J is a 4-bit input port with the pin configuration shown in figure 18.9. Each pin has an input pullup MOS, which is controlled by Port J Control Register (PJCR) in PFC.

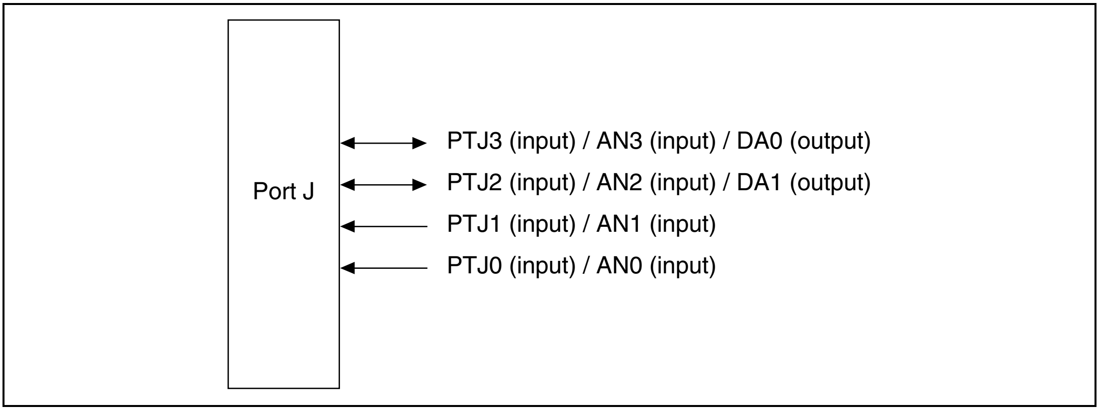

Port J
PTJ3 (input) / AN3 (input) / DA0 (output)
PTJ2 (input) / AN2 (input) / DA1 (output)
PTJ1 (input) / AN1 (input)
PTJ0 (input) / AN0 (input)

Figure 18.9 Port J

18.9.1 Description of Register

Port J has the following register. For the addresses and access size of these registers, see section 23, Control Registers Table.

- Port J data register (PJDR)

18.9.2 Port J Data Register (PJDR)

Port J data register (PJDR) is an 8-bit read register that stores data for pins PTJ7 to PTJ0. PJ3DT to PJ0DT bit corresponds to PTJ3 to PTJ0 pin. When the pin function is general output port, if the port is read the value of the corresponding PJDR bit is returned directly. When the function is 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
6	—	0	R	
5	—	0	R	
4	—	0	R	
3	PJ3DT	0	R	Table 18.9 shows the function of PJDR.
2	PJ2DT	0	R	
1	PJ1DT	0	R	
0	PJ0DT	0	R	

Table 18.9 Read/Write Operation of the Port J Data Register (PJDR)

PJnMD1	PJnMD0	Pin State	Read	Write
0	0	Other function	Low level	Ignored (no affect on pin state)
	1	Reserved (Setting prohibited)		Ignored (no affect on pin state)
1	0	Input	Pin state	Ignored (no affect on pin state)
	1	Input	Pin state	Ignored (no affect on pin state)

(n = 0 to 3)

18.10 SC Port

SC port is a 3-bit I/O, 2-bit output and 4-bit input port with the pin configuration shown in figure 18.10. Each pin has an input pullup MOS, which is controlled by SC port Control Register (SCPCR) in PFC.

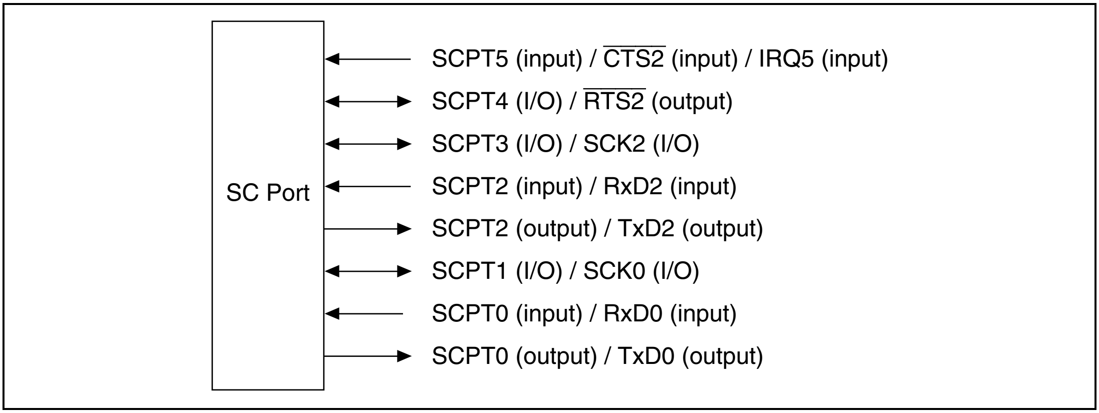

SC Port
SCPT5 (input) / $\overline{\text{CTS2}}$ (input) / IRQ5 (input)
SCPT4 (I/O) / $\overline{\text{RTS2}}$ (output)
SCPT3 (I/O) / SCK2 (I/O)
SCPT2 (input) / RxD2 (input)
SCPT2 (output) / TxD2 (output)
SCPT1 (I/O) / SCK0 (I/O)
SCPT0 (input) / RxD0 (input)
SCPT0 (output) / TxD0 (output)

Figure 18.10 SC Port

18.10.1 Description of Register

Port SC has the following register. For the addresses and access size of these registers, see section 23, Control Registers Table.

- SC Port data register (SCPDR)

18.10.2 Port SC Data Register (SCPDR)

Port SC data register (SCPDR) is a 5-bit read/write and 3-bit read register that stores data for pins SCPT5 to SCPT0. SCP5DT to SCP0DT bit corresponds to SCPT5 to SCPT0 pin. When the pin function is general output port, if the port is read, the value of the corresponding SCPDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

SCPDR is initialized to B'***00000 by a power-on reset. After initialization, the general input port function (pullup MOS on) is set as the initial pin function, and the corresponding pin levels are read from bits SCP5DT—SCP3DT and SCP1DT. SCPDR retains its previous value in standby mode and sleep mode, and in a manual reset.

Note that the low level is read if bit 7 is read except in general-purpose input.

When reading the state of the RxD2 and RxD0 pins of the SCP2DT and SCP0DT bits in SCPDR without clearing the TE or RE bit in SCSCR to 0, set the RE bit in SCSCR to 1. When the RE bit is set to 1, the RxD pin is for input and the pin state can be read before the setting of SCPCR.

Bit	Bit Name	Initial Value	R/W	Description
7	—	*	R	Reserved
6	—	*	R	
5	SCP5DT	*	R	Table 18.10 shows the function of SCPDR .
4	SCP4DT	0	R/W	
3	SCP3DT	0	R/W	
2	SCP2DT	0	R/W	
1	SCP1DT	0	R/W	
0	SCP0DT	0	R/W	

Note: * Undefined

Table 18.10 Read/Write Operation of the SC Port Data Register (SCPDR)

- For SCP4DT to SCP0DT

SCPnMD1	SCPnMD0	Pin State	Read	Write
0	0	Other function	SCPDR value	Value is written to SCPDR, but does not affect pin state.
	1	Output	SCPDR value	Write value is output from pin.
1	0	Input (Pullup MOS: on)	Pin state	Value is written to SCPDR, but does not affect pin state.
	1	Input (Pullup MOS: off)	Pin state	Value is written to SCPDR, but does not affect pin state.

(n = 0 to 4)

- For SCP5DT

SCPnMD1	SCPnMD0	Pin State	Read	Write
0	0	Other function	Low level	Ignored (no affect on pin state)
	1	Reserved (Setting prohibited)	—	Ignored (no affect on pin state)
1	0	Input (Pullup MOS: on)	Pin state	Ignored (no affect on pin state)
	1	Input (Pullup MOS: off)	Pin state	Ignored (no affect on pin state)

(n = 5)

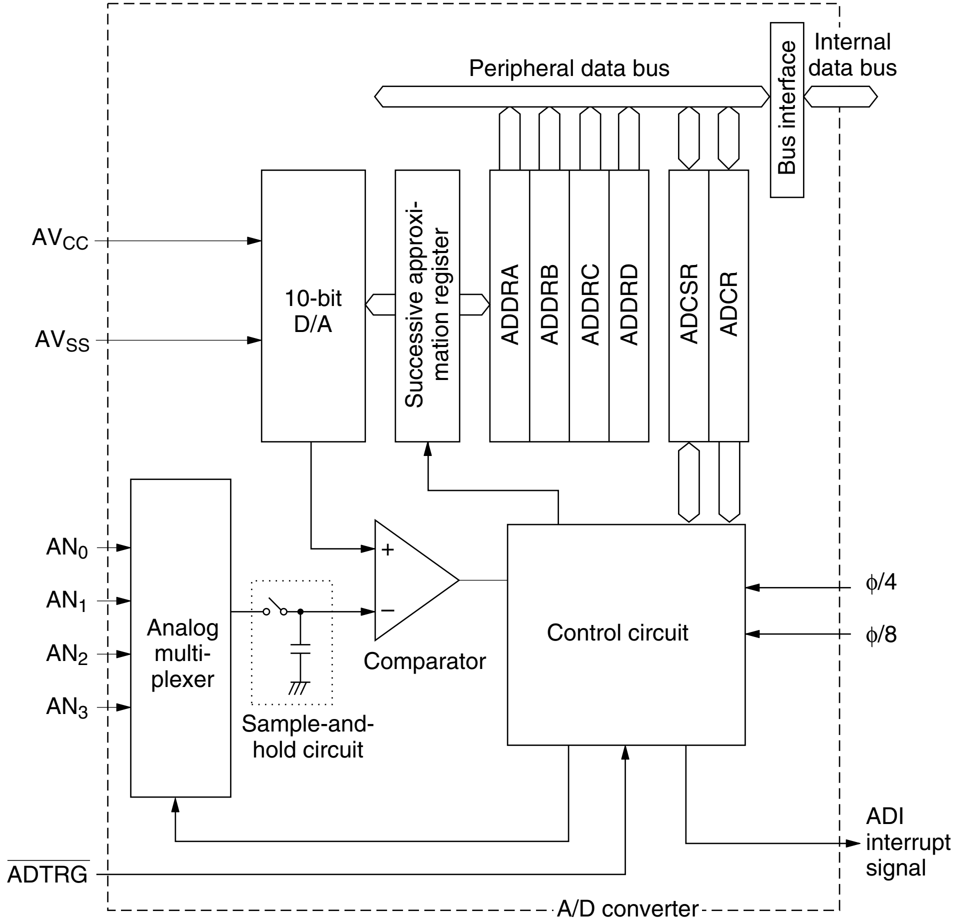
Section 19 A/D Converter (ADC)

This LSI includes a 10-bit successive-approximation A/D converter with a selection of up to four analog input channels. Figure 19.1 shows the block diagram of the A/D converter.

19.1 Features

A/D converter features are listed below.

- 10-bit resolution
- 4 input channels
- High-speed conversion
  - — Conversion time: maximum 8.1 µs per channel (with Pø = 33-MHz peripheral clock)
- Three conversion modes
  - — Single mode: A/D conversion of 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 data registers corresponding to the channels.
- Sample-and-hold function
- A/D conversion can be externally triggered
- A/D interrupt requested at the end of conversion
  - — At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.



Legend:
ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Figure 19.1 A/D Converter Block Diagram

19.2 I/O Pins

Table 19.1 summarizes the A/D converter's input pins. AV_{CC} and AV_{SS} are the power supply for the analog circuits in the A/D converter. AVcc also functions as the A/D converter reference voltage.

Table 19.1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power-supply pin	AVcc	Input	Analog power supply
Analog ground pin	AVss	Input	Analog ground and reference voltage
Analog input pin 0	AN0	Input	Group 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

19.3 Description of Registers

The A/D converter has the following registers. For the addresses and access size of these registers, see section 23, Control Register Table.

- A/D data register A (ADDRAH)

  The upper and lower bytes of ADDRA may be represented by ADDRAH and ADDRAL, respectively.
- A/D data register B(ADDRBH)

  The upper and lower bytes of ADDRB may be represented by ADDRBH and ADDRBL, respectively.
- A/D data register C (ADDRCH)

  The upper and lower bytes of ADDRC may be represented by ADDRCH and ADDRCL, respectively.
- A/D data register D (ADDRDH)

  The upper and lower bytes of ADDRD may be represented by ADDRDH and ADDRDL, respectively.
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

19.3.1 A/D Data Registers A–D (ADDRA–ADDRD)

The four A/D data registers (ADDRA–ADDRD) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The upper 8 bits of the result are stored in the upper byte (bits 15 to 8) of the A/D data register. The lower 2 bits are stored in the lower byte (bits 7 and 6). Bits 5 to 0 of an A/D data register are reserved bits that always read 0. For the reading of the data, see section 19.4, Bus Master Interface, and section 19.5, Access size and Access size of A/D Data Register. Table 19.2 indicates the pairings of analog input channels and A/D data registers.

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	AD9 to AD0	0	R	Bit data (10 bits)
5 to 0	—	0	R	Reserved These bits are always read as 0.

Table 19.2 Analog Input Channels and A/D Data Registers

Analog Input Channel	
Group 0	**A/D Data Register**
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD

19.3.2 A/D Control/Status Register (ADCSR)

ADCSR is an 8-bit read/write register that selects the mode and controls the A/D converter.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag Indicates the end of A/D conversion. 0: [Clear condition] (1) Cleared by reading ADF while ADF = 1, then writing 0 in ADF (2) Cleared when DMAC is activated by ADI interrupt and ADDR is read 1: [Set conditions] 1. Single mode: A/D conversion ends 2. Multi mode: A/D conversion ends in all selected channels 3. Scan mode: A/D conversion ends in all selected channels.
6	ADIE	0	R/W	A/D Interrupt Enable Enables or disables the interrupt (ADI) requested at the end of A/D conversion. Set the ADIE when convertion is stopped. 0: A/D end interrupt request (ADI) is disabled 1: A/D end interrupt request (ADI) is enabled
5	ADST	0	R/W	A/D Start Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin. 0: A/D conversion is stopped 1: 1. Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends. 3. Multi mode: A/D conversion stauts: ADST is automatically cleard to 0 when conversion ends in all selected channels. 3. Scan mode: A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode.

Bit	Bit Name	Initial Value	R/W	Description
4	MULTI	0	R/W	Multi Mode Selects single mode or multi mode or scan mode. For further information on operation in these modes, see section 19.6, Description of Operation. The mode is selected by the combination of this bit (MULTI) and bit 5 (SCN) of ADCR. MULTI SCAN 0 0 : Single mode 0 1 : Single mode 1 0 : Multi mode 1 1 : Single mode
3	CKS	0	R/W	Clock Select Selects the A/D conversion time. Clear the ADST bit to 0 before switching the conversion time. 0: Conversion time = 536 states (maximum) 1: Conversion time = 266 states (maximum)
2 1 0	CH2 CH1 CH0	0 0 0	R/W R/W R/W	Channel Select These bits and the MULTI bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection. Single Mode (MULTI = 0) — Multi Mode and Scan Mode (MULTI = 1) 000: AN0 — AN0 001: AN1 — AN0, AN1 010: AN2 — AN0 to AN2 011: AN3 — AN0 to AN3

Note: * Only can be written to clear the flag.

19.3.3 A/D Control Register (ADCR)

ADCR is an 8-bit read/write register that enables or disables external triggering of A/D conversion. ADCR is initialized to H'07 by a reset and in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGE1	0	R/W	Trigger Enable
6	TRGE0	0	R/W	Enables or disables external triggering of A/D conversion. 00: When an external trigger is input, the A/D conversion does not start 01: The same as above 10: The same as above 11: The A/D conversion starts at the falling edge of an input signal from the external trigger pin ($\overline{\text{ADTRG}}$).
5	SCN	0	R/W	Scan Mode Selects multi mode or scan mode when the MULTI bit is set to 1. See the description of bit 4 in section 19.3.2.
4	—	0	R/W	Reserved
3	—	0	R/W	These bits are always read as 0. The write value should always be 0.
2	—	1	R	Reserved
1	—	1	R	These bits are always read as 1. The write value should always be 0.
0	—	1	R	

19.4 Bus Master Interface

ADDRA to ADDRD are 16-bit registers, but they are connected to the bus master by the upper 8 bits of the 16-bit peripheral data bus. Therefore, although the upper byte can be accessed directly by the bus master, the lower byte is read through an 8-bit temporary register (TEMP).

An A/D data register is read as follows. When the upper byte is read, the upper-byte value is transferred directly to the bus master and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the bus master.

When reading an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, the read value is not guaranteed.

Figure 19.2 shows the data flow for access to an A/D data register.

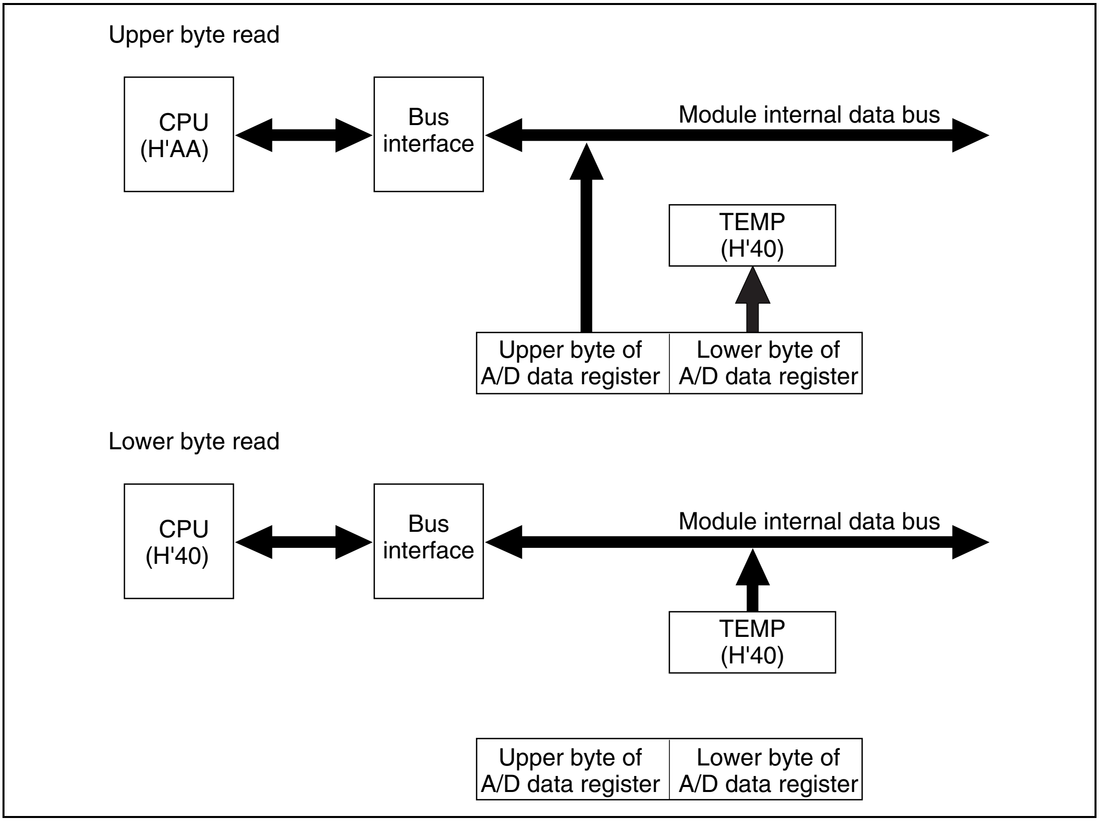


Figure 19.2 A/D Data Register Access Operation (Reading H'AA40)

19.5 Access Size of the A/D Data Register

19.5.1 Word Access

When A/D data registers (ADDRA to ADDRD) are read in word, A/D data register values are read from bits 15 to 8, and invalid data is read from bits 7 to 0.

Figure 19.3 shows an example of reading ADDRAH.

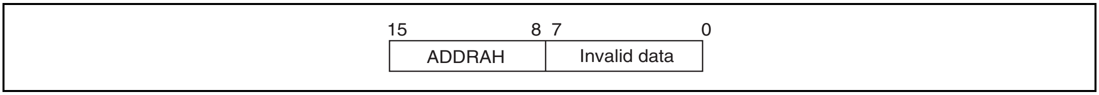
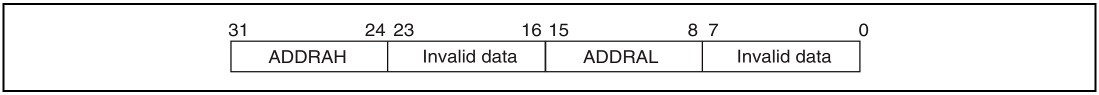


Figure 19.3 Word Access Example

19.5.2 Longword Access

When A/D data registers are read in longword, the upper byte of the A/D data register is read from bits 31 to 24, invalid data from bits 23 to 16, the lower byte of the A/D data register from bits 15 to 8, and invalid data from bits 7 to 0.

Figure 19.4 shows an example of reading ADDRAH.



31 24	23 16	15 8	7 0
ADDRAH	Invalid data	ADDRAL	Invalid data

Figure 19.4 Longword Access Example

19.6 Description of Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

19.6.1 Single Mode (MULTI = 0)

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts when the ADST bit in ADCSR is set to 1 by software, or by external trigger input. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends.

When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in ADF.

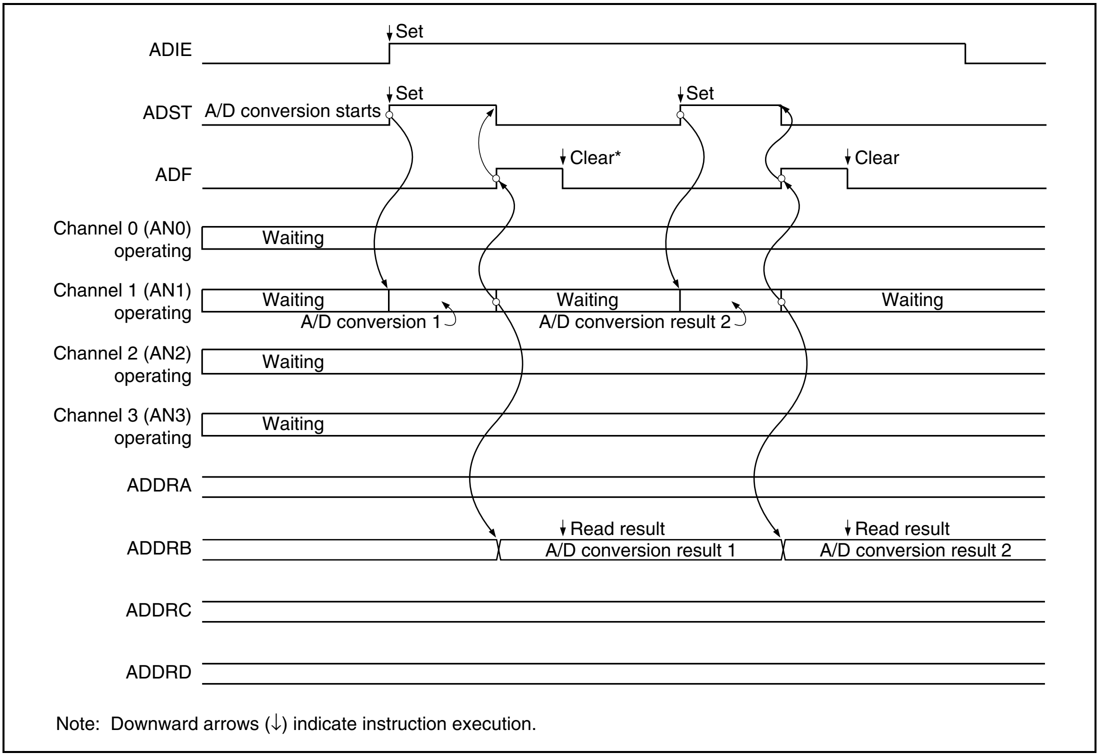
When the mode or analog input channel must be switched during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described next.

Figure 19.5 shows a timing diagram for this example.

1. Single mode is selected (MULTI = 0), input channel AN1 is selected (CH2 = CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt processing routine starts.

5. The routine reads ADCSR, then writes 0 in the ADF flag.
6. The routine reads and processes the conversion result (ADDRB = 0).
7. Execution of the A/D interrupt processing routine ends. Then, when the ADST bit is set to 1, A/D conversion starts to execute 2 to 7 above.



Note: Downward arrows (↓) indicate instruction execution.

Figure 19.5 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

19.6.2 Multi Mode (MULTI = 1, SCN = 0)

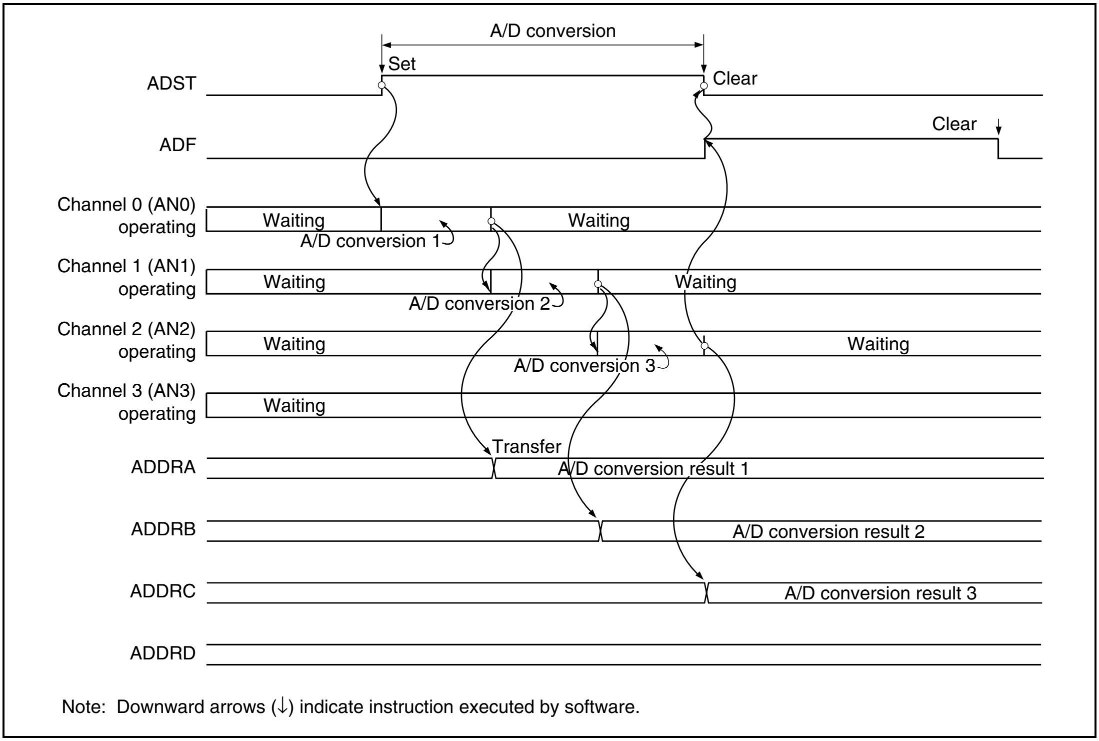
Multi mode should be selected when performing multi channel A/D conversions on one or more channels. When the ADST bit in ADCSR is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH2 = 0). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1) starts immediately. When A/D conversions end on the selected channels, the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in group 0 (AN0 to AN2) are selected in scan mode are described next. Figure 19.6 shows a timing diagram for this example.

1. Multi mode is selected (MULTI = 1, SCN = 0), channel group 0 is selected (CH2 = 0), analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN0) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion of all selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and ADST bit is cleared to 0. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.

When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).



Note: Downward arrows (↓) indicate instruction executed by software.

Figure 19.6 Example of A/D Converter Operation (Multi Mode, Channels AN0 to AN2 Selected)

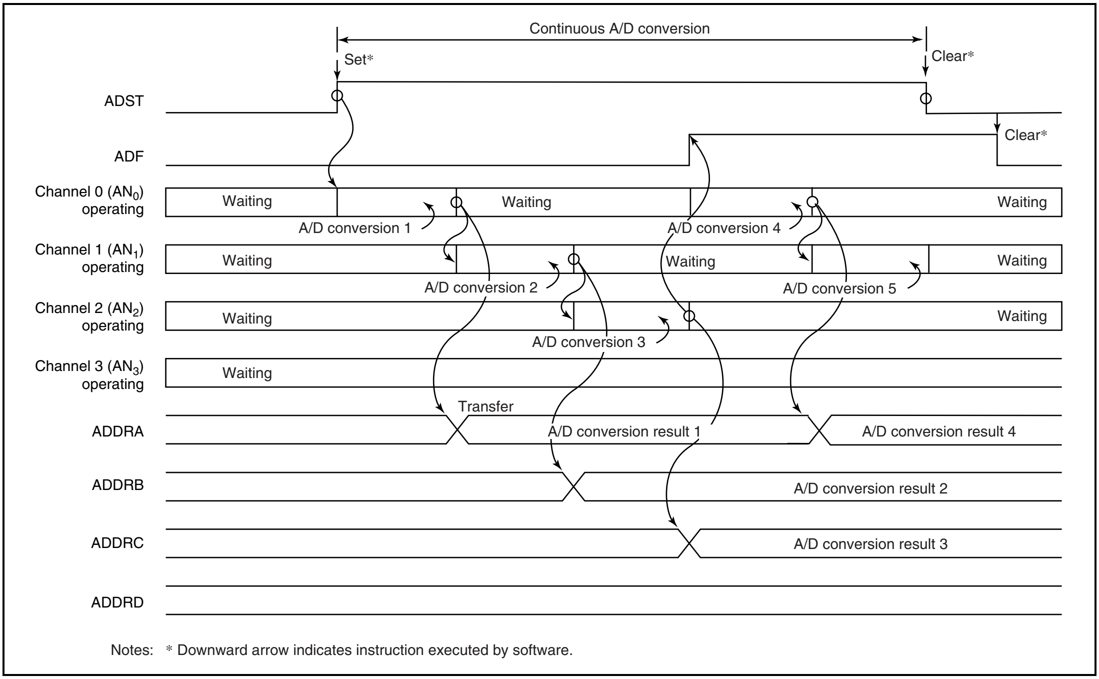
19.6.3 Scan Mode (MULTI = 1, SCN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit in the ADCSR is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH2 = 0). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the ADDRA to ADDRD corresponding to the channels.

When the mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described next. Figure 19.7 shows a timing diagram for this example.

1. Scan mode is selected (MULTI = 1, SCN = 1), channel group 2 is selected (CH2 = 0), analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN0) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion of all the selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).



Notes: * Downward arrow indicates instruction executed by software.

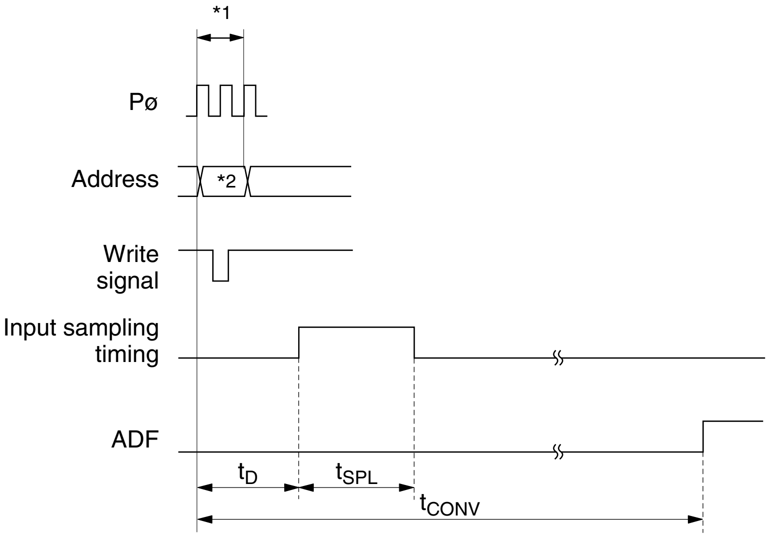
Figure 19.7 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

19.6.4 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit in ADCSR is set to 1, then starts conversion. Figure 19.5 shows the A/D conversion timing. Table 19.3 indicates the A/D conversion time.

As indicated in figure 19.8, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 19.3.

In multi mode and scan mode, the values given in table 19.3 apply to the first conversion. In the second and subsequent conversions the conversion time is fixed at 536 states when CKS = 0 or 266 states when CKS = 1.



t_D A/D conversion start delay

t_{SPL} Input sampling time

t_{CONV} A/D conversion time

Notes: *1 ADCSR write cycle

*2 ADCSR address

Figure 19.8 A/D Conversion Timing

Table 19.3 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1		
	Symbol	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay	t_D	17	—	28	10	—	17
Input sampling time	t_{SPL}	—	129	—	—	65	—
A/D conversion time	t_{CONV}	514	—	525	259	—	266

Note: Values in the table are numbers of states (t_{cyc}).

19.6.5 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE1, TRGE0 bits in ADCR are set to 1. external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A high-to-low transition at the $\overline{\text{ADTRG}}$ pin sets the ADST bit in ADCSR to 1, starting A/D conversion. Other operations, regardless of the conversion mode, are the same as if the ADST bit had been set to 1 by software. Figure 19.9 shows the timing.

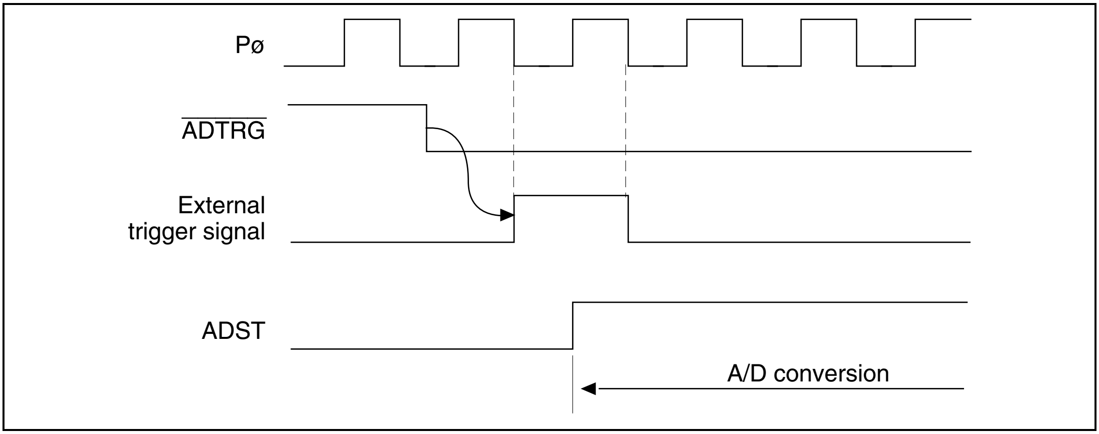


Figure 19.9 External Trigger Input Timing

19.7 Interrupt Requests

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

19.8 Definitions of A/D Conversion Accuracy

The A/D converter compares an analog value input from an analog input channel to its analog reference value and converts it into 10-bit digital data. The absolute accuracy of this A/D conversion is the deviation between the input analog value and the output digital value. It includes the following errors:

- Offset error
- Full-scale error
- Quantization error
- Nonlinearity error

These four error quantities are explained below using figure 19.10. In the figure, the 10 bits of the A/D converter have been simplified to 3 bits.

Offset error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) 0000000000 (000 in the figure) to 000000001 (001 in the figure) (figure 19.10, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the 1111111110 (110 in the figure) to the maximum 1111111111 (111 in the figure) (figure 19.10, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 19.10, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 19.10, item (4)). Note that it does not include offset, full-scale or quantization error.

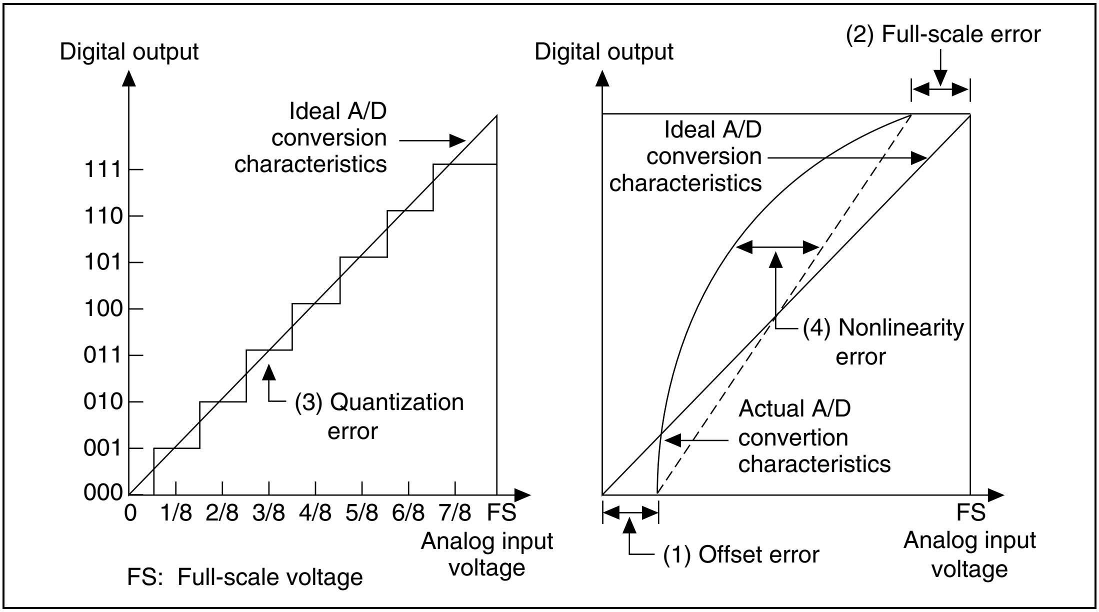


Figure 19.10 Definitions of A/D Conversion Accuracy

19.9 Notes for Usage

When using the A/D converter, note the points listed in section 19.7.1 below.

19.9.1 Setting Analog Input Voltage

- Analog Input Voltage Range: During A/D conversion, the voltages input to the analog input pins ANn should be in the range $AV_{SS} \leq ANn \leq AV_{CC}$ (n = 0–3).
- Relationships of AV_{CC} and AV_{SS} to V_{CC} and V_{SS}: AV_{CC}, AV_{SS}, V_{CC} and V_{SS} should be related as follows: $AV_{CC} = V_{CC} \pm 0.2$ V and $AV_{SS} = V_{SS}$.

19.9.2 Processing of Analog Input Pins

To prevent damage from voltage surges at the analog input pins (AN0–AN3), connect an input protection circuit like the one shown in figure 19.11. The circuit shown also includes an RC filter to suppress noise. This circuit is shown as an example; The circuit constants should be selected according to actual application conditions. Table 19.4 lists the analog input pin specifications and figure 19.12 shows an equivalent circuit diagram of the analog input ports.

19.9.3 Access Size and Read Data

Table 19.5 shows the relationship between access size and read data. Note the read data obtained with different access sizes, bus widths, and endian modes.

The case is shown here in which H'3FF is obtained when AV_{CC} is input as an analog input. FF is the data containing the upper 8 bits of the conversion result, and C0 is the data containing the lower 2 bits.

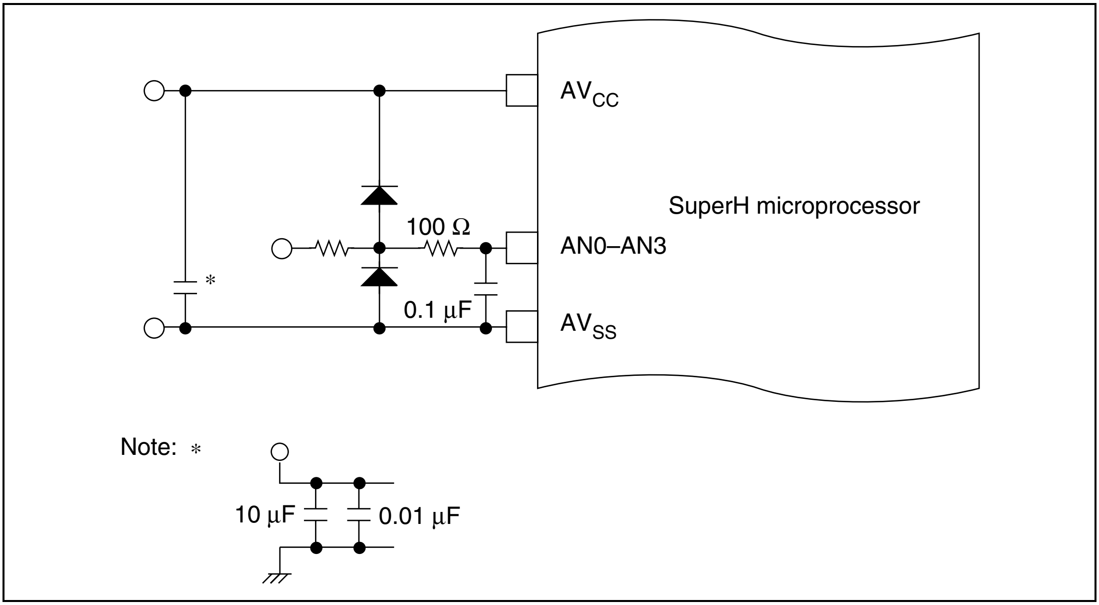


Figure 19.11 Example of Analog Input Protection Circuit

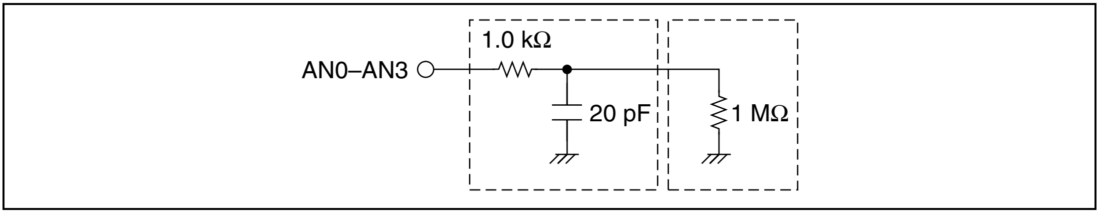


Figure 19.12 Analog Input Pin Equivalent Circuit

Table 19.4 Analog Input Pin Ratings

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	5	kΩ

Table 19.5 Relationship between Access Size and Read Data

Access		Bus Width	32 Bits (D31—D0)		16 Bits (D15—D0)		8 Bits (D7—D0)	
Size	**Command**	**Endian**	**Big**	**Little**	**Big**	**Little**	**Big**	**Little**
Byte access	`MOV.L`	`#ADDRAH,R9`						
	`MOV.B`	`@R9,R8`	FFFFFFFF	FFFFFFFF	FFFF	FFFF	FF	FF
	`MOV.L`	`#ADDRAL,R9`						
	`MOV.B`	`@R9,R8`	C0C0C0C0	C0C0C0C0	C0C0	C0C0	C0	C0
Word access	`MOV.L`	`#ADDRAH,R9`						
	`MOV.W`	`@R9,R8`	FFxxFFxx	FFxxFFxx	FFxx	FFxx	FFxx	xxFF
	`MOV.L`	`#ADDRAL,R9`						
	`MOV.W`	`@R9,R8`	C0xxC0xx	C0xxC0xx	C0xx	C0xx	C0xx	xxC0
Longword access	`MOV.L`	`#ADDRAH,R9`						
	`MOV.L`	`@R9,R8`	FFxxC0xx	FFxxC0xx	FFxxC0xx	C0xxFFxx	FFxxC0xx	xxC0xxFF

In this table:

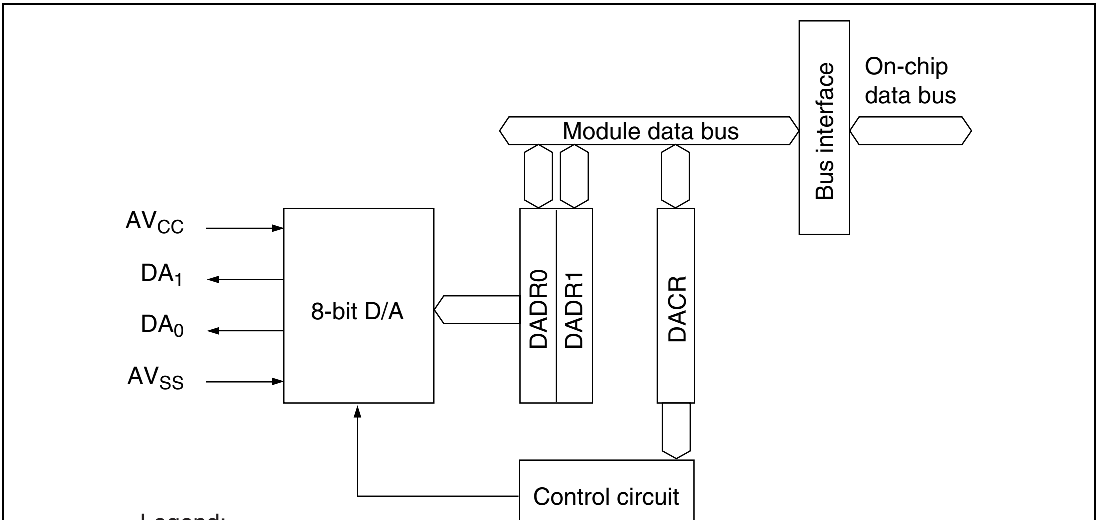
```
#ADDRAH    .EQU   H'A4000080
#ADDRAL    .EQU   H'A4000082
```

Values are shown in hexadecimal for the case where read data is output to an external device via R8.

Section 20 D/A Converter (DAC)

This LSI includes a D/A converter with two channels.

Figure 20.1 shows a block diagram of the D/A converter.



Legend:

DACR: D/A control register
DADR0: D/A data register 0
DADR1: D/A data register 1

Figure 20.1 D/A Converter Block Diagram

20.1 Features

D/A converter features are listed below.

- Eight-bit resolution
- Two output channels
- Conversion time: maximum 10 µs (with 20-pF capacitive load)
- Output voltage: 0 V to AVcc

20.2 I/O Pins

Table 20.1 summarizes the D/A converter's input and output pins.

Table 20.1 D/A Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power-supply pin	AVcc	Input	Analog power supply
Analog ground pin	AVss	Input	Analog ground and reference voltage
Analog output pin 0	DA0	Output	Analog output, channel 0
Analog output pin 1	DA1	Output	Analog output, channel 1

20.3 Description of Registers

The D/A converter has the following registers. For the addresses and access size of these registers, see section 23, Control Register Table.

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A control register (DACR)

20.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1)

The D/A data registers (DADR0 and DADR1) are 8-bit read/write registers that store the data to be converted. When analog output is enabled, the D/A data register values are constantly converted and output at the analog output pins.

The D/A data registers are initialized to H'00 by a reset.

20.3.2 D/A Control Register (DACR)

DACR is an 8-bit read/write register that controls the operation of the D/A converter.

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output. 0: DA1 analog output is disabled 1: Channel-1 D/A conversion and DA1 analog output are enabled
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output. 0: DA0 analog output is disabled 1: Channel-0 D/A conversion and DA0 analog output are enabled
5	DAE	0	R/W	D/A Enable Controls D/A conversion, together with bits DAOE0 and DAOE1. When the DAE bit is cleared to 0, D/A conversion is controlled independently in channels 0 and 1. When this LSI enters standby mode while D/A conversion is enabled, the D/A output is held and the analog power-supply current is equivalent to that during D/A conversion. To reduce the analog power-supply current in standby mode, clear the DAOE0 and DAOE1 bits and disable the D/A output. 00—: D/A conversion is disabled in channels 0 and 1 010: D/A conversion is enabled in channel 0 D/A conversion is disabled in channel 1 011: D/A conversion is enabled in channels 0 and 1 100: D/A conversion is disabled in channel 0 D/A conversion is enabled in channel 1 101: D/A conversion is enabled in channels 0 and 1 11—: D/A conversion is enabled in channels 0 and 1 When the DAE bit is set to 1, even if bits DAOE0 and DAOE1 in DACR and the ADST bit in ADCSR are cleared to 0, the same current is drawn from the analog power supply as during A/D and D/A conversion.
4 to 0	—	1	R	Reserved These bits are always read as 1. The write value should always be 1.

20.4 Description of Operation

The D/A converter has two built-in D/A conversion circuits that can perform conversion independently.

D/A conversion is performed constantly while enabled in DACR. If the DADR0 or DADR1 value is modified, conversion of the new data begins immediately. The conversion results are output when bits DAOE0 and DAOE1 are set to 1.

An example of D/A conversion on channel 0 is given next. Timing is indicated in figure 20.2.

1. Data to be converted is written in DADR0.
2. Bit DAOE0 is set to 1 in DACR. D/A conversion starts and DA0 becomes an output pin. The converted result is output after the conversion time. The output value is (DADR0 contents/256) x AVcc. Output of this conversion result continues until the value in DADR0 is modified or the DAOE0 bit is cleared to 0.
3. If the DADR0 value is modified, conversion starts immediately, and the result is output after the conversion time.
4. When the DAOE0 bit is cleared to 0, DA0 becomes an input pin.

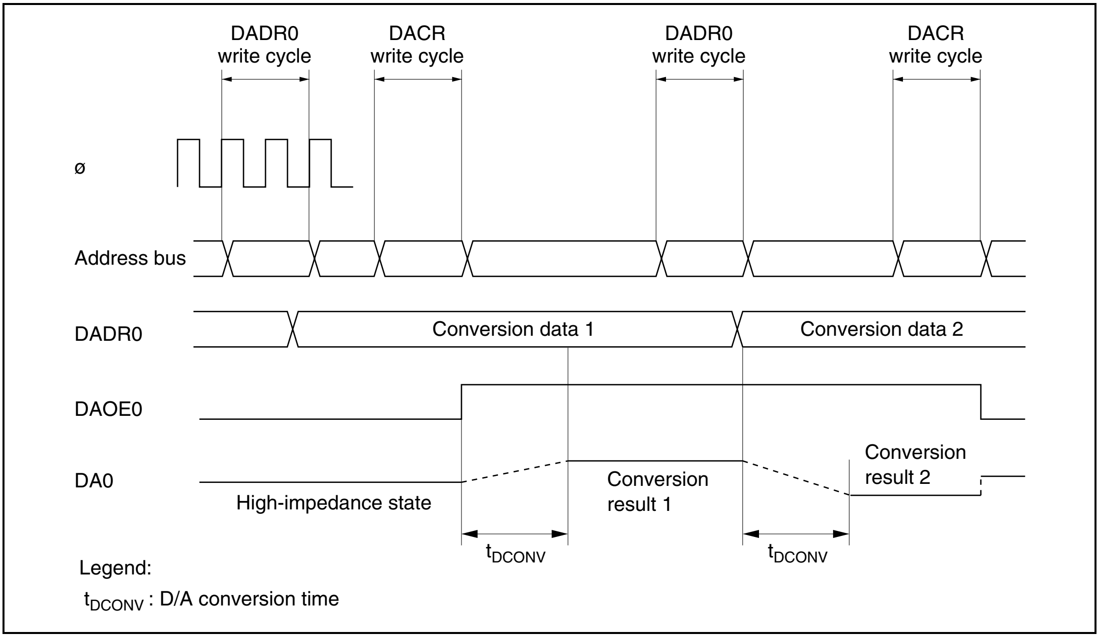


Figure 20.2 Example of D/A Converter Operation

Section 21 Hitachi User-Debugging Interface (H-UDI)

The H-UDI (Hitachi user-debugging interface) performs on-chip debugging which is supported by the SH7706. The H-UDI described here is a serial interface which is pi-compatible with JTAG (Joint Test Action Group, IEEE Standard 1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture) specifications.

The H-UDI in the SH7706 supports a boundary scan mode, and is also used for emulator connection.

When using an emulator, H-UDI functions should not be used. Refer to the emulator manual for the method of connecting the emulator.

Figure 21.1 shows the block diagram of the H-UDI.

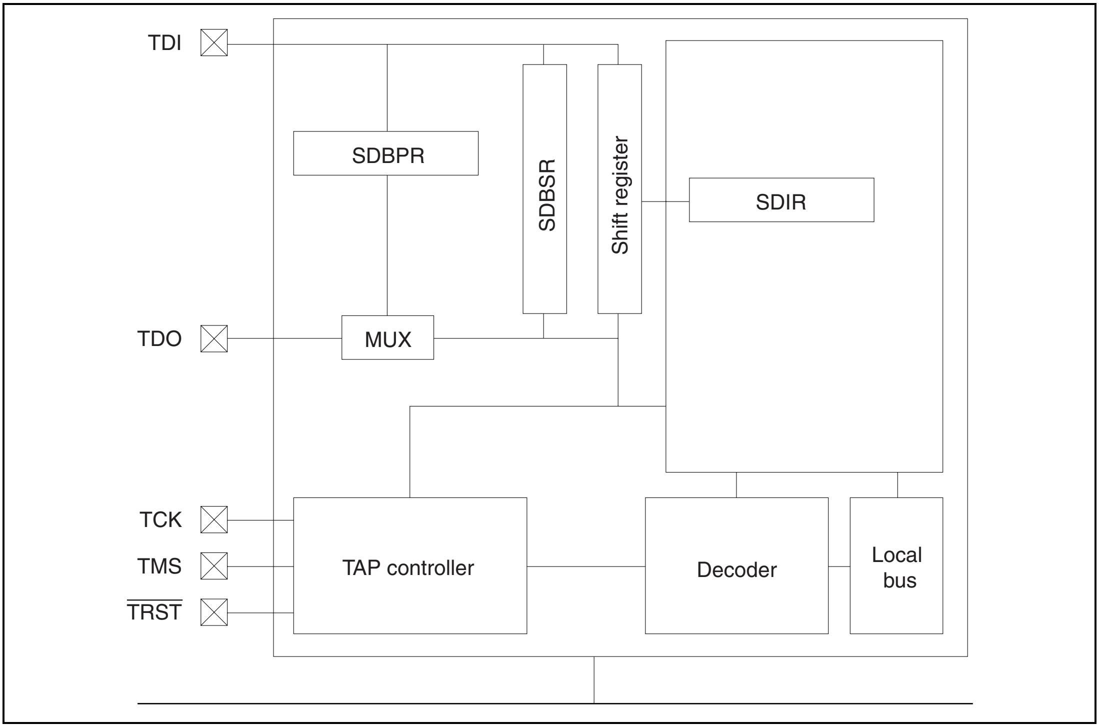


Figure 21.1 H-UDI Block Diagram

21.1 Features

The H-UDI has the following features.

- Support of the E10A emulator
- Standard pin arrangement of JTAG
- Real-time branch trace
- 1-kbyte on-chip RAM for running the high-speed emulation program

21.2 I/O Pins

Table21.1 lists the pin configuration of the H-UDI.

Table 21.1 Pin Configuraiton

Name	Description
TCK	H-UDI serial data input/output clock pin. Data is serially supplied to the H-UDI from the data input pin (TDI), and output from the data output pin (TDO), in synchronization with this clock.
TMS	Mode select input pin. The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. The protocol conforms to the JTAG standard (IEEE Std. 1140.1).
$\overline{\text{TRST}}$	H-UDI reset input pin. Input is accepted asynchronously with respect to TCK, and when low, the H-UDI is reset. See section 21.2, Reset Configuration, for more information.
TDI	H-UDI serial data input pin. Data transfer to the H-UDI is executed by changing this signal in synchronization with TCK.
TDO	H-UDI serial data output pin. Data output from the H-UDI is executed by reading this signal in synchronization with TCK.
$\overline{\text{ASEMD0}}$	ASE mode select pin. If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RESETP}}$ pin is asserted, ASE mode is entered; if a high level is input, normal operation mode is entered. $\overline{\text{ASEMD0}}$ pin should be high level when an emulator or H-UDI is not used. In ASE mode, boundary scan and emulator functions can be used. The input level at the $\overline{\text{ASEMD0}}$ pin should be held for at least one cycle after $\overline{\text{RESETP}}$ negation.
$\overline{\text{ASEBRKAK}}$	Dedicated emulator pin

21.3 Description of Registers

The H-UDI has the following registers. For the addresses and access size of these registers section 23, Control Register Table.

- Bypass register (SDBPR)

- Instruction register (SDIR)
- Boundary register (SDBSR)

21.3.1 Bypass Register (SDBPR)

The bypass register is a 1-bit register that cannot be accessed by the CPU. When the SDIR is set to the bypass mode, the SDBPR is connected between H-UDI pins TDI and TDO.

21.3.2 Instruction Register (SDIR)

The instruction register (SDIR) is a 16-bit read-only register. The register is in bypass mode in its initial state. It is initialized by $\overline{\text{TRST}}$ or in the TAP test-logic-reset state, and can be written by the H-UDI irrespective of the CPU mode. Operation is not guaranteed when a reserved command is set to this register.

Bit	Bit Name	Initial Value	R/W	Description
15	TI3	1	R	Test Instruction Bits
14	TI2	1	R	Cannot be written by the CPU.
13	TI1	1	R	0000: EXTEST
12	TI0	1	R	0100: SAMPLE/PRELOAD 0101: Reserved (Setting prohibited) 0110: H-UDI reset negate 0111: H-UDI reset assert 100X: Reserved (Setting prohibited) 101X: H-UDI interrupt 110X: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Bypass mode (initial value) 0001: Recovery from sleep
11 to 0	—	1	R	Reserved These bits are always read as 1.

Note: X: Don’t care

21.3.3 Boundary Scan Register (SDBSR)

The boundary scan register (SDBSR) is a shift register, located on the PAD, for controlling the input/output pins of this LSI.

Using the EXTEST and SAMPLE/PRELOAD commands, a boundary scan test conforming to the JTAG standard can be carried out. Table 21.2 shows the correspondence between this LSI’s pins and boundary scan register bits.

Table 21.2 This LSI's Pins and Boundary Scan Register Bits

Bit	Pin Name	I/O	Bit	Pin Name	I/O
297	D31/PTB[7]	IN	265	D31/PTB[7]	OUT
296	D30/PTB[6]	IN	264	D30/PTB[6]	OUT
295	D29/PTB[5]	IN	263	D29/PTB[5]	OUT
294	D28/PTB[4]	IN	262	D28/PTB[4]	OUT
293	D27/PTB[3]	IN	261	D27/PTB[3]	OUT
292	D26/PTB[2]	IN	260	D26/PTB[2]	OUT
291	D25/PTB[1]	IN	259	D25/PTB[1]	OUT
290	D24/PTB[0]	IN	258	D24/PTB[0]	OUT
289	D23/PTA[7]	IN	257	D23/PTA[7]	OUT
288	D22/PTA[6]	IN	256	D22/PTA[6]	OUT
287	D21/PTA[5]	IN	255	D21/PTA[5]	OUT
286	D20/PTA[4]	IN	254	D20/PTA[4]	OUT
285	D19/PTA[3]	IN	253	D19/PTA[3]	OUT
284	D18/PTA[2]	IN	252	D18/PTA[2]	OUT
283	D17/PTA[1]	IN	251	D17/PTA[1]	OUT
282	D16/PTA[0]	IN	250	D16/PTA[0]	OUT
281	D15	IN	249	D15	OUT
280	D14	IN	248	D14	OUT
279	D13	IN	247	D13	OUT
278	D12	IN	246	D12	OUT
277	D11	IN	245	D11	OUT
276	D10	IN	244	D10	OUT
275	D9	IN	243	D9	OUT
274	D8	IN	242	D8	OUT
273	D7	IN	241	D7	OUT
272	D6	IN	240	D6	OUT
271	D5	IN	239	D5	OUT
270	D4	IN	238	D4	OUT
269	D3	IN	237	D3	OUT
268	D2	IN	236	D2	OUT
267	D1	IN	235	D1	OUT
266	D0	IN	234	D0	OUT

Bit	Pin Name	I/O
233	D31/PTB[7]	Control
232	D30/PTB[6]	Control
231	D29/PTB[5]	Control
230	D28/PTB[4]	Control
229	D27/PTB[3]	Control
228	D26/PTB[2]	Control
227	D25/PTB[1]	Control
226	D24/PTB[0]	Control
225	D23/PTA[7]	Control
224	D22/PTA[6]	Control
223	D21/PTA[5]	Control
222	D20/PTA[4]	Control
221	D19/PTA[3]	Control
220	D18/PTA[2]	Control
219	D17/PTA[1]	Control
218	D16/PTA[0]	Control
217	D15	Control
216	D14	Control
215	D13	Control
214	D12	Control
213	D11	Control
212	D10	Control
211	D9	Control
210	D8	Control
209	D7	Control
208	D6	Control
207	D5	Control
206	D4	Control
205	D3	Control
204	D2	Control
203	D1	Control
202	D0	Control
201	$\overline{\text{BS}}$/PTC[0]	IN
200	$\overline{\text{WE2}}$/$\overline{\text{DQMUL}}$/$\overline{\text{ICIORD}}$/PTC[1]	IN
199	$\overline{\text{WE3}}$/$\overline{\text{DQMUU}}$/$\overline{\text{ICIOWR}}$/PTC[2]	IN
198	$\overline{\text{CS2}}$/PTC[3]	IN
197	$\overline{\text{CS3}}$/PTC[4]	IN
196	A0	OUT
195	A1	OUT
194	A2	OUT
193	A3	OUT
192	A4	OUT
191	A5	OUT
190	A6	OUT
189	A7	OUT
188	A8	OUT
187	A9	OUT
186	A10	OUT
185	A11	OUT
184	A12	OUT
183	A13	OUT
182	A14	OUT
181	A15	OUT
180	A16	OUT
179	A17	OUT
178	A18	OUT
177	A19	OUT
176	A20	OUT
175	A21	OUT
174	A22	OUT
173	A23	OUT
172	A24	OUT

Bit	Pin Name	I/O	Bit	Pin Name	I/O
171	A25	OUT	138	A22	Control
170	$\overline{\text{BS}}$/PTC[0]	OUT	137	A23	Control
169	$\overline{\text{RD}}$	OUT	136	A24	Control
168	$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	OUT	135	A25	Control
167	$\overline{\text{WE1}}$/$\overline{\text{DQMLU}}$/$\overline{\text{WE}}$	OUT	134	$\overline{\text{BS}}$/PTC[0]	Control
166	$\overline{\text{WE2}}$/$\overline{\text{DQMUL}}$/$\overline{\text{ICIORD}}$/PTC[1]	OUT	133	$\overline{\text{RD}}$	Control
165	$\overline{\text{WE3}}$/$\overline{\text{DQMUU}}$/$\overline{\text{ICIOWR}}$/PTC[2]	OUT	132	$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	Control
164	RD/$\overline{\text{WR}}$	OUT	131	$\overline{\text{WE1}}$/$\overline{\text{DQMLU}}$/$\overline{\text{WE}}$	Control
163	$\overline{\text{CS0}}$	OUT	130	$\overline{\text{WE2}}$/$\overline{\text{DQMUL}}$/$\overline{\text{ICIORD}}$/PTC[1]	Control
162	$\overline{\text{CS2}}$/PTC[3]	OUT	129	$\overline{\text{WE3}}$/$\overline{\text{DQMUU}}$/$\overline{\text{ICIOWR}}$/PTC[2]	Control
161	$\overline{\text{CS3}}$/PTC[4]	OUT	128	RD/$\overline{\text{WR}}$	Control
160	A0	Control	127	$\overline{\text{CS0}}$	Control
159	A1	Control	126	$\overline{\text{CS2}}$/PTC[3]	Control
158	A2	Control	125	$\overline{\text{CS3}}$/PTC[4]	Control
157	A3	Control	124	$\overline{\text{CS4}}$/PTC[5]	IN
156	A4	Control	123	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/PTC[6]	IN
155	A5	Control	122	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$/PTC[7]	IN
154	A6	Control	121	$\overline{\text{CE2A}}$/PTD[6]	IN
153	A7	Control	120	$\overline{\text{CE2B}}$/PTD[7]	IN
152	A8	Control	119	$\overline{\text{RASL}}$/PTD[0]	IN
151	A9	Control	118	$\overline{\text{RASU}}$/PTD[1]	IN
150	A10	Control	117	$\overline{\text{CASL}}$/PTD[2]	IN
149	A11	Control	116	$\overline{\text{CASU}}$/PTD[3]	IN
148	A12	Control	115	CKE/PTD[4]	IN
147	A13	Control	114	$\overline{\text{IOIS16}}$/PTD[5]	IN
146	A14	Control	113	$\overline{\text{BREQ}}$	IN
145	A15	Control	112	$\overline{\text{WAIT}}$	IN
144	A16	Control	111	$\overline{\text{DACK0}}$/PTE[0]	IN
143	A17	Control	110	$\overline{\text{DACK1}}$/PTE[1]	IN
142	A18	Control	109	DRAK0/PTE[2]	IN
141	A19	Control	108	DRAK1/PTE[3]	IN
140	A20	Control	107	AUDATA[0]/PTF[0]	IN
139	A21	Control	106	AUDATA[1]/PTF[1]	IN

Bit	Pin Name	I/O
105	AUDATA[2]/PTF[2]	IN
104	AUDATA[3]/PTF[3]	IN
103	$\overline{\text{AUDSYNC}}$/PTF[4]	IN
102	$\overline{\text{ASEBRKAK}}$/PTF[6]	IN
101	MD1	IN
100	$\overline{\text{CS4}}$/PTC[5]	OUT
99	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/PTC[6]	OUT
98	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$/PTC[7]	OUT
97	$\overline{\text{CE2A}}$/PTD[6]	OUT
96	$\overline{\text{CE2B}}$/PTD[7]	OUT
95	$\overline{\text{RASL}}$/PTD[0]	OUT
94	$\overline{\text{RASU}}$/PTD[1]	OUT
93	$\overline{\text{CASL}}$/PTD[2]	OUT
92	$\overline{\text{CASU}}$/PTD[3]	OUT
91	CKE/PTD[4]	OUT
90	$\overline{\text{IOIS16}}$/PTD[5]	OUT
89	$\overline{\text{BACK}}$	OUT
88	$\overline{\text{DACK0}}$/PTE[0]	OUT
87	$\overline{\text{DACK1}}$/PTE[1]	OUT
86	DRAK0/PTE[2]	OUT
85	DRAK1/PTTE[3]	OUT
84	AUDATA[0]/PTF[0]	OUT
83	AUDATA[1]/PTF[1]	OUT
82	AUDATA[2]/PTF[2]	OUT
81	AUDATA[3]/PTF[3]	OUT
80	$\overline{\text{AUDSYNC}}$/PTF[4]	OUT
79	$\overline{\text{ASEBRKAK}}$/PTF[6]	OUT
78	$\overline{\text{CS4}}$/PTC[5]	Control
77	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/PTC[6]	Control
76	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$/PTC[7]	Control
75	$\overline{\text{CE2A}}$/PTD[6]	Control
74	$\overline{\text{CE2B}}$/PTD[7]	Control
73	$\overline{\text{RASL}}$/PTD[0]	Control
72	$\overline{\text{RASU}}$/PTD[1]	Control
71	$\overline{\text{CASL}}$/PTD[2]	Control
70	$\overline{\text{CASU}}$/PTD[3]	Control
69	CKE/PTD[4]	Control
68	$\overline{\text{IOIS16}}$/PTD[5]	Control
67	$\overline{\text{BACK}}$	Control
66	$\overline{\text{DACK0}}$/PTE[0]	Control
65	$\overline{\text{DACK1}}$/PTE[1]	Control
64	DRAK0/PTE[2]	Control
63	DRAK1/PTTE[3]	Control
62	AUDATA[0]/PTF[0]	Control
61	AUDATA[1]/PTF[1]	Control
60	AUDATA[2]/PTF[2]	Control
59	AUDATA[3]/PTF[3]	Control
58	$\overline{\text{AUDSYNC}}$/PTF[4]	Control
57	$\overline{\text{ASEBRKAK}}$/PTF[6]	Control
56	STATUS0/PTE[4]	IN
55	STATUS1/PTE[5]	IN
54	TCLK/PTE[6]	IN
53	$\overline{\text{IRQOUT}}$/PTE[7]	IN
52	SCK0/SCPT[1]	IN
51	SCK2/SCPT[3]	IN
50	$\overline{\text{RTS2}}$/SCPT[4]	IN
49	RxD0/SCPT[0]	IN
48	RxD2/SCPT[2]	IN
47	$\overline{\text{CTS2}}$/IRQ5/SCPT[5]	IN
46	IRQ0/$\overline{\text{IRL0}}$/PTH[0]	IN
45	IRQ1/$\overline{\text{IRL1}}$/PTH[1]	IN
44	IRQ2/$\overline{\text{IRL2}}$/PTH[2]	IN
43	IRQ3/$\overline{\text{IRL3}}$/PTH[3]	IN
42	IRQ4/PTH[4]	IN

Bit	Pin Name	I/O
41	NMI	IN
40	AUDCK/PTG[4]	IN
39	$\overline{\text{DREQ0}}$/PTH[5]	IN
38	$\overline{\text{DREQ1}}$/PTH[6]	IN
37	$\overline{\text{ADTRG}}$/PTG[5]	IN
36	MD0	IN
35	MD2	IN
34	MD3	IN
33	MD4	IN
32	MD5	IN
31	STATUS0/PTE[4]	OUT
30	STATUS1/PTE[5]	OUT
29	TCLK/PTE[6]	OUT
28	$\overline{\text{IRQOUT}}$/PTE[7]	OUT
27	TxD0/SCPT[0]	OUT
26	SCK0/SCPT[1]	OUT
25	TxD2/SCPT[2]	OUT
24	SCK2/SCPT[3]	OUT
23	$\overline{\text{RTS2}}$/SCPT[4]	OUT
22	IRQ0/$\overline{\text{IRL0}}$/PTH[0]	OUT
21	IRQ1/$\overline{\text{IRL1}}$/PTH[1]	OUT
20	IRQ2/$\overline{\text{IRL2}}$/PTH[2]	OUT
19	IRQ3/$\overline{\text{IRL3}}$/PTH[3]	OUT
18	IRQ4/PTH[4]	OUT
17	$\overline{\text{DREQ0}}$/PTH[5]	OUT
16	$\overline{\text{DREQ1}}$/PTH[6]	OUT
15	STATUS0/PTE[4]	Control
14	STATUS1/PTE[5]	Control
13	TCLK/PTE[6]	Control
12	$\overline{\text{IRQOUT}}$/PTE[7]	Control
11	TxD0/SCPT[0]	Control
10	SCK0/SCPT[1]	Control
9	TxD2/SCPT[2]	Control
8	SCK2/SCPT[3]	Control
7	$\overline{\text{RTS2}}$/SCPT[4]	Control
6	IRQ0/$\overline{\text{IRL0}}$/PTH[0]	Control
5	IRQ1/$\overline{\text{IRL1}}$/PTH[1]	Control
4	IRQ2/$\overline{\text{IRL2}}$/PTH[2]	Control
3	IRQ3/$\overline{\text{IRL3}}$/PTH[3]	Control
2	IRQ4/PTH[4]	Control
1	$\overline{\text{DREQ0}}$/PTH[5]	Control
0	$\overline{\text{DREQ1}}$/PTH[6]	Control

21.4 H-UDI Operations

21.4.1 TAP Controller

Figure 21.2 shows the internal states of TAP controller. State transitions basically conform with the JTAG standard.

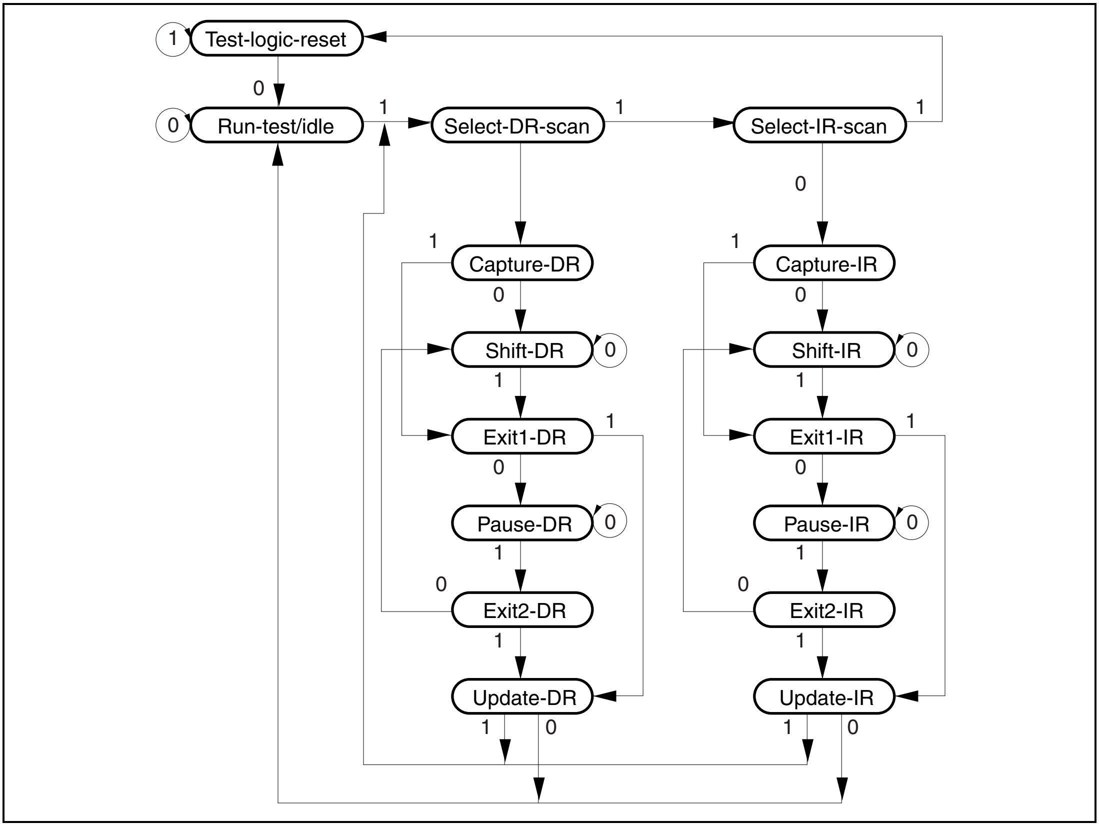


Figure 21.2 TAP Controller State Transitions

Note: The transition condition is the TMS value on the rising edge of TCK. The TDI value is sampled on the rising edge of TCK; shifting occurs on the falling edge of TCK. The TDO value changes on the TCK falling edge. The TDO is at high impedance, except with shift-DR (shift-SR) and shift-IR states. During the change to TRSTN = 0, there is a transition to test-logic-reset asynchronously with TCK.

21.4.2 Reset Configuration

Table 21.2 Reset Configuration

$\overline{\text{ASDMD0}}$*1	$\overline{\text{RESETP}}$	TRST	Chip State
H	L	L	Normal reset and H-UDI reset
		H	Normal reset
	H	L	H-UDI reset only
		H	Normal operation
L	L	L	Reset hold*[2]
		H	Normal reset
	H	L	H-UDI reset only
		H	Normal operation

Notes: *1 Performs normal operation mode and ASE mode settings

$\overline{\text{ASEMD0}}$ = H, normal operation mode

$\overline{\text{ASEMD0}}$ = L, ASE mode

$\overline{\text{ASEMD0}}$ pin should be high level when an emulator or H-UDI is not used.

*2 During ASE mode, reset hold is enabled by setting $\overline{\text{RESETP}}$ and $\overline{\text{TRST}}$ pins at low level for a constant cycle. In this state, the CPU does not start up, even if $\overline{\text{RESETP}}$ is set to high level. When $\overline{\text{TRST}}$ is set to high level, H-UDI operation is enabled, but the CPU does not start up. The reset hold state is cancelled by the following:

- Boot request from H-UDI (boot sequence)
- Another $\overline{\text{RESETP}}$ assert (power-on reset)

21.4.3 H-UDI Reset

An H-UDI reset is executed by setting an H-UDI reset assert command in SDIR. An H-UDI reset is of the same kind as a power-on reset. An H-UDI reset is released by inputting an H-UDI reset negate command.

The interval required between the H-UDI reset assert command and the H-UDI reset negate command is the same as the time for which the RESETP pin is held low in order to execute a power-on reset.

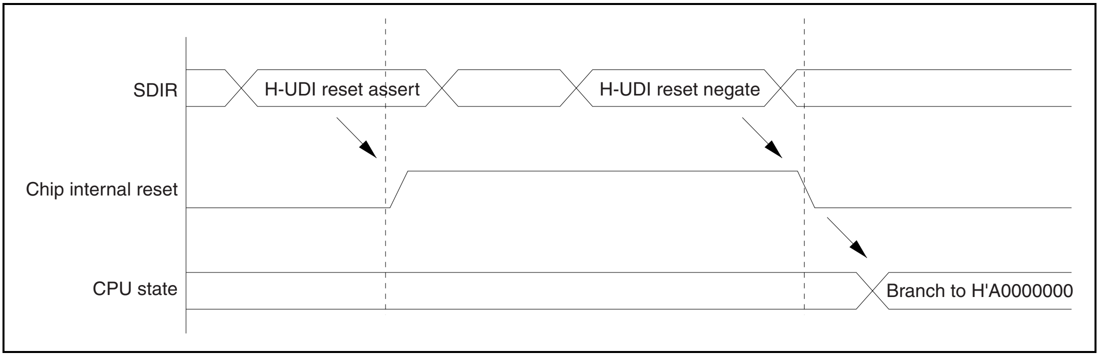


Figure 21.3 H-UDI Reset

21.4.4 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-UDI in the SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in a branch to an address based on the VBR value plus offset, and return by the RTE instruction. This interrupt request has a fixed priority level of 15.

H-UDI interrupts are not accepted in sleep mode or standby mode.

21.4.5 Bypass

The JTAG-based bypass mode for the H-UDI pins can be selected by setting a command from the H-UDI in the SDIR.

21.4.6 Using H-UDI to Recover from Sleep Mode

It is possible to recover from sleep mode by setting a command (0001) from the H-UDI in SDIR.

21.5 Boundary Scan

A command can be set in SDIR by the H-UDI to place the H-UDI pins in the boundary scan mode stipulated by JTAG.

21.5.1 Supported Instructions

This LSI supports the three essential instructions defined in the JTAG standard (BYPASS, SAMPLE/PRELOAD, and EXTEST).

BYPASS: The BYPASS instruction is an essential standard instruction that operates the bypass register. This instruction shortens the shift path to speed up serial data transfer involving other chips on the printed circuit board. While this instruction is executing, the test circuit has no effect on the system circuits. The instruction code is 1111.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction inputs values from this LSI’s internal circuitry to the boundary scan register, outputs values from the scan path, and loads data onto the scan path. When this instruction is executing, this LSI's input pin signals are transmitted directly to the internal circuitry, and internal circuit values are directly output externally from the output pins. This LSI's system circuits are not affected by execution of this instruction. The instruction code is 0100.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the internal circuitry, or a value to be transferred from the internal circuitry to an output pin, is latched into the boundary scan register and read from the scan path. Snapshot latching is performed in synchronization with the rise of TCK in the Capture-DR state. Snapshot latching does not affect normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. Without a PRELOAD operation, when the EXTEST instruction was executed an undefined value would be output from the output pin until completion of the initial scan sequence (transfer to the output latch) (with the EXTEST instruction, the parallel output latch value is constantly output to the output pin).

EXTEST: This instruction is provided to test external circuitry when this LSI is mounted on a printed circuit board. When this instruction is executed, output pins are used to output test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed circuit board, and input pins are used to latch test results into the boundary scan register from the printed circuit board. If testing is carried out by using the EXTEST instruction N times, the Nth test data is scanned-in when test data (N-1) is scanned out.

Data loaded into the output pin boundary scan register in the Capture-DR state is not used for external circuit testing (it is replaced by a shift operation).

The instruction code is 0000.

21.5.2 Notes for Boundary Scan

1. Boundary scan mode does not cover clock-related signals (EXTAL, EXTAL2, XTAL, XTAL2, CKIO).
2. Boundary scan mode does not cover reset-related signals ($\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, CA).
3. Boundary scan mode does not cover H-UDI-related signals (TCK, TDI, TDO, TMS, TRST).
4. When a boundary scan test is carried out, ensure that the CKIO clock operates constantly.

   The CKIO frequency range is as follows:

   Minimum: 1 MHz

   Maximum: Maximum frequency for respective clock mode specified in the CPG section

   Set pins MD[2:0] to the clock mode to be used.

   After powering on, wait for the CKIO clock to stabilize before performing a boundary scan test.
5. Fix the $\overline{\text{RESETP}}$ pin low.
6. Fix the CA pin high, and the $\overline{\text{ASEMD0}}$ pin low.

21.6 Notes for Usage

1. An H-UDI command other than an H-UDI interrupt, once set, will not be modified as long as another command is not re-issued from the H-UDI. An H-UDI interrupt command, however, will be changed to a bypass command once set.
2. Because chip operations are suspended in standby mode, H-UDI commands are not accepted. However, the TAP controller remains in operation at this time.
3. The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.

21.7 Advanced User Debugger (AUD)

The AUD is a function exclusively for use by an emulator. Refer to the User's Manual for the relevant emulator for details of the AUD.

Section 22 Power-Down Modes

In the power-down modes, all CPU and some on-chip supporting module functions are halted. This lowers power consumption.

The SH7706 has four power-down modes:

1. Sleep mode
2. Software standby mode
3. Module standby function (TMU, RTC, SCI, UBC, DMAC, DAC, ADC, and SCIF on-chip supporting modules)
4. Hardware standby mode

Table 22.1 shows the transition conditions for entering the modes from the program execution state, as well as the CPU and supporting module states in each mode and the procedures for canceling each mode.

Table 22.1 Power-Down Modes

		State							
Mode	**Transition Conditions**	**CPG**	**CPU**	**CPU Register**	**On-Chip Memory**	**On-Chip Peripheral Modules**	**Pins**	**External Memory**	**Canceling Procedure**
Sleep mode	Execute SLEEP instruction with STBY bit cleared to 0 in STBCR	Runs	Halts (Register: held)	Held	Held	Run	Held	Refresh	1. Interrupt 2. Reset
Software standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halts	Halts (Register: held)	Held	Held	Halt*[1]	Held	Self-refresh	1. Interrupt 2. Reset
Module standby function	Set MSTP bit of STBCR to 1	Runs	Runs *[4]	Held	Held	Specified module halts	*[2]	Refresh	1. Clear MSTP bit to 0 2. Reset
Hardware standby mode	Drive CA pin low	Halts	Halts	Held	Held	Halt*[3]	Held	Self-refresh	Power-on reset

Notes: 1. The RTC still runs if the START bit in RCR2 is set to 1 (see section 13, Realtime Clock (RTC)). TMU still runs when output of the RTC is used as input to its counter (see section 12, Timer (TMU)).

2. Depends on the on-chip supporting module.

TMU external pin: Held

SCI external pin: Reset

3. The RTC still runs if the START bit in RCR2 is set to 1. TMU does not run.

4. When the LSI enters sleep mode, the CPU halts.

22.1 I/O Pins

Table 22.2 lists the pins used for the power-down modes.

Table 22.2 Pin Configuration

Pin Name	**Symbol**	**I/O**	**Description**		
Processing state 1	STATUS1	O	Operating state of the processor.		
Processing state 0	STATUS0		STATUS1	STATUS0	state
			High-level	High-level	Reset
			High-level	Low-level	Sleep mode
			Low-level	High-level	Standby mode
			Low-level	Low-level	Normal operation

22.2 Description of Registers

These are two control registers for the power-down modes. Refer to section 23, Control Registers Table, for more detail of the addresses and access size.

- Standby control register (STBCR)
- Standby control register 2 (STBCR2)

22.2.1 Standby Control Register (STBCR)

The standby control register (STBCR) is an 8-bit read/write register that sets the power-down mode.

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby Specifies transition to software standby mode. 0: Executing SLEEP instruction puts the chip into sleep mode. 1: Executing SLEEP instruction puts the chip into software standby mode.
6	—	0	R	Reserved
5	—	0	R	These bits are always read as 0. The write value should always be 0.
4	STBXTL	0	R/W	Standby Crystal Specifies whether the crystal oscillator halts or oscillates in standby mode. 0: Halts the oscillation of the crystal oscillator in standby mode. 1: Continues the oscillation of the crystal oscillator even in standby mode.
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP2	0	R/W	Module Stop 2 Specifies halting the clock supply to the timer unit TMU (an on-chip supporting module). When the MSTP2 bit is set to 1, the supply of the clock to the TMU is halted. 0: TMU runs. 1: Clock supply to TMU is halted.
1	MSTP1	0	R/W	Module Stop 1 Specifies halting the clock supply to the realtime clock RTC (an on-chip supporting module). When the MSTP1 bit is set to 1, the supply of the clock to RTC is halted. When the clock halts, all RTC registers become inaccessible, but the counter keeps running. 0: RTC runs. 1: Clock supply to RTC is halted.
0	MSTP0	0	R/W	Module Stop 0 Specifies halting the clock supply to the serial communication interface SCI (an on-chip supporting module). When the MSTP0 bit is set to 1, the supply of the clock to the SCI is halted. 0: SCI operates. 1: Clock supply to SCI is halted.

22.2.2 Standby Control Register 2 (STBCR2)

The standby control register 2 (STBCR2) is a read/write 8-bit register that sets the power-down mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	MDCHG	0	R/W	Pin MD5 to MD0 Control Specifies whether or not pins MD5 to MD0 are changed in software standby mode. When this bit is set to 1, the MD5 to MD0 pin values are latched when returning from software standby mode by means of a reset or interrupt. 0: Pins MD5 to MD0 are not changed in software standby mode 1: Pins MD5 to MD0 are changed in software standby mode
5	MSTP8	0	R/W	Module Stop 8 (): Specifies halting the clock supply to the user break controller UBC (an on-chip supporting module). When the MSTP8 bit is set to 1, the supply of the clock to the UBC is halted. 0: UBC runs 1: Clock supply to UBC is halted
4	MSTP7	0	R/W	Module Stop 7 Specifies halting of clock supply to the DMAC (an on-chip peripheral module). When the MSTP7 bit is set to 1, the supply of the clock to the DMAC is halted. 0: DMAC runs 1: Clock supply to DMAC halted
3	MSTP6	0	R/W	Module Stop 6 Specifies halting of clock supply to the DAC (an on-chip peripheral module). When the MSTP6 bit is set to 1, the supply of the clock to the DAC is halted. 0: DAC runs 1: Clock supply to DAC halted

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP5	0	R/W	Module Stop 5 Specifies halting of clock supply to the ADC (an on-chip peripheral module). When the MSTP5 bit is set to 1, the supply of the clock to the ADC is halted and all registers are initialized. 0: ADC runs 1: Clock supply to ADC halted and all registers initialized
1	MSTP4	0	R/W	Module Stop 4 Specifies halting the clock supply to the serial communication interface with FIFO (an on-chip peripheral module). When the MSTP1 bit is set to 1, the supply of the clock to the SCIF is halted. 0: SCIF runs 1: Clock supply to SCIF halted
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

22.3 Description of Operation

22.3.1 Sleep Mode

- **Transition to Sleep Mode**

Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip supporting modules continue to run during sleep mode and the clock continues to be output to the CKIO pin. In sleep mode, the STATUS1 pin is set high and the STATUS0 pin low.

- **Canceling Sleep Mode**

Sleep mode is canceled by an interrupt (NMI, IRQ, IRL, on-chip supporting module) or reset. Interrupts are accepted during sleep mode even when the BL bit in the SR register is 1. If necessary, save SPC and SSR in the stack before executing the SLEEP instruction.

Canceling with an Interrupt: When an NMI, IRQ, IRL or on-chip supporting module interrupt occurs, sleep mode is canceled and interrupt exception processing is executed. A code indicating the interrupt source is set in the INTEVT and INTEVT2 registers.

Canceling with a Reset: Sleep mode is canceled by a power-on reset or a manual reset.

22.3.2 Software Standby Mode

- **Transition to Software Standby Mode**

To enter standby mode, set the STBY bit to 1 in STBCR, then execute the SLEEP instruction. The chip moves from the program execution state to software standby mode. In software standby mode, power consumption is greatly reduced by halting not only the CPU, but the clock and on-chip supporting modules as well. The clock output from the CKIO pin also halts. CPU and cache register contents are held, but some on-chip supporting modules are initialized. Table 22.3 lists the states of registers in software standby mode.

Table 22.3 Register States in Software Standby Mode

Module	Registers Initialized	Registers Retaining Data
Interrupt controller (INTC)	—	All registers
On-chip clock pulse generator (CPG)	—	All registers
User Break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
Timer unit (TMU)	TSTR register	Registers other than TSTR
Realtime clock (RTC)	—	All registers
A/D converter (ADC)	All registers	—
D/A converter (DAC)	—	All registers

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 WDT. Set the WDT's timer counter (WTCNT) and the CKS2–CKS0 bits of the WTCSR register to appropriate values to secure the specified oscillation settling time.
2. After the STBY bit in the STBCR register is set to 1, a SLEEP instruction is executed.
3. Software standby mode is entered and the clocks within the chip are halted. The STATUS1 pin output goes low and the STATUS0 pin output goes high.

- **Canceling Software Standby Mode**

Standby mode is canceled by an interrupt (NMI, IRQ*[1], IRL*[1], or on-chip supporting module)*[2] or a reset.

Canceling with an Interrupt: The on-chip WDT can be used for hot starts. When the chip detects an NMI, IRL, IRQ,*[1] or on-chip supporting module (except the interval timer)*[2] interrupt, the clock will be supplied to the entire chip and software standby mode canceled after the time set in the WDT's timer control/status register has elapsed. The STATUS1 and STATUS0 pins both go low. Interrupt processing then begins and a code indicating the interrupt source is set in the INTEVT and INTEVT2 registers. After branching to the interrupt processing routine occurs, clear the STBY bit in the STBCR register. The WTCNT stops automatically. If the STBY bit is not cleared, WTCNT continues operation and transits to the standby mode*[3] when it reaches H'80. This function prevents the data from being destroyed due to a rising voltage under an unstable power supply. Interrupts are accepted during software standby mode even when the BL bit in the SR register is 1. If necessary, save SPC and SSR in the stack before executing the SLEEP instruction. Immediately after an interrupt is detected, the phase of the clock output of the CKIO pin may be unstable, until the processor starts interrupt processing. (The canceling condition is that the $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ level is higher than the mask level in the I3–I0 bits in the SR register.)

Notes: 1. Software Standby mode can be canceled using $\overline{\text{IRL3}}$−$\overline{\text{IRL0}}$ or IRQ4−IRQ0.
2. Software standby mode can be canceled with an RTC or TMU (only when running on the RTC clock) interrupt.
3. Standby mode should be canceled by power-on resets. Operations at manual resets or during interrupt input are not guaranted.

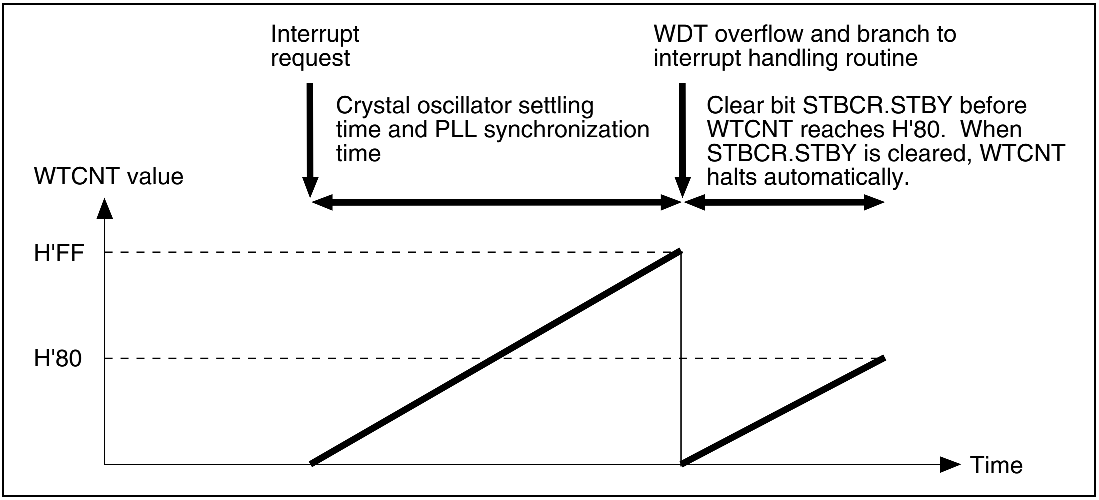


Figure 22.1 Canceling Software Standby Mode with STBCR.STBY

Canceling with a Reset: Standby mode can be canceled with a reset (power-on or manual). Keep the $\overline{\text{RESETP}}$ pin and $\overline{\text{RESETM}}$ pin low until the clock oscillation settles.

- **Clock Pause Function**

In software standby mode, the clock input from the EXTAL pin or CKIO pin can be halted and the frequency can be changed. This function is used as follows:

1. Enter software standby mode using the appropriate procedures.
2. Once software standby mode is entered and the clock stopped within the chip, the STATUS1 pin output is low and the STATUS0 pin output is high.
3. Once the STATUS1 pin goes low and the STATUS0 pin goes high, the input clock is stopped or the frequency is changed.
4. When the frequency is changed, an NMI, $\overline{\text{IRL}}$, IRQ or on-chip supporting module (except the internal timer) interrupt is input after the change. When the clock is stopped, the same interrupts are input after the clock is applied.
5. After the time set in the WDT has elapsed, the clock starts being applied internally within the chip, the STATUS1–STATUS0 pins both go low, interrupts are handled, and operation resumes.

22.3.3 Module Standby Function

- **Transition to Module Standby Function**

Setting the standby control register MSTP8–MSTP0 bits to 1 halts the supply of clocks to the corresponding on-chip supporting modules. This function can be used to reduce the power consumption in sleep mode. The module standby function holds the state prior to halt of the external pins of the on-chip supporting modules. TMU external pins hold their state prior to the halt. SCI external pins go to the reset state. With a few exceptions, all registers hold their values.

Bit	Value	Description
MSTP8	0	UBC runs.
	1	Supply of clock to UBC halted.
MSTP7	0	DMAC runs.
	1	Supply of clock to DMAC halted.
MSTP6	0	DAC runs.
	1	Supply of clock to DAC halted.
MSTP5	0	ADC runs.
	1	Supply of clock to ADC halted, and all registers initialized.
MSTP4	0	SCIF runs.
	1	Supply of clock to SCIF halted.
MSTP2	0	TMU runs.
	1	Supply of clock to TMU halted. Registers initialized.*[1]
MSTP1	0	RTC runs.
	1	Supply of clock to RTC halted. Register access prohibited.*[2]
MSTP0	0	SCI runs.
	1	Supply of clock to SCI halted.

Notes: 1. The registers initialized are the same as in the software standby mode (table 8.4).
2. The counter runs.

- **Clearing the Module Standby Function**

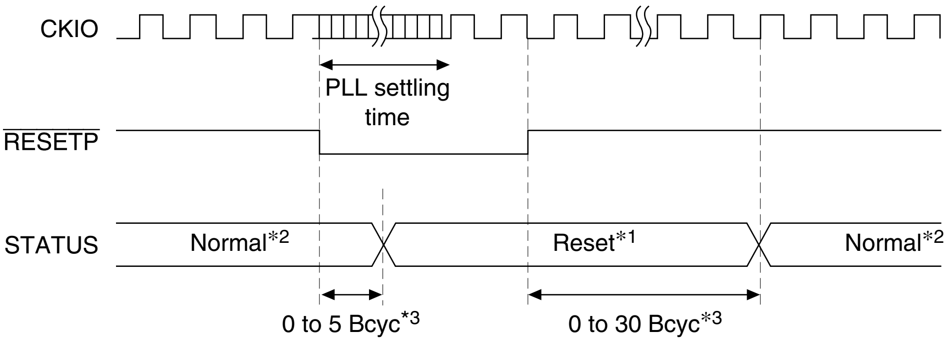
The module standby function can be cleared by clearing the MSTP8–MSTP0 bits to 0, or by a power-on reset or manual reset.

22.3.4 Timing of STATUS Pin Changes

The timing of STATUS1 and STATUS0 pin changes is shown in figures 22.2 through 22.9

- **Timing for Resets**

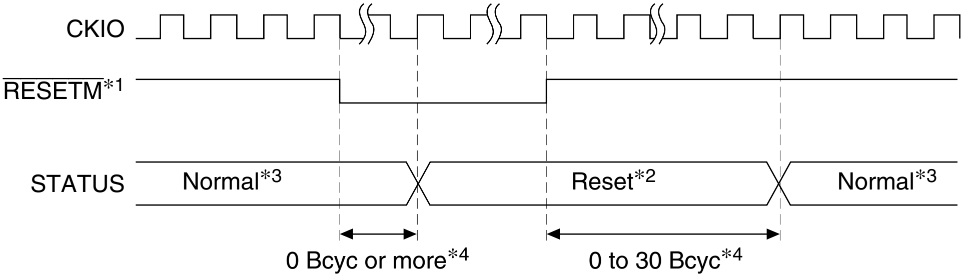
Power-On Reset:



Notes: 1. Reset: HH (STATUS1 high, STATUS0 high)
2. Normal: LL (STATUS1 low, STATUS0 low)
3. Bcyc: Bus clock cycle

Figure 22.2 Power-On Reset STATUS Output

Manual Reset:

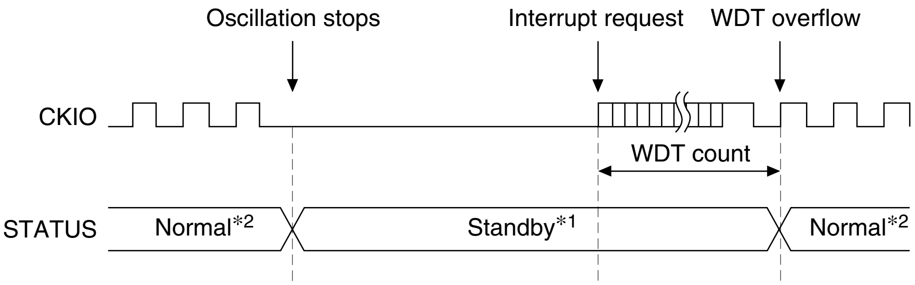


Notes: 1. During manual reset, STATUS becomes HH (reset) and the internal reset begins after waiting for the executing bus cycle to end.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Normal: LL (STATUS1 low, STATUS0 low)
4. Bcyc: Bus clock cycle

Figure 22.3 Manual Reset STATUS Output

- **Timing for Canceling Software Standbys**

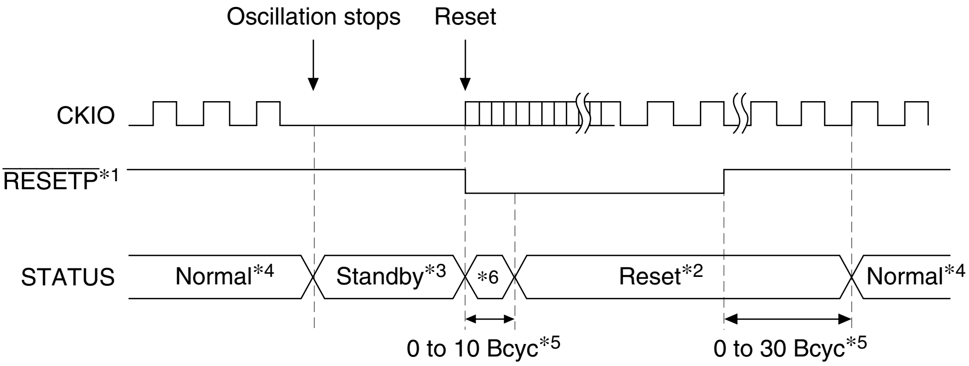
Software Standby to Interrupt:



Notes: 1. Standby: LH (STATUS1 low, STATUS0 high)
2. Normal: LL (STATUS1 low, STATUS0 low)

Figure 22.4 Software Standby to Interrupt STATUS Output

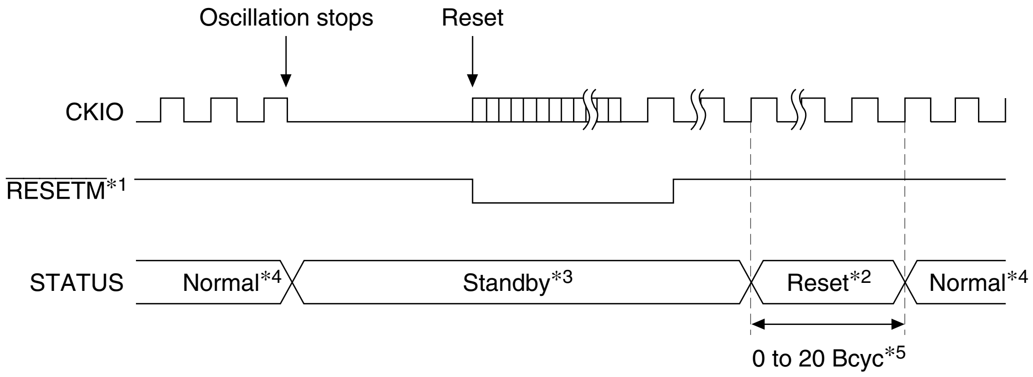
Software Standby to Power-On Reset:



Notes: 1. When software software standby mode is cleared with a power-on reset, the WDT does not count. Keep $\overline{\text{RESETP}}$ 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 22.5 Software Standby to Power-On Reset STATUS Output

Software Standby to Manual Reset:

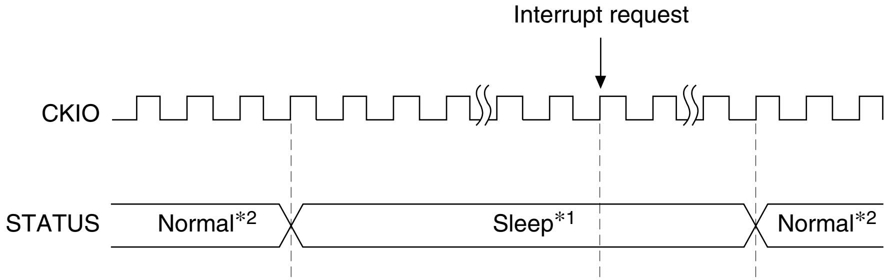


Notes: 1. When software standby mode is cleared with a manual reset, the WDT does not count.
Keep RESETM low during the PLL's oscillation settling time.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Standby: LH (STATUS1 low, STATUS0 high)
4. Normal: LL (STATUS1 low, STATUS0 low)

Figure 22.6 Software Standby to Manual Reset STATUS Output

- Timing for Canceling Sleep Mode

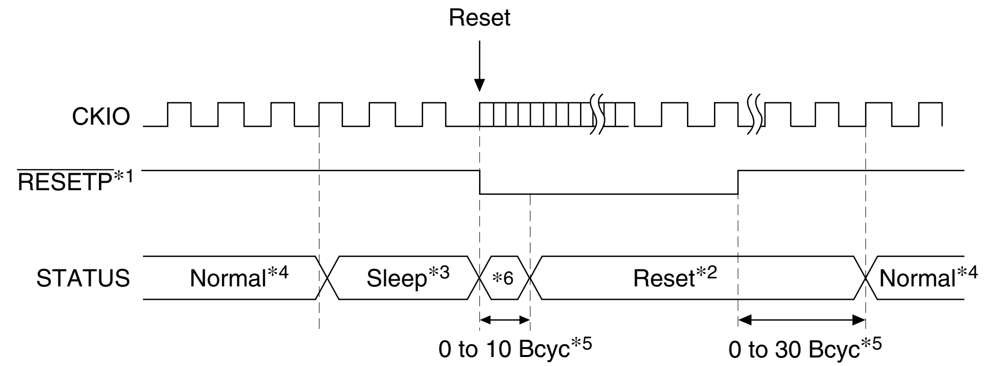
Sleep to Interrupt:



Notes: 1. Sleep: HL (STATUS1 high, STATUS0 low)
2. Normal: LL (STATUS1 low, STATUS0 low)

Figure 22.7 Sleep to Interrupt STATUS Output

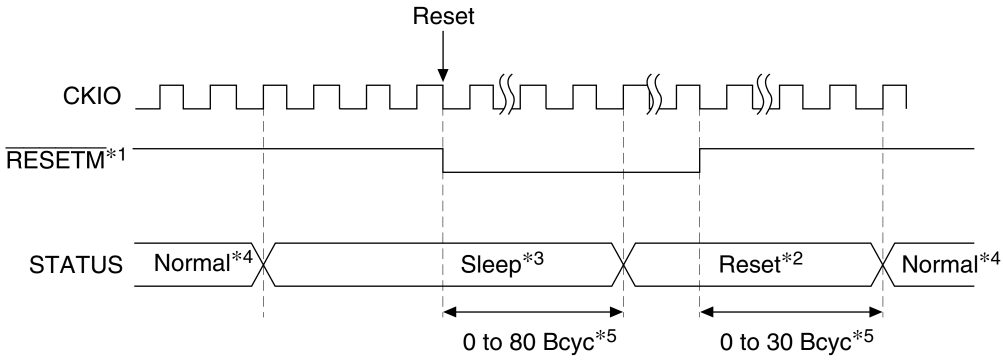
Sleep to Power-On Reset:



Notes: 1. When the PLL1's multiplication ratio is changed by a power-on reset, keep RESETP low during the PLL's oscillation settling time.
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
6. Undefined

Figure 22.8 Sleep to Power-On Reset STATUS Output

Sleep to Manual Reset:



Notes: 1. Keep 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 22.9 Sleep to Manual Reset STATUS Output

22.3.5 Hardware Standby Function

- **Transition to Hardware Standby Mode**

Driving the CA pin low causes a transition to hardware standby mode. In hardware standby mode, all modules except those operating on an RTC clock are halted, as in the software standby mode entered on execution of a SLEEP instruction ((software) standby mode).

Hardware standby mode differs from software standby mode as follows.

1. Interrupts and manual resets are not accepted.
2. The TMU does not operate.

Operation when a low-level signal is input at the CA pin depends on the CPG state, as follows.

1. In software standby mode

   The clock remains stopped and the chip enters the hardware standby state. Acceptance of interrupts and manual resets is disabled, TCLK output is fixed low, and the TMU halts.
2. During WDT operation when software standby mode is canceled by an interrupt

   The chip enters hardware standby mode after standby mode is canceled and the CPU resumes operation.
3. In sleep mode

   The chip enters hardware standby mode after sleep mode is canceled and the CPU resumes operation.

Hold the CA pin low in hardware standby mode.

In hardware standby mode, the LSI can supply power only to the RTC power-supply pin.

- **Canceling Hardware Standby Mode**

Hardware standby mode can only be canceled by a power-on reset.

When the CA pin is driven high while the $\overline{\text{RESETP}}$ pin is low, clock oscillation is started. Hold the $\overline{\text{RESETP}}$ pin low until clock oscillation stabilizes. When the $\overline{\text{RESETP}}$ pin is driven high, the CPU begins power-on reset processing.

If an interrupt or manual reset is input, correct operation cannot be guaranteed.

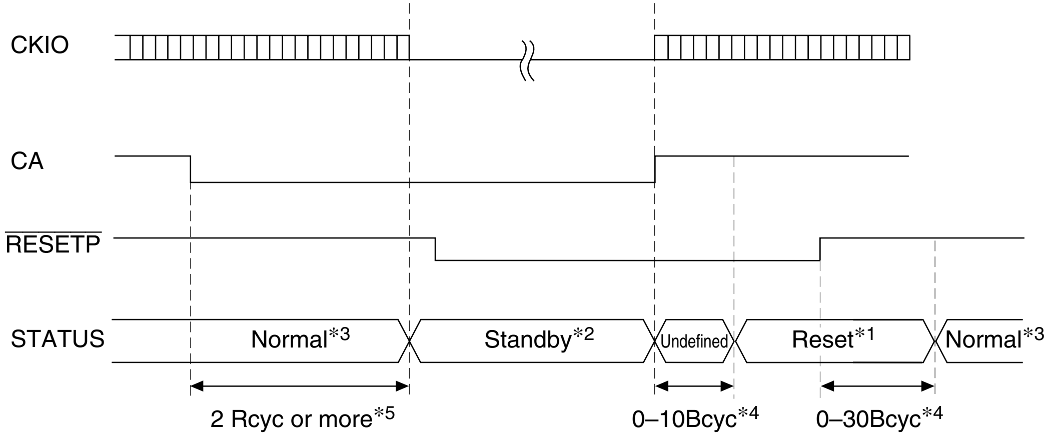
- **Hardware Standby Mode Timing**

Figures 8.10 and 8.11 show examples of pin timing in hardware standby mode.

The CA pin is sampled using EXTAL2 (32.768 kHz), and a hardware standby request is only recognized when the pin is low for two consecutive clock cycles.

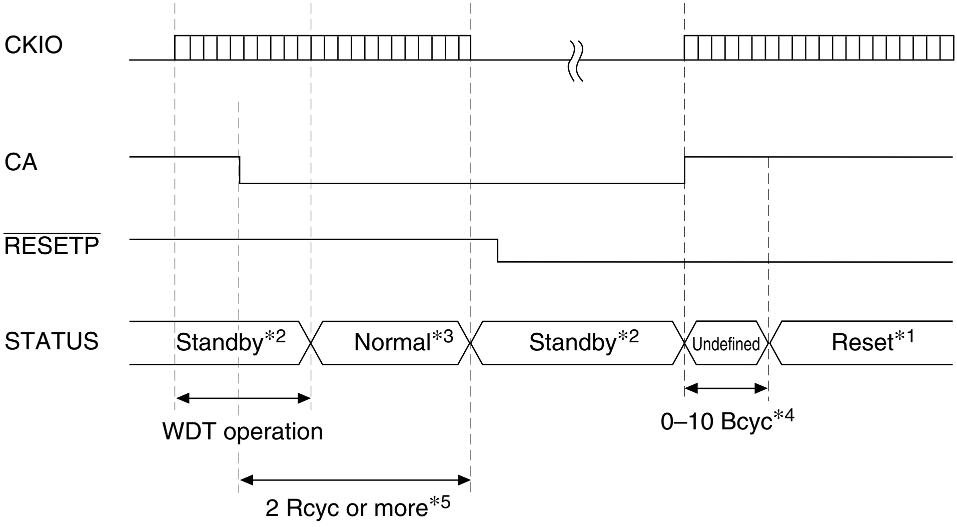
The CA pin must be held low while the chip is in hardware standby mode.

Clock oscillation starts when the CA pin is driven high after the $\overline{\text{RESETP}}$ pin is driven low.



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 22.10 Hardware Standby Mode (When CA Goes Low in Normal Operation)



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 22.11 Hardware Standby Mode Timing
(When CA Goes Low during WDT Operation on Standby Mode Cancellation)

Section 23 Control Register Table

23.1 Register Address Map

Control Register	Module*[1]	Bus*[2]	Address	Size (Bits)	Access Size (Bits)*[3]
PTEH	CCN	L	H'FFFFFFF0	32	32
PTEL		L	H'FFFFFFF4	32	32
TTB		L	H'FFFFFFF8	32	32
TEA		L	H'FFFFFFFC	32	32
MMUCR		L	H'FFFFFFE0	32	32
BASRA		L	H'FFFFFFE4	8	8
BASRB		L	H'FFFFFFE8	8	8
CCR		L	H'FFFFFFEC	32	32
CCR2		I	H'A40000B0	32	32
TRA		L	H'FFFFFFD0	32	32
EXPEVT		L	H'FFFFFFD4	32	32
INTEVT		L	H'FFFFFFD8	32	32
BARA	UBC	L	H'FFFFFFB0	32	32
BAMRA		L	H'FFFFFFB4	32	32
BBRA		L	H'FFFFFFB8	16	16
BARB		L	H'FFFFFFA0	32	32
BAMRB		L	H'FFFFFFA4	32	32
BBRB		L	H'FFFFFFA8	16	16
BDRB		L	H'FFFFFF90	32	32
BDMRB		L	H'FFFFFF94	32	32
BRCR		L	H'FFFFFF98	32	32
BETR		L	H'FFFFFF9C	16	16
BRSR		L	H'FFFFFFAC	32	32
BRDR		L	H'FFFFFFBC	32	32
FRQCR	CPG	I	H'FFFFFF80	16	16
STBCR		I	H'FFFFFF82	8	8
STBCR2		I	H'FFFFFF88	8	8
WTCNT		I	H'FFFFFF84	8	8, 16
WTCSR		I	H'FFFFFF86	8	8, 16
BCR1	BSC	I	H'FFFFFF60	16	16
BCR2		I	H'FFFFFF62	16	16

Control Register	Module*[1]	Bus*[2]	Address	Size (Bits)	Access Size (Bits)*[3]
WCR1	BSC	I	H'FFFFFF64	16	16
WCR2		I	H'FFFFFF66	16	16
MCR		I	H'FFFFFF68	16	16
PCR		I	H'FFFFFF6C	16	16
RTCSR		I	H'FFFFFF6E	16	16
RTCNT		I	H'FFFFFF70	16	16
RTCOR		I	H'FFFFFF72	16	16
RFCR		I	H'FFFFFF74	16	16
SDMR		I	H'FFFFD000– H'FFFFEFFE	—	8
R64CNT	RTC	P	H'FFFFFEC0	8	8
RSECCNT		P	H'FFFFFEC2	8	8
RMINCNT		P	H'FFFFFEC4	8	8
RHRCNT		P	H'FFFFFEC6	8	8
RWKCNT		P	H'FFFFFEC8	8	8
RDAYCNT		P	H'FFFFFECA	8	8
RMONCNT		P	H'FFFFFECC	8	8
RYRCNT		P	H'FFFFFECE	8	8
RSECAR		P	H'FFFFFED0	8	8
RMINAR		P	H'FFFFFED2	8	8
RHRAR		P	H'FFFFFED4	8	8
RWKAR		P	H'FFFFFED6	8	8
RDAYAR		P	H'FFFFFED8	8	8
RMONAR		P	H'FFFFFEDA	8	8
RCR1		P	H'FFFFFEDC	8	8
RCR2		P	H'FFFFFEDE	8	8
ICR0	INTC	I	H'FFFFFEE0	16	16
IPRA		I	H'FFFFFEE2	16	16
IPRB		I	H'FFFFFEE4	16	16

Control Register	Module*[1]	Bus*[2]	Address	Size (Bits)	Access Size (Bits)*[3]
TOCR	TMU	P	H'FFFFFE90	8	8
TSTR		P	H'FFFFFE92	8	8
TCOR_0		P	H'FFFFFE94	32	32
TCNT_0		P	H'FFFFFE98	32	32
TCR_0		P	H'FFFFFE9C	16	16
TCOR_1		P	H'FFFFFEA0	32	32
TCNT_1		P	H'FFFFFEA4	32	32
TCR_1		P	H'FFFFFEA8	16	16
TCOR_2		P	H'FFFFFEAC	32	32
TCNT_2		P	H'FFFFFEB0	32	32
TCR_2		P	H'FFFFFEB4	16	16
TCPR_2		P	H'FFFFFEB8	32	32
SCSMR	SCI	P	H'FFFFFE80	8	8
SCBRR		P	H'FFFFFE82	8	8
SCSCR		P	H'FFFFFE84	8	8
SCTDR		P	H'FFFFFE86	8	8
SCSSR		P	H'FFFFFE88	8	8
SCRDR		P	H'FFFFFE8A	8	8
SCSCMR		P	H'FFFFFE8C	8	8
INTEVT2	INTC	I	H'04000000	32	32
IRR0		I	H'A4000004	16	8
IRR1		I	H'A4000006	16	8
IRR2		I	H'A4000008	16	8
ICR1		I	H'A4000010	16	16
IPRC		I	H'A4000016	16	16
IPRD		I	H'A4000018	16	16
IPRE		I	H'A400001A	16	16
SAR_0	DMAC	P	H'A4000020	32	16,32
DAR_0		P	H'A4000024	32	16,32
DMATCR_0		P	H'A4000028	32	16,32
CHCR_0		P	H'A400002C	32	8,16,32

Control Register	Module*[1]	Bus*[2]	Address	Size (Bits)	Access Size (Bits)*[3]
SAR_1	DMAC	P	H'A4000030	32	16,32
DAR_1		P	H'A4000034	32	16,32
DMATCR_1		P	H'A4000038	32	16,32
CHCR_1		P	H'A400003C	32	8,16,32
SAR_2		P	H'A4000040	32	16,32
DAR_2		P	H'A4000044	32	16,32
DMATCR_2		P	H'A4000048	32	16,32
CHCR_2		P	H'A400004C	32	8,16,32
SAR_3		P	H'A4000050	32	16,32
DAR_3		P	H'A4000054	32	16,32
DMATCR_3		P	H'A4000058	32	16,32
CHCR_3		P	H'A400005C	32	8,16,32
DMAOR		P	H'A4000060	16	8,16
CMSTR	CMT	P	H'A4000070	16	8,16,32
CMCSR		P	H'A4000072	16	8,16,32
CMCNT		P	H'A4000074	16	8,16,32
CMCOR		P	H'A4000076	16	8,16,32
ADDRAH	A/D	P	H'A4000080	8	8,16,32*[4],*[5]
ADDRAL		P	H'A4000082	8	8,16*[4]
ADDRBH		P	H'A4000084	8	8,16,32*[4],*[5]
ADDRBL		P	H'A4000086	8	8,16*[4]
ADDRCH		P	H'A4000088	8	8,16,32*[4],*[5]
ADDRCL		P	H'A400008A	8	8,16*[4]
ADDRDH		P	H'A400008C	8	8,16,32*[4],*[5]
ADDRDL		P	H'A400008E	8	8,16*[4]
ADCSR		P	H'A4000090	8	8,16,32*[4],*[5]
ADCR		P	H'A4000092	8	8,16
DADR0	D/A	P	H'A40000A0	8	8,16,32*[4],*[5]
DADR1		P	H'A40000A2	8	8,16*[4]
DACR		P	H'A40000A4	8	8,16,32

Control Register	Module*[1]	Bus*[2]	Address	Size (Bits)	Access Size (Bits)*[3]
PACR	PORT	P	H'A4000100	16	16
PBCR		P	H'A4000102	16	16
PCCR		P	H'A4000104	16	16
PDCR		P	H'A4000106	16	16
PECR		P	H'A4000108	16	16
PFCR		P	H'A400010A	16	16
PGCR		P	H'A400010C	16	16
PHCR		P	H'A400010E	16	16
PJCR		P	H'A4000110	16	16
SCPCR		P	H'A4000116	16	16
PADR		P	H'A4000120	8	8
PBDR		P	H'A4000122	8	8
PCDR		P	H'A4000124	8	8
PDDR		P	H'A4000126	8	8
PEDR		P	H'A4000128	8	8
PFDR		P	H'A400012A	8	8
PGDR		P	H'A400012C	8	8
PHDR		P	H'A400012E	8	8
PJDR		P	H'A4000130	8	8
SCPDR		P	H'A4000136	8	8
SCSMR2	SCIF	P	H'A4000150	8	8
SCBRR2		P	H'A4000152	8	8
SCSCR2		P	H'A4000154	8	8
SCFTDR2		P	H'A4000156	8	8
SCSSR2		P	H'A4000158	16	16
SCFRDR2		P	H'A400015A	8	8
SCFCR2		P	H'A400015C	8	8
SCFDR2		P	H'A400015E	16	16
SDIR	UDI	I	H'A4000200	16	16

Notes: 1. Modules:

CCN: Cache controller

CPG: Clock pulse generator

RTC: Realtime clock

TMU: Timer unit

UBC: User break controller

BSC: Bus state controller

INTC: Interrupt controller

SCI: Serial communication interface

2. Internal buses:

L: CPU, CCN, cache, and TLB connected

I: BSC, cache, DMAC, INTC, CPG, and H-UDI connected

P: BSC and peripheral modules (RTC, TMU, SCI, SCIF, A/D, D/A, DMAC, ports, CMT) connected

3. The access size shown is for control register access (read/write). An incorrect result will be obtained if a different size from that shown is used for access.

4. With 16-bit access, it is not possible to read data in two registers simultaneously.

5. With 32-bit access, it is not possible to read data in the register at [accessed address + 2] simultaneously.

23.2 Register Bits

The following are the bit-name of each registers. The 16-bit and 32-bit registers are shown by tow and four 8-bit rows, respectively.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SCSMR	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI
SCBRR									
SCSCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
SCTDR									
SCSSR	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT	
SCRDR									
SCSCMR	—	—	—	—	SDIR	SINV	—	SMIF	
SCFRDR2									SCIF
SCFTDR2									
SCSMR2	—	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCSCR2	TIE	RIE	TE	RE	—	—	CKE1	CKE0	
SCSSR2	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCBRR2									
SCFCR2	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR2									
TOCR	—	—	—	—	—	—	—	TCOE	TMU
TSTR	—	—	—	—	—	STR2	STR1	STR0	
TCOR_0									
TCNT_0									
TCR_0	—	—	—	—	—	—	—	UNF	
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCOR_1									TMU
TCNT_1									
TCR_1	—	—	—	—	—	—	—	UNF	
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCOR_2									
TCNT_2									
TCR_2	—	—	—	—	—	—	ICPF	UNF	
	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCPR_2									
R64CNT	—	1 Hz	2 Hz	4 Hz	8 Hz	16 Hz	32 Hz	64 Hz	RTC
RSECCNT	—	10 sec			1 sec				
RMINCNT	—	10 min			1 min				
RHRCNT	—	—	10 hours		1 hour				
RWKCNT	—	—	—	—	—	day of week			
RDAYCNT	—	—	10 days		1 day				
RMONCNT	—	—	—	10 months	1 month				

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
RYRCNT	10 years				1 year				RTC
RSECAR	ENB	10 sec			1 sec				
RMINAR	ENB	10 min			1 min				
RHRAR	ENB	—	10 hours		1 hour				
RWKAR	ENB	—	—	—	—	day of week			
RDAYAR	ENB	—	10 days		1 day				
RMONAR	ENB	—	—	10 months	1 month				
RCR1	CF	—	—	CIE	AIE	—	—	AF	
RCR2	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START	
ICR0	NML	—	—	—	—	—	—	NMIE	INTC
	—	—	—	—	—	—	—	—	
IPRA	TMU0				TMU1				
	TMU2				RTC				
IPRB	WDT				REF				
	SCI				—	—	—	—	
BCR1	PULA	PULD	HIZMEM	HIZCNT	ENDIAN	A0BST1	A0BST0	A5BST1	BSC
	A5BST0	A6BST1	A6BST0	DRAMTP2	DRAMTP1	DRAMTP0	A5PCM	A6PCM	
BCR2	—	—	A6SZ1	A6SZ0	A5SZ1	A5SZ0	A4SZ1	A4SZ0	
	A3SZ1	A3SZ0	A2SZ1	A2SZ0	—	—	—	—	
WCR1	WAITSEL	—	A6IW1	A6IW0	A5IW1	A5IW0	A4IW1	A4IW0	
	A3IW1	A3IW0	A2IW1	A2IW0	—	—	A0IW1	A0IW0	
WCR2	A6W2	A6W1	A6W0	A5W2	A5W1	A5W0	A4W2	A4W1	
	A4W0	A3W1	A3W0	A2W1	A2W0	A0W2	A0W1	A0W0	
MCR	TPC1	TPC0	RCD1	RCD0	TRWL1	TRWL0	TRAS1	TRAS0	
	RASD	AMX3	AMX2	AMX1	AMX0	RFSH	RMODE	—	
PCR	A6W3	A5W3	—	—	A5TED2	A6TED2	A5THE2	A6THE2	
	A5TED1	A5TED0	A6TED1	A6TED0	A5THE1	A5THE0	A6THE1	A6THE0	
RTCSR	—	—	—	—	—	—	—	—	
	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS	
RTCNT	—	—	—	—	—	—	—	—	

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
RTCOR	—	—	—	—	—	—	—	—	BSC
RFCR	—	—	—	—	—	—	—	—	
SDMR									
FRQCR	STC2	IFC2	PFC2	—	—	—	—	—	CPG
	—	—	STC1	STC0	IFC1	IFC0	PFC1	PFC0	
STBCR	STBY	—	—	STBYTL	—	MSTP2	MSTP1	MSTP0	
STBCR2	—	MDCHG	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	—	
WTCNT									
WTCSR	TME	WT/$\overline{IT}$	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0	
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	UBC
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16	
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0	
BRCR	—	—	—	—	—	—	—	—	
	—	—	BASMA	BASMB	—	—	—	—	
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—	
	DBEB	PCBB	—	—	SEQ	—	—	ETBE	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24	
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16	
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8	
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0	

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
BBRB	—	—	—	—	—	—	—	—	UBC
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0	
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16	
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0	
BBRA	—	—	—	—	—	—	—	—	
	CDA1	CDA0	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0	
BETR	—	—	—	—					
BRSR	SVF	PID2	PID1	PID0	BSA27	BSA26	BSA25	BSA24	
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16	
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0	
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BDA24	
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16	
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0	
BASRA	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA1	BASA0	
BASRB	BASB7	BASB6	BASB5	BASB4	BASB3	BASB2	BASB1	BASB0	
TRA	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	imm		
							—	—	
EXPEVT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—					

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
INTEVT	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—					
MMUCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	SV	
	—	—	RC	RC	—	TF	IX	AT	
CCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CF	CB	WT	CE	
CCR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	W3LOAD	W3LOCK	
	—	—	—	—	—	—	W2LOAD	W2LOCK	
PTEH	VPN								
							—	—	
	ASID								
PTEL	PPN								
							—	V	
	—	PR	PR	SZ	C	D	SH	—	
TTB									
TEA									

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
INTEVT2	—	—	—	—	—	—	—	—	INTC
	—	—	—	—	—	—	—	—	
	—	—	—	—					
IRR0	—	—	IRQ5R	IRQ4R	IRQ3R	IRQ2R	IRQ1R	IRQ0R	
IRR1	—	—	—	—	DEI3R	DEI2R	DEI1R	DEI0R	
IRR2	—	—	—	ADIR	TXI2R	BRI2R	RXI2R	ERI2R	
ICR1	MAI	IRQLVL	BLMSK	—	IRQ51S	IRQ50S	IRQ41S	IRQ40S	
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
IPRC	IRQ3				IRQ2				
	IRQ1				IRQ0				
IPRD	—	—	—	—	—	—	—	—	
	IRQ5				IRQ4				
IPRE	DMAC				—	—	—	—	
	SCIF				A/D				
SAR_0									DMAC
DAR_0									
DMATCR_0	—	—	—	—	—	—	—	—	
CHCR_0	—	—	—	—	—	—	—	—	
	—	—	—	DI	RO	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SAR_1									DMAC
DAR_1									
DMATCR_1	—	—	—	—	—	—	—	—	
CHCR_1	—	—	—	—	—	—	—	—	
	—	—	—	DI	RO	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
SAR_2									
DAR_2									
DMATCR_2	—	—	—	—	—	—	—	—	

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
CHCR_2	—	—	—	—	—	—	—	—	DMAC
	—	—	—	DI	RO	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
SAR_3									
DAR_3									
DMATCR_3	—	—	—	—	—	—	—	—	
CHCR_3	—	—	—	—	—	—	—	—	
	—	—	—	DI	RO	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
DMAOR	—	—	—	—	—	—	PR1	PR0	
	—	—	—	—	—	AE	NMIF	DME	
CMSTR	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	—	STR0	
CMCSR	—	—	—	—	—	—	—	—	
	CMF	—	—	—	—	—	CKS1	CKS0	
CMCNT									
CMCOR									

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
ADDRAL	AD1	AD0	—	—	—	—	—	—	
ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRBL	AD1	AD0	—	—	—	—	—	—	
ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRCL	AD1	AD0	—	—	—	—	—	—	
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRDL	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	MULTI	CKS	CH2	CH1	CH0	
ADCR	TRGE	TRGE0	SCN	—	—	—	—	—	
DADR0									D/A
DADR1									
DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	
PACR	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0	PORT
	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0	
PBCR	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0	
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0	
PCDR	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0	
	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0	
PDCR	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0	
	PD3MD1	PD3MD0	PD2MD1	PD2D0	PD1MD1	PD1MD0	PD0MD1	PD0MD0	
PECR	PE7MD1	PE7MD0	PE6MD1	PE6MD0	PE5MD1	PE5MD0	PE4MD1	PE4MD0	
	PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0	

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PFCR	—	—	PF6MD1	PF6MD0	PF5MD1	PF5MD0	PF4MD1	PF4MD0	PORT
	PF3MD1	PF3MD0	PF2MD1	PF2MD0	PF1MD1	PF1MD0	PF0MD1	PF0MD0	
PGCR	—	—	—	—	PG5MD1	PG5MD0	PG4MD1	PG4MD0	
	PG3MD1	PG3MD0	PG2MD1	PG2MD0	PG1MD1	PG1MD0	PG0MD1	PG0MD0	
PHCR	—	—	PH6MD1	PH6MD0	PH5MD1	PH5MD0	PH4MD1	PH4MD0	
	PH3MD1	PH3MD0	PH2MD1	PH2MD0	PH1MD1	PH1MD0	PH0MD1	PH0MD0	
PJCR	—	—	—	—	—	—	—	—	
	PJ3MD1	PJ3MD0	PJ2MD1	PJ2MD0	PJ1MD1	PJ1MD0	PJ0MD1	PJ0MD0	
SCPCR	—	—	—	—	SCP5MD1	SCP5MD0	SCP4MD1	SCP4MD0	
	SCP3MD1	SCP3MD0	SCP2MD1	SCP2MD0	SCP1MD1	SCP1MD0	SCP0MD1	SCP0MD0	
PADR	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT	
PBDR	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT	
PCDR	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT	
PDDR	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT	
PEDR	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT	
PFDR	—	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF0DT	
PGDR	—	—	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT	
PHDR	—	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT	
PJDR	—	—	—	—	PJ3DT	PJ2DT	PJ1DT	PJ0DT	
SCPDR	—	—	SCP5DT	SCP4DT	SCP3DT	SCP2DT	SCP1DT	SCP0DT	
SDIR	TI3	TI2	TI1	TI0	—	—	—	—	UDI
	—	—	—	—	—	—	—	—	

23.3 Register States in Processing Mode

Register Name	Power-on Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Madule
PTEH	Initialized	Initialized	Held	Held	Held	Held	CCN
PTEL	Initialized	Initialized	Held	Held	Held	Held	
TTB	Initialized	Initialized	Held	Held	Held	Held	
TEA	Initialized	Initialized	Held	Held	Held	Held	
MMUCR	Initialized	Initialized	Held	Held	Held	Held	
BASRA	Initialized	Initialized	Held	Held	Held	Held	
BASRB	Initialized	Initialized	Held	Held	Held	Held	
CCR	Initialized	Initialized	Held	Held	Held	Held	
CCR2	Initialized	Initialized	Held	Held	Held	Held	
TRA	Initialized	Initialized	Held	Held	Held	Held	
EXPEVT	Initialized	Initialized	Held	Held	Held	Held	
INTEVT	Initialized	Initialized	Held	Held	Held	Held	
BARA	Initialized	Initialized	Held	Held	Held	Held	UBC
BAMRA	Initialized	Initialized	Held	Held	Held	Held	
BBRA	Initialized	Initialized	Held	Held	Held	Held	
BARB	Initialized	Initialized	Held	Held	Held	Held	
BAMRB	Initialized	Initialized	Held	Held	Held	Held	
BBRB	Initialized	Initialized	Held	Held	Held	Held	
BDRB	Initialized	Initialized	Held	Held	Held	Held	
BDMRB	Initialized	Initialized	Held	Held	Held	Held	
BRCR	Initialized	Initialized	Held	Held	Held	Held	
BETR	Initialized	Initialized	Held	Held	Held	Held	
BRSR	Initialized	Initialized	Held	Held	Held	Held	
BRDR	Initialized	Initialized	Held	Held	Held	Held	
FRQCR	Initialized	Held	Held	Held	Held	Held	CPG
STBCR	Initialized	Held	Held	Held	Held	Held	
STBCR2	Initialized	Held	Held	Held	Held	Held	
WTCNT	Initialized	Held	Held	Held	Held	Held	
WTCSR	Initialized	Held	Held	Held	Held	Held	

Register Name	Power-on Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Madule
BCR1	Initialized	Held	Held	Held	Held	Held	BSC
BCR2	Initialized	Held	Held	Held	Held	Held	
WCR1	Initialized	Held	Held	Held	Held	Held	
WCR2	Initialized	Held	Held	Held	Held	Held	
MCR	Initialized	Held	Held	Held	Held	Held	
PCR	Initialized	Held	Held	Held	Held	Held	
RTCSR	Initialized	Held	Held	Held	Held	Held	
RTCNT	Initialized	Held	Held	Held	Held	Held	
RTCOR	Initialized	Held	Held	Held	Held	Held	
RFCR	Initialized	Held	Held	Held	Held	Held	
R64CNT	Held	Held	Held	Held	Held	Held	RTC
RSECCNT	Held	Held	Held	Held	Held	Held	
RMINCNT	Held	Held	Held	Held	Held	Held	
RHRCNT	Held	Held	Held	Held	Held	Held	
RWKCNT	Held	Held	Held	Held	Held	Held	
RDAYCNT	Held	Held	Held	Held	Held	Held	
RMONCNT	Held	Held	Held	Held	Held	Held	
RYRCNT	Held	Held	Held	Held	Held	Held	
RSECAR	Held	Held	Held	Held	Held	Held	
RMINAR	Held	Held	Held	Held	Held	Held	
RHRAR	Held	Held	Held	Held	Held	Held	
RWKAR	Held	Held	Held	Held	Held	Held	
RDAYAR	Held	Held	Held	Held	Held	Held	
RMONAR	Held	Held	Held	Held	Held	Held	
RCR1	Initialized	Initialized	Held	Held	Held	Held	
RCR2	Initialized	Initialized	Held	Held	Held	Held	
ICR0	Initialized	Initialized	Held	Held	Held	Held	INTC
IPRA	Initialized	Initialized	Held	Held	Held	Held	
IPRB	Initialized	Initialized	Held	Held	Held	Held	
TOCR	Initialized	Initialized	Held	Held	Held	Held	TMU
TSTR	Initialized	Initialized	Held	Held	Held	Held	
TCOR_0	Initialized	Initialized	Held	Held	Held	Held	

Register Name	Power-on Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Madule
TCNT_0	Initialized	Initialized	Held	Held	Held	Held	TMU
TCR_0	Initialized	Initialized	Held	Held	Held	Held	
TCOR_1	Initialized	Initialized	Held	Held	Held	Held	
TCNT_1	Initialized	Initialized	Held	Held	Held	Held	
TCR_1	Initialized	Initialized	Held	Held	Held	Held	
TCOR_2	Initialized	Initialized	Held	Held	Held	Held	
TCNT_2	Initialized	Initialized	Held	Held	Held	Held	
TCR_2	Initialized	Initialized	Held	Held	Held	Held	
TCPR_2	Initialized	Initialized	Held	Held	Held	Held	
SCSMR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	SCI
SCBRR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
SCSCR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
SCTDR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
SCSSR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
SCRDR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
SCSCMR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
INTEVT2	Initialized	Initialized	Held	Held	Held	Held	INTC
IRR0	Initialized	Initialized	Held	Held	Held	Held	
IRR1	Initialized	Initialized	Held	Held	Held	Held	
IRR2	Initialized	Initialized	Held	Held	Held	Held	
ICR1	Initialized	Initialized	Held	Held	Held	Held	
IPRC	Initialized	Initialized	Held	Held	Held	Held	
IPRD	Initialized	Initialized	Held	Held	Held	Held	
IPRE	Initialized	Initialized	Held	Held	Held	Held	
SAR_0	Initialized	Initialized	Held	Held	Held	Held	DMAC
DAR_0	Initialized	Initialized	Held	Held	Held	Held	
DMATCR_0	Initialized	Initialized	Held	Held	Held	Held	
CHCR_0	Initialized	Initialized	Held	Held	Held	Held	
SAR_1	Initialized	Initialized	Held	Held	Held	Held	
DAR_1	Initialized	Initialized	Held	Held	Held	Held	
DMATCR_1	Initialized	Initialized	Held	Held	Held	Held	
CHCR_1	Initialized	Initialized	Held	Held	Held	Held	

Register Name	Power-on Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Madule
SAR_2	Initialized	Initialized	Held	Held	Held	Held	DMAC
DAR_2	Initialized	Initialized	Held	Held	Held	Held	
DMATCR_2	Initialized	Initialized	Held	Held	Held	Held	
CHCR_2	Initialized	Initialized	Held	Held	Held	Held	
SAR_3	Initialized	Initialized	Held	Held	Held	Held	
DAR_3	Initialized	Initialized	Held	Held	Held	Held	
DMATCR_3	Initialized	Initialized	Held	Held	Held	Held	
CHCR_3	Initialized	Initialized	Held	Held	Held	Held	
DMAOR	Initialized	Initialized	Held	Held	Held	Held	
CMSTR	Initialized	Initialized	Held	Held	Held	Held	CMT
CMCSR	Initialized	Initialized	Held	Held	Held	Held	
CMCNT	Initialized	Initialized	Held	Held	Held	Held	
CMCOR	Initialized	Initialized	Held	Held	Held	Held	
ADDRAH	Initialized	Initialized	Initialized	Initialized	Initialized	Held	ADC
ADDRAL	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ADDRBH	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ADDRBL	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ADDRCH	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ADDRCL	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ADDRDH	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ADDRDL	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ADCSR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ADCR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
DADR0	Initialized	Initialized	Held	Held	Held	Held	DAC
DADR1	Initialized	Initialized	Held	Held	Held	Held	
DACR	Initialized	Initialized	Held	Held	Held	Held	
PACR	Initialized	Held	Held	Held	Held	Held	PORT
PBCR	Initialized	Held	Held	Held	Held	Held	
PCCR	Initialized	Held	Held	Held	Held	Held	
PDCR	Initialized	Held	Held	Held	Held	Held	
PECR	Initialized	Held	Held	Held	Held	Held	
PFCR	Initialized	Held	Held	Held	Held	Held	

Register Name	Power-on Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Madule
PGCR	Initialized	Held	Held	Held	Held	Held	PORT
PHCR	Initialized	Held	Held	Held	Held	Held	
PJCR	Initialized	Held	Held	Held	Held	Held	
SCPCR	Initialized	Held	Held	Held	Held	Held	
PADR	Initialized	Held	Held	Held	Held	Held	
PBDR	Initialized	Held	Held	Held	Held	Held	
PCDR	Initialized	Held	Held	Held	Held	Held	
PDDR	Initialized	Held	Held	Held	Held	Held	
PEDR	Initialized	Held	Held	Held	Held	Held	
PFDR	Initialized	Held	Held	Held	Held	Held	
PGDR	Initialized	Held	Held	Held	Held	Held	
PHDR	Initialized	Held	Held	Held	Held	Held	
PJDR	Initialized	Held	Held	Held	Held	Held	
SCPDR	Initialized	Held	Held	Held	Held	Held	
SCSMR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held	SCIF
SCBRR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
SCSCR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
SCFTDR2	Undefined	Undefined	Undefined	Undefined	Undefined	Held	
SCSSR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
SCFRDR2	Undefined	Undefined	Undefined	Undefined	Undefined	Held	
SCFCR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
SCFDR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
SDIR*	Held	Held	Held	Held	Held	Held	UDI

Note: * Initilized on asserting state of $\overline{\text{TRST}}$ or on Test-Logic-Reset state of TAP.

Section 24 Electrical Characteristics

24.1 Absolute Maximum Ratings

Table 24.1 shows the absolute maximum ratings.

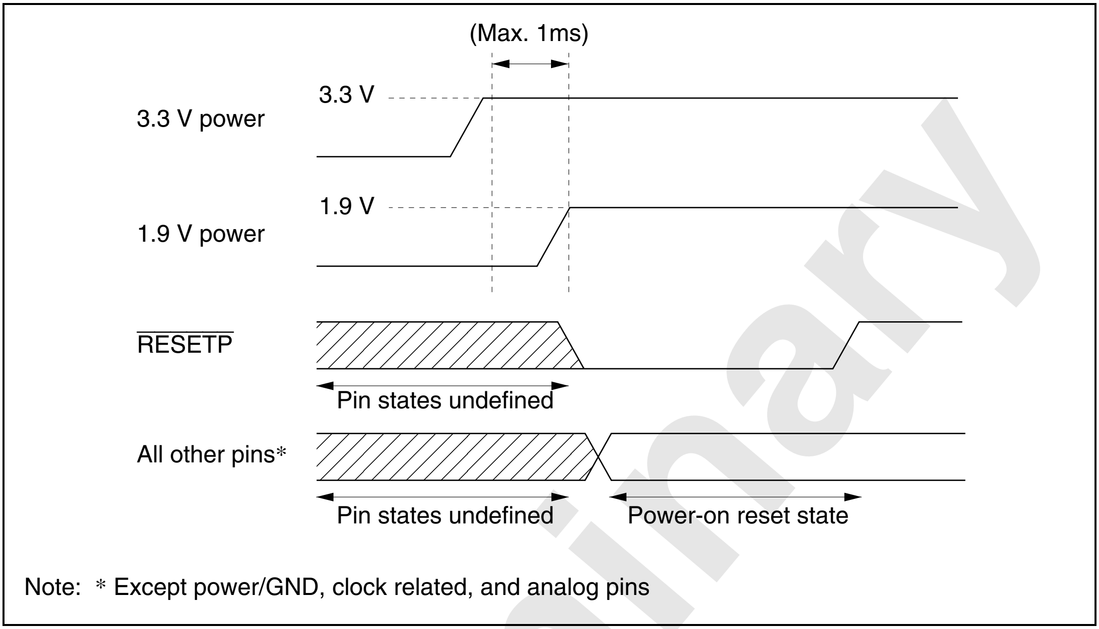
Table 24.1 Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Power supply voltage (I/O)	VccQ	–0.3 to 4.2	V
Power supply voltage (internal)	Vcc Vcc – PLL1 Vcc – PLL2 Vcc – RTC	–0.3 to 2.5	V
Input voltage (except port J)	Vin	–0.3 to VccQ + 0.3	V
Input voltage (port J)	Vin	–0.3 to AVcc + 0.3	V
Analog power-supply voltage	AVcc	–0.3 to 4.6	V
Analog input voltage	V_{AN}	–0.3 to AVcc + 0.3	V
Operating temperature	Topr	–20 to 75	°C
Storage temperature	Tstr	–55 to 125	°C

Caution: Operating the chip in excess of the absolute maximum rating may result in permanent damage.

- Order of turning on 1.9 V power (Vcc, Vcc-PLL1, Vcc-PLL2, Vcc-RTC) and 3.3 V power (VccQ, AVcc):
  1. First turn on the 3.3 V power, then turn on the 1.9 V power within 1ms. This interval should be as short as possible.
  2. Until voltage is applied to all power supplies, a high level is input at the CA pin, and a low level is input at the $\overline{\text{RESETP}}$ pin, and CKIO clocks are at least 50 clocks, internal circuits remain unsettled, and so pin states are also undefined. The system design must ensure that these undefined states do not cause erroneous system operation. When the CA pin is at a low level, the low level of the $\overline{\text{RESETP}}$ pin is not accepted.

  Waveforms at power-on are shown in the following figure.



Power-On Sequence

- Power-off order
  1. In the reverse order of powering-on, first turn off the 1.9 V power, then turn off the 3.3 V power within 1ms. This interval should be as short as possible.
  2. Pin states are undefined while only the 1.9 V power is off. The system design must ensure that these undefined states do not cause erroneous system operation.

24.2 DC Characteristics

Tables 24.2 and 24.3 list DC characteristics.

Table 24.2 DC Characteristics (Ta = –20 to 75°C)

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Power supply voltage		VccQ	3.0	3.3	3.6	V	
		Vcc, Vcc-PLL1, Vcc-PLL2, Vcc-RTC	1.75	1.90	2.05		
Current dissipation	Normal operation	Icc	—	250	400	mA	Vcc = 1.9 V, Iϕ = 133 MHz
		IccQ	—	20	40		VccQ = 3.3 V, Bϕ = 33 MHz
	In sleep mode*1	Icc	—	10	20		Bϕ = 33MHz
		IccQ	—	15	30		
	In standby mode	Icc	—	40	120	μA	Ta = 25°C (RTC on) VccQ = 3.3 V,
		IccQ	—	10	30		Vcc = 1.9 V
		Icc	—	35	110		Ta = 25°C (RTC off) VccQ = 3.3 V,
		IccQ	—	10	30		Vcc = 1.9 V
Input high voltage	$\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI, IRQ5– IRQ0, MD5–MD0, $\overline{\text{ASEMD0}}$, CA, $\overline{\text{ADTRG}}$, EXTAL, CKIO	V_{IH}	VccQ × 0.9	—	VccQ + 0.3	V	
	Port J		2.0	—	AVcc + 0.3		
	Other input pins		2.0	—	VccQ + 0.3		

Table 24.2 DC Characteristics (Ta = –20 to 75°C) (cont)

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input low voltage	$\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI, IRQ5–IRQ0, MD5–MD0, $\overline{\text{ASEMD0}}$, CA, $\overline{\text{ADTRG}}$, EXTAL, CKIO	V_{IL}	–0.3	—	VccQ × 0.1	V	
	Port J		–0.3	—	AVcc × 0.2		
	Other input pins		–0.3	—	VccQ × 0.2		
Input leak current	All input pins	I Iin I	—	—	1.0	µA	Vin = 0.5 to VccQ–0.5 V
Three-state leak current	I/O, all output pins (off condition)	I Isti I	—	—	1.0	µA	Vin = 0.5 to VccQ–0.5 V
Output high voltage	All output pins	V_{OH}	2.4	—	—	V	VccQ = 3.0 V, IOH = –200 µA
			2.0	—	—		VccQ = 3.0 V, IOH = –2 mA
Output low voltage	All output pins	V_{OL}	—	—	0.55		VccQ = 3.6 V, IOL = 1.6 mA
Pull-up resistance	Port pin	Ppull	30	60	120	kΩ	
Pin capacity	All pins	C	—	—	10	PF	
Analog power-supply voltage		AVcc	3.0	3.3	3.6	V	

Table 24.2 DC Characteristics (Ta = –20 to 75°C) (cont)

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Analog power-supply current	During A/D conversion	AIcc	—	0.8	2	mA	
	During A/D and D/A conversion		—	2.4	6	mA	
	Idle		—	0.01	5.0	μA	

Notes: 1. No external bus cycles except refresh cycles.

2. Regardless of whether PLL or RTC is used, connect Vcc – PLL and Vcc – RTC to Vcc, and Vss – PLL and Vss – RTC to Vss.

3. Conditions include software standby mode only.

4. AVcc must be under condition of VccQ – 0.3 V ≤ AVcc ≤ VccQ + 0.3 V. If the A/D and D/A converters are not used, do not leave the AVcc and AVss pins open. Connect AVcc to VccQ, and connect AVss to VssQ.

5. Current dissipation values shown are the values at which all output pins are without load
under conditions of VIHmin = VccQ – 0.5 V, VILmax = 0.5 V.

Table 24.3 Permitted Output Current Values
(VccQ = 3.3 ± 0.3 V, Vcc = 1.9 ± 0.15 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C)

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0	mA
Output low-level permissible current (total)	$\sum I_{OL}$	—	—	120	mA
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0	mA
Output high-level permissible current (total)	$\sum (-I_{OH})$	—	—	40	mA

Caution: To ensure LSI reliability, do not exceed the value for output current given in Table 24.3.

24.3 AC Characteristics

In general, inputting for this LSI should be clock synchronous. Keep the setup and hold times for each input signal unless otherwise specified.

Operating conditons are as follows:

VccQ=3.3±0.3V
Vcc=1.9±0.15V
AVcc=3.3±0.3V
Ta=–20 to 75°C

Table 24.4 Maximum Operating Frequencies

Item		Symbol	Min*	Typ	Max	Unit	Remarks
Operating frequency	CPU, cache, TLB	f	25	—	133.34	MHz	
	External bus		25	—	66.67		
	Peripheral module		6.25	—	33.34		

Note: * The Min value depends on the clock mode used. See table 10.3, Available Combinations of Clock Mode and FRQCR Values.

24.3.1 Clock Timing

**Table 24.5 Clock Timing (1)
(Maximum External Bus Operating Frequency: 25MHz)**

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	6.25	25	MHz	24.1
EXTAL clock input cycle time	t_{EXcyc}	40	160	ns	
EXTAL clock input low pulse width	t_{EXL}	8	—	ns	
EXTAL clock input high pulse width	t_{EXH}	8	—	ns	
EXTAL clock input rise time	t_{EXR}	—	4	ns	
EXTAL clock input fall time	t_{EXF}	—	4	ns	
CKIO clock input frequency	f_{CKI}	25	25	MHz	24.2
CKIO clock input cycle time	t_{CKIcyc}	40	40	ns	
CKIO clock input low pulse width	t_{CKIL}	8	—	ns	
CKIO clock input high pulse width	t_{CKIH}	8	—	ns	
CKIO clock input rise time	t_{CKIR}	—	4	ns	
CKIO clock input fall time	t_{CKIF}	—	4	ns	
CKIO clock output frequency	f_{OP}	25	25	MHz	24.3
CKIO clock output cycle time	t_{cyc}	40	40	ns	
CKIO clock output low pulse width	t_{CKOL}	20	—	ns	
CKIO clock output high pulse width	t_{CKOH}	20	—	ns	
CKIO clock output rise time	t_{CKOR}	—	7	ns	
CKIO clock output fall time	t_{CKOF}	—	7	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	24.4
$\overline{\text{RESETP}}$ setup time	t_{RESPS}	20	—	ns	24.4, 24.5
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	0	—	ns	
$\overline{\text{RESETP}}$ assert time	t_{RESPW}	20	—	tcyc	24.4, 24.5
$\overline{\text{RESETM}}$ assert time	t_{RESMW}	20	—	tcyc	
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	24.5
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	24.6
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	24.7

Table 24.5 Clock Timing (1) (cont)
(Maximum External Bus Operating Frequency: 25MHz)

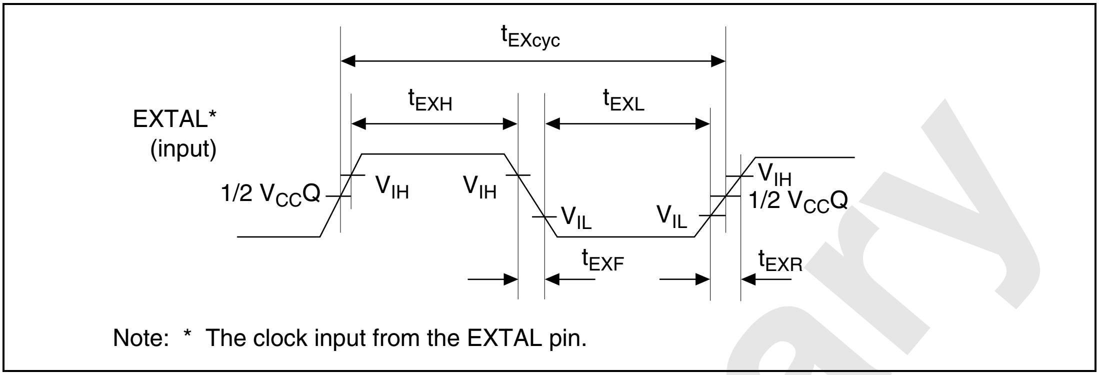
Item	Symbol	Min	Max	Unit	Figure
PLL synchronization settling time 1 (standby canceled)	t_{PLL1}	100	—	µs	24.8, 24.9
PLL synchronization settling time 2 (multiplication rate modified)	t_{PLL2}	100	—	µs	24.10
IRQ/IRL interrupt determination time (RTC used and standby mode)	t_{IRLSTB}	100	—	µs	24.10

Table 24.5 Clock Timing (2) (Maximum External Bus Operating Frequency: 33MHz)

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	6.25	33	MHz	24.1
EXTAL clock input cycle time	t_{EXcyc}	30.3	160	ns	
EXTAL clock input low pulse width	t_{EXL}	7	—	ns	
EXTAL clock input high pulse width	t_{EXH}	7	—	ns	
EXTAL clock input rise time	t_{EXR}	—	4	ns	
EXTAL clock input fall time	t_{EXF}	—	4	ns	
CKIO clock input frequency	f_{CKI}	25	33	MHz	24.2
CKIO clock input cycle time	t_{CKIcyc}	30.3	40	ns	
CKIO clock input low pulse width	t_{CKIL}	7	—	ns	
CKIO clock input high pulse width	t_{CKIH}	7	—	ns	
CKIO clock input rise time	t_{CKIR}	—	3	ns	
CKIO clock input fall time	t_{CKIF}	—	3	ns	
CKIO clock output frequency	f_{OP}	25	33	MHz	24.3
CKIO clock output cycle time	t_{cyc}	30.3	40	ns	
CKIO clock output low pulse width	t_{CKOL}	8	—	ns	
CKIO clock output high pulse width	t_{CKOH}	8	—	ns	
CKIO clock output rise time	t_{CKOR}	—	6	ns	
CKIO clock output fall time	t_{CKOF}	—	6	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	24.4
$\overline{\text{RESETP}}$ setup time	t_{RESPS}	20	—	ns	24.4, 24.5
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	0	—	ns	
$\overline{\text{RESETP}}$ assert time	t_{RESPW}	20	—	tcyc	
$\overline{\text{RESETM}}$ assert time	t_{RESMW}	20	—	tcyc	
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	24.5
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	24.6
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	24.7
PLL synchronization settling time 1 (standby canceled)	t_{PLL1}	100	—	µs	24.8, 24.9
PLL synchronization settling time 2 (multiplication rete modified)	t_{PLL2}	100	—	µs	24.10
IRQ/IRL interrupt determination time (RTC used and standby mode)	t_{IRLSTB}	100	—	µs	24.10

Table 24.5 Clock Timing (3) (Maximum External Bus Operating Frequency: 66 MHz)

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	6.25	66	MHz	24.1
EXTAL clock input cycle time	t_{EXcyc}	15.2	160	ns	
EXTAL clock input low pulse width	t_{EXL}	4	—	ns	
EXTAL clock input high pulse width	t_{EXH}	4	—	ns	
EXTAL clock input rise time	t_{EXR}	—	2	ns	
EXTAL clock input fall time	t_{EXF}	—	2	ns	
CKIO clock input frequency	f_{CKI}	25	66	MHz	24.2
CKIO clock input cycle time	t_{CKIcyc}	15.2	40	ns	
CKIO clock input low pulse width	t_{CKIL}	4	—	ns	
CKIO clock input high pulse width	t_{CKIH}	4	—	ns	
CKIO clock input rise time	t_{CKIR}	—	2	ns	
CKIO clock input fall time	t_{CKIF}	—	2	ns	
CKIO clock output frequency	f_{OP}	25	66	MHz	24.3
CKIO clock output cycle time	t_{cyc}	15.2	40	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	24.4
$\overline{\text{RESETP}}$ setup time	t_{RESPS}	20	—	ns	24.4, 24.5
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	0	—	ns	
$\overline{\text{RESETP}}$ assert time	t_{RESPW}	20	—	tcyc	
$\overline{\text{RESETM}}$ assert time	t_{RESMW}	20	—	tcyc	
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	24.5
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	24.6
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	24.7
PLL synchronization settling time 1 (standby canceled)	t_{PLL1}	100	—	µs	24.8, 24.9
PLL synchronization settling time 2 (multiplication rete modified)	t_{PLL2}	100	—	µs	24.10
IRQ/IRL interrupt determination time (RTC used and standby mode)	t_{IRLSTB}	100	—	µs	24.10



Note: * The clock input from the EXTAL pin.

Figure 24.1 EXTAL Clock Input Timing

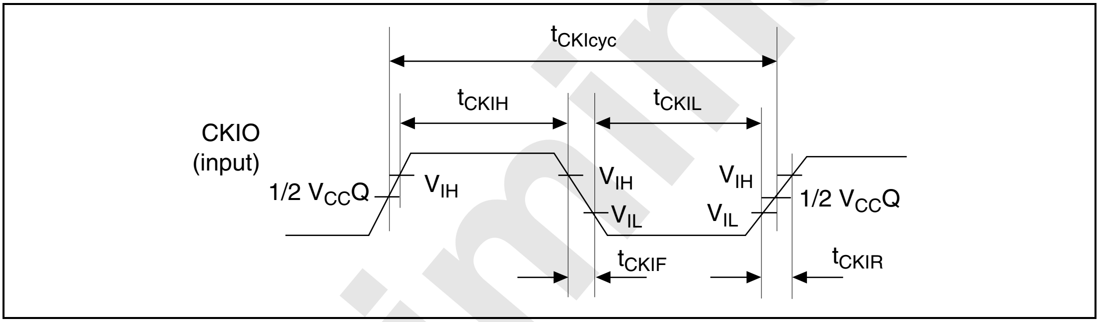


Figure 24.2 CKIO Clock Input Timing

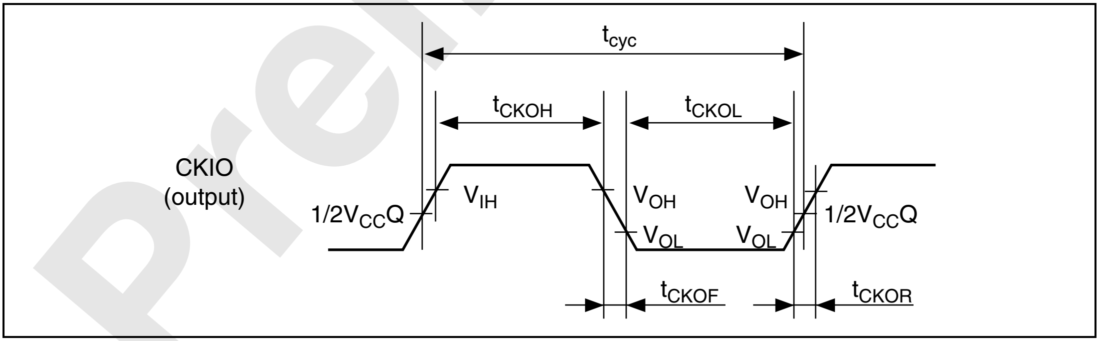


Figure 24.3 CKIO Clock Output Timing

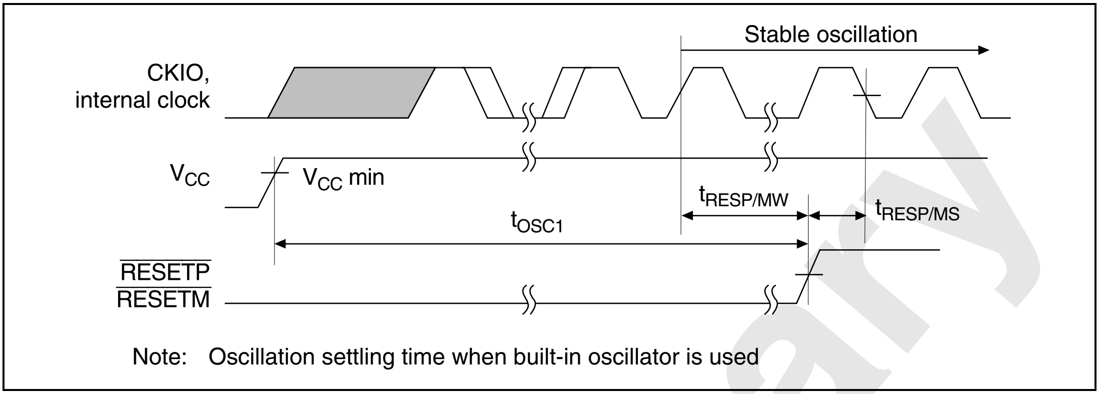


Figure 24.4 Power-on Oscillation Settling Time

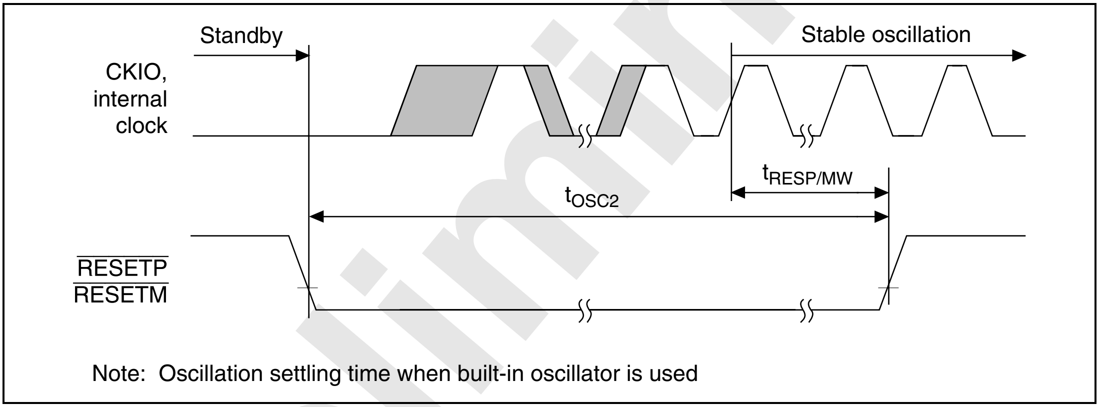


Figure 24.5 Oscillation Settling Time at Standby Return (Return by Reset)

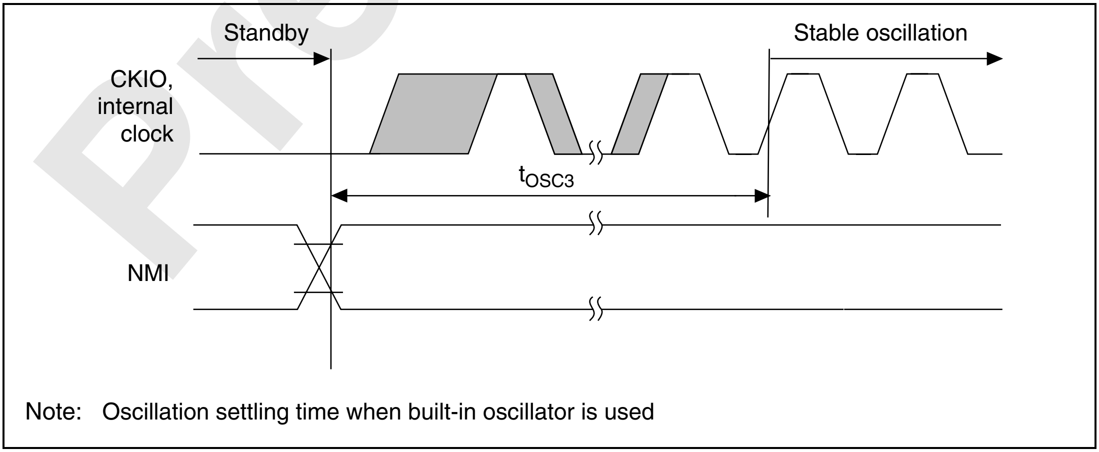
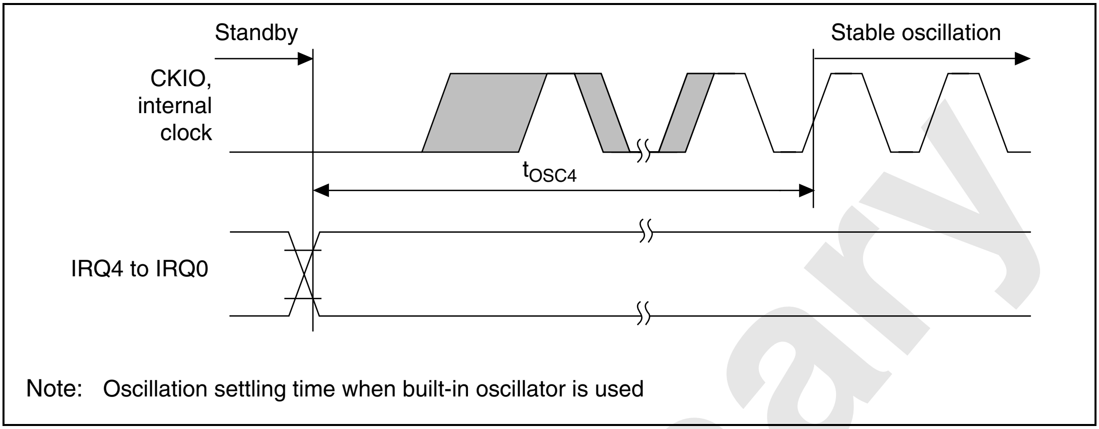
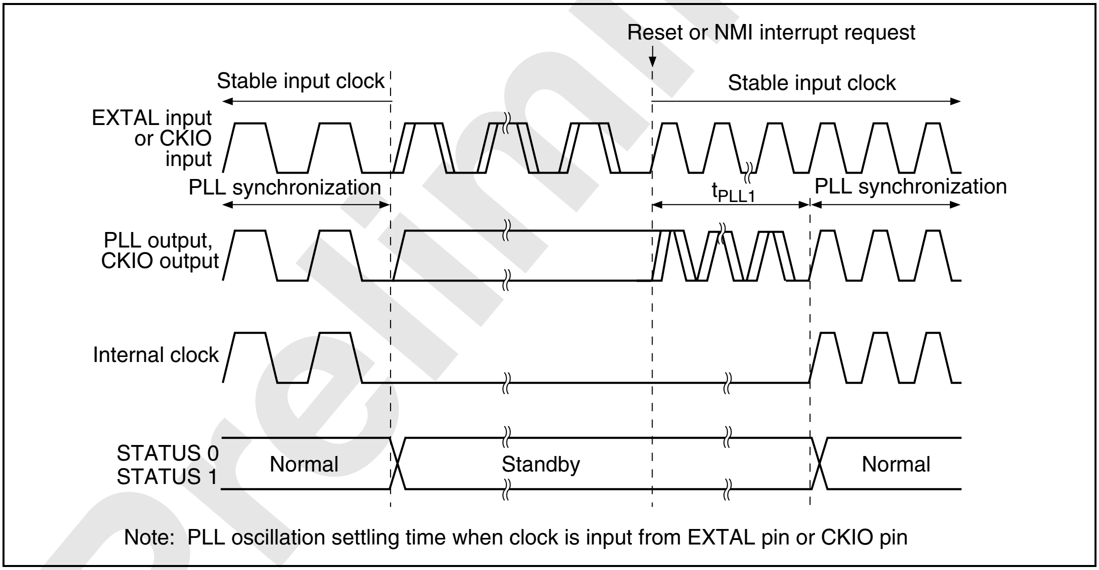

Standby
Stable oscillation
CKIO,
internal
clock
t_{OSC3}
NMI
Note: Oscillation settling time when built-in oscillator is used

Figure 24.6 Oscillation Settling Time at Standby Return (Return by NMI)



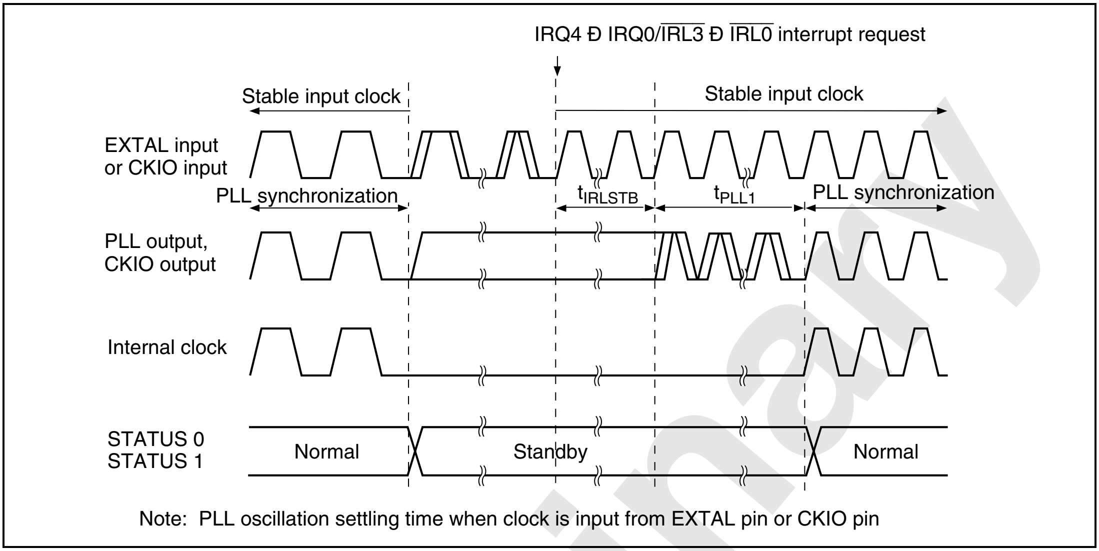
Note: Oscillation settling time when built-in oscillator is used

Figure 24.7 Oscillation Settling Time at Standby Return (Return by IRQ4 to IRQ0)



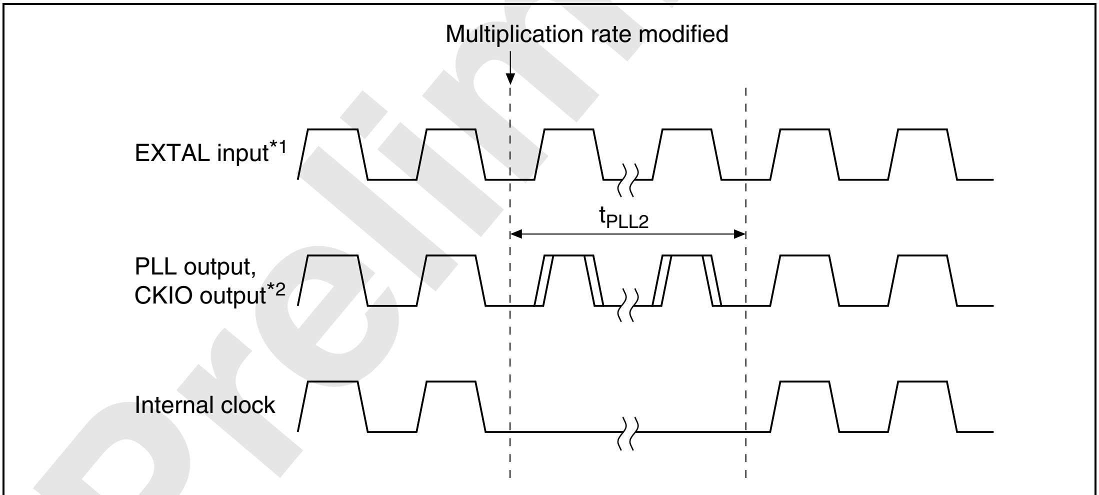
Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin

Figure 24.8 PLL Synchronization Settling Time by Reset or NMI



Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin

Figure 24.9 PLL Synchronization Settling Time by IRQ/IRL Interrupt



Notes: *1 CKIO input in clock mode 7
*2 PLL output in clock mode 7

Figure 24.10 PLL Synchronization Settling Time when Frequency Multiplication Rate Modified

24.3.2 Control Signal Timing

Table 24.6 Control Signal Timing

		–66*2			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
$\overline{\text{RESETP}}$ pulse width	t_{RESPW}	20 *3	—	tcyc	24.11, 24.12
$\overline{\text{RESETP}}$ setup time*1	t_{RESPS}	23	—	ns	
$\overline{\text{RESETP}}$ hold time	t_{RESPH}	2	—	ns	
$\overline{\text{RESETM}}$ pulse width	t_{RESMW}	12 *4	—	tcyc	
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	6	—	ns	
$\overline{\text{RESETM}}$ hold time	t_{RESMH}	34	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	6	—	ns	24.14
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	4	—	ns	
NMI setup time *1	t_{NMIS}	10	—	ns	24.12, 24.13
NMI hold time	t_{NMIH}	4	—	ns	
IRQ5–IRQ0 setup time *1	t_{IRQS}	10	—	ns	
IRQ5–IRQ0 hold time	t_{IRQH}	4	—	ns	
$\overline{\text{IRQOUT}}$ delay time	t_{IRQOD}	—	10	ns	
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	10	ns	24.14, 24.15
STATUS1, STATUS0 delay time	t_{STD}	—	10	ns	
Bus tri-state delay time 1	t_{BOFF1}	0	15	ns	
Bus tri-state delay time 2	t_{BOFF2}	0	15	ns	
Bus buffer-on time 1	t_{BON1}	0	15	ns	
Bus buffer-on time 2	t_{BON2}	0	15	ns	

Notes: *1. $\overline{\text{RESETP}}$, NMI, and IRQ5 to IRQ0 are asynchronous. Changes are detected at the clock fall when the setup shown is used. When the setup cannot be used, detection can be delayed until the next clock falls.

*2. The upper limit of the external bus clock is 66 MHz.

*3. In the standby mode, $t_{RESPW} = t_{OSC2}$ (10 ms). In the sleep mode, $t_{RESPW} = t_{PLL1}$ (100 μs). When the clock multiplication ratio is changed, $t_{RESPW} = t_{PLL1}$ (100 μs).

*4. In the standby mode, $t_{RESMW} = t_{OSC2}$ (10 ms). In the sleep mode, $\overline{\text{RESETM}}$ must be kept low until $\overline{\text{STATUS (0-1}}$) changes to reset (HH). When the clock multiplication ratio is changed, $\overline{\text{RESETM}}$ must be kept low until STATUS (0-1) changes to reset (HH).

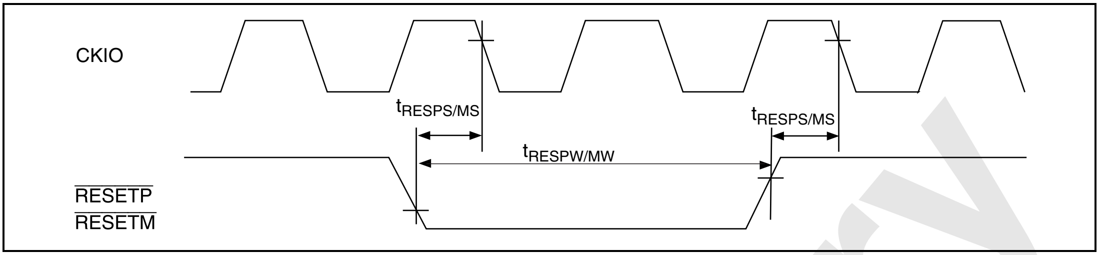


Figure 24.11 Reset Input Timing

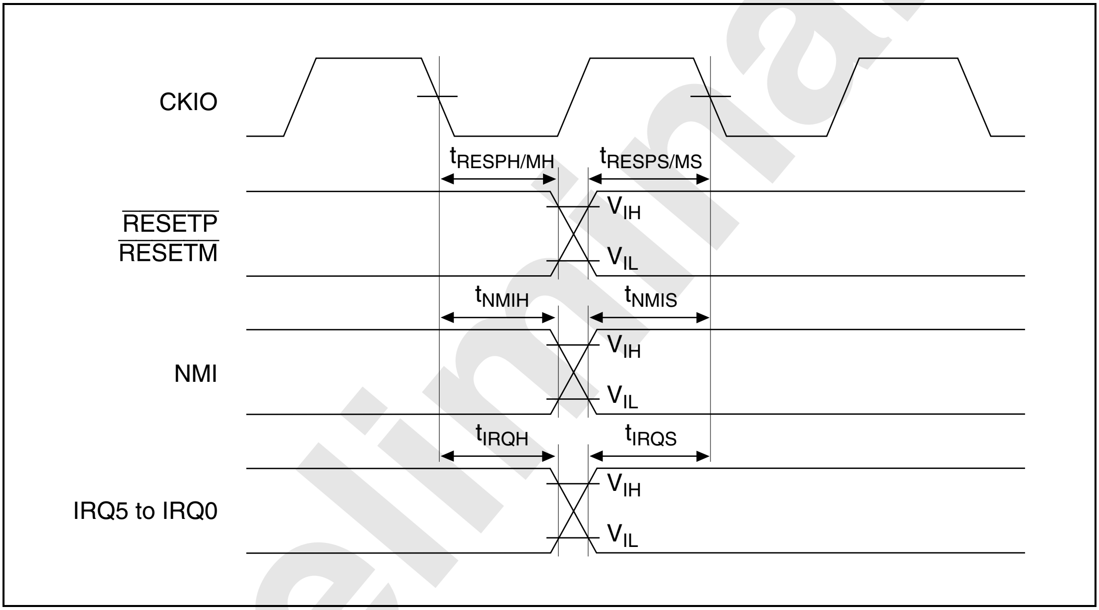


Figure 24.12 Interrupt Signal Input Timing

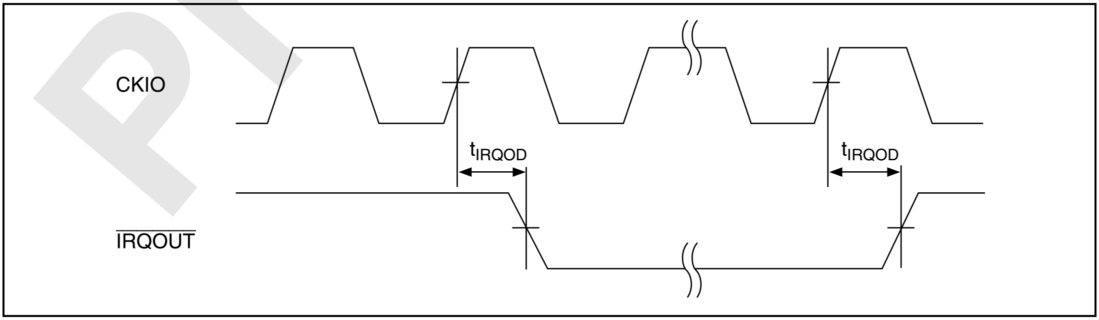


Figure 24.13 $\overline{\text{IRQOUT}}$ Timing

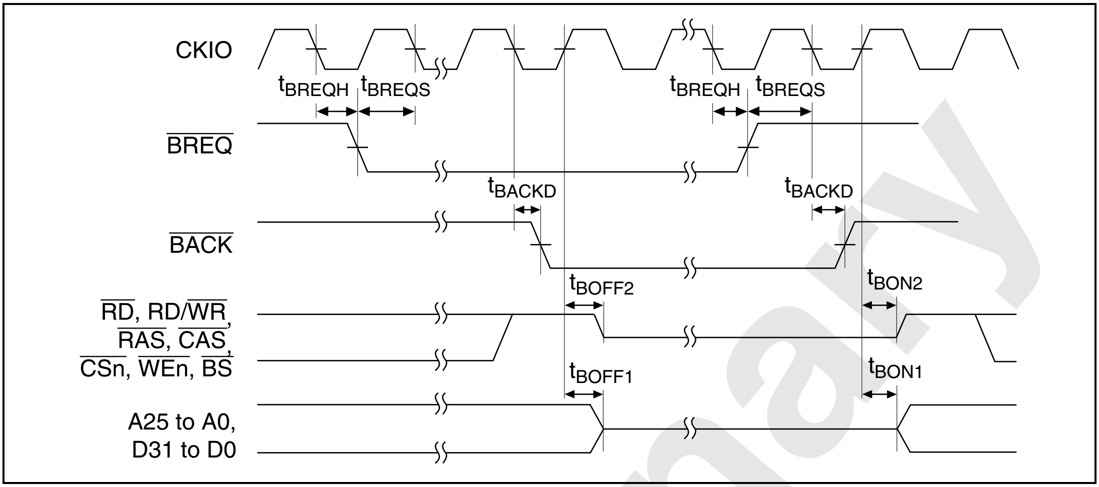


Figure 24.14 Bus Release Timing

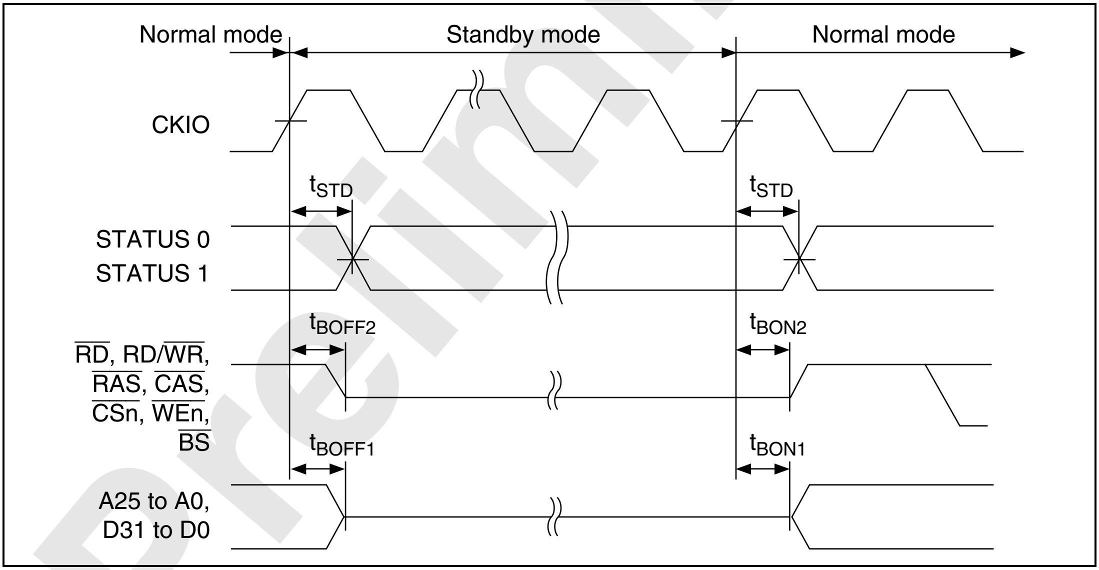


Figure 24.15 Pin Drive Timing at Standby

24.3.3 AC Bus Timing

Table 24.7 Bus Timing
(Clock Modes 0/1/2/7)

		–66*			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
Address delay time	t_{AD}	1.5	12	ns	24.16–24.36, 24.39–24.46
Address setup time	t_{AS}	0	—	ns	24.16–24.18
Address hold time	t_{AH}	4	—	ns	24.16–24.21
$\overline{\text{BS}}$ delay time	t_{BSD}	—	10	ns	24.16–24.36, 24.40–24.46
$\overline{\text{CS}}$ delay time 1	t_{CSD1}	—	10	ns	24.16–24.21, 24.40–24.46
$\overline{\text{CS}}$ delay time 2	t_{CSD2}	—	10	ns	24.16–24.21
$\overline{\text{CS}}$ delay time 3	t_{CSD3}	1.5	10	ns	24.24–24.39
Read/write delay time	t_{RWD}	1.5	10	ns	24.16–24.46
Read/write hold time	t_{RWH}	0	—	ns	24.16–24.21
Read strobe delay time	t_{RSD}	—	10	ns	24.16–24.21 24.40–24.43
Read data setup time 1	t_{RDS1}	6	—	ns	24.16–24.21, 24.40–24.46
Read data setup time 2	t_{RDS2}	5	—	ns	24.22–24.25, 24.30–24.33
Read data hold time 1	t_{RDH1}	0	—	ns	24.16–24.21, 24.40–24.46
Read data hold time 2	t_{RDH2}	1	—	ns	24.22–24.25, 24.30–24.33
Write enable delay time	t_{WED}	—	10	ns	24.16–22.18, 24.40–24.41
Write data delay time 1	t_{WDD1}	—	14	ns	24.16–24.18, 24.40–24.41, 24.44–24.46
Write data delay time 2	t_{WDD2}	1.5	12	ns	24.20–24.29
Write data hold time 1	t_{WDH1}	1.5	—	ns	24.16–24.18, 24.40–24.41, 24.44–24.46
Write data hold time 2	t_{WDH2}	1.5	—	ns	24.26–24.29
Write data hold time 3	t_{WDH3}	2	—	ns	24.16–24.18
Write data hold time 4	t_{WDH4}	2	—	ns	24.40–24.41, 24.44–24.46
$\overline{\text{WAIT}}$ setup time	t_{WTS}	5	—	ns	24.17–24.21, 24.41, 24.43, 24.45, 24.46
$\overline{\text{WAIT}}$ hold time	t_{WTH}	0	—	ns	24.17–24.21, 24.41, 24.43, 24.45, 24.46
$\overline{\text{RAS}}$ delay time	t_{RASD}	1	10	ns	24.22–24.39
$\overline{\text{CAS}}$ delay time	t_{CASD}	1	10	ns	24.22–24.39
$\overline{\text{DQM}}$ delay time	t_{DQMD}	1.5	10	ns	24.22–24.36

**Table 24.7 Bus Timing (cont)
(Clock Modes 0/1/2/7)**

		–66*			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
CKE delay time	t_{CKED}	1.5	10	ns	22.38
$\overline{\text{ICIORD}}$ delay time	t_{ICRSD}	—	10	ns	22.44–22.46
$\overline{\text{ICIOWR}}$ delay time	t_{ICWSD}	—	10	ns	22.44–22.46
$\overline{\text{IOIS16}}$ setup time	t_{IO16S}	6	—	ns	22.45, 22.46
$\overline{\text{IOIS16}}$ hold time	t_{IO16H}	4	—	ns	22.45, 22.46
DACK delay time 1	t_{DAKD1}	—	12	ns	22.16–22.36, 22.39–22.46
DACK delay time 2	t_{DAKD2}	—	10	ns	22.16–22.18, 22.20–22.21

Note: * The upper limit of the external bus clock is 66 MHz.

24.3.4 Basic Timing

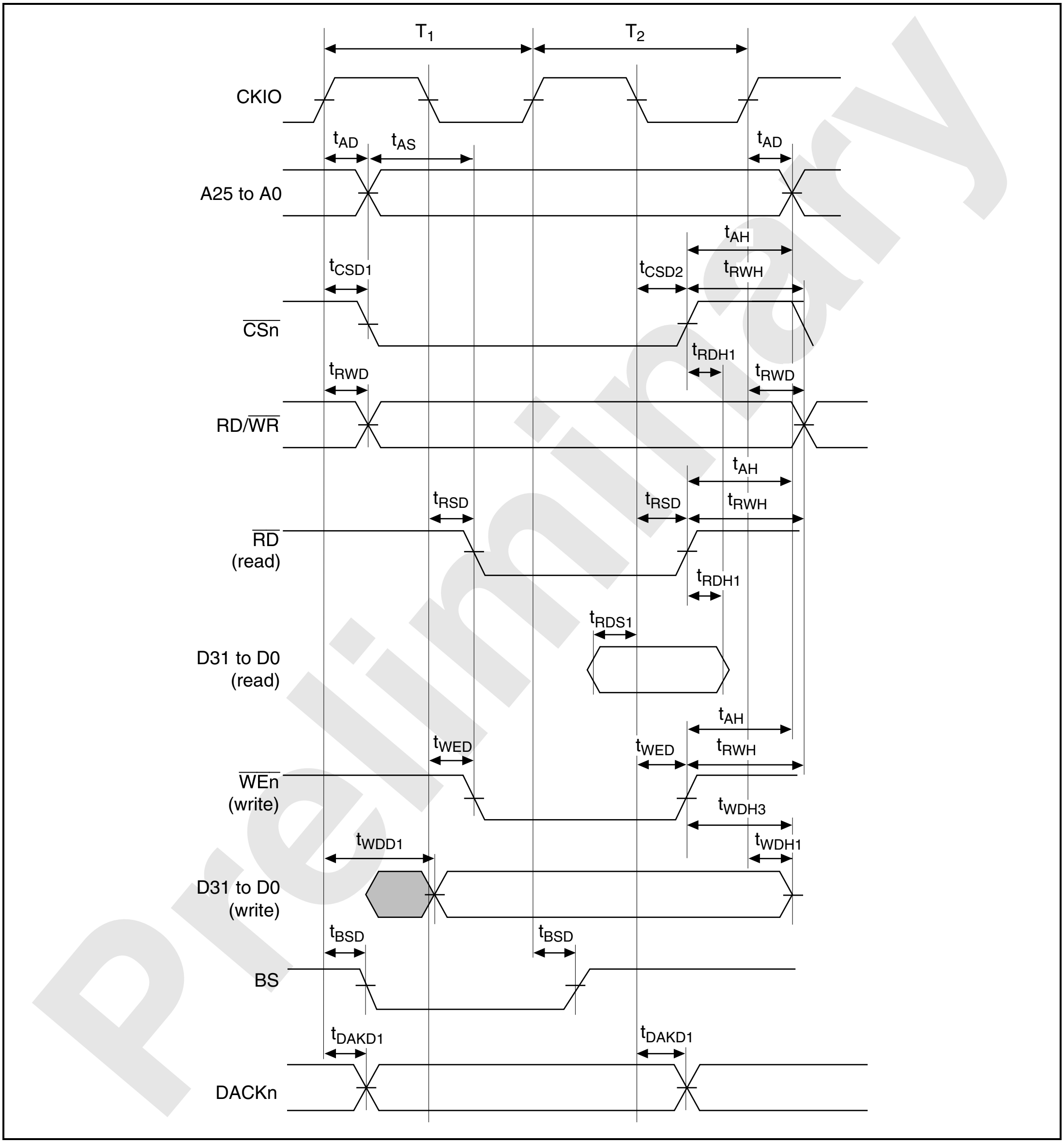


Figure 24.16 Basic Bus Cycle (No Wait)

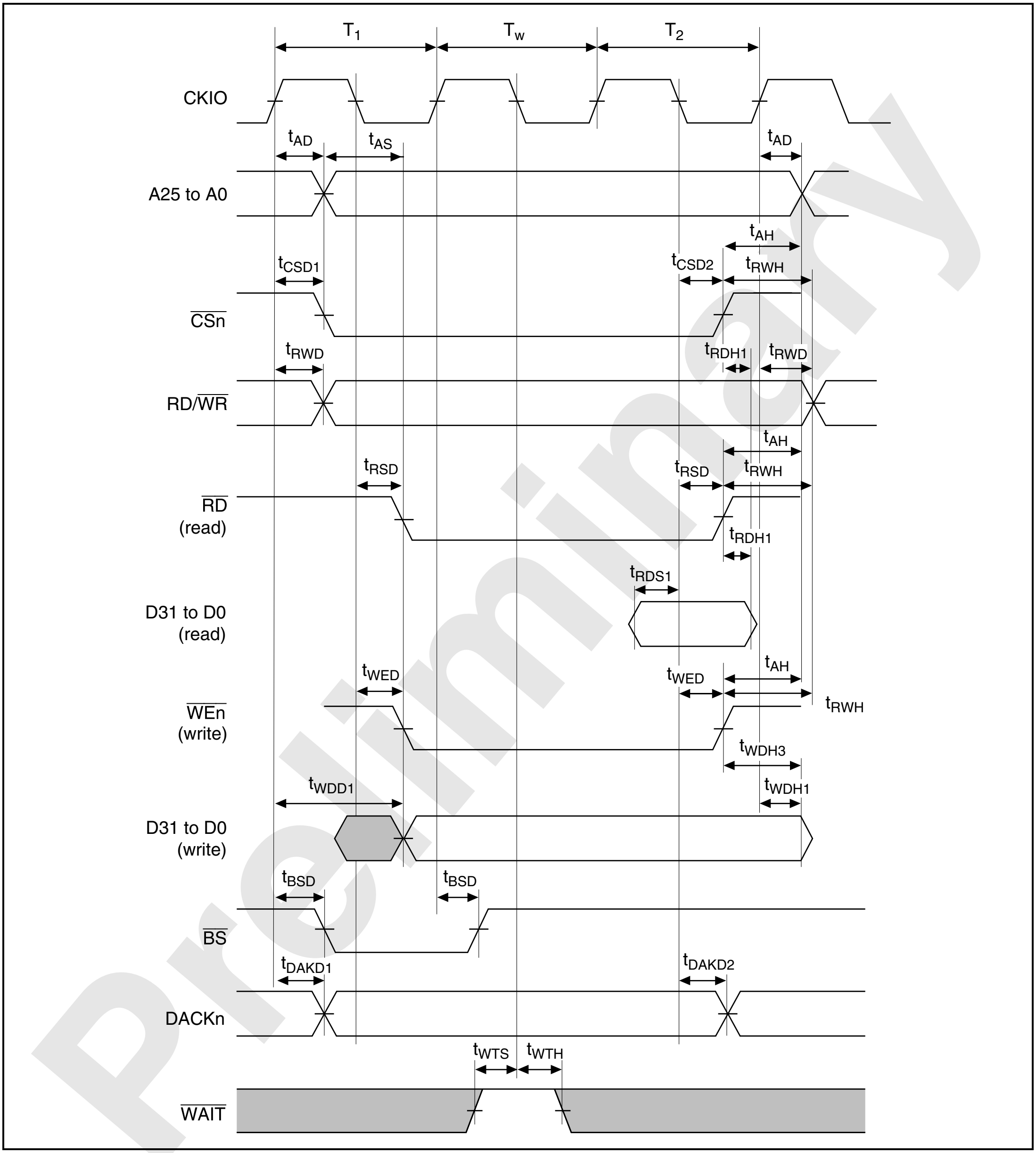


Figure 24.17 Basic Bus Cycle (One Wait)

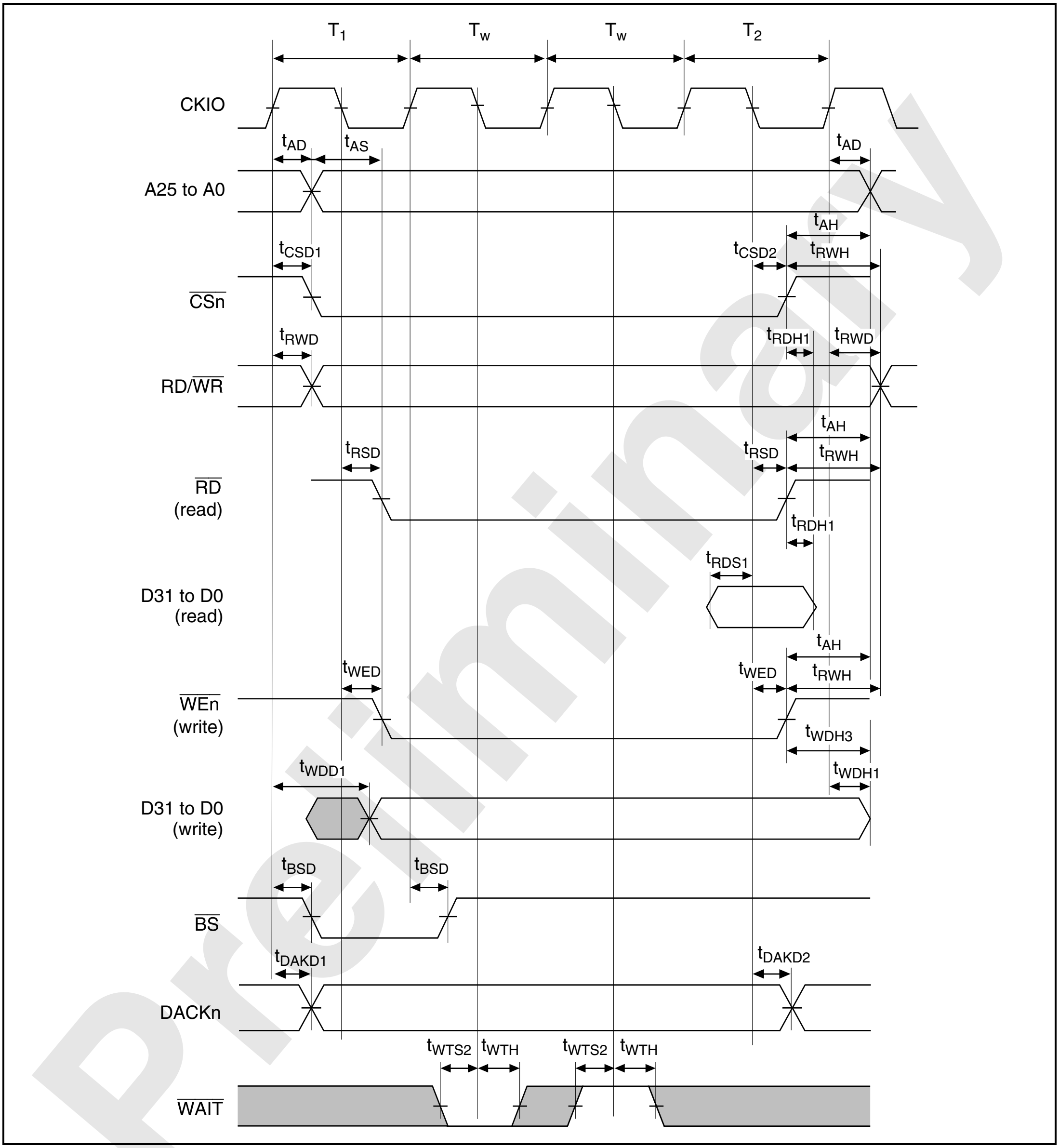
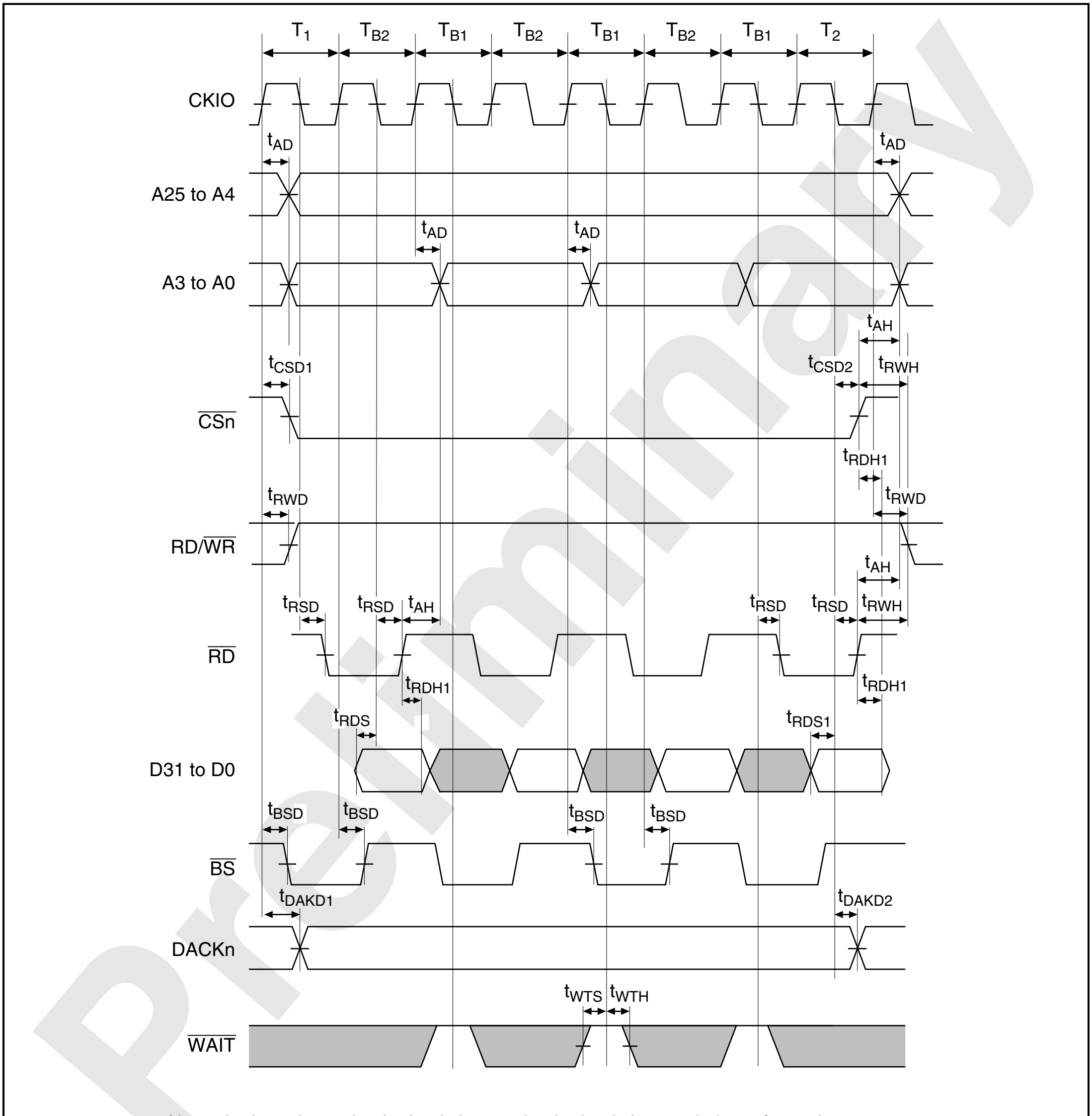


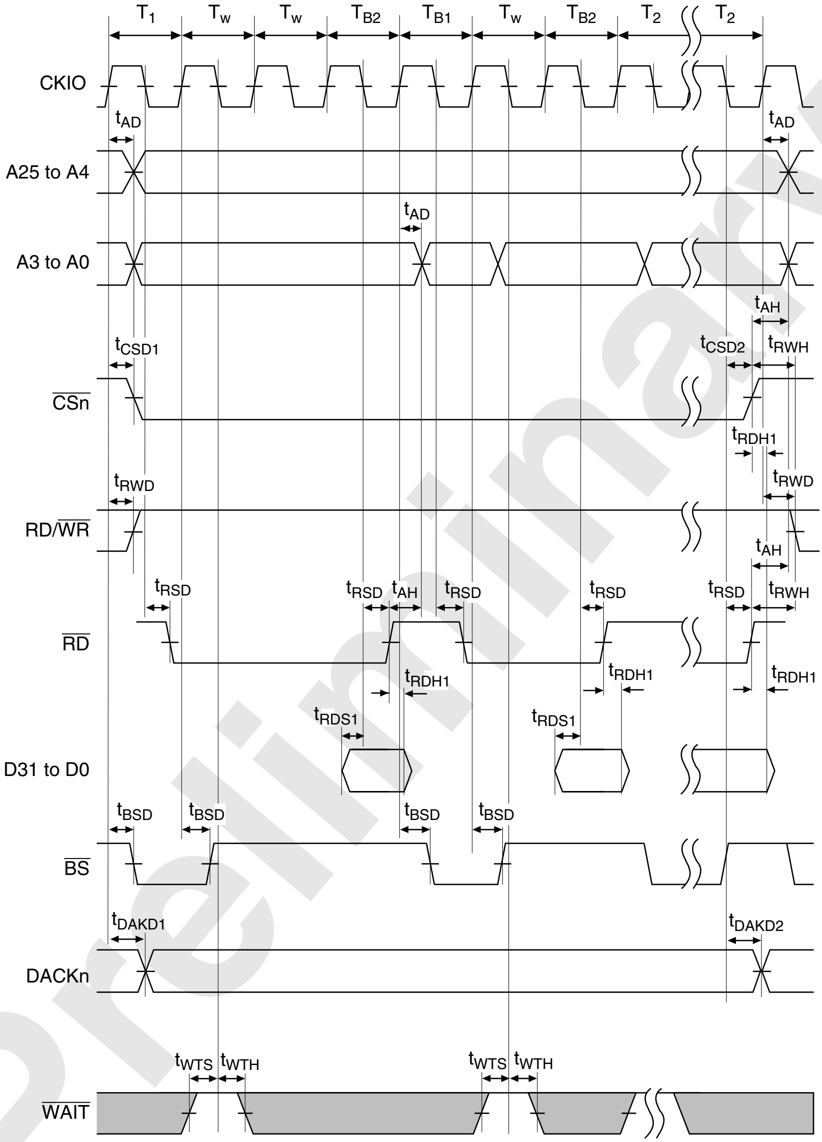
Figure 24.18 Basic Bus Cycle (External Wait)

24.3.5 Burst ROM Timing



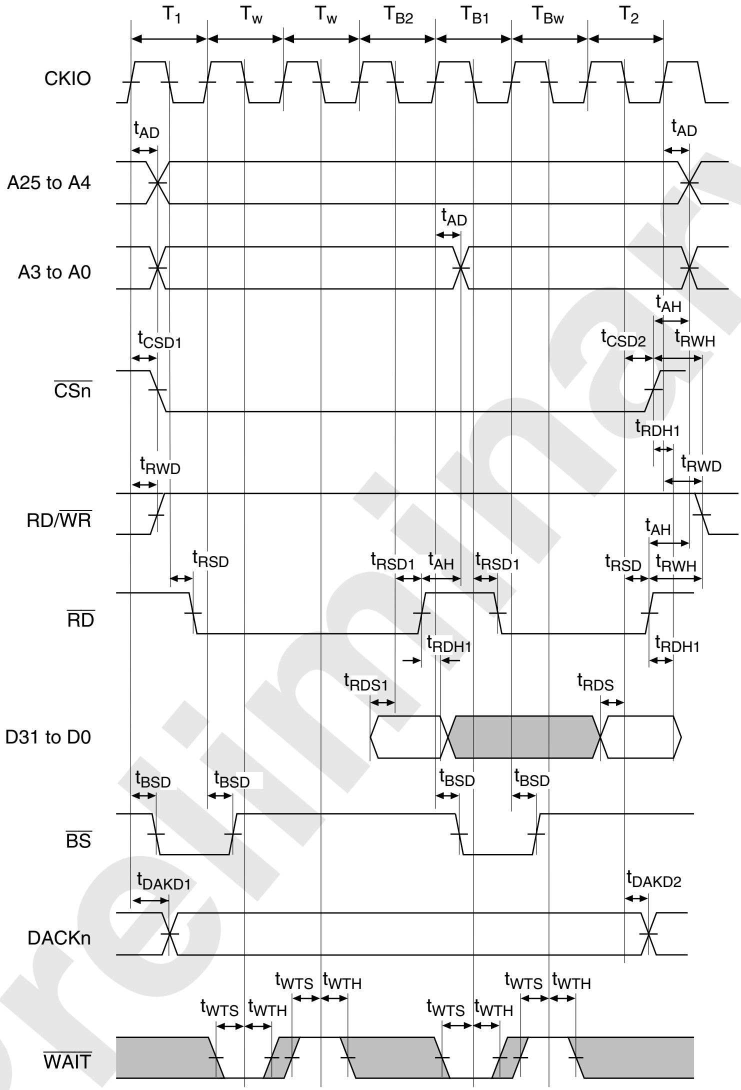
Note: In the write cycle, the basic bus cycle, the basic bus cycle is performed.

Figure 24.19 Burst ROM Bus Cycle (No Wait)



Note: In the write cycle, the basic bus cycle is performed.

Figure 24.20 Burst ROM Bus Cycle (Two Waits)



Note: In the write cycle, the basec bus cycle is performed.

Figure 24.21 Burst ROM Bus Cycle (External Wait)

24.3.6 Synchronous DRAM Timing

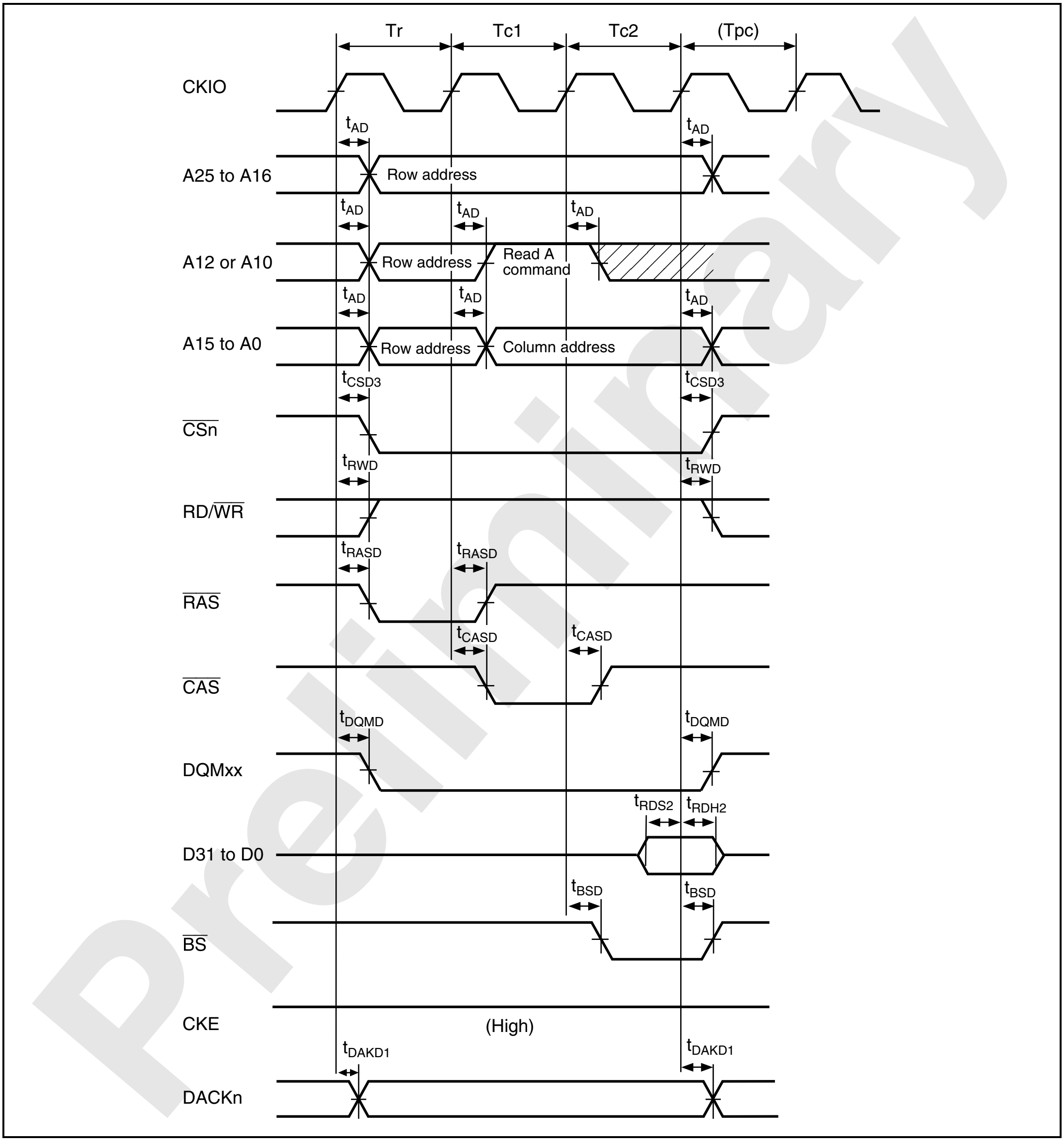


Figure 24.22 Synchronous DRAM Read Bus Cycle (RCD = 0, CAS Latency = 1, TPC = 0)

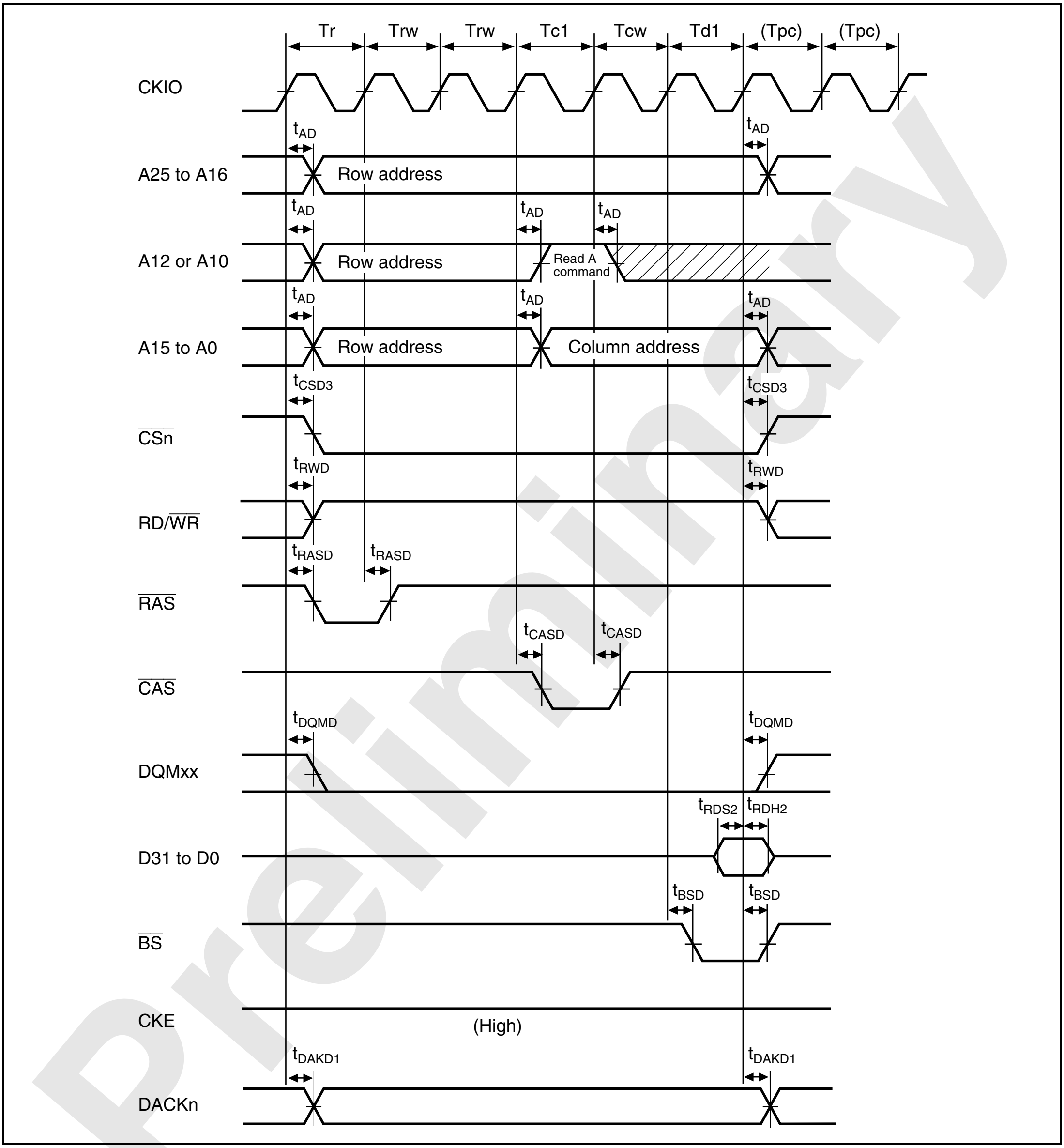


Figure 24.23 Synchronous DRAM Read Bus Cycle (RCD = 2, CAS Latency = 2, TPC = 1)

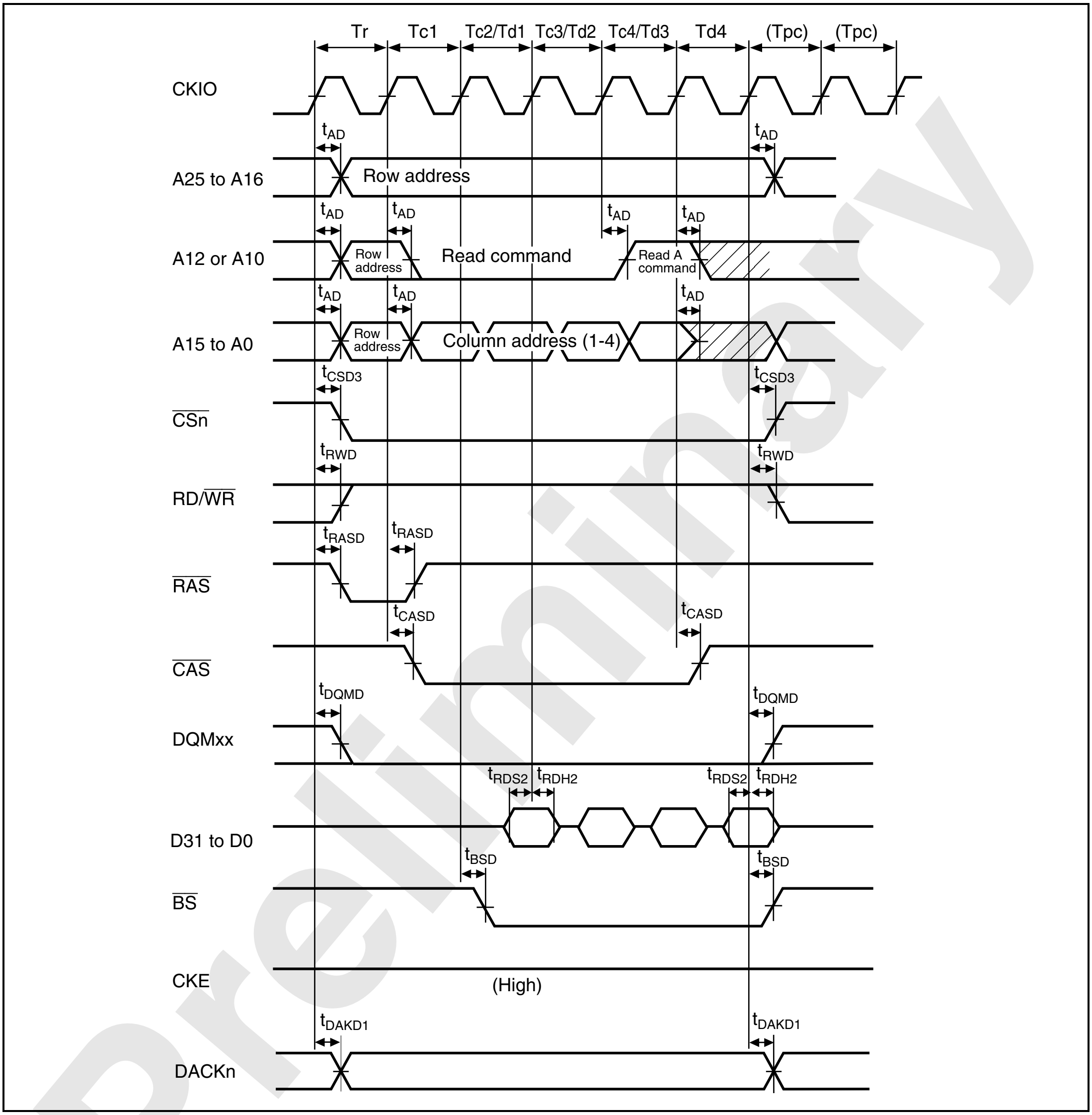


Figure 24.24 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 0, CAS Latency = 1, TPC = 1)

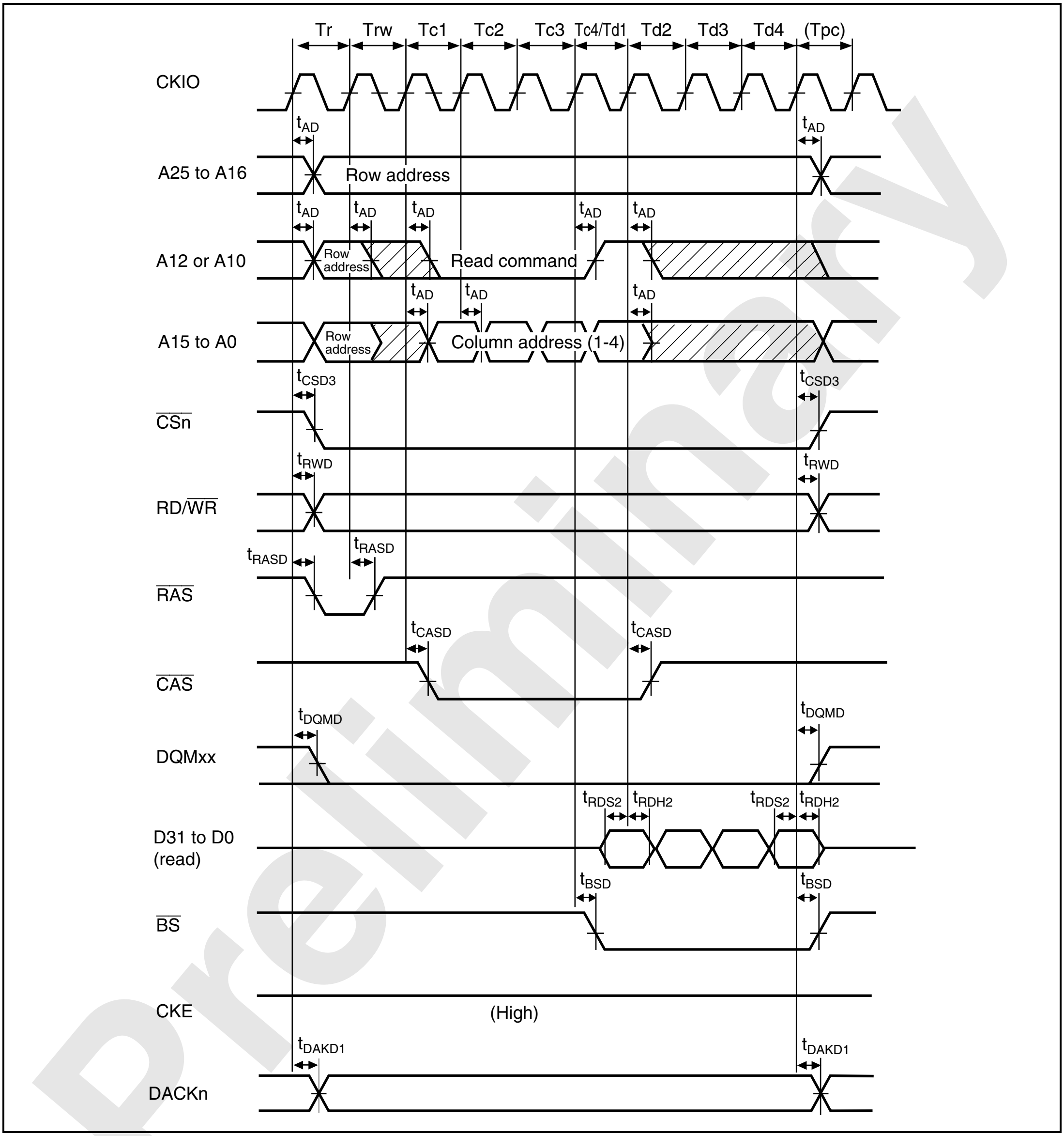


Figure 24.25 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 1, CAS Latency = 3, TPC = 0)

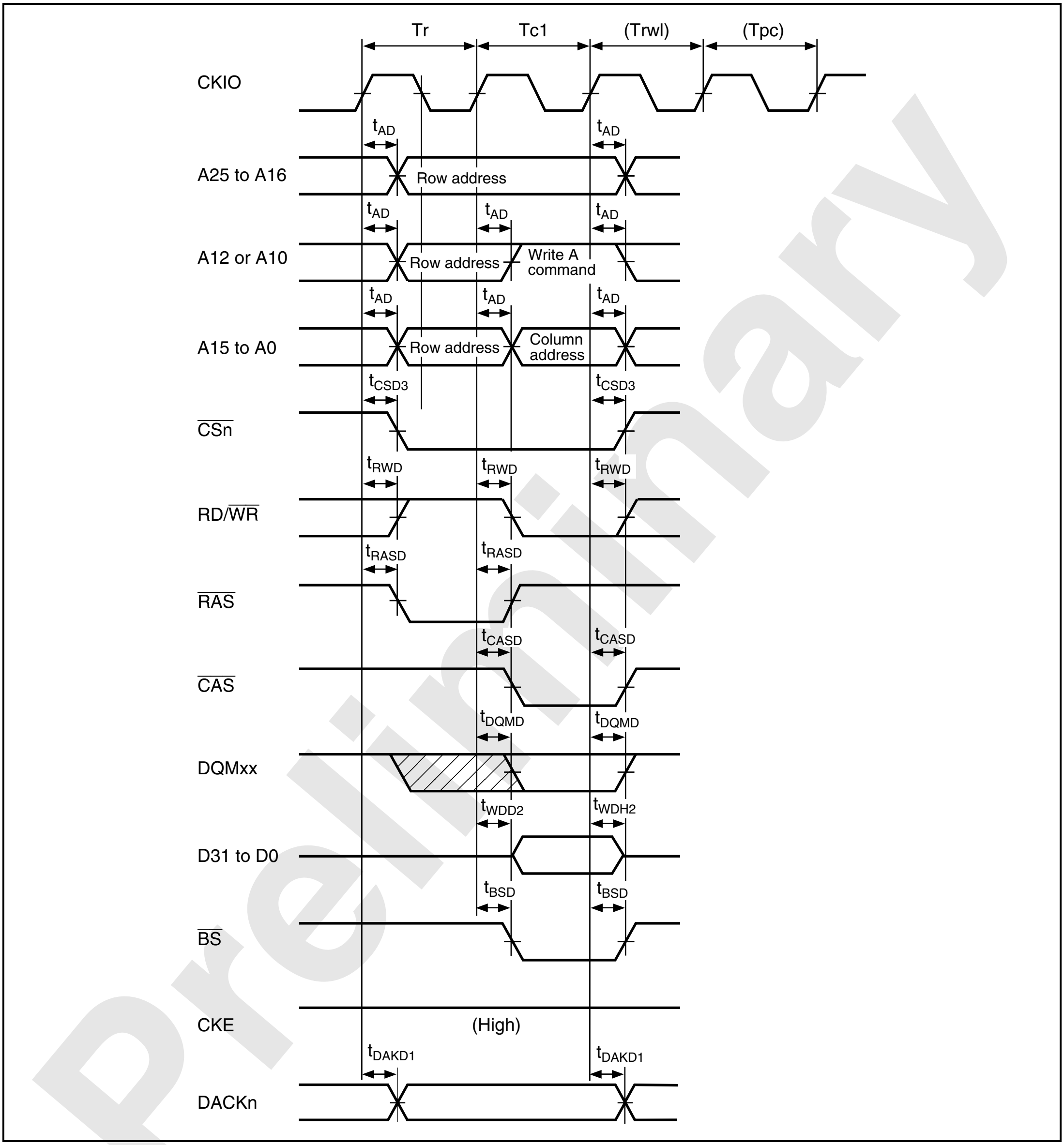


Figure 24.26 Synchronous DRAM Write Bus Cycle (RCD = 0, TPC = 0, TRWL = 0)

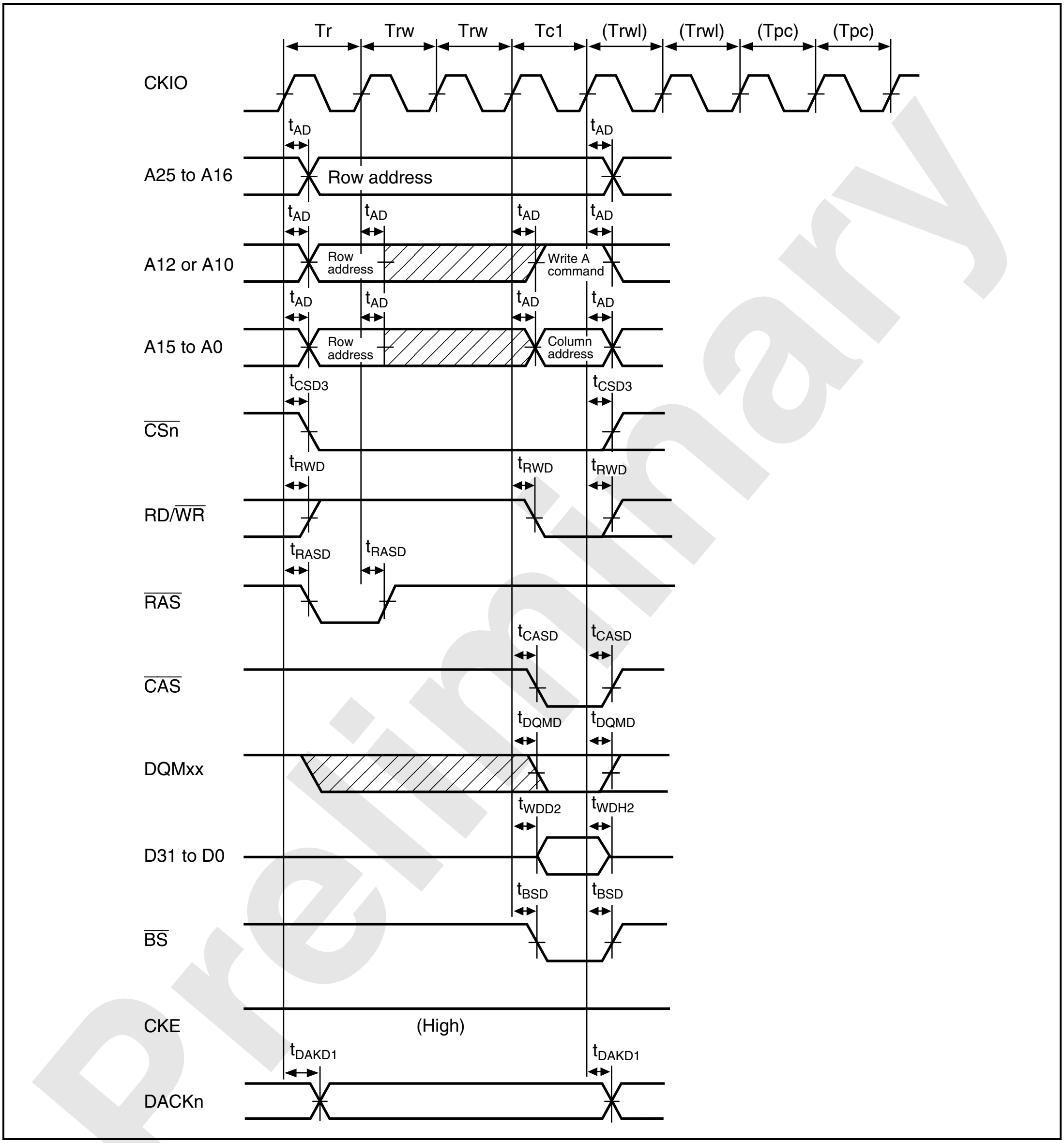


Figure 24.27 Synchronous DRAM Write Bus Cycle (RCD = 2, TPC = 1, TRWL = 1)

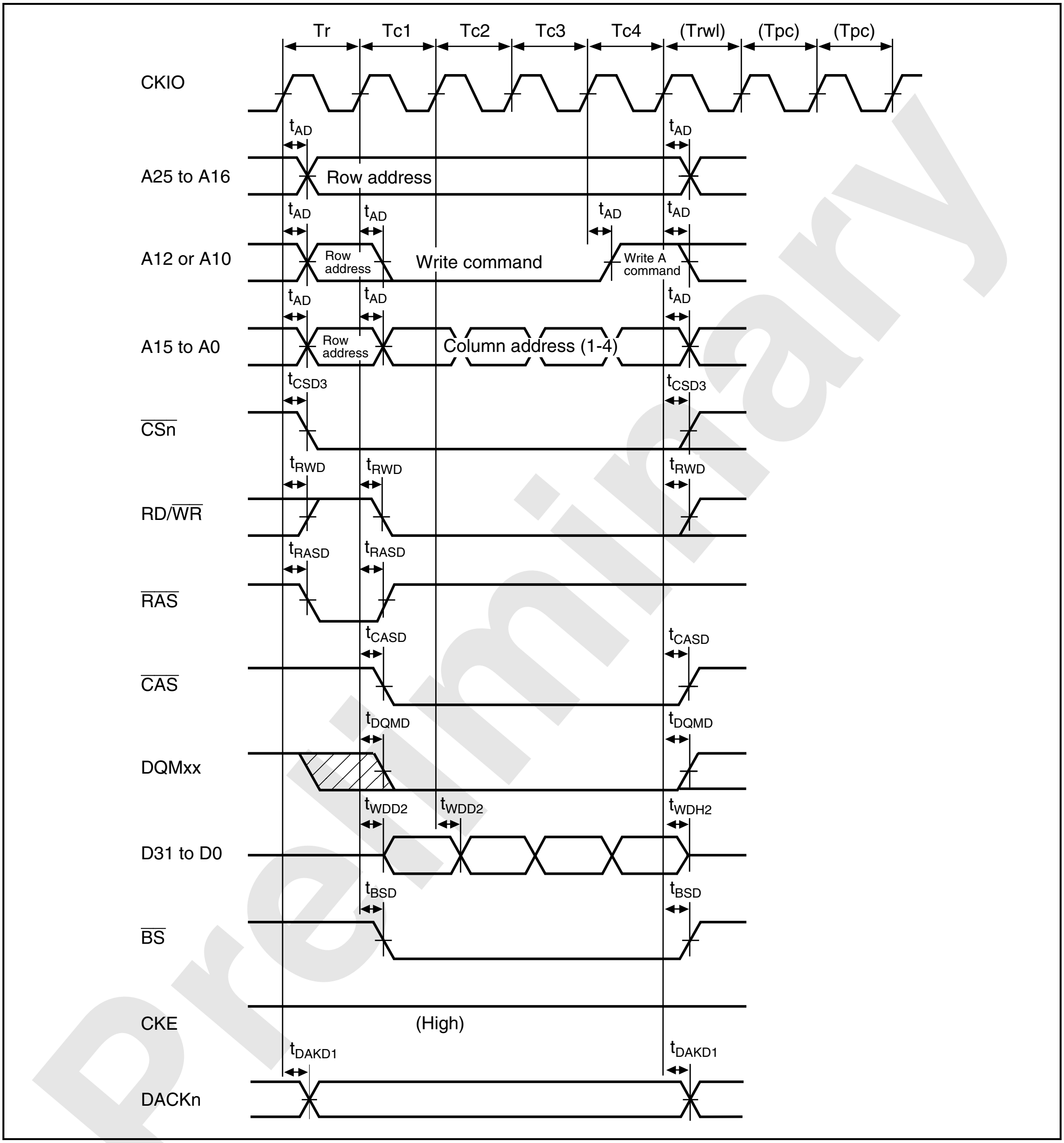


Figure 24.28 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 0, TPC = 1, TRWL = 0)

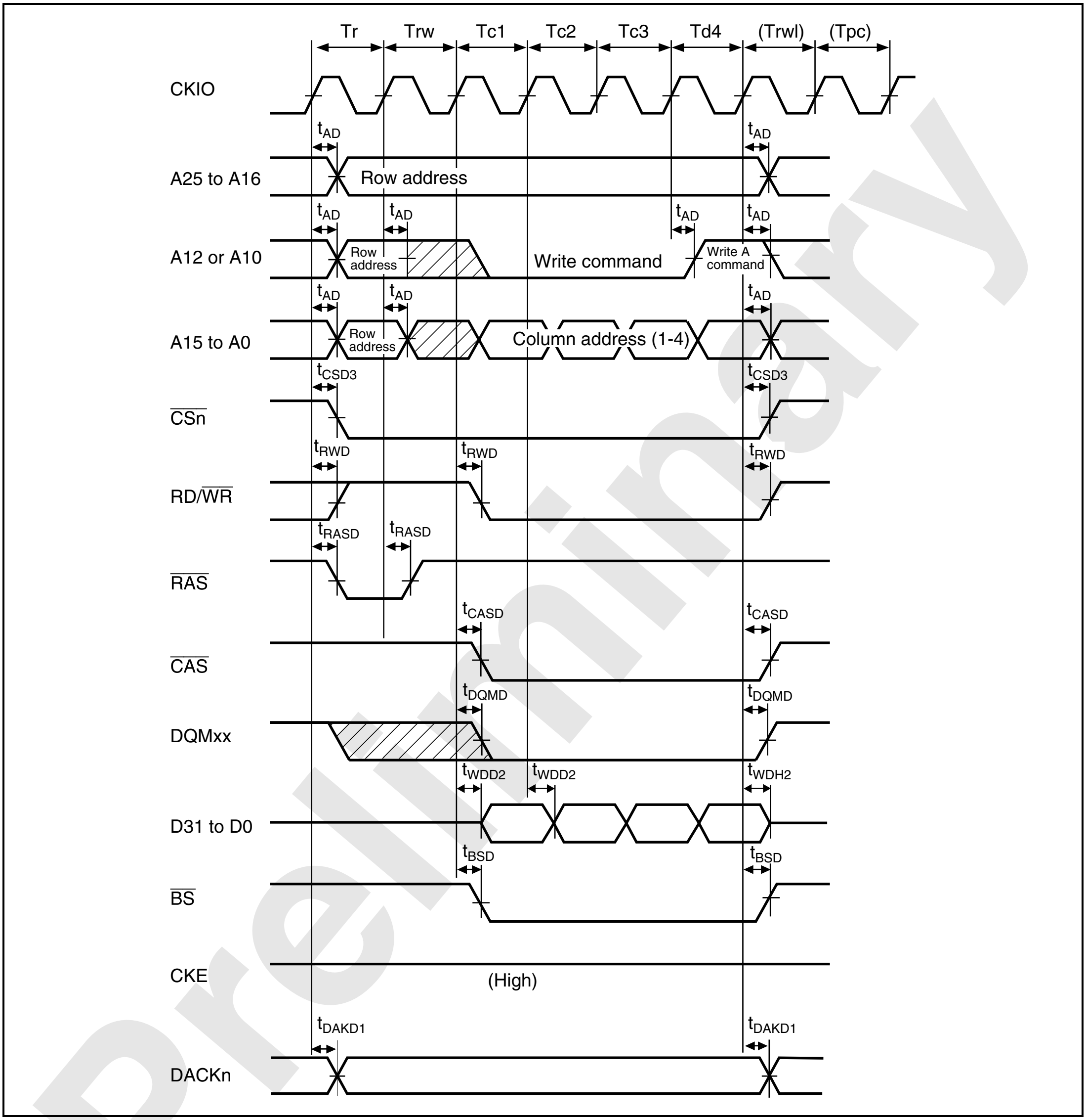


Figure 24.29 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 1, TPC = 0, TRWL = 0)

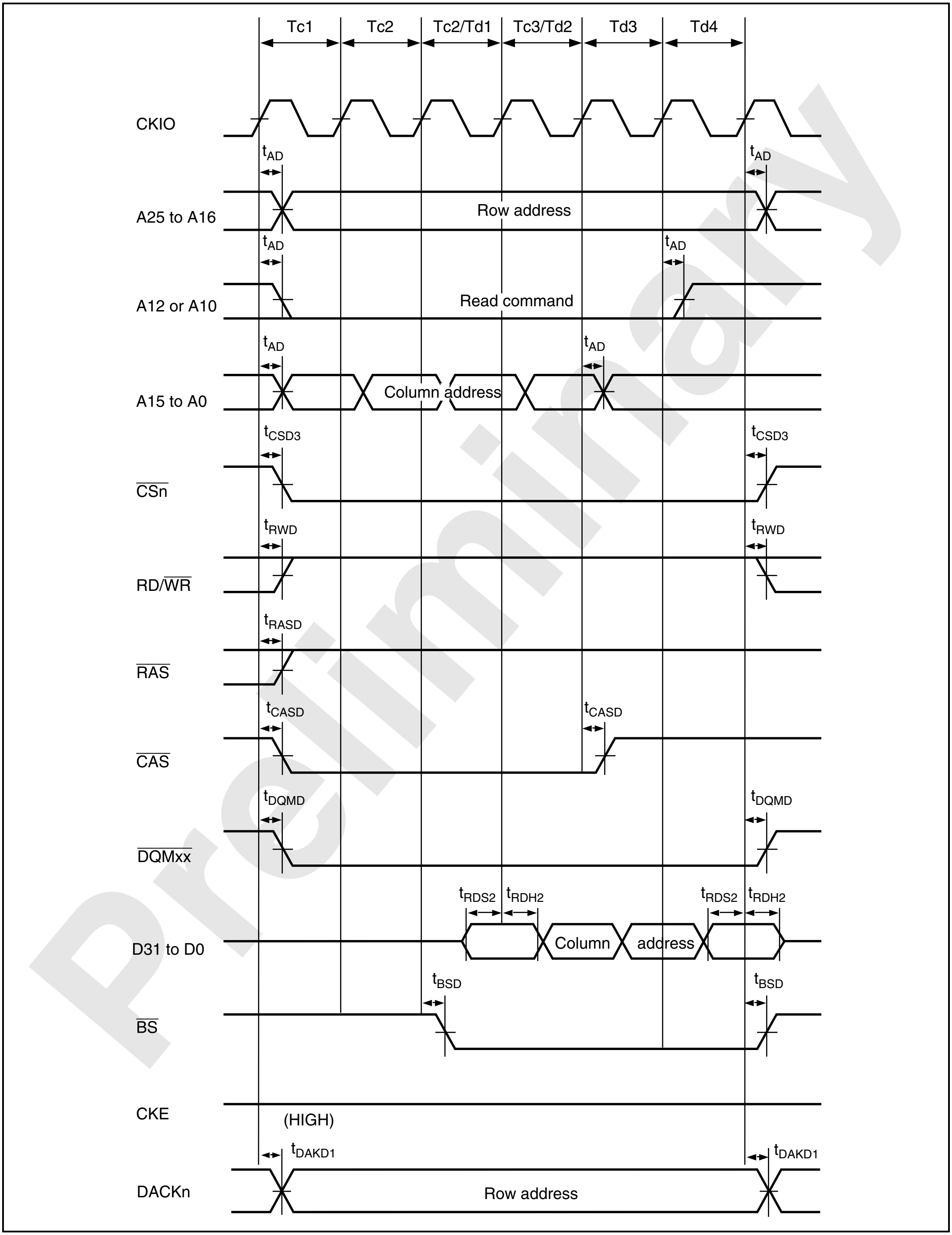


Figure 24.30 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Same Row Address, CAS Latency = 1)

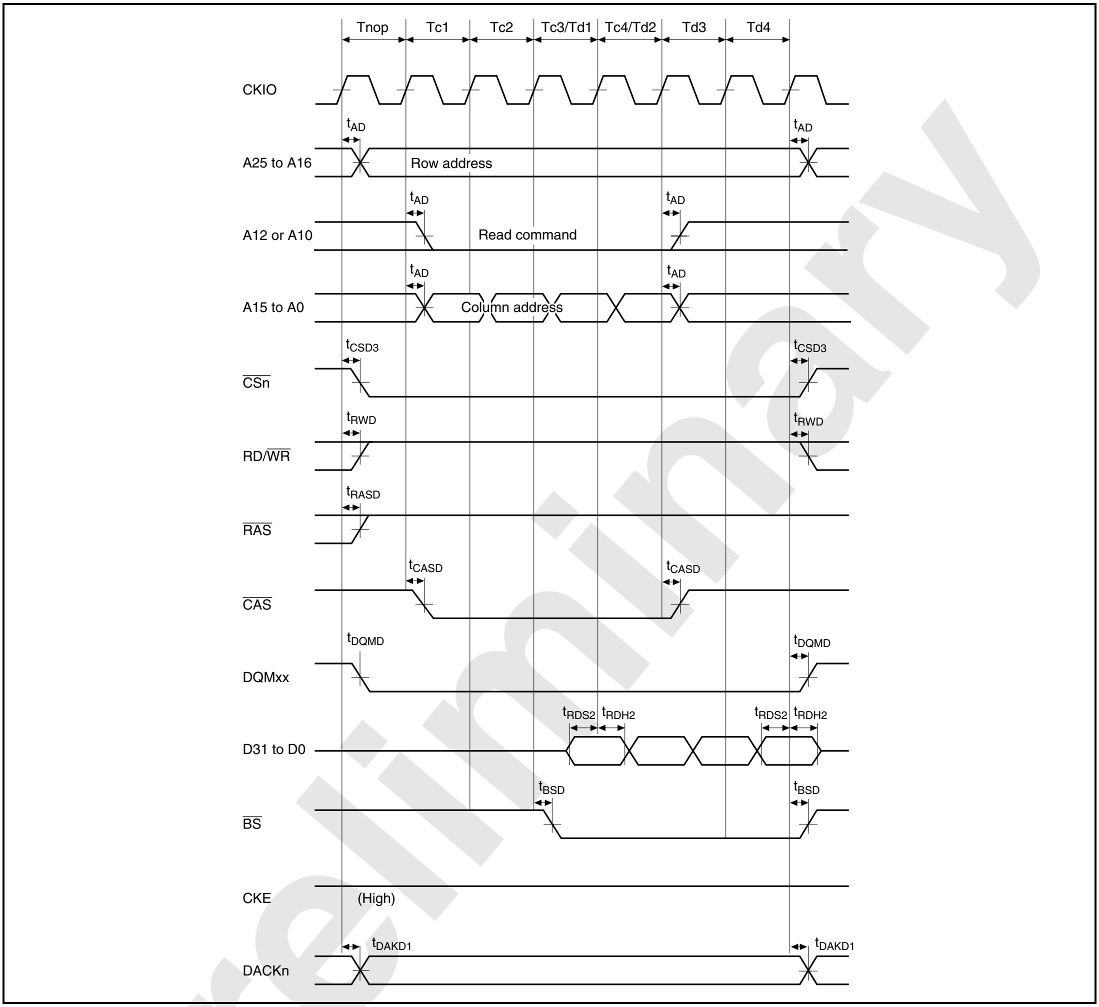


Figure 24.31 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Same Row Address, CAS Latency = 2)

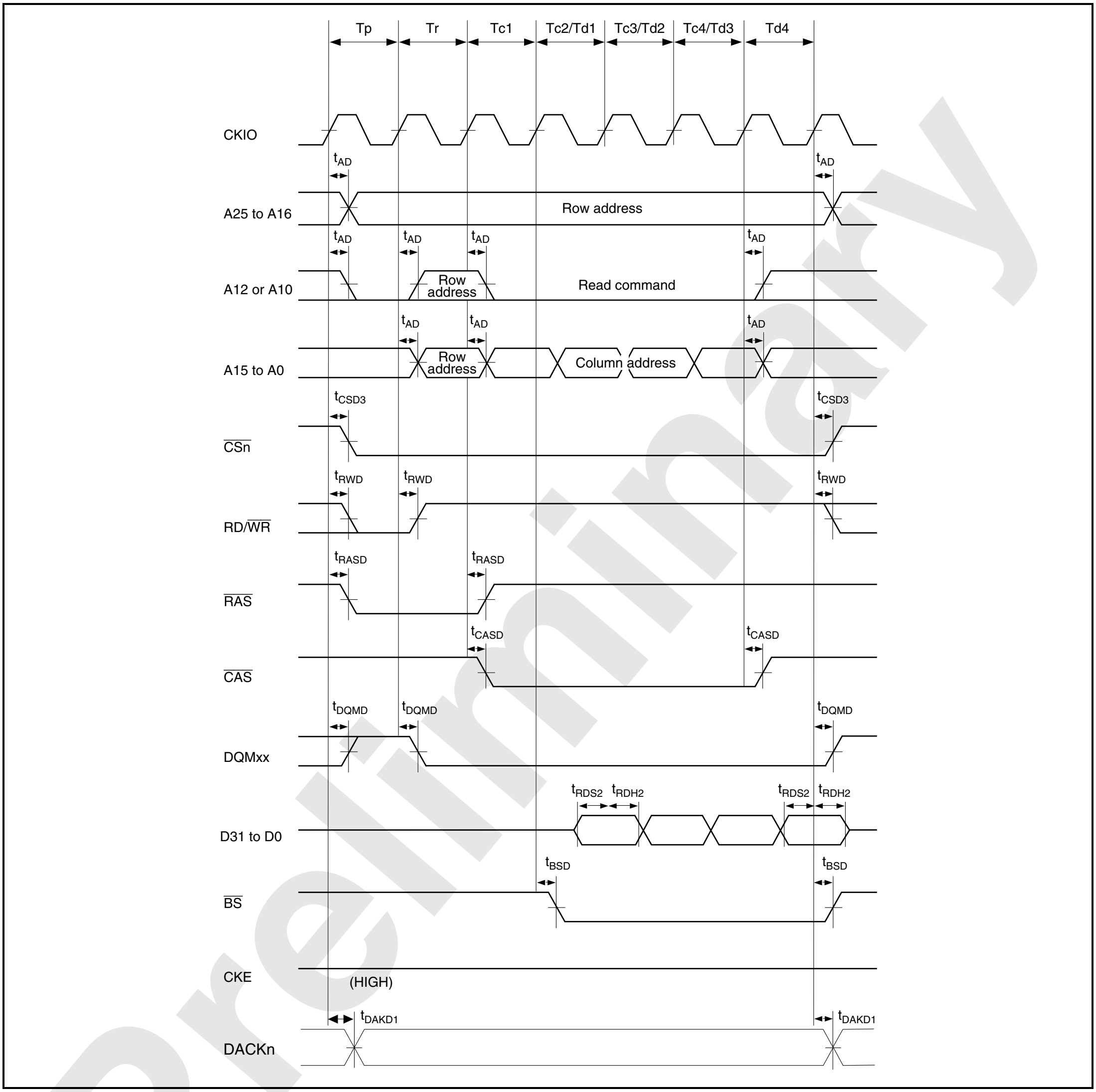


Figure 24.32 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Different Row Address, TPC = 0, RCD = 0, CAS Latency = 1)

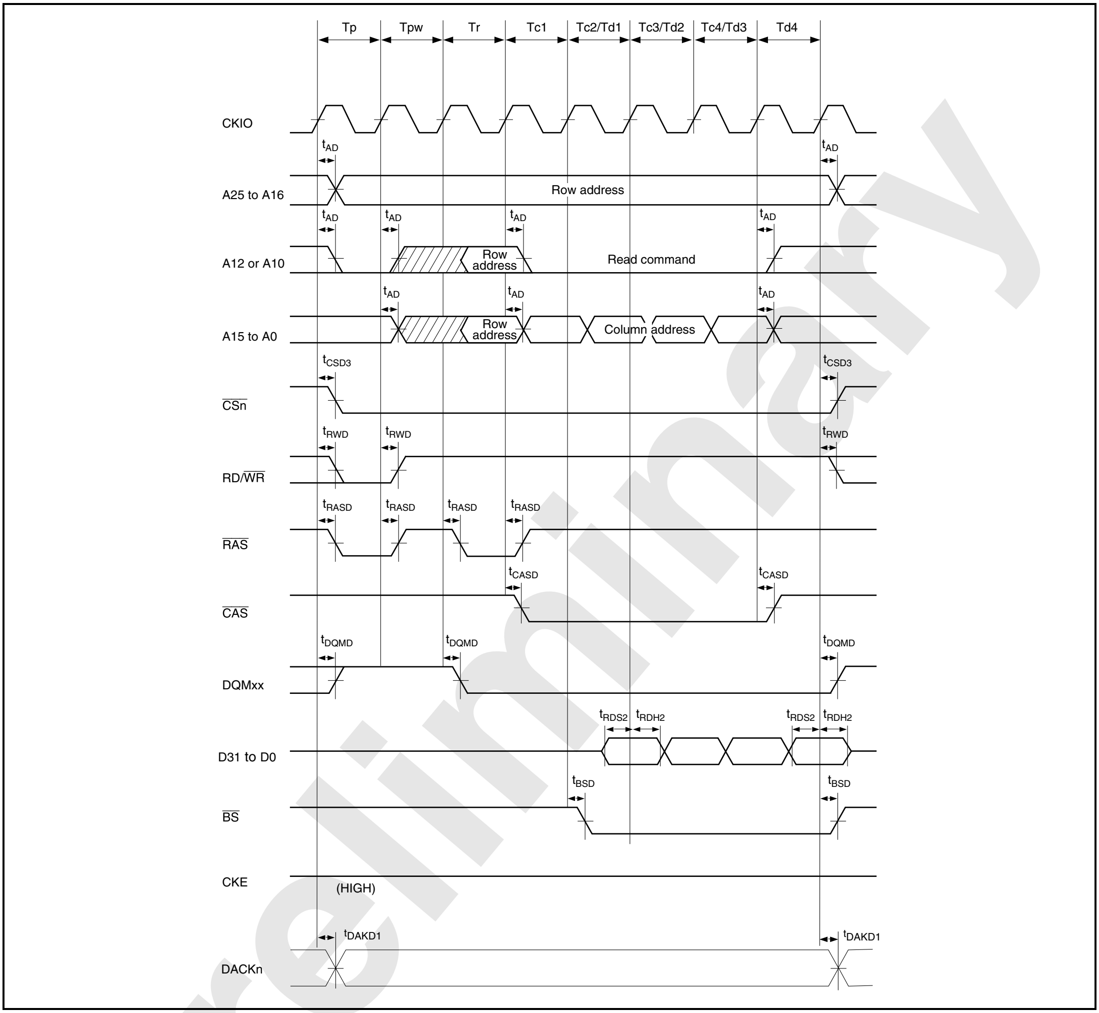


Figure 24.33 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Different Row Address, TPC = 1, RCD = 0, CAS Latency = 1)

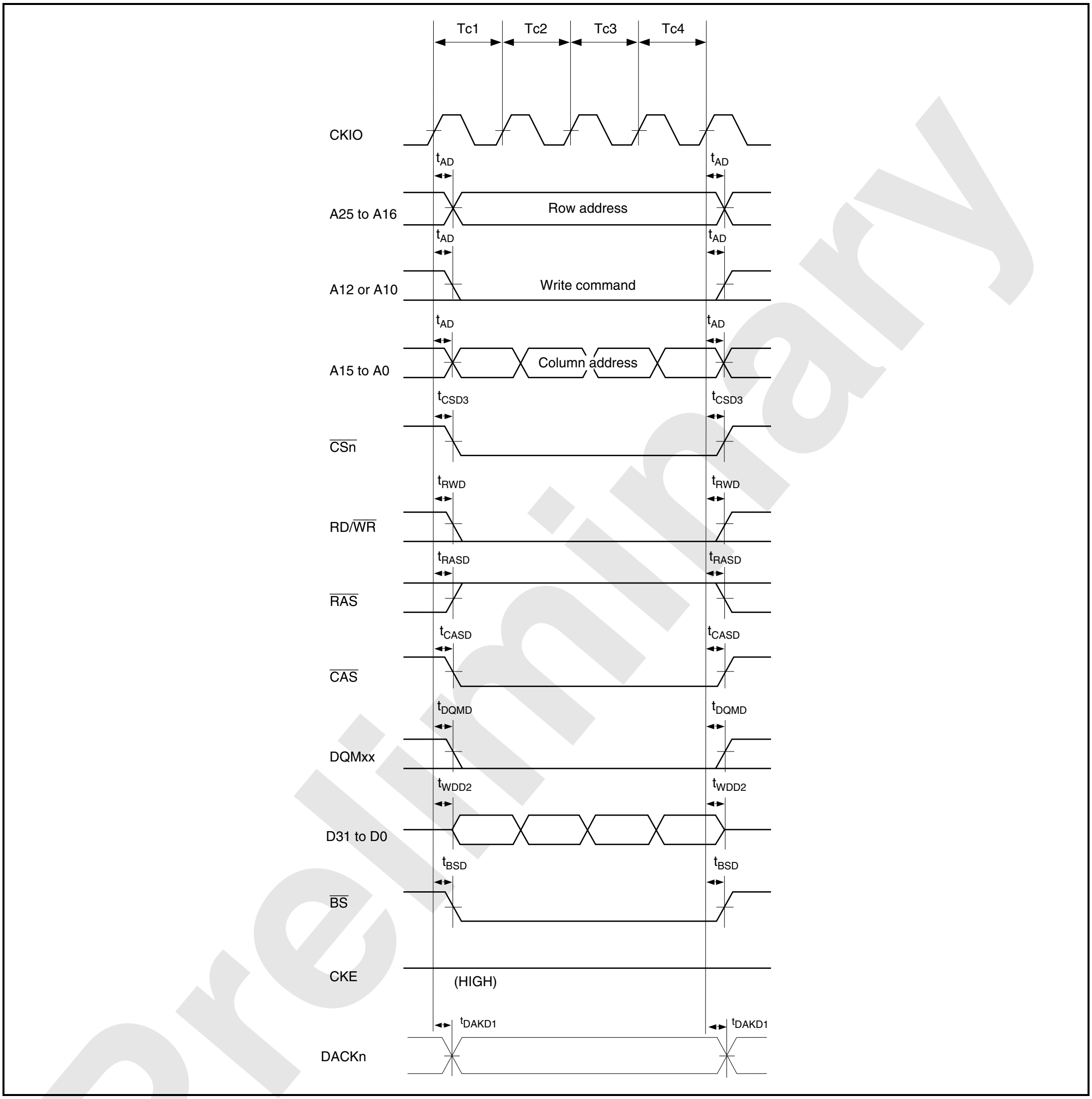


Figure 24.34 Synchronous DRAM Burst Write Bus Cycle (RAS Down, Same Row Address)

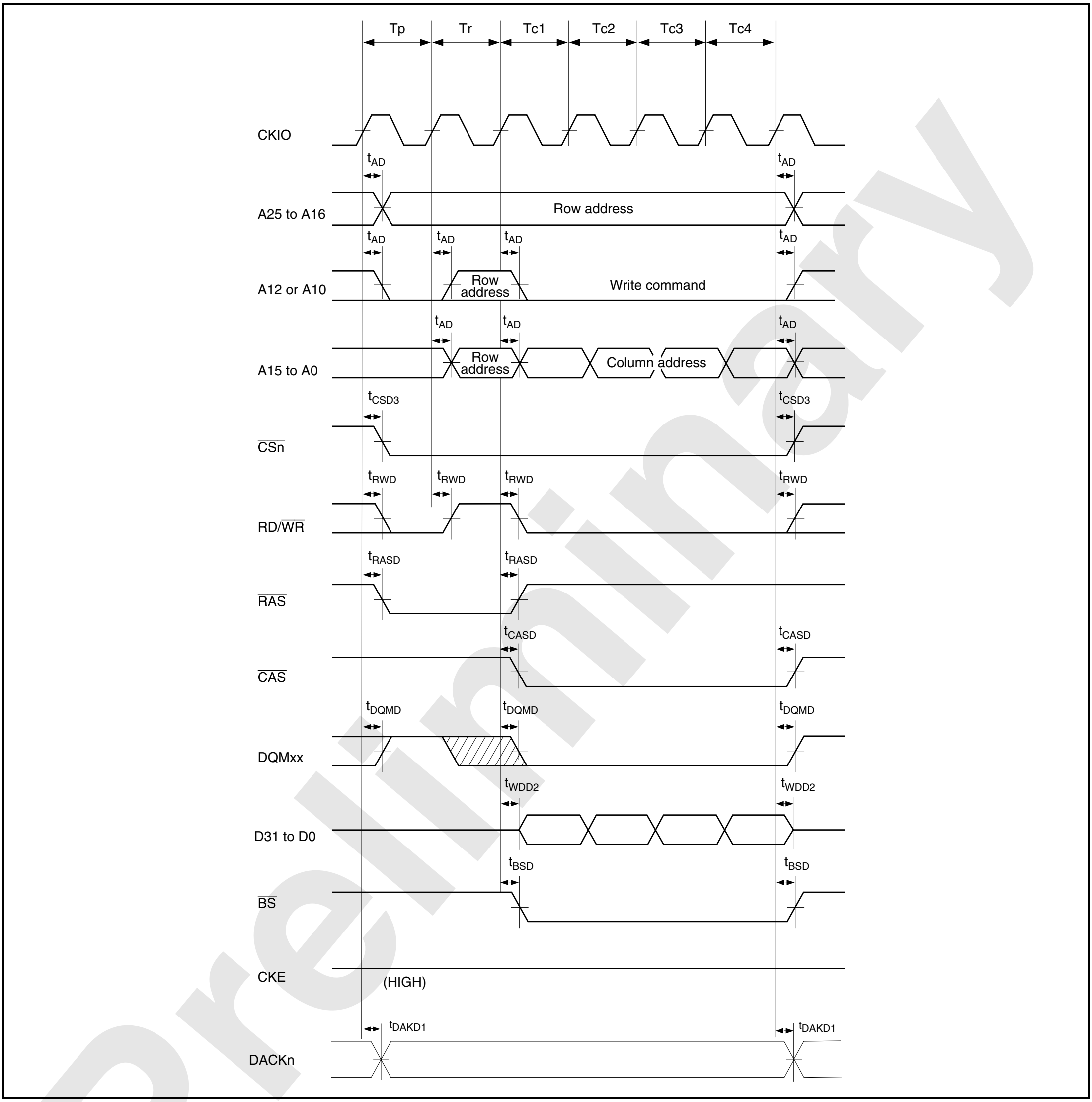


Figure 24.35 Synchronous DRAM Burst Write Bus Cycle (RAS Down, Different Row Address, TPC = 0, RCD = 0)

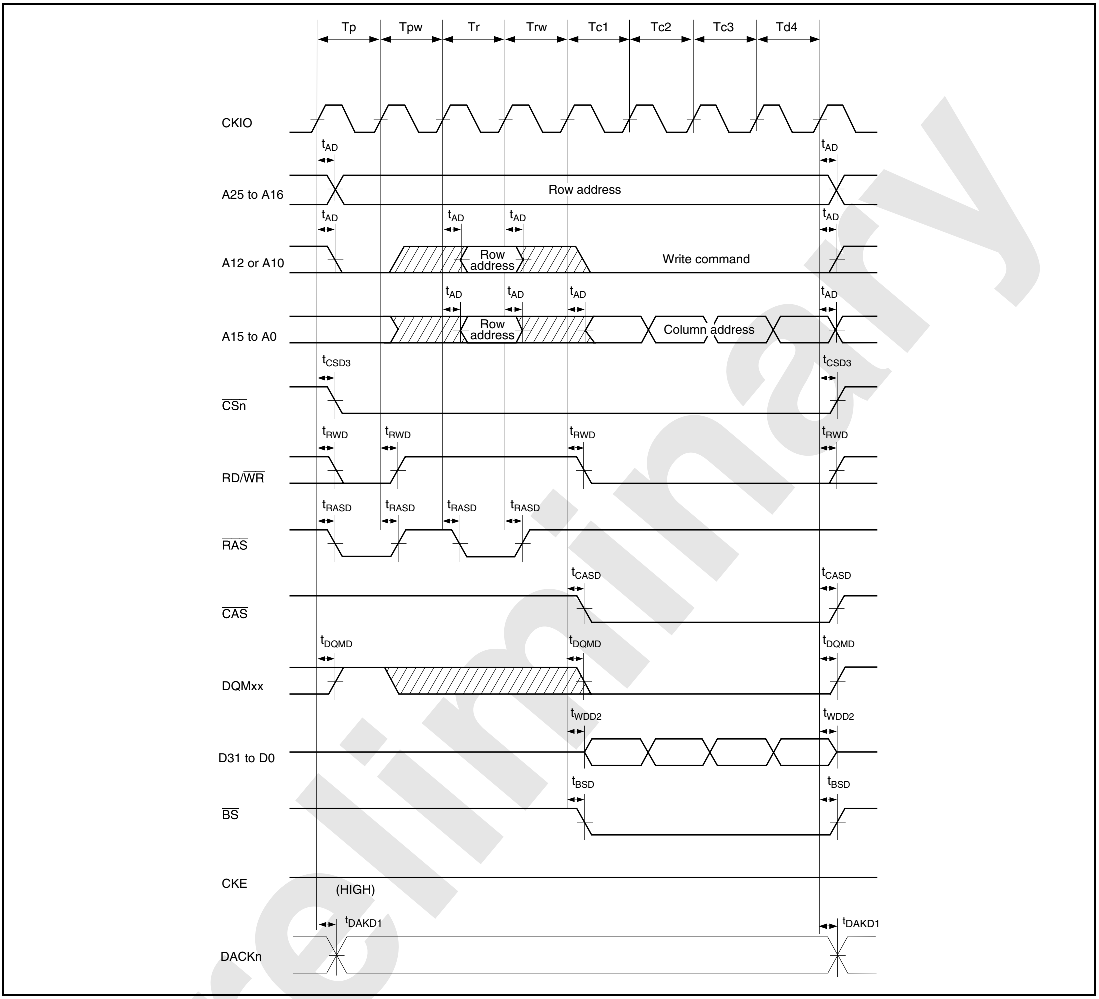


Figure 24.36 Synchronous DRAM Burst Write Bus Cycle (RAS Down, Different Row Address, TPC = 1, RCD = 1)

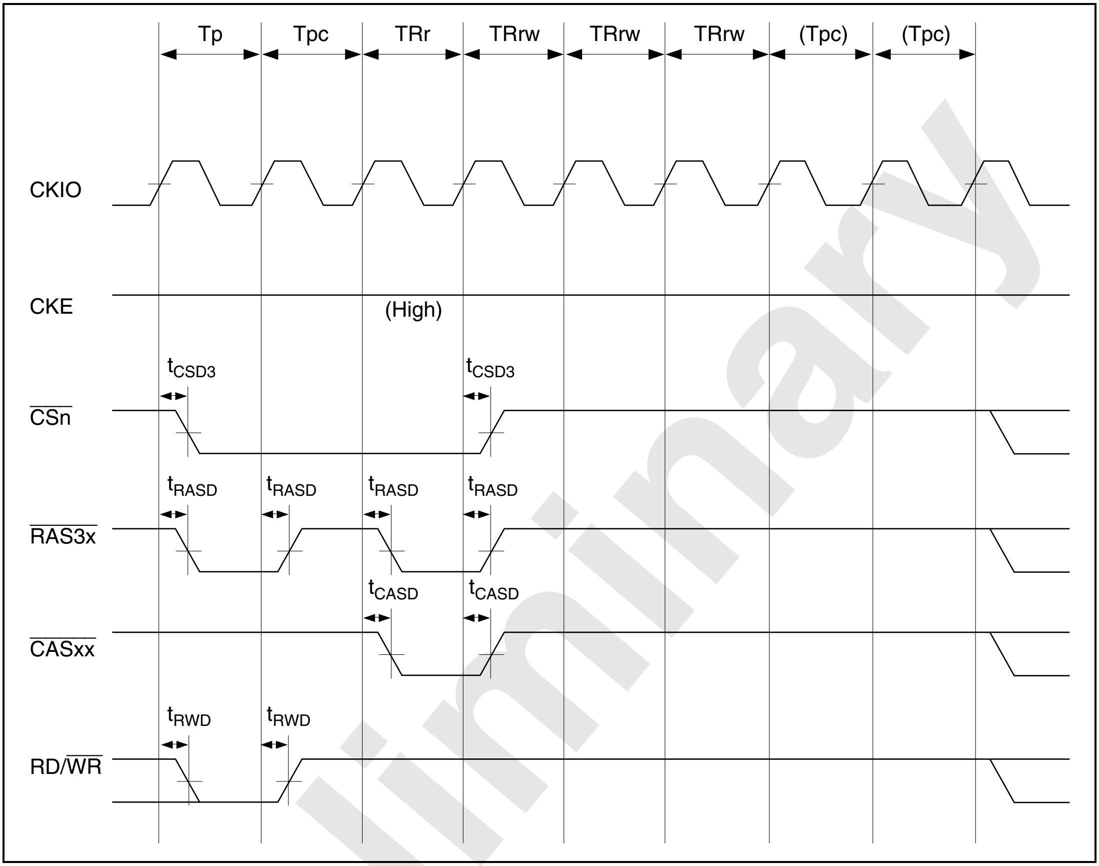


Figure 24.37 Synchronous DRAM Auto-Refresh Timing (TRAS = 1, TPC = 1)

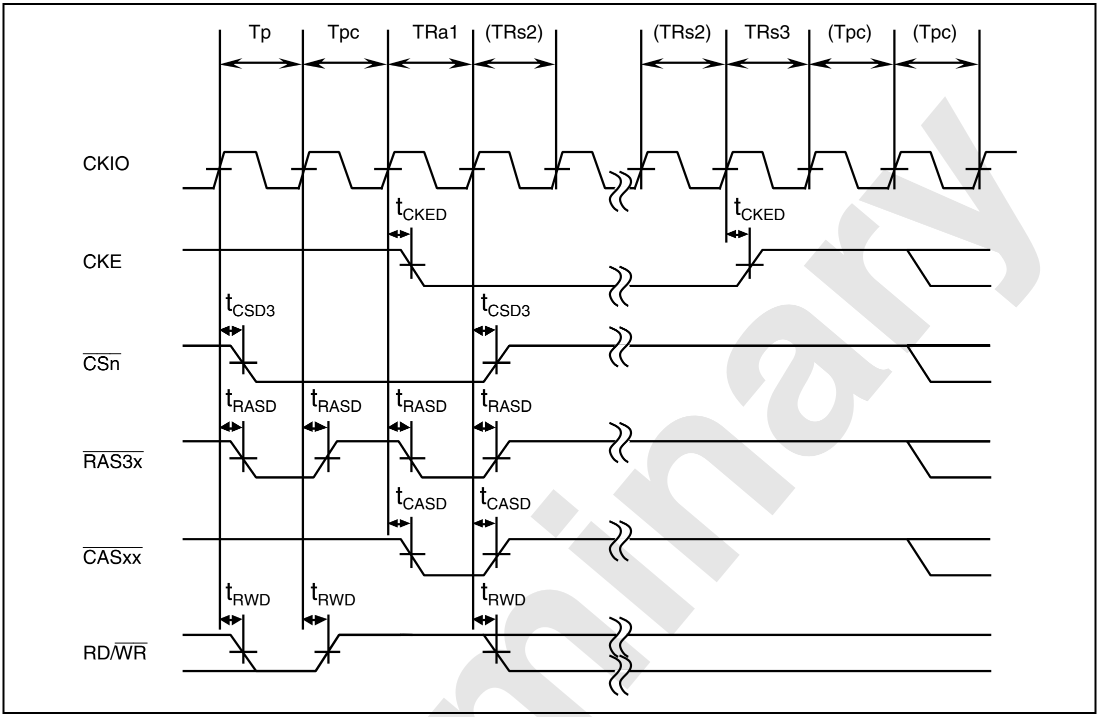


Figure 24.38 Synchronous DRAM Self-Refresh Cycle (TPC = 0)

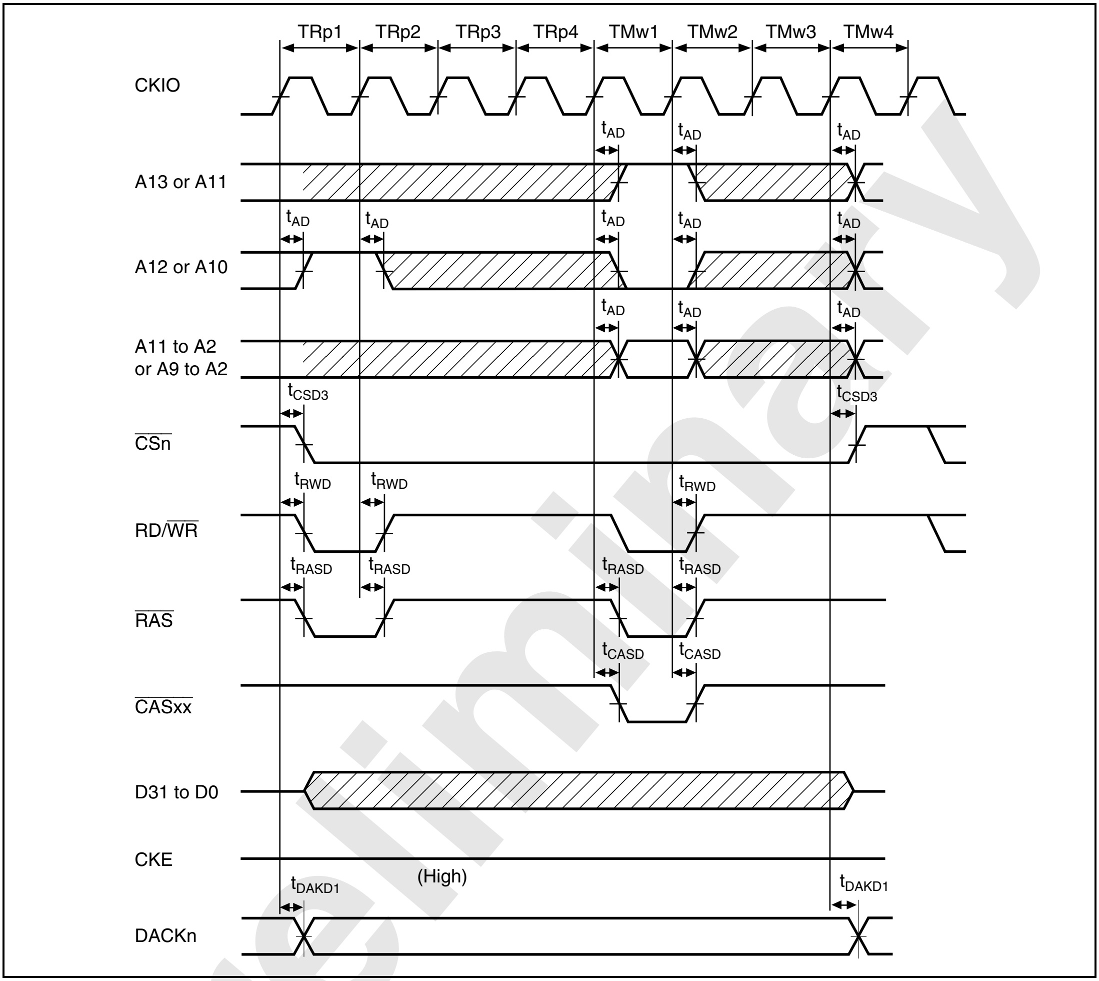


Figure 24.39 Synchronous DRAM Mode Register Write Cycle

24.3.7 PCMCIA Timing

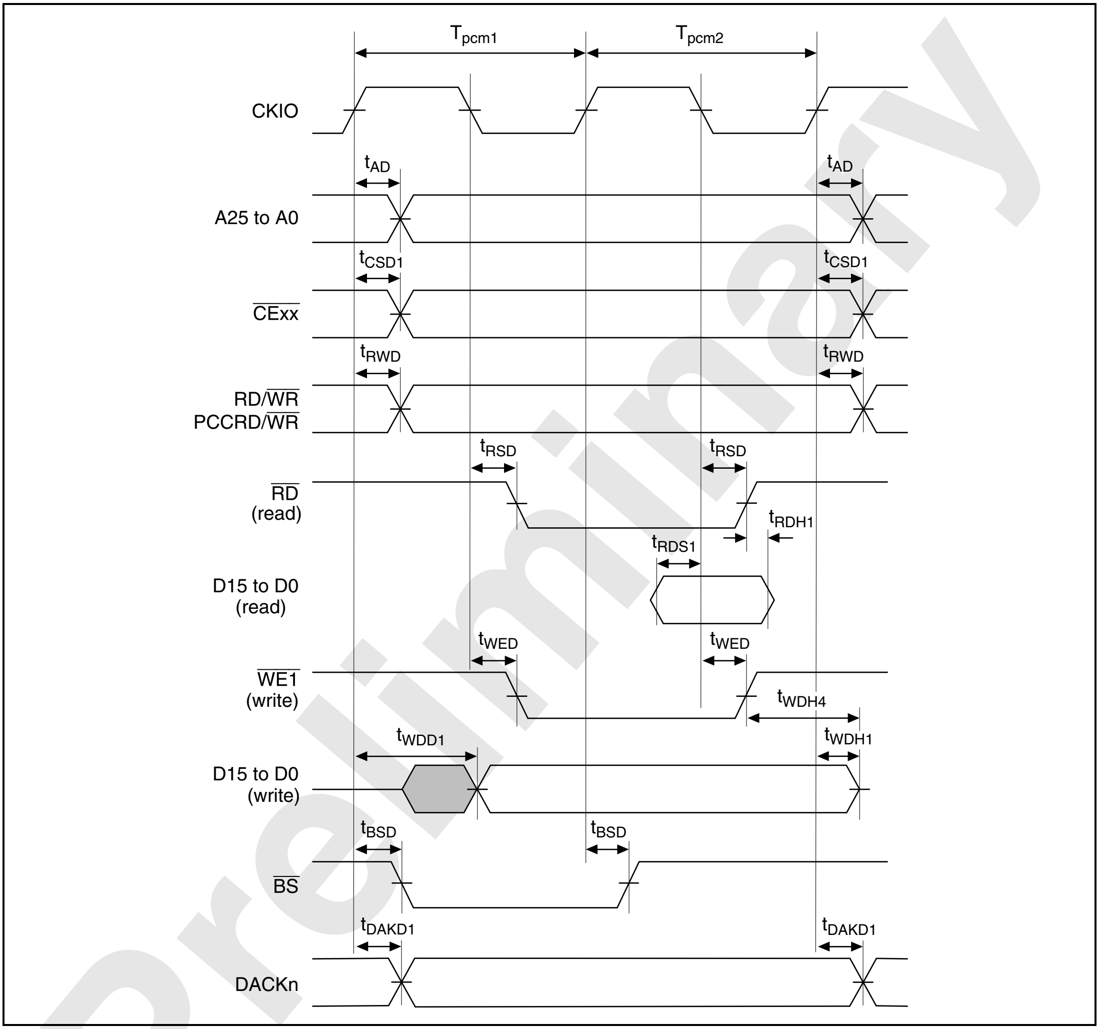


Figure 24.40 PCMCIA Memory Bus Cycle (TED = 0, TEH = 0, No Wait)

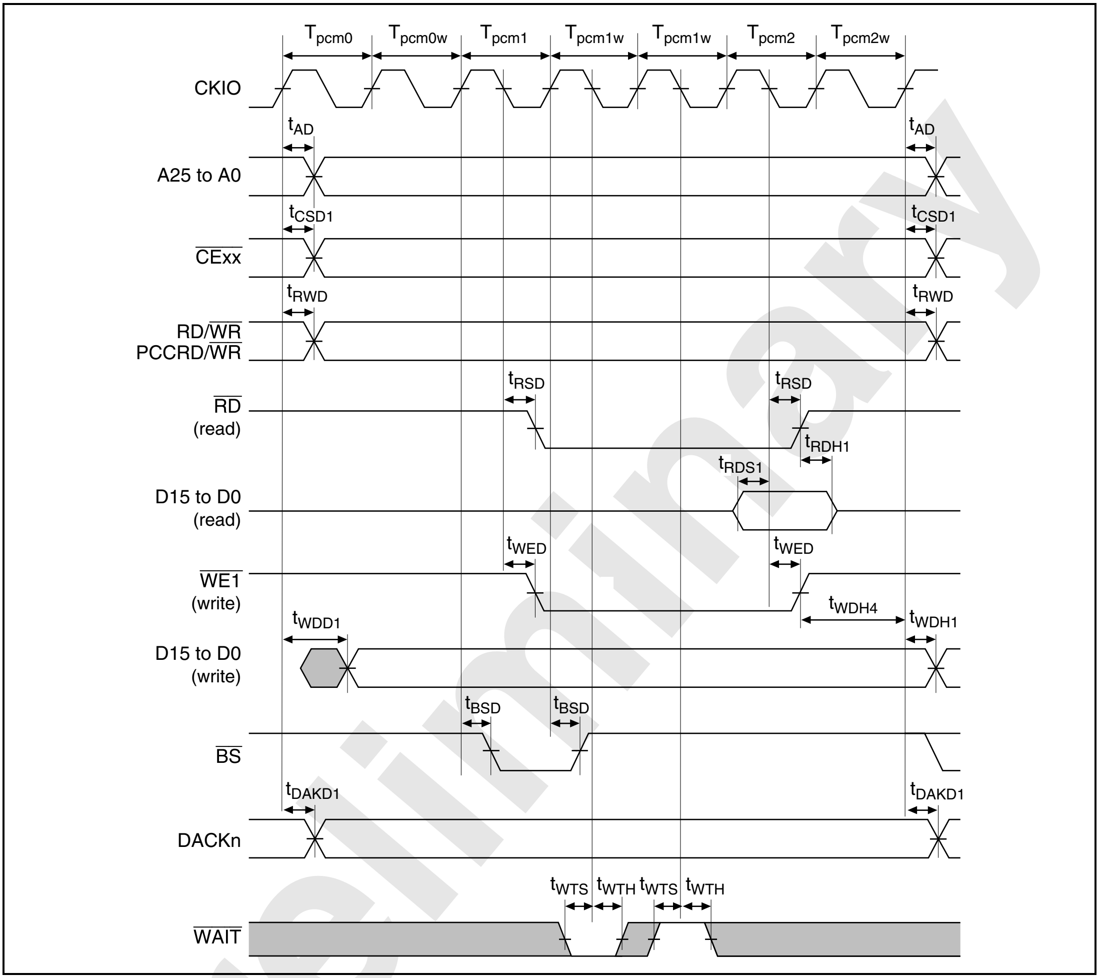
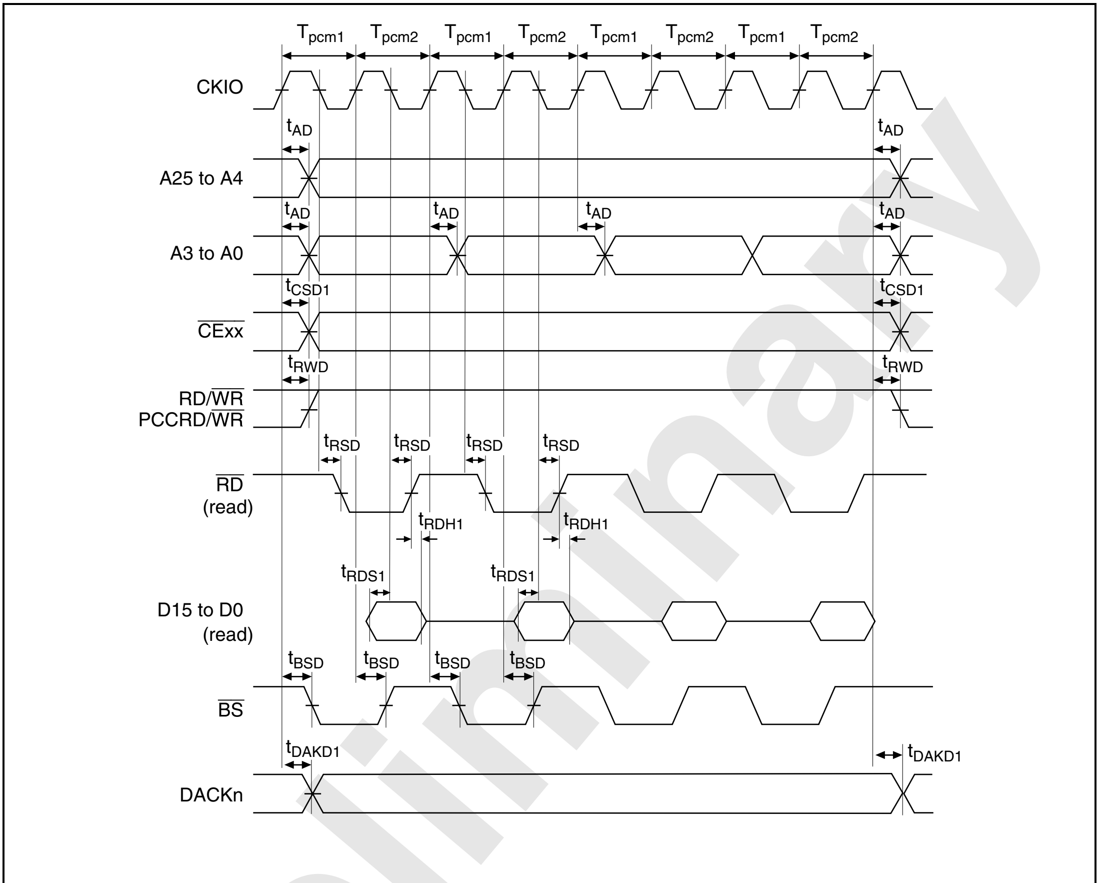
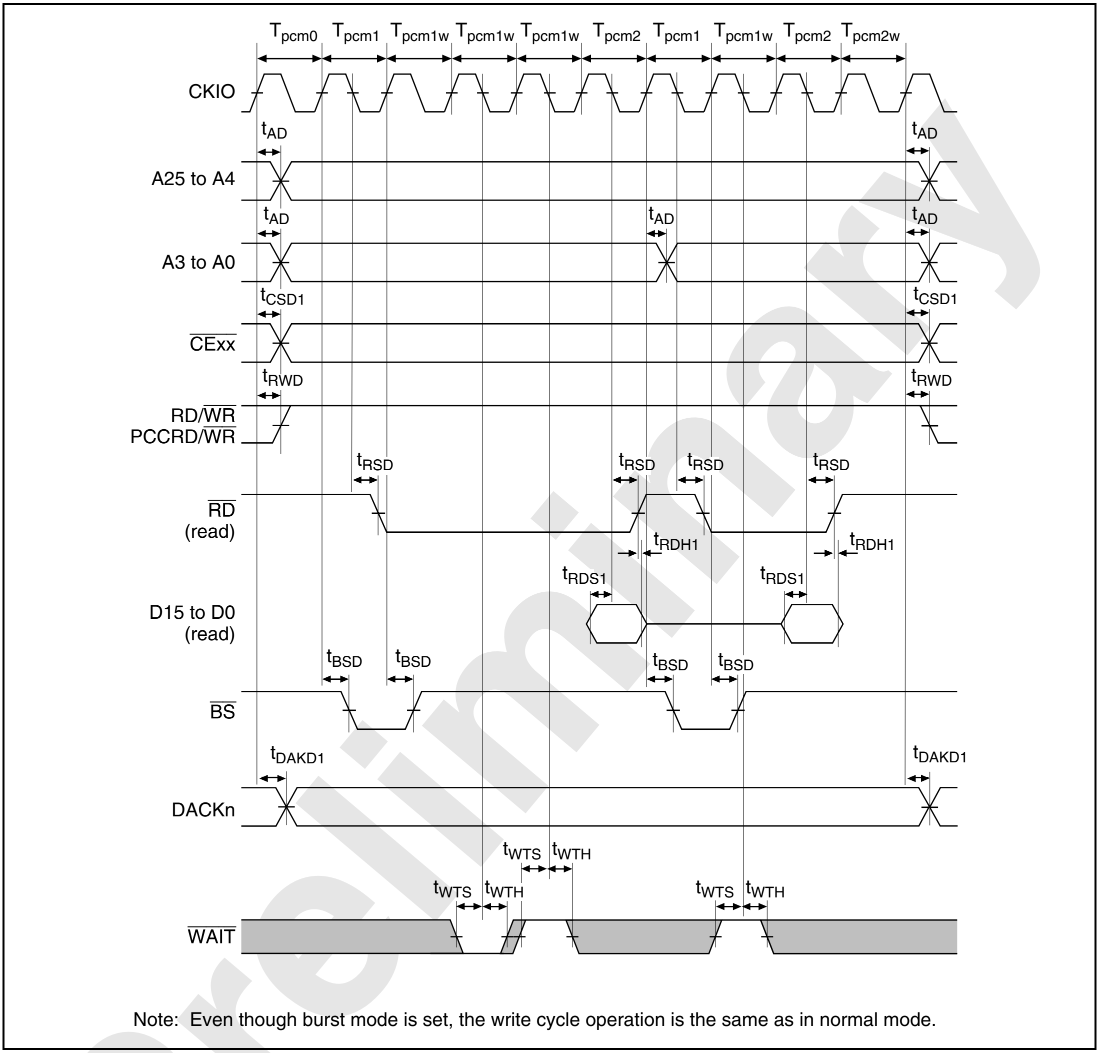


Figure 24.41 PCMCIA Memory Bus Cycle
(TED = 2, TEH = 1, One Wait, External Wait)



Note: Even though burst mode is set, write cycle operation is the same as in normal mode.

Figure 24.42 PCMCIA Memory Bus Cycle (Burst Read, TED = 0, TEH = 0, No Wait)



Note: Even though burst mode is set, the write cycle operation is the same as in normal mode.

Figure 24.43 PCMCIA Memory Bus Cycle (Burst Read, TED = 1, TEH = 1, Two Waits, Burst Pitch = 3)

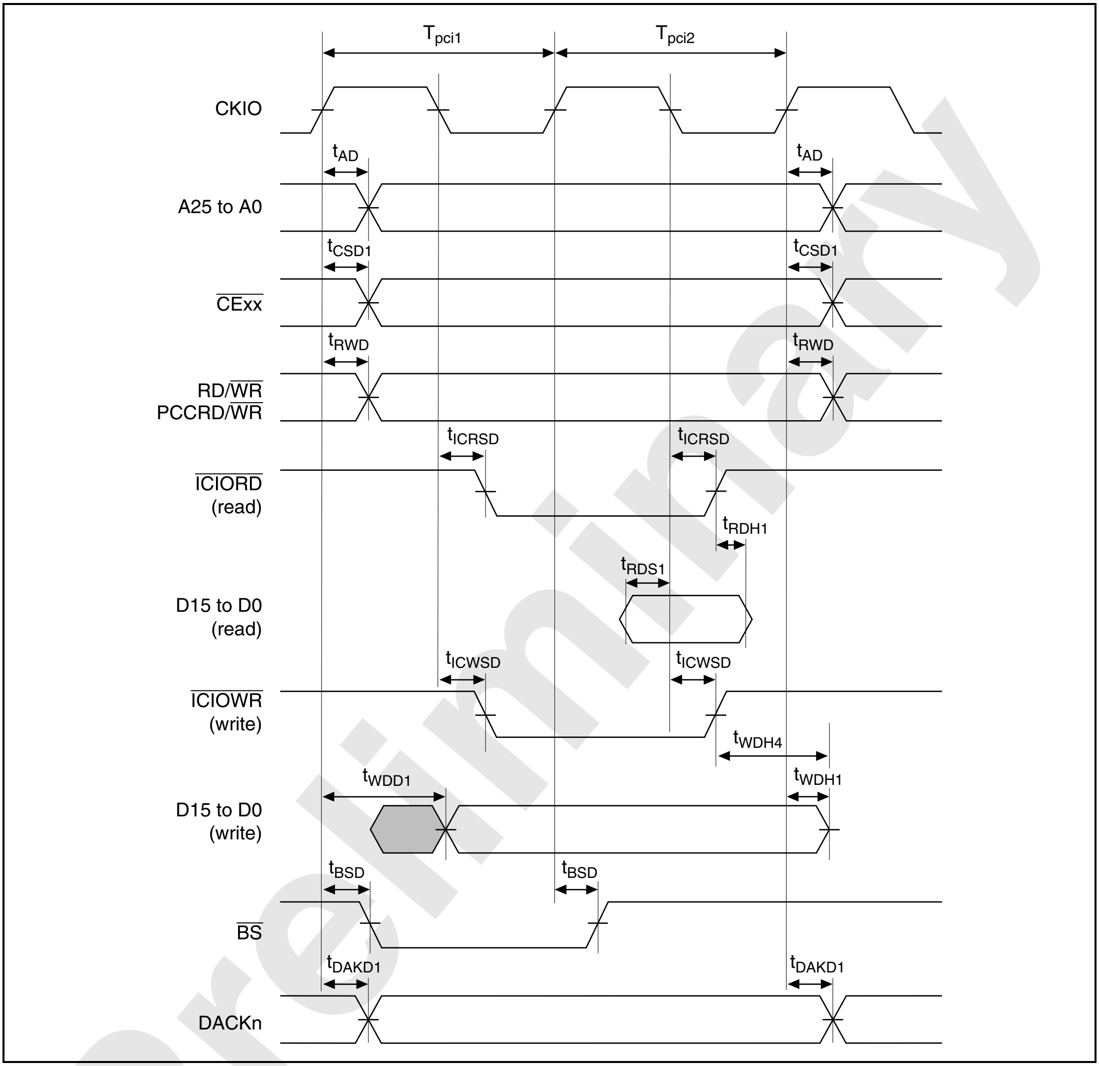


Figure 24.44 PCMCIA I/O Bus Cycle (TED = 0, TEH = 0, No Wait)

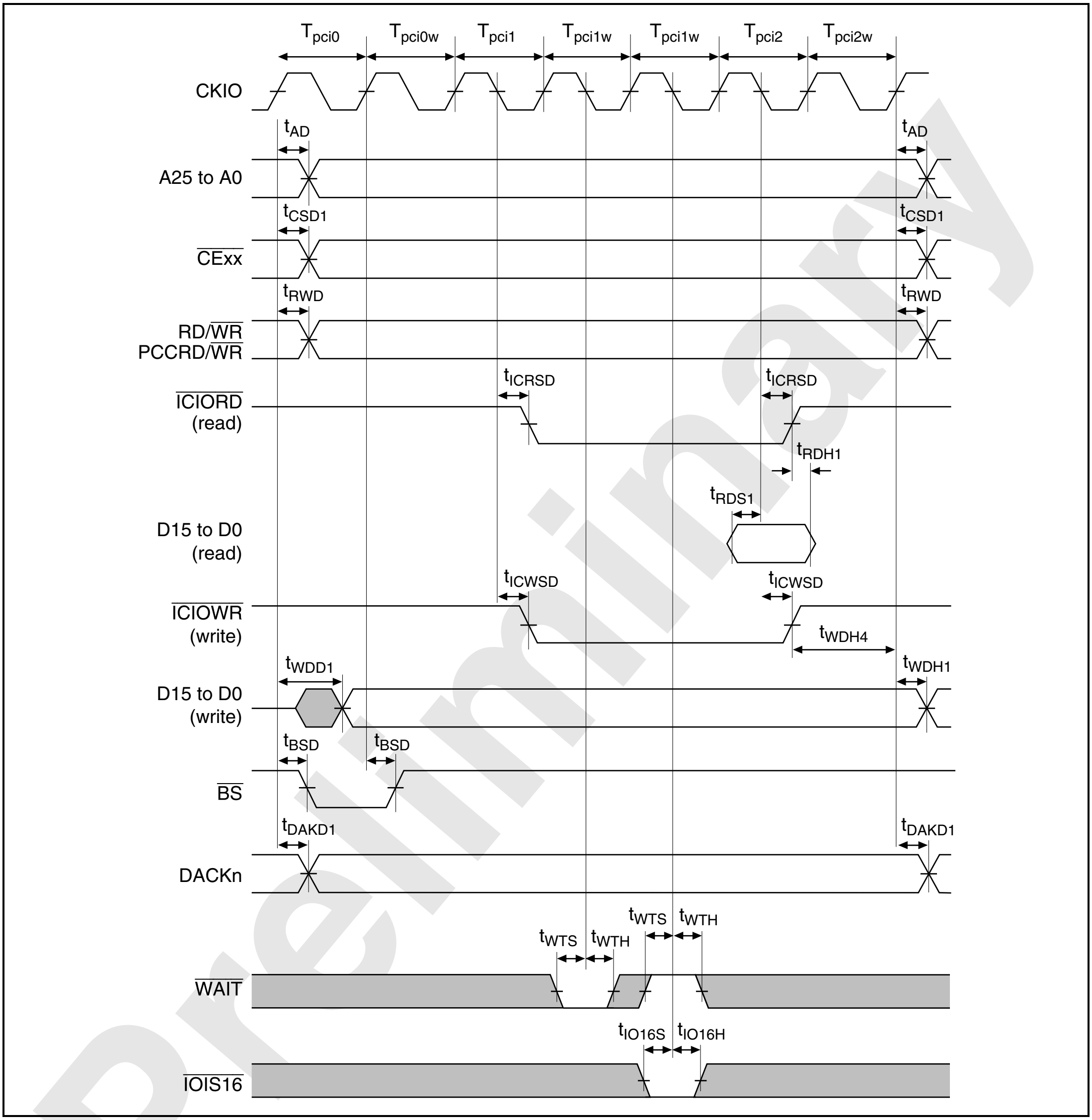


Figure 24.45 PCMCIA I/O Bus Cycle (TED = 2, TEH = 1, One Wait, External Wait)

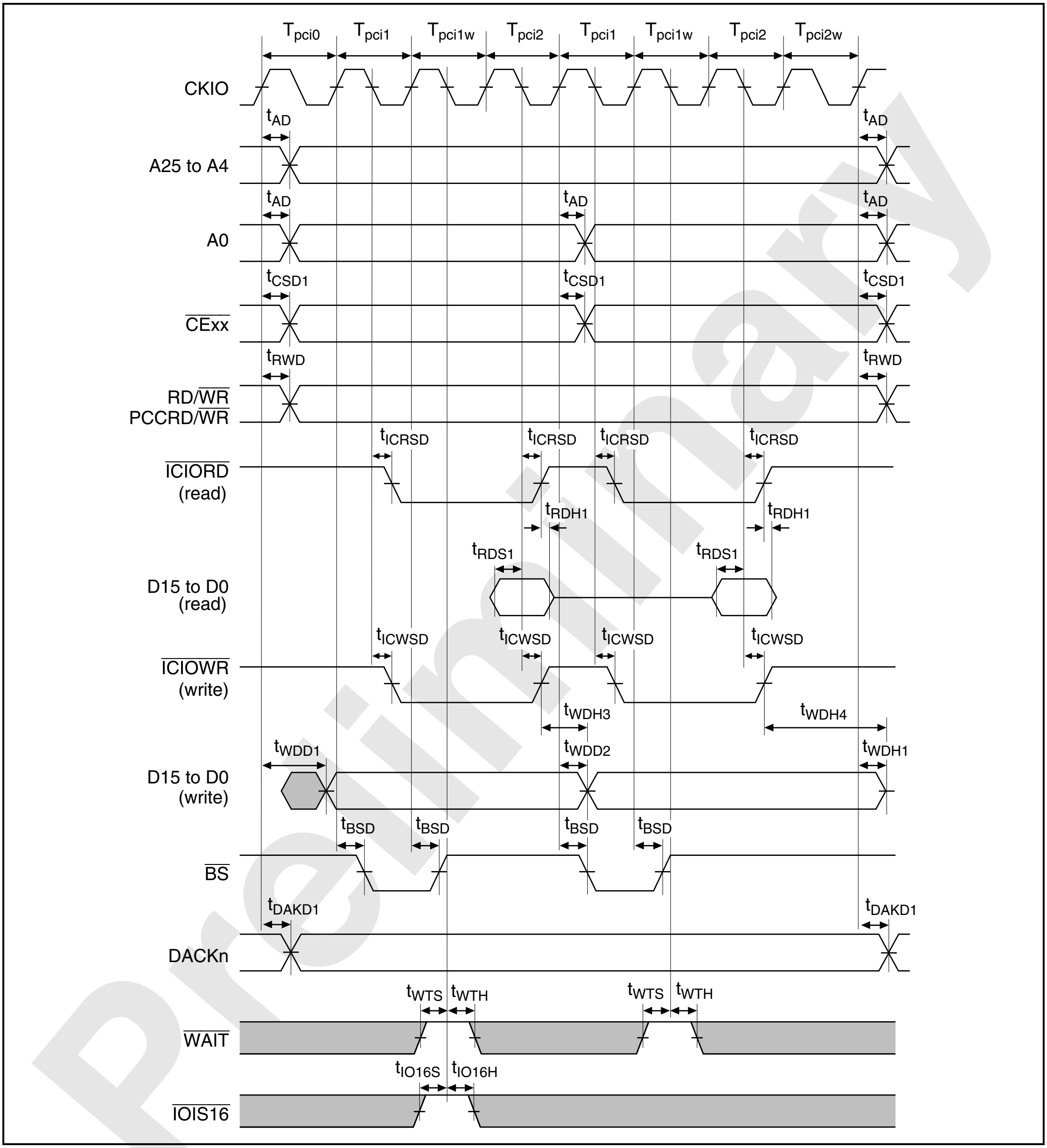


Figure 24.46 PCMCIA I/O Bus Cycle
(TED = 1, TEH = 1, One Wait, Bus Sizing)

24.3.8 Peripheral Module Signal Timing

Table 24.8 Peripheral Module Signal Timing

				–66			
Module	**Item**		**Symbol**	**Min**	**Max**	**Unit**	**Figure**
TMU, RTC	Timer input setup time		t_{TCLKS}	15	—	ns	24.47
	Timer clock input setup time		t_{TCKS}	15	—		24.48
	Timer clock pulse width	Edge specification	t_{TCKWH}	1.5	—	tcyc	
		Both edge specification	t_{TCKWL}	2.5	—		
	Oscillation settling time		t_{ROSC}	—	3	s	24.44
SCI	Input clock cycle	Asynchronization	t_{SCYC}	4	—	tcyc	22.50, 22.51
		Clock synchronization		6	—		
	Input clock rise time		t_{SCKR}	—	1.5		24.50
	Input clock fall time		t_{SCKF}	—	1.5		
	Input clock pulse width		t_{SCKW}	0.4	0.6	tscyc	
	Transmission data delay time		t_{TXD}	—	100	ns	24.51
	Receive data setup time (clock synchronization)		t_{RXS}	100	—		
	Receive data hold time (clock synchronization)		t_{RXH}	100	—		
	$\overline{\text{RTS}}$ delay time		t_{RTSD}	—	100		
	$\overline{\text{CTS}}$ setup time (clock synchronization)		t_{CTSS}	100	—		
	$\overline{\text{CTS}}$ hold time (clock synchronization)		t_{CTSH}	100	—		
Port	Output data delay time		t_{PORTD}	—	17	ns	24.52
	Input data setup time		t_{PORTS1}	15	—		
	Input data hold time		t_{PORTH1}	8	—		
	Input data setup time		t_{PORTS2}	17	—		
	Input data hold time		t_{PORTH2}	10	—		
DMAC	$\overline{\text{DREQ}}$ setup time		t_{DREQ}	6	—	ns	24.53
	$\overline{\text{DREQ}}$ hold time		t_{DREQH}	4	—		
	DRAK delay time		t_{DRAKD}	—	10		22.54

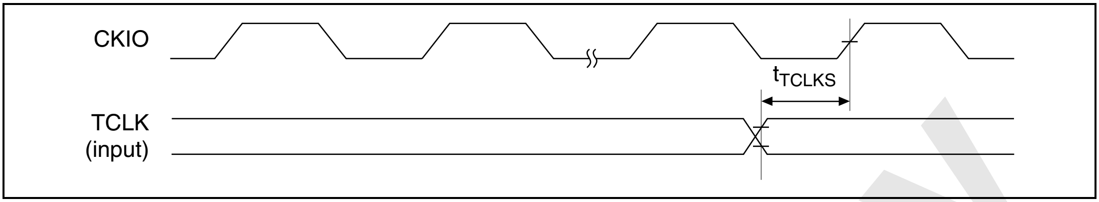


Figure 24.47 TCLK Input Timing

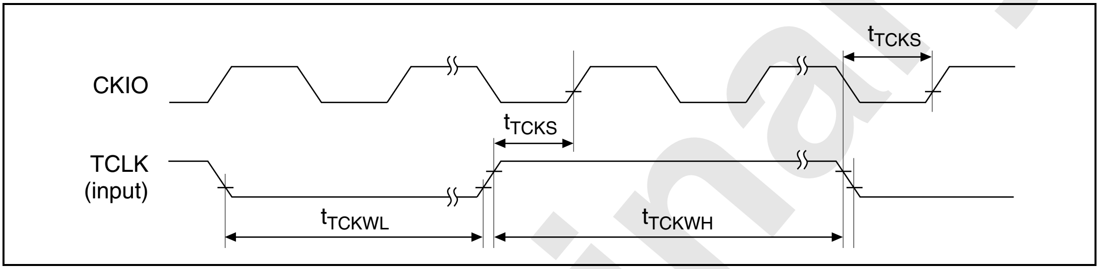


Figure 24.48 TCLK Clock Input Timing

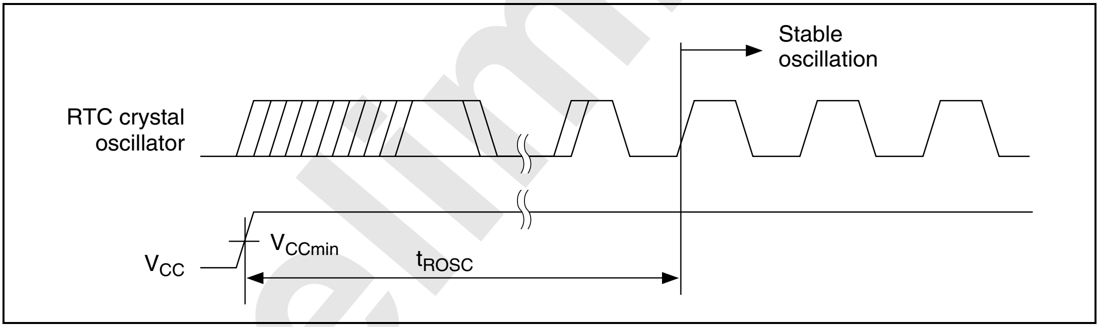


Figure 24.49 Oscillation Settling Time at RTC Crystal Oscillator Power-on

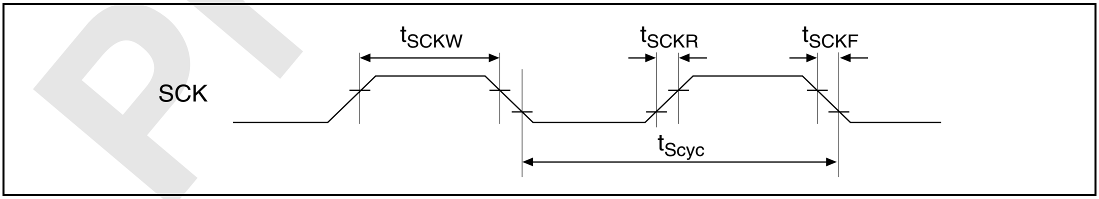


Figure 24.50 SCK Input Clock Timing

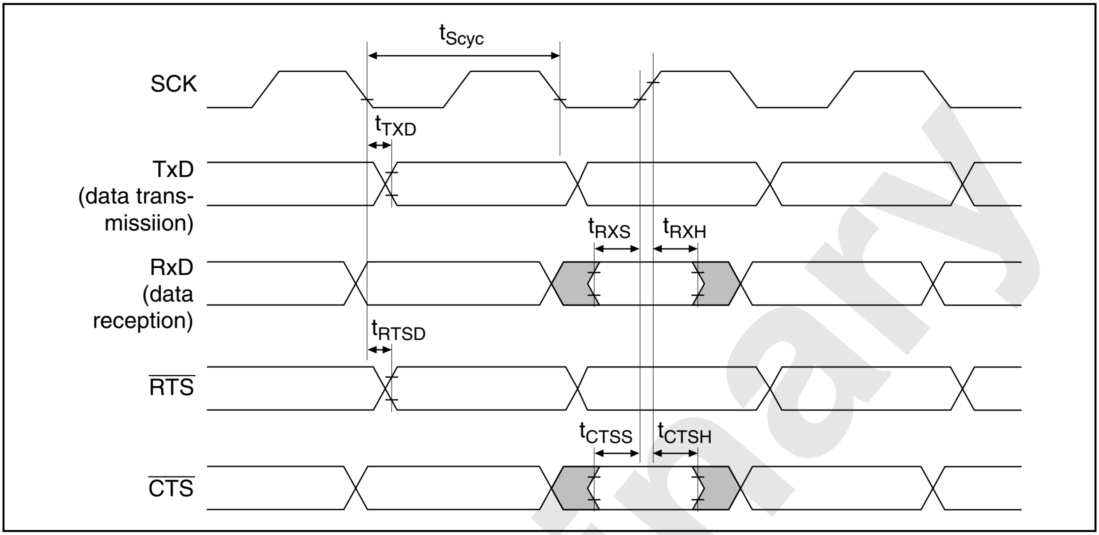


Figure 24.51 SCI I/O Timing in Clock Synchronous Mode

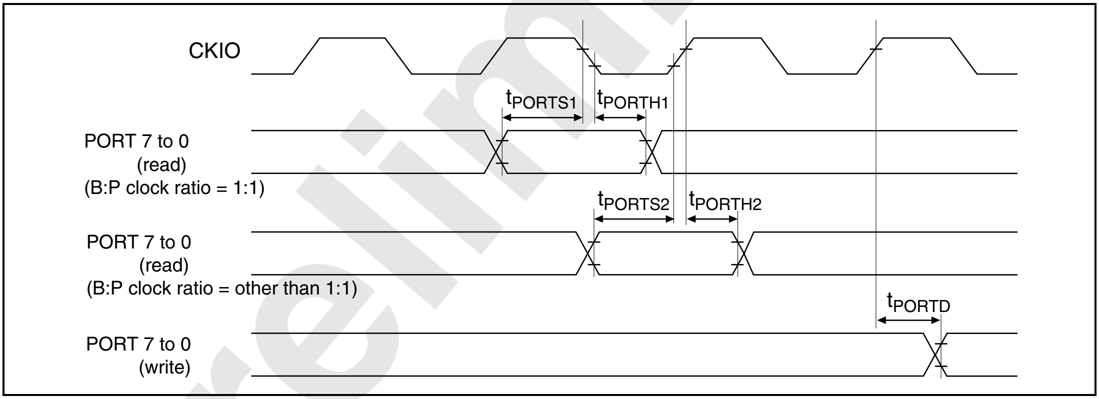


Figure 24.52 I/O Port Timing

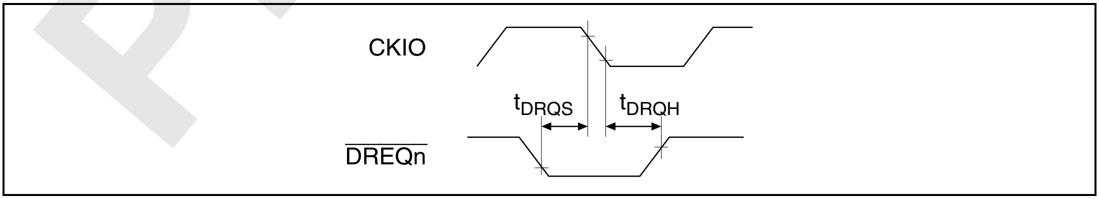


Figure 24.53 $\overline{\text{DREQ}}$ Input Timing

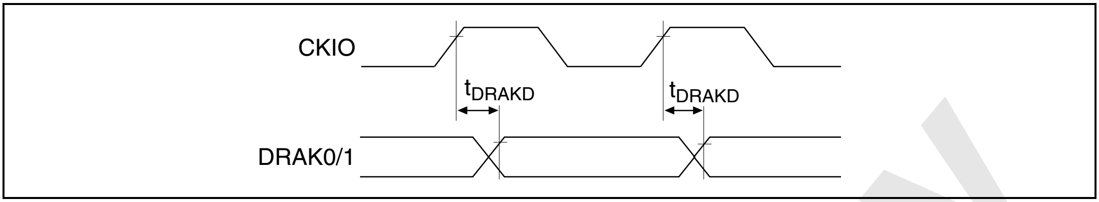


Figure 24.54 DRAK Output Timing

24.3.9 H-UDI, AUD Related Pin Timing

Table 24.9 H-UDI, AUD Related Pin Timing

Item	Symbol	Min	Max	Unit	Figure
TCK cycle time	t_{TCKcyc}	50	—	ns	24.55
TCK high pulse width	t_{TCKH}	12	—	ns	
TCK low pulse width	t_{TCKL}	12	—	ns	
TCK rise/fall time	t_{TCKf}	—	4	ns	
TRST setup time	t_{TRSTS}	12	—	ns	24.56
TRST hold time	t_{TRSTH}	50	—	t_{cyc}	
TDI setup time	t_{TDIS}	10	—	ns	24.57
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}	—	16	ns	
$\overline{\text{ASEMD0}}$ setup time	t_{ASEMDH}	12	—	ns	24.58
$\overline{\text{ASEMD0}}$ hold time	t_{ASEMDS}	12	—	ns	
AUDATA delay time	t_{AUDD}	—	10	ns	24.59
AUDSYNC delay time	t_{AUSYD}	—	10	ns	24.59

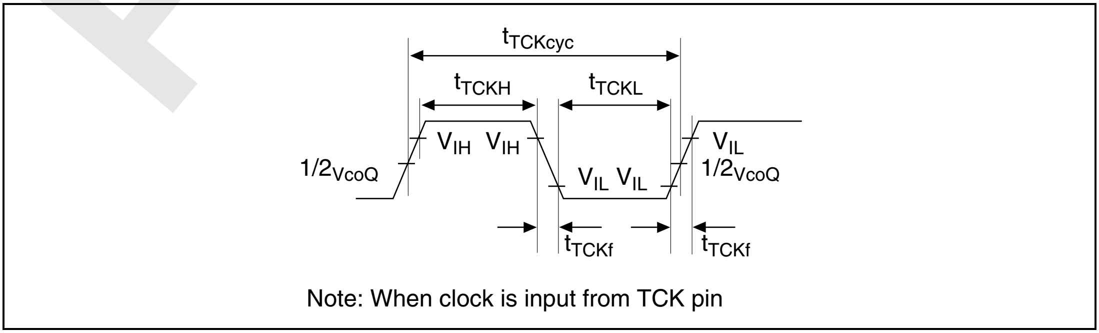


Figure 24.55 TCK Input Timing

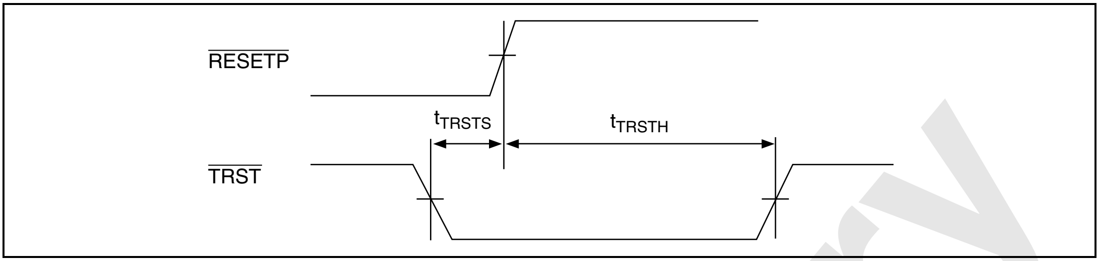


Figure 24.56 TRST Input Timing (Reset Hold)

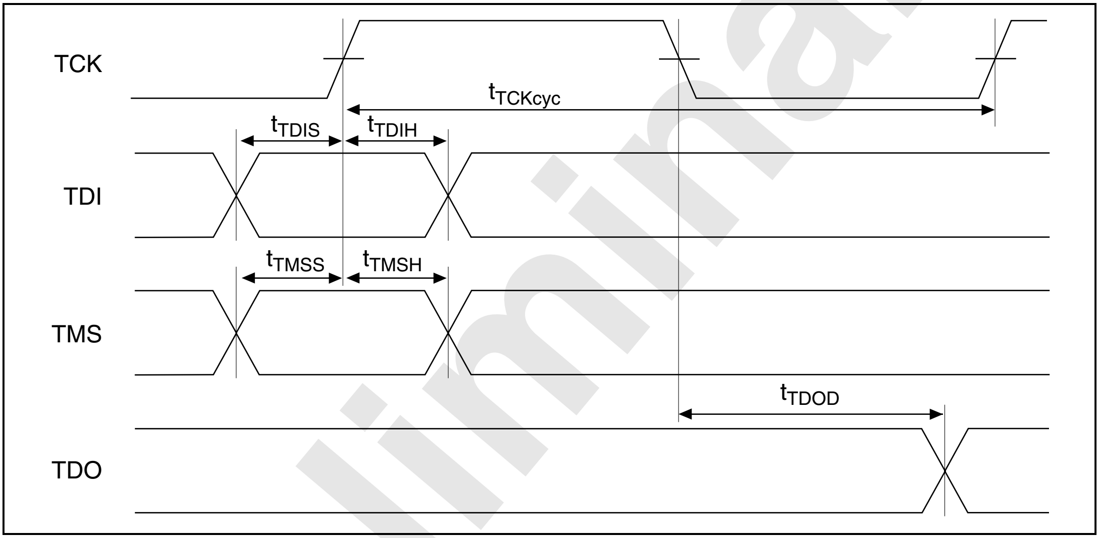


Figure 24.57 H-UDI Data Transfer Timing

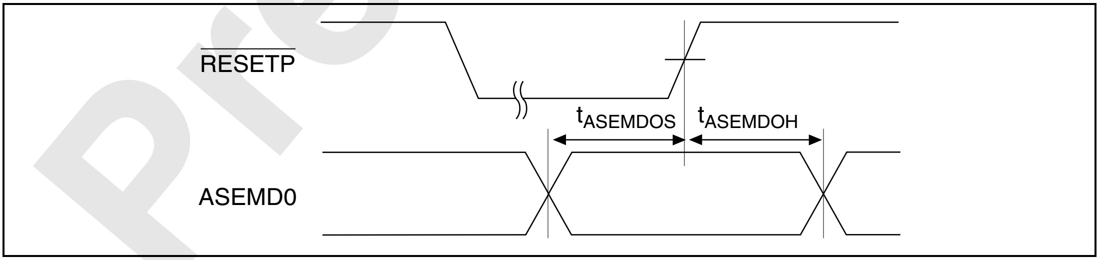


Figure 24.58 ASEMD0 Input Timing

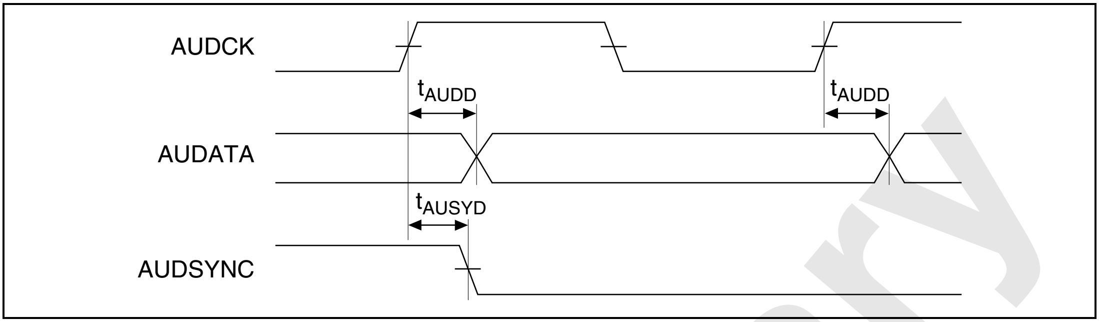


Figure 24.59 AUD Timing

24.3.10 A/D Converter Timing

Table 24.10 A/D Converter Timing

Item		Symbol	Min	Typ	Max	Unit	Figure
External trigger input pulse width		t_{TRGW}	2	—	—	tcyc	24.60
External trigger input start delay time		t_{TRGS}	50	—	—	ns	
Input sampling time	(CKS = 0)	t_{SPL}	—	129	—	tcyc	24.61
	(CKS = 1)		—	65	—		
A/D conversion start delay time	(CKS = 0)	t_D	17	—	28	tcyc	
	(CKS = 1)		10	—	17		
A/D conversion time	(CKS = 0)	t_{CONV}	514	—	525	tcyc	
	(CKS = 1)		259	—	266		

tcyc: Pϕ cycle

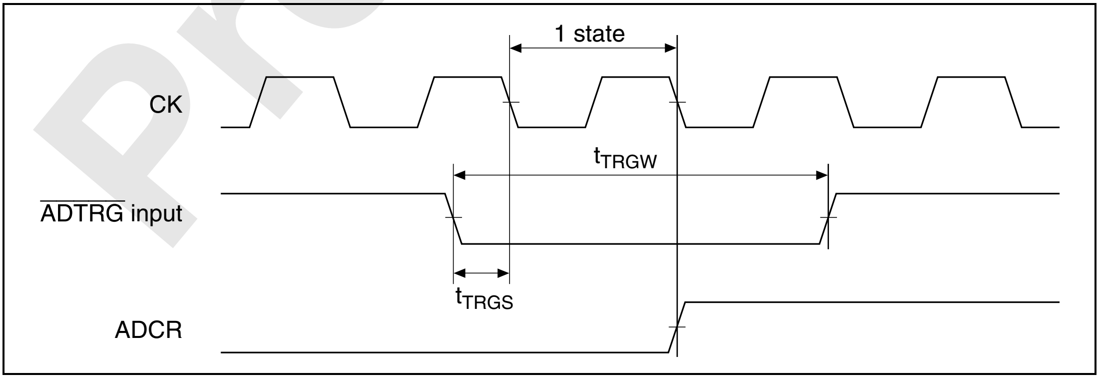
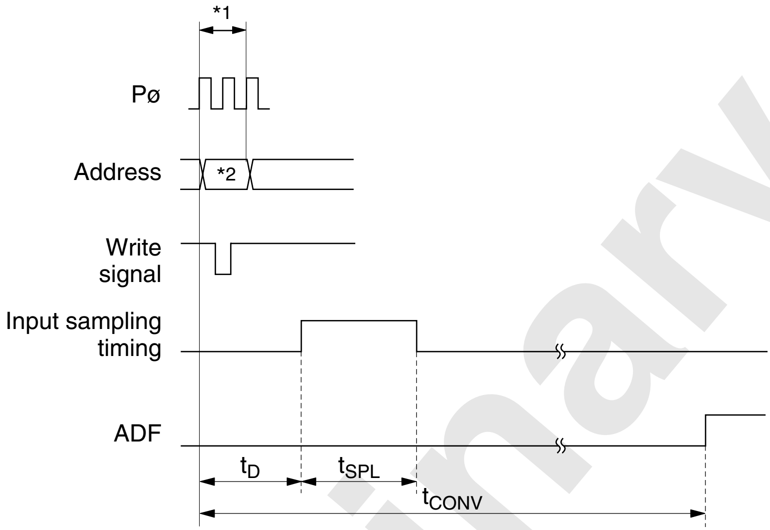


Figure 24.60 External Trigger Input Timing



t_D A/D conversion start delay

t_{SPL} Input sampling time

t_{CONV} A/D conversion time

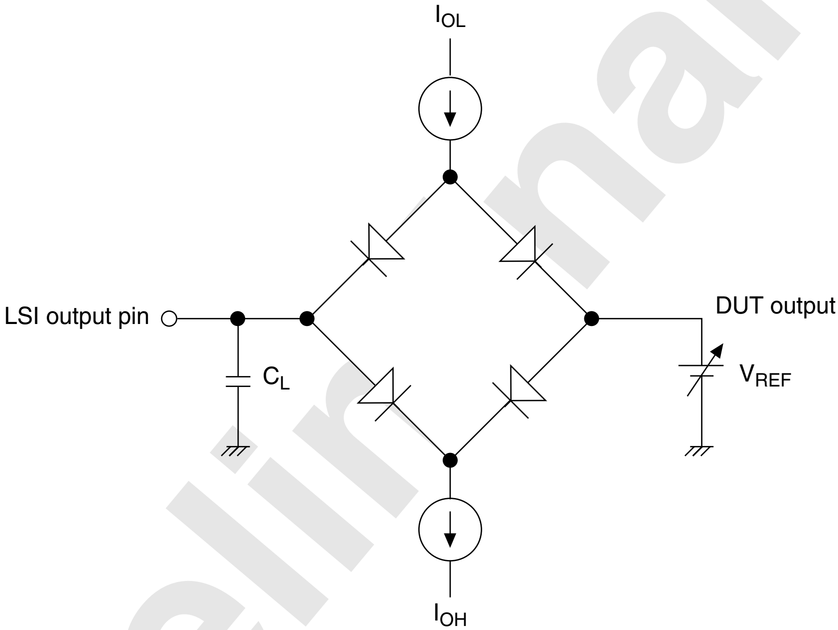
Notes: *1 ADCSR write cycle

*2 ADCSR address

Figure 24.61 A/D Conversion Timing

24.3.11 AC Characteristics Measurement Conditions

- I/O signal reference level: 1.5 V (VccQ = 3.3 ± 0.3 V, Vcc = 1.9 ± 0.15 V)
- Input pulse level: Vss to 3.0 V (where $\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI, $\overline{\text{IRQ5}}$–$\overline{\text{IRQ0}}$, CKIO, and MD5–MD0 are within Vss to Vcc)
- Input rise and fall times: 1 ns



Notes: 1. C_L is the total value that includes the capacitance of measurement instruments, etc., and is set as follows for each pin.
30pF: CKIO, $\overline{\text{RAS}}$, $\overline{\text{CASxx}}$, $\overline{\text{CS0}}$, $\overline{\text{CS2}}$–$\overline{\text{CS6}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$, $\overline{\text{BACK}}$
50pF: All other pins
2. I_{OL} and I_{OH} are the values shown in table 23.3.

Figure 24.62 Output Load Circuit

24.3.12 Delay Time Variation Due to Load Capacitance

A graph (reference data) of the variation in delay time when a load capacitance greater than that stipulated (30 pF) is connected to this LSI's pins is shown below. The graph shown in Figure 24.63 should be taken into consideration in the design process if the stipulated capacitance is exceeded in connecting an external device.

If the connected load capacitance exceeds the range shown in Figure 24.63 the graph will not be a straight line.

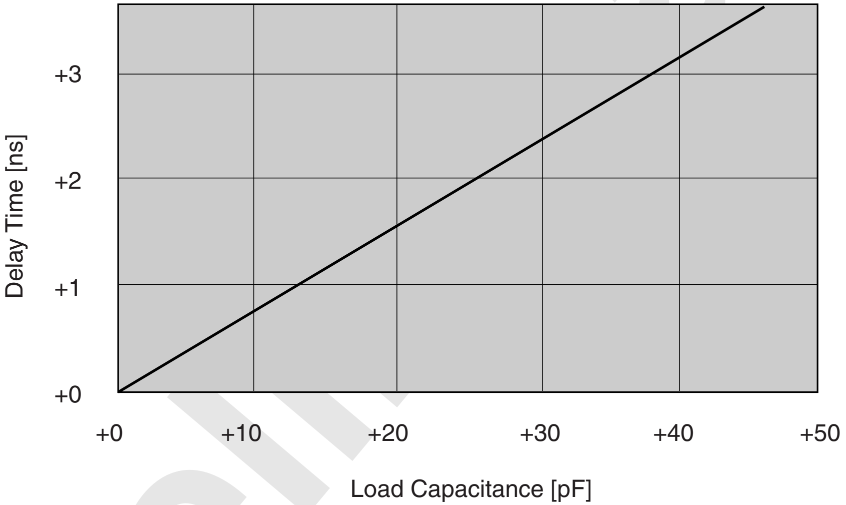


Figure 24.63 Load Capacitance vs. Delay Time

24.4 A/D Converter Characteristics

Table 24.11 lists the A/D converter characteristics.

Table 24.11 A/D Converter Characteristics
(VccQ = 3.3 ± 0.3 V, Vcc = 1.9 ± 0.15 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C)

Item	Min	Typ	Max	Unit
Resolution	10	10	10	bits
Conversion time	8.1	—	—	μs
Analog input capacitance	—	—	20	pF
Permissible signal-source (single-source) impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.0	LSB
Offset error	—	—	±2.0	LSB
Full-scale error	—	—	±2.0	LSB
Quantization error	—	—	±0.5	LSB
Absolute accuracy	—	—	±4.0	LSB

24.5 D/A Converter Characteristics

Table 24.12 lists the D/A converter characteristics.

Table 24.12 D/A Converter Characteristics
(VccQ = 3.3 ± 0.3 V, Vcc = 1.9 ± 0.15 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C)

Item	Min	Typ	Max	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time	—	—	10.0	μs	20-pF capacitive load
Absolute accuracy	—	±2.5	±4.0	LSB	2-MΩ resistance load

Appendix A Pin Functions

A.1 Pin States

Table A.1 shows pin states during resets, power-down states, and the bus-released states.

Table A.1 Pin States during Resets, Power-Down States, and Bus-Released State

Category	Pin	Reset		Power-Down		
		Power-On Reset	Manual Reset	Standby	Sleep	Bus Released
Clock	EXTAL	I	I	I	I	I
	XTAL	O*1	O*1	O*1	O*1	O*1
	CKIO	IO*1	IO*1	IO*1	IO*1	IO*1
	EXTAL2	I	I	I	I	I
	XTAL2	O	O	O	O	O
	CAP1, CAP2	—	—	—	—	—
System control	$\overline{\text{RESETP}}$	I	I	I	I	I
	$\overline{\text{RESETM}}$	I	I	I	I	I
	$\overline{\text{BREQ}}$	I	I	I	I	
	$\overline{\text{BACK}}$	O	O	O	O	L
	MD[5:0]	I	I	I	I	I
	CA	I	I	I	I	
	STATUS[1:0]/PTE[5:4]	O	OP*3	OP*3	OP*3	OP*3
Interrupt	IRQ[3:0]/$\overline{\text{IRL[3:0]}}$/ PTH[3:0]	I	I	I	I	I
	IRQ4/ PTH[4]	I	I	I	I	I
	NMI	I	I	I	I	I
	TCK/PTG[1]	IV	I	IZ	I	I
	TDI/PTG[0]	IV	I	IZ	I	I
	TMS/PTG[2]	IV	I	IZ	I	I
	$\overline{\text{TRST}}$/PTG[3]	IV	I	IZ	I	I
	$\overline{\text{IRQOUT}}$/PTE[7]	H	OP*3	ZK*3	OP*3	OP*3
Address bus	A[25:0]	Z	O	ZL*10	O	Z

Table A.1 Pin States during Resets, Power-Down States, and Bus-Released State (cont)

		Reset		Power-Down		
Category	**Pin**	**Power-On Reset**	**Manual Reset**	**Standby**	**Sleep**	**Bus Released**
Data bus	D[15:0]	Z	I	ZL[*10]	IO	Z
	D[23:16]/PTA[7:0]	Z	IP[*3]	ZK[*3]	IOP[*3]	ZP[*3]
	D[31:24]/PTB[7:0]	Z	IP[*3]	ZK[*3]	IOP[*3]	ZP[*3]
Bus control	$\overline{\text{CS0}}$	H	O	ZH[*11]	O	Z
	$\overline{\text{CS[2:4]}}$/PTC[5:3]	H	OP[*3]	ZH[*11] K[*3]	OP[*3]	ZP[*3]
	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/PTC[6]	H	OP[*3]	ZH[*11] K[*3]	OP[*3]	ZP[*3]
	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$/PTC[7]	H	OP[*3]	ZH[*11]K[*3]	OP[*3]	ZP[*3]
	$\overline{\text{BS}}$/PTC[0]	H	OP[*3]	ZH[*11] K[*3]	OP[*3]	ZP[*3]
	$\overline{\text{RASL}}$/PTD[0]	H	OP[*3]	ZOK[*4]	OP[*3]	ZOP[*4]
	$\overline{\text{RASU}}$/PTD[1]	H	OP[*3]	ZOK[*4]	OP[*3]	ZOP[*4]
	$\overline{\text{CASL}}$/PTD[2]	H	OP[*3]	ZOK[*4]	OP[*3]	ZOP[*4]
	$\overline{\text{CASU}}$/PTD[3]	H	OP[*3]	ZOK[*4]	OP[*3]	ZOP[*4]
	$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	H	O	ZH[*11]	O	Z
	$\overline{\text{WE1}}$/$\overline{\text{DQMLU}}$/$\overline{\text{WE}}$	H	O	ZH[*11]	O	Z
	$\overline{\text{WE2}}$/$\overline{\text{DQMUL}}$/$\overline{\text{ICIORD}}$/ PTC[1]	H	OP[*3]	ZH[*11] K[*3]	OP[*3]	ZP[*3]
	WE3/$\overline{\text{DQMUU}}$/$\overline{\text{ICIOWR}}$/ PTC[2]	H	OP[*3]	ZH[*11] K[*3]	OP[*3]	ZP[*3]
	RD/$\overline{\text{WR}}$	H	O	ZH[*11]	O	Z
	$\overline{\text{RD}}$	H	O	ZH[*11]	O	Z
	CKE/PTD[4]	H	OP[*3]	OK[*3]	OP[*3]	OP[*3]
	$\overline{\text{WAIT}}$	Z	I	Z	I	Z
DMAC	$\overline{\text{DREQ0}}$/PTH[5]	I	ZI[*7]	Z	I	I
	$\overline{\text{DACK0}}$/PTE[0]	O	OP[*3]	ZK[*3]	OP[*3]	OP[*3]
	DRAK0/PTE[2]	O	OP[*3]	ZH[*11] K[*3]	OP[*3]	OP[*3]
	$\overline{\text{DREQ1}}$/PTH[6]	I	ZI[*7]	Z	I	I
	$\overline{\text{DACK1}}$/PTE[1]	O	OP[*3]	ZK[*3]	OP[*3]	OP[*3]
	DRAK1/PTE[3]	O	OP[*3]	ZH[*11] K[*3]	OP[*3]	OP[*3]
Timer	TCLK/PTE[6]	I	ZI[*7]	IOP[*5]	IOP[*5]	IOP[*5]

Table A.1 Pin States during Resets, Power-Down States, and Bus-Released State (cont)

		Reset		Power-Down		
Category	**Pin**	**Power-On Reset**	**Manual Reset**	**Standby**	**Sleep**	**Bus Released**
SCI/Smart card without FIFO	RxD0/SCPT[0]	Z	ZI*7	Z	IZ*6	IZ*6
	TxD0/SCPT[0]	Z	ZO*7	ZK*3	OZ*6	OZ*6
	SCK0/SCPT[1]	V	ZP*3	ZK*3	IOP*5	IOP*5
SCIF with FIFO	RxD2/SCPT[2]	Z	ZI*7	Z	IZ*6	IZ*6
	TxD2/SCPT[2]	Z	ZO*7	ZK*3	OZ*6	OZ*6
	SCK2/SCPT[3]	V	ZP*3	ZK*3	IOP*5	IOP*5
	$\overline{\text{RTS2}}$/SCPT[4]	V	OP*3	ZK*3	OP*3	OP*3
	$\overline{\text{CTS2}}$/IRQ5/SCPT[5]	V*8	ZI*7	I	I	I
Port	$\overline{\text{AUDSYNC}}$/PTF[4]	OV	OP*3	OK*3	OP*3	OP*3
	$\overline{\text{CE2B}}$/PTD[7]	H	OP*3	ZH*11 K*3	OP*3	ZP*3
	$\overline{\text{CE2A}}$/PTD[6]	H	OP*3	ZH*11 K*3	OP*3	ZP*3
	TDO/PTF[5]	OV	OP*3	OK*3	OP*3	OP*3
	$\overline{\text{IOIS16}}$/PTD[5]	I	I	Z	I	I
	AUDCK/PTG[4]	IV	I	IZ	I	I
	$\overline{\text{ADTRG}}$/PTG[5]	V*8	I	IZ	I	I
	AUDATA[3:0]/PTF[3:0]	IV	I	IZ	I	I
	$\overline{\text{ASEBRKAK}}$/PTF[6]	OV	OP*3	OP*3	OP*3	OP*3
	$\overline{\text{ASEMD0}}$	I	I	Z	I	I
Analog	AN[1:0]/PTJ[1:0]	Z	ZI*7	Z	I	I
	AN[3:2]/DA[0:1]/ PTJ[3:2]	Z	ZI*7	OZ*2	IO*9	IO*9

I: Input
O: Output
H: High-level output
L: Low-level output
Z: High impedance
P: Input or output depending on register setting
K: Input pin is high impedance, output pin holds the state
V: I/O buffer off, pullup MOS on

Notes: 1. Depending on the clock mode (MD2–MD0 setting)
2. 0 when DA output is enabled: otheruise Z.
3. K or P when the port function is used.
4. K or P when the port function is used. Z or O when the port function is not used depending on register setting.

5. K or P when the port function is used. I or O when the port function is not used depending on register setting.
6. Depending on register setting
7. I or O when the port function is used.
8. Input Schmidt buffers and pullup MOS of IRQ[5:0] and ADTRG are on; other inputs are off.
9. O when DA output is enabled; otherwise depends on a register setting.
10. In the standby mode, Z or L depending on register setting.
11. In the standby mode, Z or H depending on register setting.

A.2 Pin Specifications

Table A.2 shows the pin specifications.

Table A.2 Pin Specifications

Pin	Pin No.	I/O	Function
MD0	163	I	Clock mode setting
MD1	129	I	Clock mode setting
MD2	164	I	Clock mode setting
MD3	167	I	Area 0 bus width setting
MD4	168	I	Area 0 bus width setting
MD5	169	I	Endian setting
D31 to D24/ PTB[7] to PTB[0]	5, 6, 7, 8, 9, 10, 12, 14	I/O	Data bus / input/output port B
D23 to D16/ PTA[7] to PTA[0]	15, 16, 17, 18, 20, 22, 23, 24	I/O	Data bus / input/output port A
D15 to D0	26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 38, 40, 41, 42, 43, 44	I/O	Data bus
A25 to A0	76, 75, 74, 72, 70, 69, 68, 67, 66, 65, 64, 62, 60, 59, 58, 57, 56, 55, 54, 53, 52, 50, 48, 47, 46, 45	O	Address bus
$\overline{\text{BS}}$/PTC[0]	77	O / I/O	Bus cycle start signal / input/output port C
$\overline{\text{RD}}$	78	O	Read strobe
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	79	O	D7–D0 select signal / DQM (SDRAM)
$\overline{\text{WE1}}$/$\overline{\text{DQMLU}}$/$\overline{\text{WE}}$	80	O	D15–D8 select signal / DQM (SDRAM) / write strobe (PCMCIA)
$\overline{\text{WE2}}$/$\overline{\text{DQMUL}}$/ $\overline{\text{ICIORD}}$/PTC[1]	81	O / O / O / I/O	D23–D16 select signal / DQM (SDRAM) / PCMCIA input/output read / input/output port C
$\overline{\text{WE3}}$/$\overline{\text{DQMUU}}$/ $\overline{\text{ICIOWR}}$/PTC[2]	82	O / O / O / I/O	D31–D24 select signal / DQM (SDRAM) / PCMCIA input/output write / input/output port C
RD/$\overline{\text{WR}}$	83	O	Read/write
$\overline{\text{CS0}}$	85	O	Chip select

Table A.2 Pin Specifications (cont)

Pin	Pin No.	I/O	Function
CS2/PTC[3]	87	O / I/O	Chip select 2 / input/output port C
CS3/PTC[4]	88	O / I/O	Chip select 3 / input/output port C
CS4/PTC[5]	89	O / I/O	Chip select 4 / input/output port C
CS5/CE1A/PTC[6]	90	O / O / I/O	Chip select 5 / CE1 (area 5 PCMCIA) / input/output port C
CS6/CE1B/PTC7]	91	O / O / I/O	Chip select 6 / CE1 (area 6 PCMCIA) / input/output port C
CE2A/PTD[6]	92	O / I/O	Area 5 PCMCIA CE2 / input/output port D
CE2B/PTD[7]	94	O / I/O	Area 6 PCMCIA CE2 / input/output port D
RASL/PTD[0]	96	O / I/O	Lower 32 Mbytes address RAS (SDRAM) / input/output port D
RASU/PTD[1]	97	O / I/O	Upper 32 Mbytes address RAS (SDRAM) / input/output port D
CASL/PTD[2]	98	O / I/O	Lower 32 Mbytes address CAS (SDRAM) / input/output port D
CASU/PTD[3]	99	O / I/O	Upper 32 Mbytes address CAS (SDRAM) / input/output port D
CKE/PTD[4]	100	O / I/O	CK enable (SDRAM) / input/output port D
IOIS16/PTD[5]	101	I / I/O	IOIS16 (PCMCIA) / input port D
BACK	102	O	Bus acknowledge
BREQ	103	I	Bus request
WAIT	104	I	Hardware wait request
DACK0/PTE[0]	105	O / I/O	DMA acknowledge 0 / input/output port E
DACK1/PTE[1]	106	O / I/O	DMA acknowledge 1 / input/output port E
DRAK0/PTE[2]	107	O / I/O	DMA request acknowledge / input/output port E
DRAK1/PTE[3]	108	O / I/O	DMA request acknowledge / input/output port E
AUDATA[0]/PTF[0]	109	I/O	AUD data / input/output port F
AUDATA[1]/PTF[1]	110	I/O	AUD data / input/output port F
AUDATA[2]/PTF[2]	111	I/O	AUD data / input/output port F
AUDATA[3]/PTF[3]	112	I/O	AUD data / input/output port F
AUDSYNC/PTF[4]	113	O / I/O	AUD synchronous / input/output port F
TDI/PTG[0]	114	I	Data input (H-UDI) / input port G
TCK/PTG[1]	116	I	Clock (H-UDI) / input port G
TMS/PTG[2]	118	I	Mode select (H-UDI) / input port G
TRST/PTG[3]	119	I	Reset (H-UDI) / input port G
TDO/PTF[5]	120	O / I/O	Data output (H-UDI) / input/output port F

Table A.2 Pin Specifications (cont)

Pin	Pin No.	I/O	Function
$\overline{\text{ASEBRKAK}}$/PTF[6]	121	O / I/O	ASE break acknowledge (H-UDI) / input/output port F
$\overline{\text{ASEMDO}}$	122	I	ASE mode (H-UDI)
CAP1	124	—	PLL1 external capacitance pin
CAP2	127	—	PLL2 external capacitance pin
XTAL	131	O	Clock oscillator pin
EXTAL	132	I	External clock / crystal oscillator pin
XTAL	2	O	On-chip RTC crystal oscillator pin
EXTAL2	3	I	On-chip RTC crystal oscillator pin
STATUS0/PTE[4]	133	O / I/O	Processor status / input/output port E
STATUS1/PTE[5]	134	O / I/O	Processor status / input/output port E
TCLK/PTE[6]	135	I/O	TMU or RTC clock input/output / input/output port E
$\overline{\text{IRQOUT}}$/PTE[7]	136	O / I/O	Interrupt request notification / input/output port E
CKIO	138	I/O	System clock input/output
TxD0/SCPT[0]	140	O	SCI transmit data 0 / SC port
TxD2/SCPT[2]	142	O	SCIF transmit data 2 / SC port
SCK0/SCPT[1]	141	I/O	SCI clock 0 / SC port
SCK2/SCPT[3]	143	I/O	SCIF clock 2 / SC port
RxD0/SCPT[0]	145	I	SCI receive data 0 / SC port
RxD2/SCPT[2]	146	I	SCIF receive data 2 / SC port
$\overline{\text{RTS2}}$/SCPT[4]	144	O / I/O	SCIF transmit request 2 / SC port
$\overline{\text{CTS2}}$/IRQ5/ SCPT[5]	147	I	SCIF transmit clear / external interruption request / SC port
$\overline{\text{RESETM}}$	149	I	Manual reset request
IRQ[3:0]/$\overline{\text{IRL}}$[3:0]/ PTH[[3:0]]	151, 152, 153, 154	I / I / I/O	External interrupt request / input/output port H
IRQ4/PTH[4]	155	I / I/O	External interrupt request / input/output port H
NMI	157	I	Nonmaskable interrupt request
AUDCK/PTG[4]	159	I	AUD clock / input port G
$\overline{\text{RESETP}}$	165	I	Power-on reset request
CA	166	I	Chip activate / hardware standby request
AN[0]/PTJ[0]	171	I	A/D converter input / input port J
AN[1]/PTJ[1]	172	I	A/D converter input / input port J
AN2[2]/DA[1]/PTJ[2]	173	I / O / I	A/D converter input / D/A converter output / input port J
AN3[3]/DA[0]/PTJ[3]	174	I / O / I	A/D converter input / D/A converter output / input port J

Table A.2 Pin Specifications (cont)

Pin	Pin No.	I/O	Function
$\overline{\text{ADTRG}}$/PTG[5]	162	I	Analog trigger / input port G
$\overline{\text{DREQ0}}$/PTH[5]	160	I / I/O	DMA request / input/output port H
$\overline{\text{DREQ1}}$/PTH[6]	161	I / I/O	DMA request / input/output port H
$V_{CC}Q$	13, 27, 39, 51, 63, 86, 95, 139, 158	Power supply	Input/output power supply (3.3 V)
V_{CC}	21, 73, 117, 150	Power supply	Internal power supply (1.9 V)
V_{CC}-RTC	1	Power supply	RTC power supply (1.9 V)
V_{CC}-PLL1	123	Power supply	PLL1 power supply (1.9 V)
V_{CC}-PLL2	128	Power supply	PLL2 power supply (1.9 V)
AV_{CC}	175	Power supply	Analog power supply (3.3 V)
$V_{SS}Q$	11, 25, 37, 49, 61, 84, 93, 137, 156	Power supply	Input/output power supply (0 V)
V_{SS}	19, 71, 115, 130, 148	Power supply	Internal power supply (0 V)
V_{SS}-RTC	4	Power supply	RTC power supply (0 V)
V_{SS}-PLL1	125	Power supply	PLL1 power supply (0 V)
V_{SS}-PLL2	126	Power supply	PLL2 power supply (0 V)
AV_{SS}	170, 176	Power supply	Analog power supply (0 V)

A.3 Processing of Unused Pins

- When RTC is not used
  - EXTAL2: Pull up to V_{CC} (1.9)
  - XTAL2: Leave unconnected
  - V_{CC} – RTC: Power supply (1.9)
  - V_{SS} – RTC: Power supply (0 V)
- When PLL1 is not used
  - CAP1: Leave unconnected
  - V_{CC} – PLL1: Power supply (1.9)
  - V_{SS} – PLL1: Power supply (0 V)
- When PLL2 is not used
  - CAP2: Leave unconnected
  - V_{CC} – PLL2: Power supply (1.9/1.8 V)
  - V_{SS} – PLL2: Power supply (0 V)
- When on-chip crystal oscillator is not used
  - XTAL: Leave unconnected
- When EXTAL terminal is not used
  - EXTAL: Pull up to V_{CC}Q (3.3 V)
- When A/D converter is not used
  - AN[3:0]: Leave unconnected
  - AV_{CC}: Power supply (3.3 V)
  - AV_{SS}: Power supply (0 V)

A.4 Pin States in Access to Each Address Space

Table A.3 Pin States (Normal Memory/Little Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
CS6 to CS2, CS0		Enabled	Enabled	Enabled	Enabled
RD	R	Low	Low	Low	Low
	W	High	High	High	High
RD/WR	R	High	High	High	High
	W	Low	Low	Low	Low
BS		Enabled	Enabled	Enabled	Enabled
RASU/PTD[1]		High	High	High	High
RASL/PTD[0]		High	High	High	High
CASL/PTD[2]		High	High	High	High
CASU/PTD[3]		High	High	High	High
WE0/DQMLL	R	High	High	High	High
	W	Low	Low	High	Low
WE1/WE/DQMLU	R	High	High	High	High
	W	High	High	Low	Low
WE2/ICIORD/DQMUL/ PTC[1]	R	High	High	High	High
	W	High	High	High	High
WE3/ICIOWR/DQMUU/ PTC[2]	R	High	High	High	High
	W	High	High	High	High
CE2A/PTD[6]		High	High	High	High
CE2B/PTD[7]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
WAIT		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
IOIS16		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z[*2]	Invalid data	Valid data	Valid data
D31 to D16		High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]

Table A.3 Pin States (Normal Memory/Little Endian) (cont)

		32-Bit Bus Width						
Pin		**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	Low	Low	Low
	W	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High	High	High	High	High	High	High
$\overline{\text{RASL}}$/PTD[0]		High	High	High	High	High	High	High
$\overline{\text{CASL}}$/PTD[2]		High	High	High	High	High	High	High
$\overline{\text{CASU}}$/PTD[3]		High	High	High	High	High	High	High
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/$\overline{\text{DQMLU}}$	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/$\overline{\text{DQMUL}}$/ PTC[1]	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/$\overline{\text{DQMUU}}$/ PTC[2]	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
$\overline{\text{CE2A}}$/PTD[6]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTD[7]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: *1 Disabled when WCR2 register wait setting is 0.

*2 Unused data pins should be switched to the port function, or pulled up or down.

Table A.4 Pin States (Normal Memory/Big Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
CS6 to CS2, CS0		Enabled	Enabled	Enabled	Enabled
RD	R	Low	Low	Low	Low
	W	High	High	High	High
RD/WR	R	High	High	High	High
	W	Low	Low	Low	Low
BS		Enabled	Enabled	Enabled	Enabled
RASU/PTD[1]		High	High	High	High
RASL/PTD[0]		High	High	High	High
CASL/PTD[2]		High	High	High	High
CASU/PTD[3]		High	High	High	High
WE0/DQMLL	R	High	High	High	High
	W	Low	High	Low	Low
WE1/WE/DQMLU	R	High	High	High	High
	W	High	Low	High	Low
WE2/ICIORD/DQMUL/PTC[1]	R	High	High	High	High
	W	High	High	High	High
WE3/ICIOWR/DQMUU/PTC[2]	R	High	High	High	High
	W	High	High	High	High
CE2A/PTD[6]		High	High	High	High
CE2B/PTD[7]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
WAIT		Enabled*1	Enabled*1	Enabled*1	Enabled*1
IOIS16		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z*2	Valid data	Invalid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.4 Pin States (Normal Memory/Big Endian) (cont)

		32-Bit Bus Width						
Pin		Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	Low	Low	Low
	W	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High	High	High	High	High	High	High
$\overline{\text{RASL}}$/PTD[0]		High	High	High	High	High	High	High
$\overline{\text{CASL}}$/PTD[2]		High	High	High	High	High	High	High
$\overline{\text{CASU}}$/PTD[3]		High	High	High	High	High	High	High
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/$\overline{\text{DQMLU}}$	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/$\overline{\text{DQMUL}}$/ PTC[1]	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/$\overline{\text{DQMUU}}$/ PTC[2]	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
$\overline{\text{CE2A}}$/PTD[6]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTD[7]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data
D15 to D8		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: *1 Disabled when WCR2 register wait setting is 0.

*2 Unused data pins should be switched to the port function, or pulled up or down.

Table A.5 Pin States (Burst ROM/Little Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low
	W	—	—	—	—
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High	High	High	High
$\overline{\text{RASL}}$/PTD[0]		High	High	High	High
$\overline{\text{CASL}}$/PTD[2]		High	High	High	High
$\overline{\text{CASU}}$/PTD[3]		High	High	High	High
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/$\overline{\text{DQMLU}}$	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/$\overline{\text{DQMUL}}$/PTC[1]	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/$\overline{\text{DQMUU}}$/PTC[2]	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{CE2A}}$/PTD[6]		High	High	High	High
$\overline{\text{CE2B}}$/PTD[7]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z*2	Invalid data	Valid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.5 Pin States (Burst ROM/Little Endian) (cont)

		32-Bit Bus Width						
Pin		Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	Low	Low	Low
	W	—	—	—	—	—	—	—
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High	High	High	High	High	High	High
$\overline{\text{RASL}}$/PTD[0]		High	High	High	High	High	High	High
$\overline{\text{CASL}}$/PTD[2]		High	High	High	High	High	High	High
$\overline{\text{CASU}}$/PTD[3]		High	High	High	High	High	High	High
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/$\overline{\text{DQMLU}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/$\overline{\text{DQMUL}}$/ PTC[1]	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/$\overline{\text{DQMUU}}$/ PTC[2]	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{CE2A}}$/PTD[6]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTD[7]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: *1 Disabled when WCR2 register wait setting is 0.

*2 Unused data pins should be switched to the port function, or pulled up or down.

Table A.6 Pin States (Burst ROM/Big Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low
	W	—	—	—	—
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High	High	High	High
$\overline{\text{RASL}}$/PTD[0]		High	High	High	High
$\overline{\text{CASL}}$/PTD[2]		High	High	High	High
$\overline{\text{CASU}}$/PTD[3]		High	High	High	High
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/$\overline{\text{DQMLU}}$	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/$\overline{\text{DQMUL}}$/PTC[1]	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/$\overline{\text{DQMUU}}$/PTC[2]	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{CE2A}}$/PTD[6]		High	High	High	High
$\overline{\text{CE2B}}$/PTD[7]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z*2	Valid data	Invalid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.6 Pin States (Burst ROM/Big Endian) (cont)

		32-Bit Bus Width						
Pin		**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	Low	Low	Low
	W	—	—	—	—	—	—	—
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High	High	High	High	High	High	High
$\overline{\text{RASL}}$/PTD[0]		High	High	High	High	High	High	High
$\overline{\text{CASL}}$/PTD[2]		High	High	High	High	High	High	High
$\overline{\text{CASU}}$/PTD[3]		High	High	High	High	High	High	High
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/$\overline{\text{DQMLU}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/$\overline{\text{DQMUL}}$/PTC[1]	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/$\overline{\text{DQMUU}}$/PTC[2]	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{CE2A}}$/PTD[6]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTD[7]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data
D15 to D8		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: *1 Disabled when WCR2 register wait setting is 0.

*2 Unused data pins should be switched to the port function, or pulled up or down.

Table A.7 Pin States (Synchronous DRAM/Little Endian)

		32-Bit Bus Width						
Pin		**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1
$\overline{\text{RASL}}$/PTD[0]		Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1
$\overline{\text{CASL}}$/PTD[2]		High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1
$\overline{\text{CASU}}$/PTD[3]		Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1
$\overline{\text{DQMLL}}$/$\overline{\text{WE0}}$	R	Low	High	High	High	Low	High	Low
	W	Low	High	High	High	Low	High	Low
$\overline{\text{DQMLU}}$/$\overline{\text{WE1}}$	R	High	Low	High	High	Low	High	Low
	W	High	Low	High	High	Low	High	Low
$\overline{\text{DQMUL}}$/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	Low	High	High	Low	Low
	W	High	High	Low	High	High	Low	Low
$\overline{\text{DQMUU}}$/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	Low	High	Low	Low
	W	High	High	High	Low	High	Low	Low
$\overline{\text{CE2A}}$/PTD[6]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTD[7]		High	High	High	High	High	High	High
CKE		High*2	High*2	High*2	High*2	High*2	High*2	High*2
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address command	Address command	Address command	Address command	Address command	Address command	Address command
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: *1 Lower 32-Mbyte access / Upper 32-Mbyte access

*2 Normally high. Low in self-refreshing.

Table A.8 Pin States (Synchronous DRAM/Big Endian)

		32-Bit Bus Width						
Pin		Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
CS6 to CS2, CS0		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
RD	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/WR	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
BS		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
RASU/PTD[1]		High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]
RASL/PTD[0]		Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]
CASL/PTD[2]		High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]
CASU/PTD[3]		Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]
DQMLL/WE0	R	High	High	High	Low	High	Low	Low
	W	High	High	High	Low	High	Low	Low
DQMLU/WE1	R	High	High	Low	High	High	Low	Low
	W	High	High	Low	High	High	Low	Low
DQMUL/WE2/ICIORD	R	High	Low	High	High	Low	High	Low
	W	High	Low	High	High	Low	High	Low
DQMUU/WE3/ICIOWR	R	Low	High	High	High	Low	High	Low
	W	Low	High	High	High	Low	High	Low
CE2A/PTD[6]		High	High	High	High	High	High	High
CE2B/PTD[7]		High	High	High	High	High	High	High
CKE		High[*2]	High[*2]	High[*2]	High[*2]	High[*2]	High[*2]	High[*2]
WAIT		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
IOIS16		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address command	Address command	Address command	Address command	Address command	Address command	Address command
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: [*1] Lower 32-Mbyte access/ Upper 32-Mbyte access

[*2] Normally high. Low in self-refreshing.

Table A.9 Pin States (PCMCIA/Little Endian)

		PCMCIA Memory Interface (Area 5)				PCMCIA/IO Interface (Area 5)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access	Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High	High	High	High	High	High	High	High
$\overline{\text{RASL}}$/PTD[0]		High	High	High	High	High	High	High	High
$\overline{\text{CASL}}$/PTD[2]		High	High	High	High	High	High	High	High
$\overline{\text{CASU}}$/PTD[3]		High	High	High	High	High	High	High	High
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/$\overline{\text{DQMLU}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/ $\overline{\text{DQMUL}}$/PTC[1]	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/ $\overline{\text{DQMUU}}$/PTC[2]	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
$\overline{\text{CE2A}}$/PTD[6]		High	High	Low	Low	High	High	Low	Low
$\overline{\text{CE2B}}$/PTD[7]		High	High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z[*2]	Invalid data	Valid data	Valid data	High-Z[*2]	Invalid data	Valid data	Valid data
D31 to D16		High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]

Table A.9 Pin States (PCMCIA/Little Endian) (cont)

		PCMCIA Memory Interface (Area 6)				PCMCIA/IO Interface (Area 6)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access	Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High	High	High	High	High	High	High	High
$\overline{\text{RASL}}$/PTD[0]		High	High	High	High	High	High	High	High
$\overline{\text{CASL}}$/PTD[2]		High	High	High	High	High	High	High	High
$\overline{\text{CASU}}$/PTD[3]		High	High	High	High	High	High	High	High
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/$\overline{\text{DQMLU}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/$\overline{\text{DQMUL}}$/ PTC[1]	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/$\overline{\text{DQMUU}}$/ PTC[2]	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
$\overline{\text{CE2A}}$/PTD[6]		High	High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTD[7]		High	High	Low	Low	High	High	Low	Low
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled *1	Enabled *1	Enabled *1	Enabled *1	Enabled *1	Enabled *1	Enabled *1	Enabled *1
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z*2	Invalid data	Valid data	Valid data	High-Z*2	Invalid data	Valid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2

Notes: *1 Disabled when WCR2 register wait setting is 0.

*2 Unused data pins should be switched to the port function, or pulled up or down.

Table A.10 Pin States (PCMCIA/Big Endian)

		PCMCIA Memory Interface (Area 5)				PCMCIA/IO Interface (Area 5)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/ Word/ Long-word Access	Byte Access (Ad-dress 2n)	Byte Access (Ad-dress 2n + 1)	Word/ Long-word Access	Byte/ Word/ Long-word Access	Byte Access (Ad-dress 2n)	Byte Access (Ad-dress 2n + 1)	Word/ Long-word Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High	High	High	High	High	High	High	High
$\overline{\text{RASL}}$/PTD[0]		High	High	High	High	High	High	High	High
$\overline{\text{CASL}}$/PTD[2]		High	High	High	High	High	High	High	High
$\overline{\text{CASU}}$/PTD[3]		High	High	High	High	High	High	High	High
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/$\overline{\text{DQMLU}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/ $\overline{\text{DQMUL}}$/PTC[1]	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/ $\overline{\text{DQMUU}}$/PTC[2]	R	High	High	High	High	High	High	High	High
	W	High	High	Low	Low	Low	Low	Low	Low
$\overline{\text{CE2A}}$/PTD[6]		High	High	Low	Low	High	High	Low	Low
$\overline{\text{CE2B}}$/PTD[7]		High	High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z*2	Valid data	Invalid data	Valid data	High-Z*2	Valid data	Invalid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.10 Pin States (PCMCIA/Big Endian) (cont)

		PCMCIA Memory Interface (Area 6)				PCMCIA/IO Interface (Area 6)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access	Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RASU}}$/PTD[1]		High	High	High	High	High	High	High	High
$\overline{\text{RASL}}$/PTD[0]		High	High	High	High	High	High	High	High
$\overline{\text{CASL}}$/PTD[2]		High	High	High	High	High	High	High	High
$\overline{\text{CASU}}$/PTD[3]		High	High	High	High	High	High	High	High
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/$\overline{\text{DQMLU}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/$\overline{\text{DQMUL}}$/ PTC[1]	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/$\overline{\text{DQMUU}}$/ PTC[2]	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
$\overline{\text{CE2A}}$*[3]/PTD[6]		High	High	High	High	High	High	High	High
$\overline{\text{CE2B}}$*[3]/PTD[7]		High	High	Low	Low	High	High	Low	Low
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled *[1]	Enabled *[1]	Enabled *[1]	Enabled *[1]	Enabled *[1]	Enabled *[1]	Enabled *[1]	Enabled *[1]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Invalid data	Valid data
D15 to D8		High-Z*[2]	Valid data	Invalid data	Valid data	High-Z*[2]	Valid data	Invalid data	Valid data
D31 to D16		High-Z*[2]	High-Z*[2]	High-Z*[2]	High-Z*[2]	High-Z*[2]	High-Z*[2]	High-Z*[2]	High-Z*[2]

Notes: *1 Disabled when WCR2 register wait setting is 0.

*2 Unused data pins should be switched to the port function, or pulled up or down.

*3 The behavior of the CE pin in the big endian is the same as that in the little endian.

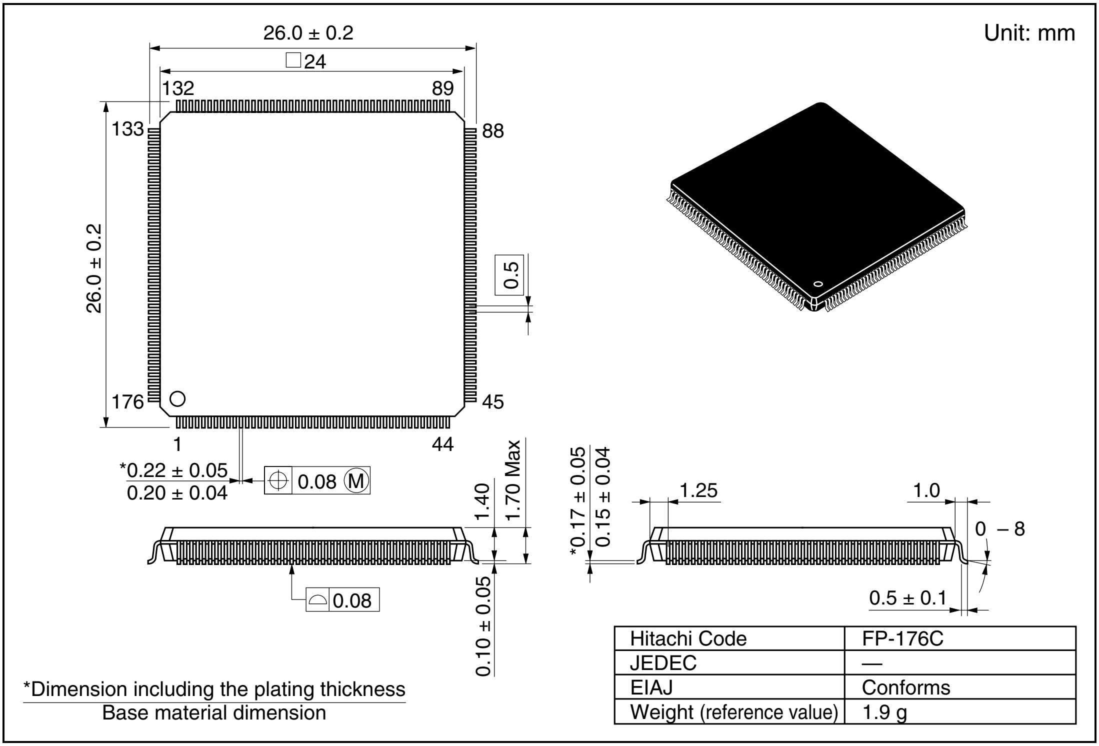
Appendix B Product Lineup

Table B.1 SH7706 Product Lineup

Abbr.	Model Marking	Package
SH7706	HD6417706F133	176-pin plastic LQFP (FP-176C)
	HD6417706BP133V	208-pin TFBGA (TBP-208AV)

Appendix C Package Dimensions

Figures C.1 and C.2 show the SH7706 package dimensions.



Hitachi Code	FP-176C
JEDEC	—
EIAJ	Conforms
Weight (reference value)	1.9 g

Figure C.1 Package Dimensions (FP-176C)

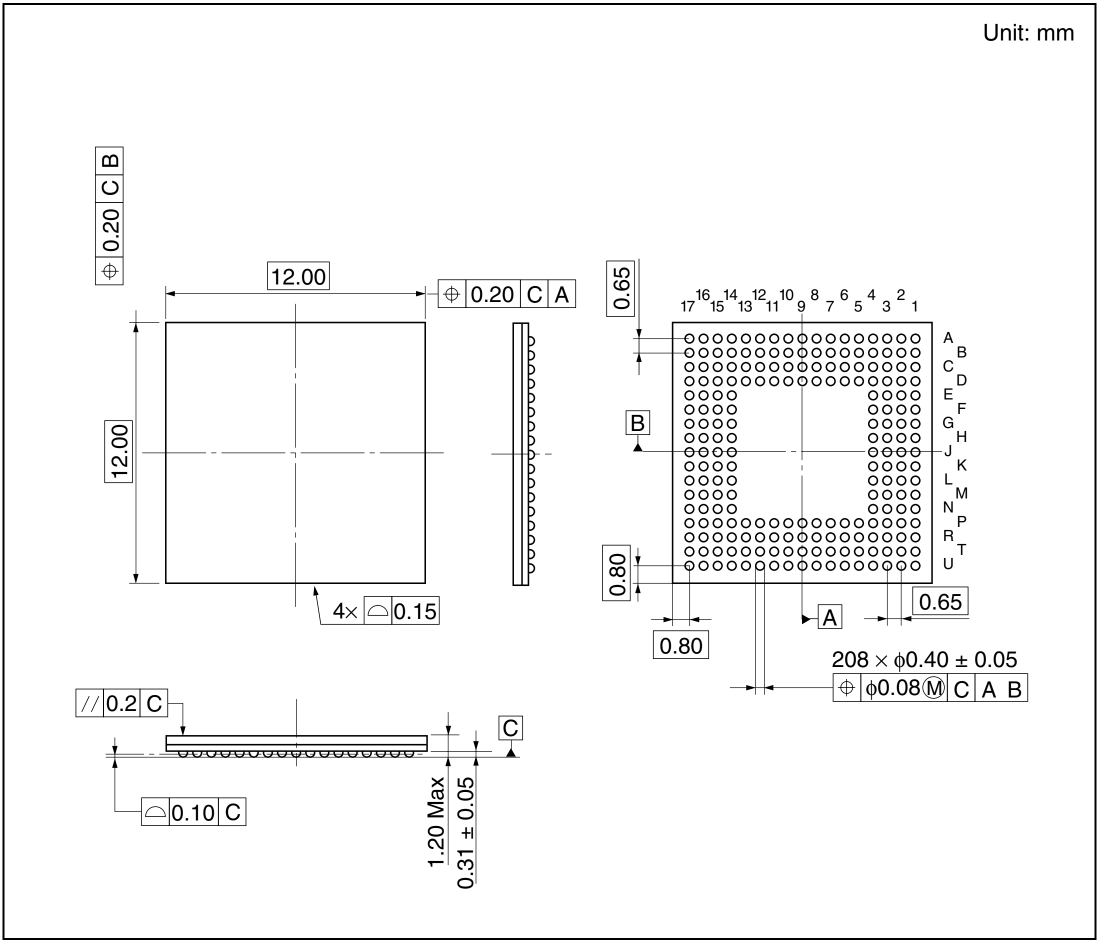


Figure C.2 Package Dimensions (TBP-208AV)

Index

SH7706 Hardware Manual

Publication Date: 1st Edition, September 2001
Published by: Customer Service Division
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.